

# NSX vSphere API Guide

NSX 6.1 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-001545-02

**vmware®**

You can find the most up-to-date technical documentation on the VMware Web site at:

<http://www.vmware.com/support/>

The VMware Web site also provides the latest product updates.

If you have comments about this documentation, submit your feedback to:

[docfeedback@vmware.com](mailto:docfeedback@vmware.com)

Copyright © 2012 - 2014 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at <http://www.vmware.com/go/patents>.

VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

**VMware, Inc.**  
3401 Hillview Ave.  
Palo Alto, CA 94304  
[www.vmware.com](http://www.vmware.com)

# Contents

About This Book	19
<b>1 Overview of NSX</b>	<b>21</b>
NSX Components	21
Data Plane	22
Control Plane	22
Management Plane	22
Consumption Platform	22
NSX Services	22
Logical Switches	23
Logical Routers	23
Logical Firewall	23
Logical Virtual Private Networks (VPN)s	23
Logical Load Balancer	23
Service Composer	23
NSX Extensibility	24
An Introduction to REST API for NSX Users	24
How REST Works	24
About the REST API	24
RESTful Workflow Patterns	24
For More Information About REST	25
Using the NSX REST API	25
Ports Required for NSX REST API	25
<b>2 User Management</b>	<b>27</b>
Configuring SSO on NSX Manager	27
Query SSO Details	27
Query SSO Configuration Status	28
Delete SSO Configuration	28
User Management	28
Get Information About a User	28
Enable or Disable a User Account	29
Remove Role Assignment	29
Role Management	29
Get Role for a User	30
Get Role for a NSX Manager User	30
Add Role and Resources for a User	31
Change User Role	31
Get List of Possible Roles	32
Get List of Scoping Objects	32
Delete User Role	33
<b>3 Managing the NSX Manager Appliance</b>	<b>35</b>
Upgrading the Appliance Manager	35
Upload Upgrade Bundle	35
Query Upgrade Information	35
Begin Upgrade	36
Query Upgrade Status	37

Configuring NSX Manager with vCenter Server	37
Configure vCenter Server with NSX Manager	37
Query Configuration Details	37
Certificate Management	38
Generate CSR Certificate	38
Download CSR Certificate	38
Upload Certificate Chain	38
Query Certificates	39
Upload Keystore File	39
Resource Management	39
Query Global Appliance Manager Information	39
Query Summary Appliance Manager Information	40
Query Component Information	40
Reboot Appliance Manager	41
Query Appliance Manager CPU	41
Query Appliance Manager Uptime	42
Query Appliance Manager Memory	42
Query Appliance Manager Storage	42
Working with Network Settings	42
Query Network Information	42
Configure DNS Servers	43
Delete DNS Servers	43
Working with Time Settings	43
Configure Time Settings	43
Query Time Settings	44
Delete Time Settings	44
Working with Locale Settings	44
Configure Locale	44
Query Locale	44
Working with Syslog Servers	45
Configure Syslog Servers	45
Query Syslog Servers	45
Retrieves syslog servers.	45
Delete Syslog Servers	45
Deletes syslog servers.	45
Components Management	46
Query Components	46
Query Specific Component	46
Query Component Dependencies	47
Query Specific Component Dependents	47
Query Component Status	47
Toggle Specific Component Status	48
Working with Backup and Restore	48
Configure Backup Settings	48
Configure On-Demand Backup	49
Query Backup Settings	49
Delete Backup Configuration	50
Query Available Backups	50
Restore Data	50
Working with Tech Support Logs	50
Generate Tech Support Logs	50
Download Tech Support Logs	51
Querying NSX Manager Logs	51
Get NSX Manager System Events	51
Get NSX Manager Audit Logs	51

Working with Support Notifications	51
Query Notifications	51
Delete all Notifications	52
Acknowledge Notifications	52

## 4 Grouping Objects 53

Working with Security Groups	53
Create Security Group	53
Query Security Groups	55
Query Members for a Scope	57
Query Security Group Objects	58
Query Security Groups that contain a Virtual Machine	58
Modify a Security Group	58
Delete a Security Group	59
Add Member to Security Group	59
Delete Member from Security Group	59
Working with Tags	59
Create Security Tag	59
Query Security Tags	60
Apply Tag to Virtual Machine	60
Detach Tag from Virtual Machine	60
Delete Tag from Virtual Machine	60
Working with IPsets	60
Create an IPset	60
Query IPsets	61
Query Details of an IPset	61
Modify an IPset	61
Delete an IPset	62
Working with MACsets	62
Create a MACset on a Scope	62
List MACsets Created on a Scope	62
Get Details of a MACset	63
Modify an Existing MACset	63
Delete a MACset	63
Working with Services	64
List Services on a Scope	64
Add Service to a Scope	64
Get Details of a Service	65
Modify Service Details	65
Delete Service	66
Working with Service Groups	66
Add Service Group	66
Query Service Groups	66
Query Details of a Service Group	67
Modify Service Group Details	67
Delete Service Group from Scope	68
Working with the Members of a Service Group	69
Query Service Group Members	69
Add a Member to the Service Group	70
Delete a Member from the Service Group	70
Working with IP Pools	70
Add an IP Pool	70
Query IP Pool Details	70
Modify an IP Pool	71
Allocating a New IP Address	72
Allocating a Specific IP Address	72

Query all IP Pools on Scope	72
Query Allocated IP Addresses	73
Release an IP Address	74
Delete an IP Pool	74
Querying Object IDs	74
Query Datacenter MOID	74
Query Datacenter ID	74
Query Host ID	74
Query Portgroup ID	75

## 5 Installing NSX Components 77

Installing Licenses	77
Working with Network Virtualization Components	78
Install Network Virtualization Components	78
Upgrade Network Virtualization Components	78
Delete Network Virtualization Components	79
Working with VXLAN for Logical Switches	79
Working with Controllers	80
Add Controller	80
Query Controllers	80
Query Controller Addition or Deletion Details	81
Query Controller Tech Support Logs	81
Delete Controller	81
Query Cluster Information	81
Modify Cluster Configuration	81
Add Controller Syslog Exporter	82
Query Controller Syslog Exporter	82
Delete Controller Syslog Exporter	82
Backup Controller Data	83
Working with Segment IDs	83
Add a new Segment ID Range	83
Query all Segment ID Ranges	83
Query a Specific Segment ID Range	84
Update a Segment ID Range	84
Delete a Segment ID Range	84
Configure VXLAN	85
Install VXLAN	85
Delete VXLAN	86
Delete VXLAN with vdsContext	86
Working with Network Scopes	86
Create a Network Scope	86
Edit a Network Scope	86
Update Attributes on a Network Scope	87
Query existing Network Scopes	87
Query a Specific Network Scope	88
Delete a Network Scope	88
Reset Communication	88
Query Features on Cluster	88
Query Status of Specific Resources	89
Query Status of Child Resources	90
Query Status of Resources by Criterion	91
Working with Services	92
Install Security Fabric	93
Service Dependency	93

Deploying a Service with a Dependency	94
Identify Service Dependency	94
Uninstall Service Dependency	94
Query Installed Services	94
Query Details about a Service	95
Query Clusters	95
Upgrade Service	96
Query Agents on Host	96
Query Agent Information	97
Query Agents for Deployment	98
Working with Conflicting Agencies	99
Query Conflicts	99
Restore Conflicting Agencies	99
Delete Conflicting Agencies	100
Delete Deployment Units	100
Uninstalling Services	100

## 6 Working with Logical Switches 103

Preparing for Logical Switches	104
Configuring Switches	104
Prepare Switch	104
Query Configured Switches	104
Query Configured Switches on Datacenter	105
Query Specific Switch	105
Delete Switch	105
Working with Segment IDs	106
Add a new Segment ID Range	106
Query all Segment ID Ranges	106
Query a Specific Segment ID Range	107
Update a Segment ID Range	107
Delete a Segment ID Range	107
Working with Multicast Address Ranges	107
Add a new Multicast Address Range	107
Query all Multicast Address Ranges	108
Get a Specific Multicast Address Range	108
Update a Multicast Address Range	109
Delete a Multicast Address Range	109
Working with Network Scopes	109
Create a Network Scope	109
Edit a Network Scope	109
Update Attributes on a Network Scope	110
Query existing Network Scopes	110
Query a Specific Network Scope	111
Delete a Network Scope	111
Working with Logical Switches	111
Create a Logical Switch	111
Query all Logical Switches on a Network Scope	112
Query all Logical Switches on all Network Scopes	112
Query a Specific Logical Switch	113
Modify Control Plane Mode	113
Delete a Logical Switch	114
Working with ARP Suppression and MAC Learning for Logical Switches	114
Managing the Logical Switch UDP Port	115
Get UDP Port	115
Update UDP Port	115

Querying Allocated Resources	115
Testing Multicast Group Connectivity	116
Test Multicast Group Connectivity in a Network Scope	116
Test Multicast Group Connectivity in a Logical Switch	116
Performing Ping Test	117
<b>7 NSX Edge Logical Router Installation and Management</b>	<b>119</b>
Installing a Logical Router	119
Query a Logical Router	120
Modify a Router	122
Deleting a Router	122
Working with Interfaces	122
Working with Management Interfaces	122
Configure Management Interfaces	122
Query Management Interfaces	123
Working with all Interfaces	123
Add Interfaces	123
Query Interfaces for a NSX Edge Router	124
Delete Interfaces	125
Delete all Interfaces	125
Manage an NSX Edge Router Interface	126
Retrieve Interface with Specific Index	126
Modify an Interface	126
Delete Interface Configuration	126
Configure Routes	127
Query Routes	129
Delete Routes	132
Manage Global Routing Configuration	132
Specify Global Configuration	132
Query Global Route	132
Manage Static Routing	133
Configure Static Routes	133
Query Static Routes	133
Delete Static Routes	134
Manage OSPF Routes for NSX Edge	134
Configure OSPF	134
Query OSPF	135
Delete OSPF	136
Manage ISIS Routes for NSX Edge	136
Configure ISIS	136
Query ISIS	137
Delete ISIS	138
Manage BGP Routes for NSX Edge	139
Configure BGP	139
Query BGP	140
Delete BGP	141
Working with Bridging	141
Configure a Bridge	141
Query Bridge Configuration	142
Query BGP	142
Delete Bridge Configuration	142



## 8 NSX Edge Services Gateway Installation, Upgrade, and Management 143

- Installing NSX Edge Services Gateway 145
- Upgrading vShield Edge 5.1.x or 5.5 to NSX Edge 147
- Query Installed Edges 147
- Modifying NSX Edge Configuration 151
- Deleting NSX Edge 155
- Configuring Edge Services in Async Mode 155
  - Query Async Job Status 155
  - Query all Jobs 156
  - Query active Jobs 156
- Configuring Certificates 157
  - Working with Certificates 157
    - Create Certificate 157
    - Create Certificate or Certificate Chain for CSR 157
    - Query Certificates 157
    - Delete Certificate 157
  - Working with Certificate Signing Requests (CSRs) 158
    - Create CSR 158
    - Create Self Signed Certificate for CSR 158
    - Query CSRs 158
  - Working with Certificate Revocation List (CRL) 159
    - Create a CRL 159
    - Query CRL 159
    - Delete CRL 159
- Working with NSX Edge Firewall 159
  - Configure Firewall 160
  - Query Firewall Configuration 161
  - Query Firewall Configuration with PreRules for Edge 163
    - Append Firewall Rules 165
    - Add a Firewall Rule Above a Specific Rule 165
    - Query Specific Rule 166
    - Modify Firewall Rule 166
    - Delete a Firewall Rule 167
  - Delete Firewall Configuration 167
  - Manage Global Firewall Configuration 167
    - Query Global Firewall Configuration 167
    - Modify Global Configuration 168
  - Manage Default Firewall Policy 168
    - Query Default Firewall Policy 168
    - Modify Default Firewall Policy 169
  - Query Firewall Statistics 169
  - Query Firewall Statistics for Rule 169
  - Disable Firewall 170
- Working with NAT 170
  - Configure NAT 170
    - Query NAT Rules for a Edge Edge 171
    - Delete all NAT Rules 172
    - Add a NAT Rule above a Specific Rule 172
    - Append NAT Rules 172
    - Modify a NAT Rule 173
    - Delete a NAT Rule 173
- Working with Routing 173
  - Configure Routes 173
  - Query Routes 177

Delete Routes	177
Manage Global Routing Configuration	177
Specify Global Configuration	177
Query Global Route	178
Manage Static Routing	178
Configure Static Routes	178
Query Static Routes	179
Delete Static Routes	179
Manage OSPF Routes for NSX Edge	180
Configure OSPF	180
Query OSPF	181
Delete OSPF	182
Manage ISIS Routes for NSX Edge	182
Configure ISIS	182
Query ISIS	183
Delete ISIS	184
Manage BGP Routes for NSX Edge	184
Configure BGP	184
Query BGP	185
Delete BGP	186
Working with Load Balancer	187
Configure Load Balancer	187
Query Load Balancer Configuration	193
Delete Load Balancer Configuration	193
Manage Application profiles	194
Append Application Profile	194
Modify Application Profile	194
Query Application Profile	194
Query all Application Profiles	195
Delete Application Profile	195
Delete all Application Profiles	196
Manage Application Rules	196
Append Application Rule	196
Modify Application Rule	196
Query Application Rule	196
Query all Application Rules	196
Delete Application Rule	197
Delete all Application Rules	197
Manage Load Balancer Monitors	197
Append Monitor	197
Modify Monitor	197
Query Monitor	198
Query all Monitors	198
Delete Monitor	199
Delete all Monitors	199
Manage Virtual Servers	199
Append Virtual Server	199
Query a Virtual Server	200
Query all Virtual Servers	200
Delete a Virtual Server	201
Delete all Virtual Server	201
Manage Backend Pools	201
Append Backend Pool	201

Modify a Backend Pool	202
Query Backend Pool Details	202
Query all Backend Pools	203
Delete a Backend Pool	205
Delete all Backend Pools	205
Query Statistics	205
Update LoadBalancer Acceleration Mode	207
Update Load Balancer Member Condition	207
Working with DHCP Service	207
Configure DHCP	208
Query DHCP Configuration	209
Delete DHCP Configuration	209
Retrieve DHCP Lease Information	210
Append IP Pool to DHCP Configuration	210
Append Static Binding to DHCP Configuration	210
Delete DHCP Pool	211
Delete DHCP Static Binding	211
Working with DHCP Relay	211
Query DHCP Relay	212
Delete DHCP Relay Configuration	212
Working with High Availability (HA)	213
Retrieve High Availability Configuration	213
Delete High Availability Configuration	213
Working with Syslog	214
Configure Syslog	214
Query Syslog	214
Delete Syslog	214
Managing SSL VPN	215
Enable or Disable SSL VPN	215
Query SSL VPN Details	215
Manage Server Settings	215
Apply Server Settings	215
Query Server Settings	216
Configure Private Networks	216
Add Private Network	216
Modify Private Network	216
Query Specific Private Network	217
Delete Private Network	217
Delete all Private Networks	218
Apply All Private Networks	218
Configure Web Resource	218
Add Portal Web Resource	218
Modify Portal Web Resource	218
Query Portal Web Resource	219
Query all Web Resources	219
Delete Portal Web Resource	220
Deletes all Web Resources	220
Apply All Web Resources	220
Configure Users	220
Add User	220
Modify User	221
Query User Details	221
Delete User	221
Delete all Users	222

Apply all Users	222
Configure IP Pool	222
Add IP Pool	222
Modify IP Pool	223
Query IP Pool	223
Query all IP Pools	223
Delete IP Pool	224
Deletes all IP Pools	224
Apply all IP Pools	224
Configure Network Extension Client Parameters	225
Apply Client Configuration	225
Get Client Configuration	225
Configure Network Extension Client Installation Package	225
Add Client Installation Package	225
Modify Client Installation Package	226
Query Client Installation Package	227
Query all Client Installation Packages	227
Delete Client Installation Package	228
Delete all Client Installation Packages	228
Apply all Installation Packages	228
Configure Portal Layouts	229
Upload Portal Logo	229
Upload Phat Banner	229
Upload Client Connected Icon	229
Upload Client Disconnected Icon	229
Upload Client Desktop Icon	229
Upload Error Connected Icon	230
Apply Layout Configuration	230
Query Portal Layout	230
Configure Authentication Parameters	231
Upload RSA Config File	231
Apply Authentication Configuration	231
Query Authentication Configuration	232
Configure SSL VPN Advanced Configuration	233
Apply advanced configuration	233
Query Advanced Configuration	233
Working with Active Clients	234
Query Active Clients	234
Disconnect Active Client	234
Manage Logon and Logoff scripts	235
Upload Script	235
Configure Script Parameters	235
Modify Script Configuration	235
Query Script Configuration	235
Query All Script Configurations	236
Delete Script Configuration	236
Delete All Script Configuragtions	236
Apply All Script Configurations	237
Reconfigure SSL VPN	237
Query SSL VPN Configuration	240
Delete SSL VPN Configuration	243
Query SSL VPN Statistics	243
Working with L2 VPN	244

Configure L2VPN	244
Query L2VPN	246
Query L2VPN Statistics	246
Enable L2VPN	247
Delete L2VPN	247
Working with IPSEC VPN	248
Retrieve IPSec Configuration	249
Retrieve IPSec Statistics	250
Query Tunnel Traffic Statistics	251
Delete IPSec Configuration	252
Managing an NSX Edge	252
Force Sync Edge	252
Redeploy Edge	252
Update DNS Settings	252
Modify AESNI Setting	253
Modify Edge Appliance Core Dump Setting	253
Modify FIPs Setting	253
Modify Log Setting	253
Query Edge Summary	253
Query Edge Status	256
Query Edge Tech Support Logs	257
Manage CLI Credentials and Access	258
You can modify the CLI credentials and enable or disable SSH services for a Edge Edge.	258
Modify CLI Credentials	258
Change CLI Remote Access	258
Manage Auto Configuration Settings	258
Modify Auto Configuration Settings	258
Query Auto Configuration Settings	259
Working with Appliances	259
Query Appliance Configuration	259
Modify Appliance Configuration	260
Change Appliance Size	260
Manage an Appliance	260
Working with Interfaces	262
Add Interfaces or Sub Interfaces	262
Retrieve Interfaces for a Edge	264
Retrieve Specified Interface	266
Modify Specified Interface	267
Delete Interfaces	269
Manage a Edge Interface	269
Retrieve Interface with Specific Index	269
Modify an Interface	270
Delete Interface Configuration	270
Query Interface Statistics	271
Query Statistics for all Interfaces	271
Query Statistics for Uplink Interfaces	271
Query Statistics for Internal Interfaces	272
Query Dashboard Statistics	273

## 9 Firewall Management 275

Configuring Firewall	277
Query Firewall Configuration	277
Filter Firewall Configuration	278
Modify Firewall Configuration	278

Delete Firewall Configuration	281
Working with Firewall Sections	281
Query Firewall Sections	281
Add Firewall Section	283
Modify Firewall Section	284
Delete Firewall Section	286
Working with Firewall Rules	286
Query Firewall Rule	286
Add Firewall Rule	287
Modify Firewall Rule	288
Working with Layer3 Redirect Sections and Rules	289
Query Layer3 Redirect Section	289
Add Layer3 Redirect Section	290
Modify Layer3 Redirect Section	290
Delete Layer3 Redirect Section	291
Query Layer3 Redirect Rules	291
Add Layer3 Redirect Rule	292
Modify Layer3 Redirect Rule	292
Delete Layer3 Redirect Rule	293
Query Service Insertion Profiles	294
Query Status	296
Query Firewall Configuration Status	296
Query Layer3 Section Status	297
Query Layer2 Section Status	297
Working with Memory and CPU Thresholds	298
Configure Thresholds	298
Query Thresholds	299
Tuning Firewall Performance	299
Synchronizing and Enabling Firewall	300
Force Sync Host	300
Force Sync Cluster	300
Enable or Disable APIs for a Cluster	300
Importing and Exporting Firewall Configurations	301
Save a Configuration	301
Query all Saved Configurations	301
Query a Saved Configuration	302
Modify a Saved Configuration	303
Delete a Saved Configuration	303
Export a Saved Configuration	304
Import a Saved Configuration	304
Working with Local Rules	305
Configure Firewall	305
Query Firewall Configuration	306
Query Pre Rules	308
Append Firewall Rules	309
Add a Firewall Rule Above a Specific Rule	309
Query Specific Rule	310
Modify Firewall Rule	310
Delete a Firewall Rule	311
Delete Firewall Configuration	311
Manage Global Firewall Configuration	311
Query Global Firewall Configuration	311
Modify Global Configuration	312
Manage Default Firewall Policy	312
Query Default Firewall Policy	312
Modify Default Firewall Policy	313

Query Firewall Statistics	313
Query Firewall Statistics for Rule	313
Disable Firewall	314
Working with NAT	314
Configure NAT	314
Query NAT Rules for a Edge Edge	315
Delete all NAT Rules	316
Add a NAT Rule above a Specific Rule	316
Append NAT Rules	316
Modify a NAT Rule	317
Delete a NAT Rule	317
Firewall Migration Switch	317
Configuring Fail-Safe Mode for Distributed Firewall	318
Configure Fail-Safe Mode for Firewall	318
Query Fail-Safe Mode Configuration for Firewall	318
Working with SpoofGuard	318
Create SpoofGuard Policy	319
Modify SpoofGuard Policy	319
Query SpoofGuard Policy	320
Query all SpoofGuard Policies	320
Delete SpoofGuard Policy	321
Getting Flow Statistic Details	321
Get Flow Statistics	321
Get Flow Meta-Data	323
Query Flow Summary	324
Query Flow Table	325
Query Flow Details	325
Query Paged Flow Details	325
Query Flow Details Application	326
Query Paged Flow Details Application	326
Flow Exclusion	327
Exclude Flows	327
Query Excluded Flows	328
Working with IPFix	328
Configure IPFix	328
Query IPFix Configuration	329
Delete IPFix Configuration	329
Excluding Virtual Machines from Firewall Protection	329
Add a Virtual Machine to the Exclusion List	330
Get Virtual Machine Exclusion List	330
Delete a Virtual Machine from Exclusion List	330

## 10 Service Composer Management 333

Working with Security Policies	334
Creating a Security Policy	334
Description of Tags	336
Querying Security Policies	337
Edit a Security Policy	340
Delete a Security Policy	341
Export a Security Policy Configuration	341
Import a Security Policy Configuration	341
Query Security Actions for a Security Policy	342
Working with Security Actions	342
Query Virtual Machines for a Security Action	342
Query Security Actions Applicable on a Security Group	343

- Query Security Action Applicable on A Virtual Machine 347
- Query Security Policies Mapped to a Security Group 347
- Query Service Provider Data 348
- Query Security Group Effective Membership 348
- Query Security Groups to which a VM Belongs 348

## 11 Data Security Configuration 351

- Data Security User Roles 351
- Defining a Data Security Policy 352
  - Query Regulations 352
  - Enable a Regulation 352
  - Query Classification Value 353
  - Configure a Customized Regex as a Classification Value 353
  - View the List of Excludable Areas 353
  - Exclude Areas from Policy Inspection 354
  - Specify Security Groups to be Scanned 355
  - Query Security Groups Being Scanned 355
  - Configure File Filters 356
- Saving and Publishing Policies 357
  - Query Saved Policy 357
  - Query Published Policy 358
  - Publish the Updated Policy 358
- Data Security Scanning 358
  - Start, Pause, Resume, or Stop a Scan Operation 359
  - Query Status for a Scan Operation 359
- Querying Scan Results 359
  - Get List of Virtual Machines Being Scanned 359
  - Get Number of Virtual Machines Being Scanned 360
  - Get Summary Information about the Last Five Scans 360
  - Get Information for Virtual Machines Scanned During Previous Scan 361
  - Retrieve Information About Previous Scan Results 361
  - Get XML Representation of Policy Used for Previous Scan 361
- Querying Violation Details 363
  - Get List of Violation Counts 363
  - Get List of Violating Files 364
  - Get List of Violating Files in CSV Format 365
  - Get Violations in Entire Inventory 365

## 12 Activity Monitoring 367

- Data Collection 367
  - Enable Data Collection on a Single Virtual Machine 368
  - Disable Data Collection on a Single Virtual Machine 368
  - Override Data Collection 368
    - Turn On Kill Switch 368
    - Turn Off Kill Switch 369
  - Query Per Virtual Machine Data Collection 369
- Query Resources 370
  - Prerequisites 370
  - View Outbound Activity 370
    - Parameter Values 370
  - View Inbound Activity 371
    - Parameter Values 371
  - View Interaction between Inventory Containers 372
    - Parameter Values 372



View Outbound AD Group Activity	372
Parameter Values	372
Query User Details	373
View Outbound Activity	373
Parameter Values	373
View Inbound Activity	374
Parameter Values	374
View Interaction between Inventory Containers	374
Parameter Values	375
View Outbound AD Group Activity	375
Parameter Values	375
View Virtual Machine Activity Report	376
Parameter Values	376
Query Discovered User Details	377
Working with Domains	378
Register a Domain with NSX Manager	378
Parameter Values for Register/Update Domain	379
Query Domains	379
Delete Domain	380
Working with LDAP Servers	380
Working with EventLog Servers	380
Working with Mapping Lists	381
Working with Activity Monitoring Syslog Support	381
<b>13 Task Framework Management</b>	<b>383</b>
About Task Framework	383
Query Job Instances for Job ID	384
Query Latest Job Instances for Job ID	385
Block REST Thread	385
Query Job Instances by Criterion	385
<b>14 Object IDs</b>	<b>387</b>
Query Datacenter MOID	387
Query Datacenter ID	387
Query Host ID	387
Query Portgroup ID	388
Query VMID	388
<b>15 vShield Endpoint Management</b>	<b>389</b>
Overview of Solution Registration	389
Registering a Solution with vShield Endpoint Service	390
Register a Vendor	390
Register a Solution	390
Altitude of a Solution	390
IP Address and Port for a Solution	390
Activate a Solution	391
Querying Registration Status of vShield Endpoint	391
Get Vendor Registration	391
Get Solution Registration	391
Get IP Address of a Solution	392
Get Activation Status of a Solution	392
Querying Activated Security Virtual Machines for a Solution	392
Query Activated Security Virtual Machines	392
Query Activation Information	393

Unregistering a Solution with vShield Endpoint	393
Unregister a Vendor	393
Unregister a Solution	393
Unset IP Address	394
Deactivate a Solution	394
Status Codes and Error Schema	394
Return Status Codes	394
Error Schema	395

## 16 Deprecated APIs 397

### Appendix A: Schemas 399

Firewall Schemas	399
Firewall Configuration Schema	399
Firewall Section Schema	400
Firewall Sections Schema	401
Deprecated: vShield Manager Global Configuration Schema	401
Deprecated: ESX Host Preparation and Uninstallation Schema	406
Deprecated: vShield App Schemas	407
vShield App Configuration Schema	407
vShield App Firewall Schema	407
vShield App SpoofGuard Schema	410
vShield App Namespace Schema	412
Error Message Schema	413

# 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 repurposed 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 21
- [“NSX Services”](#) on page 22
- [“An Introduction to REST API for NSX Users”](#) on page 24

## 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** All NSX REST requests require authentication. The default NSX Manager login credentials are user admin password default. Unless you changed these, you can use the following basic authentication, where YWRtaW46ZGVmYXVsdA== is the Base 64 encoding of the default credentials admin:default.

Authorization: Basic YWRtaW46ZGVmYXVsdA==

---

### 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.

### To use the REST API in Firefox

- 1 Locate the RESTClient Mozilla add-on, and add it to Firefox.
- 2 Click **Tools > REST Client** to start the add-on.
- 3 Click **Login** and enter the NSX login credentials, which then appear encoded in the Request Header.
- 4 Select a method such as GET, POST, or PUT, and type the URL of a REST API. You might be asked to accept or ignore the lack of SSL certificate. Click **Send**.

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://<vsm-ip>/api/2.0/services/usermgmt/user/admin
```

## Ports Required for NSX REST API

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



# 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 27
- [“User Management”](#) on page 28
- [“Role Management”](#) on page 29

---

**IMPORTANT** All NSX REST requests require authentication. See [“Using the NSX REST API”](#) on page 25 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 2-1. Configure SSO

---

Request:

POST https://<nsxmgr-ip>/api/2.0/services/ssoconfig

Request Body:

```
<ssoconfig>
  <ssoLookupServiceUrl></ssoLookupServiceUrl>
  <ssoAdminUsername></ssoAdminUsername>
  <ssoAdminUserpassword></ssoAdminUserpassword>
</ssoConfig>
```

## Query SSO Details

### Example 2-2. Get SSO details

---

Request:

GET https://<nsxmgr-ip>/api/2.0/services/ssoconfig

Response Body:

```
<ssoConfig>
  <vsmSolutionName></vsmSolutionName>
  <ssoLookupServiceUrl></ssoLookupServiceUrl>
  <ssoAdminUsername></ssoAdminUsername>
</ssoConfig>
```

---

## Query SSO Configuration Status

**Example 2-3.** Get SSO configuration status

---

Request:

GET https://<nsxmgr-ip>/api/2.0/services/ssoconfig/status

Response Body:

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

---

## Delete SSO Configuration

**Example 2-4.** Delete SSO configuration

---

Request:

DELETE https://<nsxmgr-ip>/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 2-5.** Get information about a user

---

Request:

GET https://<nsxmgr-ip>/api/2.0/services/usermgmt/user/<userId>

Request Body:

```
<userInfo>
  <objectId></objectId>
  <type>
    <typeName></typeName>
  </type>
  <name></name>
  <revision></revision>
  <objectTypeName></objectTypeName>
  <userId></userId>
  <fullName></fullName>
  <email></email>
  <isLocal></isLocal>
  <isEnabled></isEnabled>
  <isGroup></isGroup>
  <hasGlobalObjectAccess></hasGlobalObjectAccess>
  <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>
    ...
  </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 2-6. Enable or disable a user account

---

Request:

```
PUT https://<nsxmgr-ip>/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.

## 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 2-7. Remove role assignment

---

Request:

```
DELETE https://<nsxmgr-ip>/api/2.0/services/usermgmt/user/<userId>
```

---

### Example 2-8. Delete a user role

---

Request:

```
DELETE https://<nsxmgr-ip>/api/2.0/services/usermgmt/role/<userId>
```

---

Both APIs return “204 No Content” if successful.<sup>7</sup>

## 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 2-9. Get user role

---

Request:

GET https://<nsxmgr-ip>/api/2.0/services/usermgmt/role/<userId>

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 2-10. Get user role

---

Request:

GET https://<nsxmgr-ip>/api/2.0/services/usermgmt/users/vsm

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<users>
  <userInfo>
    <objectId></objectId>
    <type>
      <typeName></typeName>
    </type><name></name>
    <revision></revision>
    <objectTypeName></objectTypeName>
    <userId></userId>
    <fullname></fullname>
    <email></email>
    <isLocal></isLocal>
    <isEnabled></isEnabled>
    <isGroup>false</isGroup>
    <hasGlobalObjectAccess></hasGlobalObjectAccess>
    <accessControlEntry>
      <role></role>
```

```

    <resource>
      <objectId></objectId>
      <type>
        <typeName></typeName>
      </type>
      <name></name>
      <revision></revision>
      <objectTypeName></objectTypeName>
      <scope>
        <id>group-d1</id>
        <objectTypeName></objectTypeName>
        <name></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 2-11. Update user role

---

Request Header:

POST https://<nsxmgr-ip>/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.

## 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 2-12. Change user role

---

Request Header:

PUT https://<nsxmgr-ip>/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 2-13. Get possible roles

---

Request:

GET https://<nsxmgr-ip>/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 2-14. Get scoping objects

---

Request:

GET https://<nsxmgr-ip>/api/2.0/services/usermgmt/scopingobjects

Response Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<scopingObjects>
  <object>
    <objectId></objectId>
    <type>
      <typeName></typeName>
    </type>
    <name></name>
    <revision></revision>
    <objectTypeName></objectTypeName>
    <scope>
      <id></id>
      <objectTypeName></objectTypeName>
      <name></name>
    </scope>
  </object>
  <object>
    <objectId></objectId>
    <type>
      <typeName></typeName>
    </type>
    <name></name>
    <revision></revision>
    <objectTypeName></objectTypeName>
    <scope>
      <id></id>
      <objectTypeName></objectTypeName>
      <name></name>
    </scope>
  </object>

```



```
</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 2-15. Delete role

---

Request:

```
DELETE https://<nsxmgr-ip>/api/2.0/usermgmt/role/<user Id>
```

---



# 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:

- [“Upgrading the Appliance Manager”](#) on page 35
- [“Configuring NSX Manager with vCenter Server”](#) on page 37
- [“Certificate Management”](#) on page 38
- [“Resource Management”](#) on page 39
- [“Components Management”](#) on page 46
- [“Working with Backup and Restore”](#) on page 48
- [“Working with Tech Support Logs”](#) on page 50
- [“Working with Support Notifications”](#) on page 51

---

**IMPORTANT** All NSX REST requests require authentication. See [“Using the NSX REST API”](#) on page 25 for details about basic authorization.

---

## Upgrading the Appliance Manager

You can upgrade NSX Manager to a later version.

### Upload Upgrade Bundle

**Example 3-1.** Upload upgrade bundle

---

Request:

POST <https://<nsxmgr-ip>/api/1.0/appliance-management/upgrade/uploadbundle/<component-id>>

---

### 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 3-2. Query upgrade information**

---

Request:

GET https://<nsxmgr-ip>/api/1.0/appliance-management/upgrade/uploadbundle/<component-id>

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 3-3. Start upgrade**

---

Request:

GET https://<nsxmgr-ip>/api/1.0/appliance-management/upgrade/start/<component-id>

Response Body:

```

<preUpgradeQuestionsAnswers>
  <preUpgradeQuestionAnswer>
    <questionId></questionId>
    <question></question>
    <questionAnserType></questionAnserType>
    <answer></answer>
  </preUpgradeQuestionAnswer>
  ...
</preUpgradeQuestionsAnswers>

```

---

## Query Upgrade Status

Retrieves upgrade status.

### Example 3-4. Query upgrade status

---

Request:

GET https://<nsxmgr-ip>/api/1.0/appliance-management/status/<component-id>

Response Body:

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

---

## 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 3-5. Synchronize NSX Manager with vCenter server

---

Request:

PUT https://<nsxmgr-ip>/api/2.0/services/vcconfig

Request Body:

```
<vcInfo>
  <ipAddress></ipAddress>
  <userName></userName>
  <password></password>
  <certificateThumbprint></certificateThumbprint>
  <assignRoleToUser></assignRoleToUser>
  <pluginDownloadServer></pluginDownloadServer>
  <pluginDownloadPort></pluginDownloadPort>
</vcInfo>
```

---

## Query Configuration Details

#### Example 3-6. Get vCenter Server configuration details on NSX Manager

---

Request:

GET https://<nsxmgr-ip>/api/2.0/services/vcconfig

Response Body:

```
<vcInfo>
  <ipAddress></ipAddress>
  <userName></userName>
  <certificateThumbprint></certificateThumbprint>
```

```

    <assignRoleToUser></assignRoleToUser>
    <vcInventoryLastUpdateTime></vcInventoryLastUpdateTime>
</vcInfo>

```

---

### Example 3-7. Get default vCenter Server connection status

---

Request:

GET https://<nsxmgr-ip>/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 3-8. Generate CSR

---

Request:

PUT https://<nsxmgr-ip>/api/1.0/appliance-management/certificatemanager/csr/nsx

Request Body:

```

<csr>
  <algorithm></algorithm>
  <keySize></keySize>
  <subjectDto>
    <commonName></commonName>
    <organizationUnit></organizationUnit>
    <organizationName></organizationName>
    <localityName></localityName>
    <stateName></stateName>
    <countryCode></countryCode>
  </subjectDto>
</csr>

```

---

### Download CSR Certificate

Downloads generated CSR from appliance.

#### Example 3-9. Download CSR

---

Request:

GET https://<nsxmgr-ip>/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 3-10. Upload certificate chain**

---

Request:

PUT https://<nsxmgr-ip>/api/1.0/appliance-management/certificatemanager/uploadchain/nsx

---

## Query Certificates

Retrieves certificates.

**Example 3-11. Query certificates**

---

Request:

GET https://<nsxmgr-ip>/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>
    <isValid></isValid>
  </x509certificate>
  ....
</x509Certificates>

```

---

## Upload Keystore File

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

**Example 3-12. Upload file**

---

Request:

POST https://<nsxmgr-ip>/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 3-13. Query global information**

---

Request:

GET https://<nsxmgr-ip>/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 3-14. Query summary

---

#### Request:

GET https://<nsx-ip>/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 3-15. Query global information**

---

Request:

GET https://<nsx-ip>/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 3-16. Reboot appliance**

---

Request:

POST https://<nsxmgr-ip>/api/1.0/appliance-management/system/restart

---

## Query Appliance Manager CPU

**Example 3-17. Query CPU**

---

Request:

GET https://<nsxmgr-ip>/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 3-18. Query uptime

---

Request:

GET https://<nsxmgr-ip>/api/1.0/appliance-management/system/uptime

Response Body

<> days, <> hours, <> minutes

---

## Query Appliance Manager Memory

### Example 3-19. Query memory

---

Request:

GET https://<nsxmgr-ip>/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 3-20. Query storage

---

Request:

GET https://<nsxmgr-ip>/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 3-21. Query network details

---

Request:

GET https://<nsxmgr-ip>/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 3-22. Configure DNS

Request:

PUT https://<nsxmgr-ip>/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 3-23. Configure DNS

Request:

DELETE https://<nsxmgr-ip>/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 3-24. Configure time

Request:

PUT https://<nsxmgr-ip>/api/1.0/appliance-management/system/timesettings

Response 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 3-25. Query time settings

---

Request:

GET https://<nsxmgr-ip>/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 3-26. Delete NTP

---

Request:

DELETE https://<nsxmgr-ip>/api/1.0/appliance-management/system/timesettings/ntp

---

## Working with Locale Settings

### Configure Locale

Configures locale.

### Example 3-27. Configure locale

---

Request:

PUT https://<nsxmgr-ip>/api/1.0/appliance-management/system/locale

Request Body

```

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

```

---

### Query Locale

Retrieves locale information.

### Example 3-28. Query locale

---

Request:

GET https://<nsxmgr-ip>/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 3-29. Configure syslog

---

Request:

PUT https://<nsxmgr-ip>/api/1.0/appliance-management/system/syslogserver

Request Body

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

---

### Query Syslog Servers

Retrieves syslog servers.

#### Example 3-30. Query syslog

---

Request:

GET https://<nsxmgr-ip>/api/1.0/appliance-management/system/syslogserver

Response Body

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

---

### Delete Syslog Servers

Deletes syslog servers.

#### Example 3-31. Delete syslog

---

Request:

DELETE https://<nsxmgr-ip>/api/1.0/appliance-management/system/syslogserver

---

# Components Management

## Query Components

Retrieves all Appliance Manager components.

### Example 3-32. Query components

---

Request:

GET https://<nsxmgr-ip>/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>
    </componentGroup>
  </component>
</components>
```

---

## Query Specific Component

Retrieves details for the specified component ID.

### Example 3-33. Query component

---

Request:

GET https://<nsxmgr-ip>/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 3-34. Query component dependency details

---

Request:

GET https://<nsxmgr-ip>/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 3-35. Query component dependents

---

Request:

GET https://<nsxmgr-ip>/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 3-36. Query component status

---

Request:

GET https://<nsxmgr-ip>/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 3-37. Toggle status

---

Request:

POST https://<nsxmgr-ip>/api/1.0/appliance-management/components/*componentID*/toggleStatus/*command*

---

## 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 83.

## Configure Backup Settings

Configures backup on the Appliance Manager.

### Example 3-38. Configure backup

---

Request:

PUT https://<nsxmgr-ip>/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://<nsxmgr-ip>/1.0/appliance-management/backuprestore/backupsettings/ftpsettings
- **Specify tables that need not be backed up:**  
PUT https://<nsxmgr-ip>/1.0/appliance-management/backuprestore/backupsettings/excludedata
- **Set backup schedule:**  
PUT https://<nsxmgr-ip>/1.0/appliance-management/backuprestore/backupsettings/schedule
- **Delete backup schedule**  
DELETE https://<nsxmgr-ip>/1.0/appliance-management/backuprestore/backupsettings/schedule

## Configure On-Demand Backup

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

### Example 3-39. On-demand backup

---

Request:

```
POST https://<nsxmgr-ip>/api/1.0/appliance-management/backuprestore/backup
```

---

## Query Backup Settings

Retrieves backup settings.

### Example 3-40. Query backup

---

Request:

```
GET https://<nsxmgr-ip>/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 3-41. Delete backup settings

---

Request:

DELETE https://<nsxmgr-ip>/api/1.0/appliance-management/backuprestore/backupsettings

---

## Query Available Backups

Retrieves list of all backups available at configured backup location.

### Example 3-42. Query backup

---

Request:

GET https://<nsxmgr-ip>/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 3-43. Restore data

---

Request:

POST https://<nsxmgr-ip>/api/1.0/appliance-management/backuprestore/restore?restoreFile=*filename*

---

## Working with Tech Support Logs

### Generate Tech Support Logs

Generates tech support logs. Response header contains the location of the created tech support file.

### Example 3-44. Generate tech support log

---

Request:

POST <https://<nsxmgr-ip>/api/1.0/appliance-management/techsupportlogs/componentID>

---

## Download Tech Support Logs

Downloads tech support logs. Response header contains the location of the created tech support file.

### Example 3-45. Generate tech support log

---

Request:

POST <https://<nsxmgr-ip>/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 3-46. Get NSX Manager system events

---

Request:

GET <https://<vsm-ip>/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 3-47. Get NSX Manager audit logs

---

Request:

GET <https://<nsxmgr-ip>/api/2.0/logging/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 3-48. Query notifications**

---

Request:

GET https://<nsxmgr-ip>/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 3-49. Delete notifications**

---

Request:

DELETE https://<nsxmgr-ip>/api/1.0/appliance-management/notifications

---

## Acknowledge Notifications

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

**Example 3-50. Acknowledge notification**

---

Request:

POST https://<nsxmgr-ip>/api/1.0/appliance-management/notifications/*NotificationId*/acknowledge

---

# 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 53
- [“Working with Tags”](#) on page 59
- [“Working with IPsets”](#) on page 60
- [“Working with MACsets”](#) on page 62
- [“Working with Services”](#) on page 64
- [“Working with Service Groups”](#) on page 66
- [“Working with IP Pools”](#) on page 70
- [“Querying Object IDs”](#) on page 74

---

**IMPORTANT** All NSX REST requests require authentication. See [“Using the NSX REST API”](#) on page 25 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 4-1. Create new security group

---

Request:

POST https://<nsxmgr-ip>/api/2.0/services/securitygroup/bulk/<scopeID>

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/>

<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/>
</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/>
</excludeMember>

<excludeMember>
  ...
</excludeMember>

<excludeMember>
  ...
</excludeMember>

<dynamicMemberDefinition>
  <dynamicSet>
    <operator></operator>
    <dynamicCriteria>
      <operator></operator>
      <key></key>
      <criteria></criteria>
      <value></value>
    </dynamicCriteria>
  </dynamicSet>

```

```

    </dynamicCriteria>
    <dynamicCriteria>
    ....
    </dynamicCriteria>

</dynamicSet>

<dynamicSet>
    ....
</dynamicSet>

</dynamicMemberDefinition>

</securitygroup>

```

---

where <scopeID> is globalroot-0 and 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 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 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 a delay of a few seconds between a security group modification and running a GET call on it.

### Example 4-2. Query all security groups on NSX Manager

---

Request:

GET https://<nsxmgr-ip>/api/2.0/services/securitygroup/scope/<scopeID>

Response Body

```

<?xml version="1.0" encoding="UTF-8"?>
<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/>
<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/>
</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/>
</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>

```

---

where <scopeID> is the globalroot-0.

The following command retrieves details for the specified security group:

```
GET https://<nsxmgr-ip>/api/2.0/services/securitygroup/<securityGroupID>
```

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://<nsxmgr-ip>/api/2.0/services/securitygroup/internal/scope/<scopeID>
```

## Query Members for a Scope

You can retrieve a list of applicable member elements that can be added to security groups created on a particular scope. 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 4-3. Get members for a security group scope

---

Request:

```
GET https://<nsxmgr-ip>/api/2.0/services/securitygroup/scope/<scopeID>/memberTypes
```

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 />
    <extendedAttributes />
  </basicinfo>
  ....
</list>

```

```

</basicinfo>
<basicinfo>
...
</basicinfo>
<basicinfo>
...
</basicinfo>
</list>

```

---

where <scopeID> is globalroot-0.

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://<nsxmgr-ip>/api/2.0/services/securitygroup/scope/<scopeID>/members/*memberType*

## Query Security Group Objects

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

### Example 4-4. Query security group members

---

Request:

```

GET https://<nsxmgr-ip>/api/2.0/services/securitygroup/{securityGroupId}/translation/virtualmachines
GET https://<nsxmgr-ip>/api/2.0/services/securitygroup/{securityGroupId}/translation/ipaddresses
GET https://<nsxmgr-ip>/api/2.0/services/securitygroup/{securityGroupId}/translation/macaddresses
GET https://<nsxmgr-ip>/api/2.0/services/securitygroup/{securityGroupId}/translation/vnics

```

---

## Query Security Groups that contain a Virtual Machine

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

### Example 4-5. Query Security Groups that contain a Virtual Machine

---

Request:

GET https://<nsxmgr-ip>/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 4-6. Modify a security group

---

Request:

PUT https://<nsxmgr-ip>/api/2.0/services/securitygroup/bulk/<securitygroup-id>

Request Body:

See [Example 4-1](#).

---

## Delete a Security Group

You can delete an existing security group.

### Example 4-7. Delete a security group

---

Request:

```
DELETE https://<nsxmgr-ip>/api/2.0/services/securitygroup/<securitygroup-id>
```

---

## Add Member to Security Group

You can add a new member to a security group.

### Example 4-8. Add a member to a security group

---

Request:

```
PUT https://<vsm-ip>/api/2.0/services/securitygroup/<securitygroup-id>/members/<member-moref>
```

---

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

## Delete Member from Security Group

This API removes a member from a security group.

### Example 4-9. Delete member from a security group

---

Request:

```
DELETE https://<vsm-ip>/api/2.0/services/securitygroup/<securitygroup-id>/members/<member-moref>
```

---

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

## 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 4-10. Create tag

---

Request:

```
POST https://<nsxmgr-ip>/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/>
</securityTag>
```

---

## Query Security Tags

Retrieves security tags.

### Example 4-11. Query tag

---

Request:

GET https://<nsxmgr-ip>/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>description of the tag</description>
<extendedAttributes/>
</securityTag>
</securityTags>
```

---

## Apply Tag to Virtual Machine

Applies security tag to virtual machine.

### Example 4-12. Apply tag

---

Request:

PUT https://<nsxmgr-ip>/api/2.0/services/securitytags/tag/{TagIdentifierString}/vm/{vmMoid}

---

## Detach Tag from Virtual Machine

Detaches security tag from virtual machine.

### Example 4-13. Detach tag

---

Request:

DELETE https://<nsxmgr-ip>/api/2.0/services/securitytags/tag/{TagIdentifierString}/vm/{vmMoid}

---

## Delete Tag from Virtual Machine

Deletes tags.

### Example 4-14. Delete tag

---

Request:

DELETE https://<nsxmgr-ip>/api/2.0/services/securitytags/tag/{TagIdentifierString}

---

## 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 4-15. Create IPset**

---

Request:

POST https://<nsxmgr-ip>/api/2.0/services/ipset/<scope-moref>

Request Body Example:

```

<ipset>
  <objectId />
  <type>
  <typeName />
  </type>
  <description>
    New Description
  </description>
  <name>TestIPSet2</name>
  <revision>0</revision>
  <objectTypeName />
  <value>10.112.201.8-10.112.201.14</value>
</ipset>

```

---

where <scope-moref> is globalroot-0.

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 4-16. List IPsets on a scope**

---

Request:

GET https://<nsxmgr-ip>/api/2.0/services/ipset/scope/<scope-moref>

---

where <scope-moref> is globalroot-0.

## Query Details of an IPset

You can retrieve details about an IPset.

**Example 4-17. Get details of an IPset**

---

Request:

GET https://<nsxmgr-ip>/api/2.0/services/ipset/<ipset-id>

---

The <ipset-id> 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 4-18. Modify an IPset**

---

Request:

PUT https://<nsxmgr-ip>/api/2.0/services/ipset/<ipset-id>

Request Body Example:

```

<ipset>
  <objectId />

```

```

<type>
<typeName />
</type>
<description>
New Description
</description>
<name>TestIPSet2</name>
<revision>0</revision>
<objectTypeName />
<value>10.112.201.8-10.112.201.21</value>
</ipset>

```

---

The <ipset-id> is as returned by listing the IPset on a scope. In the request body example, the IP address range is doubled.

## 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 4-19. Delete an IPset

---

Request:

```
DELETE https://<nsxmgr-ip>/api/2.0/services/ipset/<ipset-id>?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 4-20. Create MACset on a scope

---

Request:

```
POST https://<nsxmgr-ip>/api/2.0/services/macset/scope/<scope-moref>
```

Request Body Example:

```

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

```

---

where <scope-moref> is globalroot-0. In the request body example, a comma-separated list of MAC addresses is specified.

### List MACsets Created on a Scope

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

**Example 4-21. List MACsets on a scope**

---

Request:

GET https://<nsxmgr-ip>/api/2.0/services/macset/<scope-moref>

---

where <scope-moref> is globalroot-0.

**Get Details of a MACset**

You can retrieve details about a MACset.

**Example 4-22. Get details of a MACset**

---

Request:

GET https://<nsxmgr-ip>/api/2.0/services/macset/<macset-id>

---

The <MACset-id> 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 4-23. Modify details of a MACsets**

---

Request:

PUT https://<nsxmgr-ip>/api/2.0/services/MACset/<MACset-id>

Request Body:

```
<macset>
  <objectId />
  <type>
    <typeName />
  </type>
  <description>Some description</description>
  <name>TestMACSet1</name>
  <revision>1</revision>
  <objectTypeName />
  <value>22:33:44:55:66:77,00:11:22:33:44:55</value>
</macset>
```

---

The <MACset-id> is as returned by listing the MACset on a scope. In the request body example, one MAC address fewer is specified.

**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 4-24. Delete a MACset**

---

Request:

DELETE https://<nsxmgr-ip>/api/2.0/services/macset/<macset-id>

---

## 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 4-25.** List services on a given scope

---

Request:

GET https://<nsxmgr-ip>/api/2.0/services/application/scope/<moref>

---

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

### Add Service to a Scope

You can create a new service on the specified scope.

---

**Example 4-26.** Add a service to a scope

---

Request:

POST https://<nsxmgr-ip>/api/2.0/services/application/<moref>

Request Body:

```
<application>
  <objectId/>
  <type>
    <typeName/>
  </type>
  <description>Some description</description>
  <name>TestApplication1 </name>
  <revision>0</revision>
  <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 <applicationgroup-id> as returned by the call shown in [Example 4-26](#).

---

### Example 4-27. Retrieve details about a service

---

Request:

GET https://<nsxmgr-ip>/api/2.0/services/application/<application-id>

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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>
      AmolDC
    </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 Details

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

---

### Example 4-28. Modify application

---

Request:

PUT https://<nsxmgr-ip>/api/2.0/services/application/<application-id>

Request Body:

```
<application>
  <objectId>Application-1</objectId>
  <type>
    <typeName>Application</typeName>
  </type>
  <description>Some description</description>
  <name>TestApplication</name>
  <revision>2</revision>
  <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 4-29. Delete service

---

Request:

DELETE https://<nsxmgr-ip>/api/2.0/services/application/<application-id>?force=<true|false>

---

## Working with Service Groups

### Add Service Group

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

#### Example 4-30. Add a service group to a scope

---

Request:

POST https://<nsxmgr-ip>/api/2.0/services/applicationgroup/<scope-moref>

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 4-31.** List service groups on a given scope

Request:

GET https://<nsxmgr-ip>/api/2.0/services/applicationgroup/<scope-moref>

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 />
    <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 />
    </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 <applicationgroup-id> as returned by the call shown in [Example 4-26](#).

**Example 4-32.** Retrieve details about a service group

Request:

GET https://<nsxmgr-ip>/api/2.0/services/applicationgroup/<applicationgroup-id>

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 4-33. Modify service group**

---

Request:

PUT https://&lt;nsxmgr-ip&gt;/api/2.0/services/applicationgroup/&lt;applicationgroup-id&gt;

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<applicationGroup>
  <objectId>applicationgroup-1</objectId>
  <type>
    <typeName>ApplicationGroup</typeName>
  </type>
  <name>testglobalAG-updated</name>
  <description>Updated with description</description>
  <revision>2</revision>
  <objectTypeName>ApplicationGroup</objectTypeName>
  <scope>
    <id>globalroot-0</id>
    <objectTypeName>GlobalRoot</objectTypeName>
    <name>Global</name>
  </scope>
  <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 />
  </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 4-34. Delete service group**

---

Request:

DELETE https://&lt;nsxmgr-ip&gt;/api/2.0/services/applicationgroup/&lt;applicationgroup-id&gt;?force=&lt;true|false&gt;

## 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 4-35. Retrieve member elements

---

Request:

GET https://<nsxmgr-ip>/api/2.0/services/applicationgroup/scope/<scope-moref>/members

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 />
  </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 />
  </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 />
  </basicinfo>
</list>
```

---

## Add a Member to the Service Group

You can add a member to the service group.

### Example 4-36. Add member

---

Request:

```
PUT https://<nsxmgr-ip>/api/2.0/services/applicationgroup/<applicationgroup-id>/members/
      <member-moref>
```

---

## Delete a Member from the Service Group

You can delete a member from the service group.

### Example 4-37. Delete member

---

Request:

```
DELETE https://<nsxmgr-ip>/api/2.0/services/applicationgroup/<applicationgroup-id>/members/
      <member-moref>
```

---

## Working with IP Pools

You can create a pool of IP addresses.

### Add an IP Pool

#### Example 4-38. Add IP pool

---

Request:

```
POST https://<nsxmgr-ip>/api/2.0/services/ipam/pools/scope/<scopeId>
```

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>
```

---

where <scopeID> is globalroot-0.

### Query IP Pool Details

Retrieves details about the specified IP pool.

#### Example 4-39. Query IP Pool

---

Request:

```
GET https://<nsxmgr-ip>/api/2.0/services/ipam/pools/<pool-ID>
```

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/>
  <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 4-40. Query IP Pool

---

Request:

PUT https://<nsxmgr-ip>/api/2.0/services/ipam/pools/<pool-ID>

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/>
  <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 4-41. Allocate new address

---

Request:

POST https://<nsxmgr-ip>/api/2.0/services/ipam/pools/<pool-ID>/ipaddresses

Request Body:

```
<ipAddressRequest>
  <allocationMode>ALLOCATE</allocationMode>
</ipAddressRequest>
Response Body:
<?xml version="1.0" encoding="UTF-8"?>
<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 4-42. Allocate new address

---

Request:

POST https://<nsxmgr-ip>/api/2.0/services/ipam/pools/<pool-ID>/ipaddresses

Request Body:

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

Response Body:

See [Example 4-41](#).

---

## Query all IP Pools on Scope

Retrieves all IP pools on the specified scope.

### Example 4-43. Query IP pools on scope

---

Request:

GET https://<nsxmgr-ip>/api/2.0/services/ipam/pools/<pool-ID>/ipaddresses

Response Body:

```
<ipamAddressPools>
  <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/>
<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>
<ipPoolType>IPv4</ipPoolType>
<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>
<subnetId>subnet-1</subnetId>
</ipamAddressPool>
</ipamAddressPools>

```

---

## Query Allocated IP Addresses

Retrieves all allocated IP addresses from the specified pool.

### Example 4-44. Query allocated addresses

---

Request:

GET https://<nsxmgr-ip>/api/2.0/services/ipam/pools/<pool-ID>/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 4-45. Release IP address

---

Request:

DELETE https://<nsxmgr-ip>/api/2.0/services/ipam/pools/<pool-ID>/ipaddresses/<allocated-ip-address>

---

## Delete an IP Pool

### Example 4-46. Delete IP Pool

---

Request:

DELETE https://<nsxmgr-ip>/api/2.0/services/ipam/pools/<pool-ID>

---

## Querying Object IDs

This section describes how to retrieve the IDs for the objects in your virtual inventory.

### Query Datacenter MOID

- 1 In a web browser, type the following:

http://<vCenter-IP>/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>/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>/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>/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.



# Installing NSX Components

---

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

This chapter includes the following topics:

- [“Installing Licenses”](#) on page 77
- [“Working with Network Virtualization Components”](#) on page 78
- [“Working with VXLAN for Logical Switches”](#) on page 79
- [“Working with Services”](#) on page 92
- [“Working with Conflicting Agencies”](#) on page 99
- [“Uninstalling Services”](#) on page 100

---

**IMPORTANT** All NSX REST requests require authentication. See [“Using the NSX REST API”](#) on page 25 for details about basic authorization.

---

## Installing Licenses

You can install and assign an NSX for vSphere license after NSX Manager installation is complete by using the vSphere Web Client.

Before purchasing and activating an NSX for vSphere license, you can install and run the software in evaluation mode. When run in evaluation mode, intended for demonstration and evaluation purposes, NSX components are completely operational immediately after installation, do not require any licensing configuration, and provide full functionality for 60 days from the time you first activate them.

- 1 Log in to the vSphere Web Client.
- 2 Click **Administration** and then click **Licenses**.
- 3 Click the **Solutions** tab.
- 4 From the drop-down menu at the top, select **Assign a new license key**.
- 5 Type the license key and an optional label for the new key.
- 6 Click **Decode**.  
Decode the license key to verify that it is in the correct format, and that it has enough capacity to license the assets.
- 7 Click **OK**.

### What to do next

Obtain and install an NSX for vSphere license within the evaluation period.

## 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 VLAN's.

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 vmknicpg 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 5-1. Install network virtualization

---

Request

POST https://<nsxmgr-ip>/api/2.0/nwfabric/configure

Request Body

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

---

### Upgrade Network Virtualization Components

After NSX Manager is upgraded to NSX Manager, previously prepared clusters must have the 6.0 network virtualization components installed.

#### Example 5-2. Upgrade network virtualization

---

Request

PUT https://<nsxmgr-ip>/api/2.0/nwfabric/configure

Request Body

See [Example 5-1](#).

---

## Delete Network Virtualization Components

Remove 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 5-3. Delete network virtualization

Request

DELETE https://<nsxmgr-ip>/api/2.0/nwfabric/configure

## 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 78.
- 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 cluster Id or resourcePool Id.

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.

#### Example 5-4. Add controller

---

Request

POST <https://<nsxmgr-ip>/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 5-5. Query controllers

---

Request

GET <https://<nsxmgr-ip>/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 5-6. Query controller addition or deletion details

---

Request

GET https://<nsxmgr-ip>/api/2.0/vdn/controller/progress/<job\_id>

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 5-7. Query controller logs

---

Request

GET https://<nsxmgr-ip>/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 5-8. Delete controller

---

Request

DELETE https://<nsxmgr-ip>/api/2.0/vdn/controller/<controller-id>?forceRemoval=<true/false>

---

## Query Cluster Information

Retrieves cluster wise configuration information for controller.

### Example 5-9. Query cluster details

---

Request

GET https://<nsxmgr-ip>/api/2.0/vdn/controller/cluster

Response Body:

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

---

## Modify Cluster Configuration

Modifies cluster wise configuration information for controller.

**Example 5-10. Modify cluster configuration**

---

**Request**

PUT https://<nsxmgr-ip>/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 5-11. Add controller syslog exporter**

---

**Request**

POST https://<nsxmgr-ip>/api/2.0/vdn/controller/{controller-id}/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 5-12. Query controller syslog exporter**

---

**Request**

GET https://<nsxmgr-ip>/api/2.0/vdn/controller/{controller-id}/syslog

**Response Body:**

```
<?xml version="1.0" encoding="UTF-8"?>
<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 5-13. Delete controller syslog exporter**

---

**Request**

DELETE https://<nsxmgr-ip>/api/2.0/vdn/controller/{controller-id}/syslog

---

## Backup Controller Data

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

### Example 5-14. Backup controller data

---

Request:

GET

`https://NSXManagerIPAddress/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://NSXManagerIPAddress/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 5-15. Add a segment ID range

---

Request:

POST `https://<vsm-ip>/api/2.0/vdn/config/segments`

Request Body:

```
<segmentRanges>
  <segmentRange>
    <id>1</id>
    <name>name</name>
    <desc>desc</desc>
    <begin>1000</begin>
    <end>1500</end>
  </segmentRange>
  <segmentRange>
    ....
  </segmentRange>
  ....
</segmentRanges>
```

---

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 5-16. Get all Segment ID Ranges**

---

Request:

GET https://<vsm-ip>/api/2.0/vdn/config/segments

Response Body:

```
<segmentRanges>
  <segmentRange>
    <id>1</id>
    <name>name</name>
    <desc>desc</desc>
    <begin>5000</begin>
    <end>9000</end>
  </segmentRange>
  <segmentRange>
    ....
  </segmentRange>
</segmentRanges>
```

---

**Query a Specific Segment ID Range**

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

**Example 5-17. Get a specific Segment ID Range**

---

Request:

GET https://<vsm-ip>/api/2.0/vdn/config/segments/*SegmentID*

Response Body:

```
<segmentRange>
  <id>1</id>
  <name>name</name>
  <desc>desc</desc>
  <begin>10000</begin>
  <end>11000</end>
</segmentRange>
```

---

**Update a Segment ID Range**

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

**Example 5-18. Update a Segment ID Range**

---

Request:

PUT https://<vsm-ip>/api/2.0/vdn/config/segments/*SegmentID*

Request Body:

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

---

**Delete a Segment ID Range**

You can delete a segment ID range.

**Example 5-19. Delete a Segment ID Range**

---

Request:

DELETE https://<vsm-ip>/api/2.0/vdn/config/segments/*SegmentID*

---

## Configure VXLAN

### Example 5-20. Install VXLAN

---

Request

POST https://<nsxmgr-ip>/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_LOADBASE
            D|LOADBALANCE_SRCID|LOADBALANCE_SRCMAC|LACP_V2 -->
      <teaming>ETHER_CHANNEL</teaming>
    </configSpec>
  </resourceConfig>
</nwFabricFeatureConfig>
```

---

## Install VXLAN

### Example 5-21. Install VXLAN with LACPv2

---

Request

POST https://<nsxmgr-ip>/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>
      <mtu>1600</mtu>
      <teaming>LACP_V2</teaming>
      <!-- uplinkPortName should be as specified in vCenter. -->
      <uplinkPortName>{LAG NAME}</uplinkPortName>
    </configSpec>
  </resourceConfig>
</nwFabricFeatureConfig>
```

```

</configSpec>
</resourceConfig>
</nwFabricFeatureConfig>

```

---

## Delete VXLAN

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

### Example 5-22. Delete VXLAN

---

Request

```
DELETE https://<nsxmgr-ip>/api/2.0/nwfabric/configure
```

---

## Delete VXLAN with vdsContext

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

### Example 5-23. Delete VXLAN

---

Request

```
DELETE https://<nsxmgr-ip>/api/2.0/nwfabric/configure
```

---

## 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 5-24. Create a network scope

---

Request:

```
POST https://<vsm-ip>/api/2.0/vdn/scopes
```

Request Body:

```

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

```

---

### Edit a Network Scope

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

### Example 5-25. Create a network scope

---

Request:

```
POST https://<vsm-ip>/api/2.0/vdn/scopes/scopeID?action=patch
```

Request Body:

```

<vdnScope>
  <objectId>{id}</objectId>

```

```

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

```

---

## Update Attributes on a Network Scope

You can update the attributes of a network scope.

### Example 5-26. Update attributes of a network scope

---

Request:

PUT https://<vsm-ip>/api/2.0/vdn/scopes/scopeID/attributes

Request Body:

```

<vdnScope>
  <objectId>vdnScope-1</objectId>
  <name>new name</name>
  <description>new description</description>
</vdnScope>

```

---

## Query existing Network Scopes

You can retrieve all existing network scopes.

### Example 5-27. Get all network scopes

---

Request:

GET https://<vsm-ip>/api/2.0/vdn/scopes

Response Body:

```

<vdnScopes>
  <vdnScope>
    <objectId>vdnscope-2</objectId>
    <type><typeName>VdnScope</typeName></type>
    <name>My Name</name>
    <description>My Description</description>
    <revision>0</revision>
    <objectTypeName>VdnScope</objectTypeName>
    <extendedAttributes/>
    <id>vdnscope-2</id>
    <clusters>
      <cluster>
        <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/>
        </cluster>
      </cluster>
    ...
  </clusters>
  <virtualWireCount>10</virtualWireCount>
</vdnScope>
...
<vdnScope>...</vdnScope>
...
</vdnScopes>

```

---

## Query a Specific Network Scope

You can retrieve a specific network scope.

### Example 5-28. Get a network scope

---

Request:

GET https://<vsm-ip>/api/2.0/vdn/scopes/*scopeID*

Response Body:

```
<vdnScope>
<objectId>vdnscope-2</objectId>
<type><typeName>VdnScope</typeName></type>
<name>My Name</name>
<description>My description</description>
<revision>0</revision>
<objectTypeName>VdnScope</objectTypeName>
<extendedAttributes/>
<id>vdnscope-2</id>
<clusters>
<cluster>
<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/>
</cluster>
</cluster>
...
</clusters>
<virtualWireCount>10</virtualWireCount>
</vdnScope>
```

---

## Delete a Network Scope

You can delete a network scope.

### Example 5-29. Delete network scope

---

Request:

DELETE https://<vsm-ip>/api/2.0/vdn/scopes/*scopeID*

---

## Reset Communication

Resets communication between NSX Manager and a host or cluster.

### Example 5-30. Reset communication

---

Request

POST https://<nsxmgr-ip>/api/2.0/nwfabric/configure?action=synchronize

---

## Query Features on Cluster

Retrieves all features available on the cluster.

### Example 5-31. Query features

---

Request



POST https://<nsxmgr-ip>/api/2.0/nwfabric/features

Response Body:

```
<featureInfos>
  <!-- Contains multiple featureInfo -->
  <featureInfo>
    <name>{FEATURE NAME}</name>
    <featureId>{FEATURE ID}</featureId>
    <version>{FEATURE VERSION}</version>
  </featureInfo>
</featureInfos>
```

---

## Query Status of Specific Resources

### Example 5-32. Query status

---

Request

GET https://<nsxmgr-ip>/api/2.0/nwfabric/status?resource=<RESOURCE ID>

Response Body:

```
<resourceStatuses>
  <resourceStatus>
    <resource>
      <objectId>{resource id}</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/>
    </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 5-33. Query status

---

#### Request

GET https://<nsxmgr-ip>/api/2.0/nwfabric/status/child/<PARENT RESOURCE ID>

#### 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>ClusterComputeResource</objectTypeName>
        <name>c-1</name>
      </scope>
      <clientHandle>
      </clientHandle>
      <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 Resources by Criterion

### Example 5-34. Query status

---

#### Request

GET https://<nsxmgr-ip>/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>
      </clientHandle>
      <extendedAttributes/>
    </resource>
    <nwFabricFeatureStatus>
      <featureId>com.vmware.vshield.nsxmgr.nwfabric.hostPrep</featureId>
      <featureVersion>5.5</featureVersion>
      <updateAvailable>false</updateAvailable>
      <status>RED</status>
      <message>
        <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>

```

```

</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/>
  </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>

```

---

## 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 5-35. Install service

---

Request

POST https://<nsxmgr-ip>/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 5-36. Deploy service

Request

POST https://<nsxmgr-ip>/api/2.0/si/deploy

## Identify Service Dependency

Lists the service on which the specified service depends on.

### Example 5-37. Identify service dependency

Request

GET https://<nsxmgr-ip>/api/2.0/si/deploy/service/<service-id>/dependsOn

## Uninstall Service Dependency

Lists the service on which the specified service depends on.

### Example 5-38. Uninstall service dependency

Request

DELETE https://<nsxmgr-ip>/api/2.0/si/deploy/clutser/<cluster-id>

If you try to remove a service on which a service depends on and it is already installed, the un-installation 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 5-39. Query services

Request

GET https://<nsxmgr-ip>/api/2.0/si/deploy/cluster/<cluster-id>

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/>
</cluster>
<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/>
</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/>
</dvPortGroup>
<serviceStatus>SUCCEEDED</serviceStatus>
</deployedService>
</deployedServices>

```

---

## Query Details about a Service

Retrieves detailed information about the service.

### Example 5-40. Query service

---

Request

GET https://<nsxmgr-ip>/api/2.0/si/deploy/cluster/<cluster-id>/service/<service-id>

Response Body

See [Example 5-39](#).

---

## Query Clusters

Retrieves all clusters on which the specified service is installed.

### Example 5-41. Query clusters

---

Request

GET https://<nsxmgr-ip>/api/2.0/si/deploy/service/<service-id>

Response Body

See [Example 5-39](#).

## Upgrade Service

Upgrades service to recent version.

### Example 5-42. Query clusters

#### Request

PUT https://<nsxmgr-ip>/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>{dvpg ID}</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.

## 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 5-43. Query agents on host

#### Request

GET https://<nsxmgr-ip>/api/2.0/si/host/<host-id>/agents

#### Response Body

```
<fabricAgents>
  <agent>
    <agentId>nsxmragent-1</agentId>
    <agentName>agent name if present</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>
    </allocatedIpAddress>
  </agent>
</fabricAgents>
```



```

    <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/>
  </hostInfo>
  <initialData>partner data if present</initialData>
</agent>
</fabricAgents>

```

---

## Query Agent Information

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

### Example 5-44. Query agent details

---

#### Request

GET https://<nsxmgr-ip>/api/2.0/si/agent/<agent-id>

#### Response Body

```

<agent>
  <agentId>nsxmgragent-1</agentId>
  <agentName>agent name if present</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/>
  </hostInfo>
  <initialData>partner data if present</initialData>
</agent>

```

---

## Query Agents for Deployment

Retrieves all agents for the specified deployment.

### Example 5-45. Query agents for deployment

---

#### Request

GET https://<nsxmgr-ip>/api/2.0/si/deployment/<deploymentunit-id>/agents

#### Response Body

```

<fabricAgents>
  <agent>
    <agentId>nsxmragent-1</agentId>
    <agentName>agent name if present</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>
  </agent>
  <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>
  </hostInfo>

```

```

    <scope>
      <id>domain-c7</id>
      <objectTypeName>ClusterComputeResource</objectTypeName>
      <name>Kaustubh-CL</name>
    </scope>
    <clientHandle>
    </clientHandle>
    <extendedAttributes/>
  </hostInfo>
  <initialData>partner data if present</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 5-46. Query conflicts

---

##### Request

GET https://<nsxmgr-ip>/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 5-47. Query conflicts

---

##### Request

PUT https://<nsxmgr-ip>/api/2.0/si/fabric/sync/conflicts

##### Request Body

```

<conflictResolverInfo>
  <agencyAction>RESTORE</agencyAction>

```

---

```
</conflictResolverInfo>
```

---

## Delete Conflicting Agencies

Deletes conflicting EAM Agencies.

### Example 5-48. Delete conflicts

---

Request

PUT https://<nsxmgr-ip>/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 5-49. Query conflicts

---

Request

PUT https://<nsxmgr-ip>/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 5-50. Uninstall services from a cluster

---

Request:

DELETE https://<vsm-ip>/api/2.0/si/deploy/cluster/<cluster-id>?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 5-51. Uninstall specified service from specified clusters

---

Request:

```
DELETE
    https://<vsm-ip>/api/2.0/si/deploy/service/<service-id>?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.



## 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 25 for details about basic authorization.

This chapter includes the following topics:

- [“Preparing for Logical Switches”](#) on page 104
- [“Configuring Switches”](#) on page 104
- [“Working with Segment IDs”](#) on page 106
- [“Working with Multicast Address Ranges”](#) on page 107
- [“Working with Network Scopes”](#) on page 109
- [“Working with Logical Switches”](#) on page 111
- [“Managing the Logical Switch UDP Port”](#) on page 115
- [“Querying Allocated Resources”](#) on page 115
- [“Testing Multicast Group Connectivity”](#) on page 116

- [“Performing Ping Test”](#) on page 117

## 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 6-1. Prepare switch

---

Request:

POST https://<vsm-ip>/api/2.0/vdn/switches

Request Body:

```
<vdsContext>
  <switch>
    <objectId>dvs-26</objectId>
    <type><typeName>DistributedVirtualSwitch</typeName></type>
    <name>My 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 6-2. Get all configured switches

---

Request:

GET https://<vsm-ip>/api/2.0/vdn/switches

Response Body:

```
<vdsContexts>
  <vdsContext>
    <switch>
      <objectId>dvs-26</objectId>
      <type><typeName>DistributedVirtualSwitch</typeName></type>
      <name>My Name</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 6-3. Get configured switches on a datacenter

---

Request:

GET https://<vsm-ip>/api/2.0/vdn/switches/datacenter/datacenterID

Response Body:

```
<vdsContexts>
  <vdsContext>
    <switch>
      <objectId>dvs-26</objectId>
      <type><typeName>DistributedVirtualSwitch</typeName></type>
      <name>My Name</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 6-4. Get specific switch

---

Request:

GET https://<vsm-ip>/api/2.0/vdn/switches/switchID

Response Body:

```
<vdsContext>
  <switch>
    <objectId>dvs-26</objectId>
    <type><typeName>DistributedVirtualSwitch</typeName></type>
    <name>My Name</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 6-5. Delete switch**

---

Request:

DELETE https://<vsm-ip>/api/2.0/vdn/switches/*switchID*

---

## Working with Segment IDs

You can specify a segment ID pool to isolate your network traffic.

### Add a new Segment ID Range

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

**Example 6-6. Add a segment ID range**

---

Request:

POST https://&lt;vsm-ip&gt;/api/2.0/vdn/config/segments

Request Body:

```

<segmentRanges>
  <segmentRange>
    <id>1</id>
    <name>name</name>
    <desc>desc</desc>
    <begin>1000</begin>
    <end>1500</end>
  </segmentRange>
  <segmentRange>
    ....
  </segmentRange>
  ....
</segmentRanges>

```

---

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 6-7. Get all Segment ID Ranges**

---

Request:

GET https://&lt;vsm-ip&gt;/api/2.0/vdn/config/segments

Response Body:

```

<segmentRanges>
  <segmentRange>
    <id>1</id>
    <name>name</name>
    <desc>desc</desc>
    <begin>5000</begin>
    <end>9000</end>
  </segmentRange>
  <segmentRange>
    ....
  </segmentRange>
</segmentRanges>

```

---

## Query a Specific Segment ID Range

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

### Example 6-8. Get a specific Segment ID Range

---

Request:

GET https://<vsm-ip>/api/2.0/vdn/config/segments/*SegmentID*

Response Body:

```
<segmentRange>
  <id>1</id>
  <name>name</name>
  <desc>desc</desc>
  <begin>10000</begin>
  <end>11000</end>
</segmentRange>
```

---

## Update a Segment ID Range

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

### Example 6-9. Update a Segment ID Range

---

Request:

PUT https://<vsm-ip>/api/2.0/vdn/config/segments/*SegmentID*

Request Body:

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

---

## Delete a Segment ID Range

You can delete a segment ID range.

### Example 6-10. Delete a Segment ID Range

---

Request:

DELETE https://<vsm-ip>/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 6-11. Add a multicast address range

---

Request:

POST https://<vsm-ip>/api/2.0/vdn/config/multicasts

Request Body:

```

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

```

---

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 6-12. Get all multicast ranges

---

Request:

GET https://<vsm-ip>/api/2.0/vdn/config/multicasts

Response Body:

```

<multicastRanges>
  <multicastRange>
    <id>1</id>
    <name>name</name>
    <desc>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 6-13. Get a multicast range

---

Request:

GET https://<vsm-ip>/api/2.0/vdn/config/multicasts/*multicastAddressRangeID*

Response Body:

```

<multicastRange>
  <id>1</id>
  <name>name</name>
  <desc>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 6-14. Update a multicast range

---

Request Header:

PUT https://<vsm-ip>/api/2.0/vdn/config/multicasts/*multicastAddressRangeID*

Request Body:

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

---

## Delete a Multicast Address Range

You can delete a multicast address range.

### Example 6-15. Delete multicast address range

---

Request:

DELETE https://<vsm-ip>/api/2.0/vdn/config/multicasts/<multicasts/multicasts/  
*multicastAddressRangeID*

---

## 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 6-16. Create a network scope

---

Request:

POST https://<vsm-ip>/api/2.0/vdn/scopes

Request Body:

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

---

### Edit a Network Scope

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

#### Example 6-17. Create a network scope

---

Request:

POST https://<vsm-ip>/api/2.0/vdn/scopes/*scopeID*?action=patch

Request Body:

```

<vdnScope>
  <objectId>{id}</objectId>
  <clusters>
    <cluster><cluster><objectId>domain-c59</objectId></cluster></cluster>
  </clusters>
</vdnScope>

```

---

## Update Attributes on a Network Scope

You can update the attributes of a network scope.

### Example 6-18. Update attributes of a network scope

---

Request:

PUT https://<vsm-ip>/api/2.0/vdn/scopes/*scopeID*/attributes

Request Body:

```

<vdnScope>
  <objectId>vdnScope-1</objectId>
  <name>new name</name>
  <description>new description</description>
</vdnScope>

```

---

## Query existing Network Scopes

You can retrieve all existing network scopes.

### Example 6-19. Get all network scopes

---

Request:

GET https://<vsm-ip>/api/2.0/vdn/scopes

Response Body:

```

<vdnScopes>
  <vdnScope>
    <objectId>vdnscope-2</objectId>
    <type><typeName>VdnScope</typeName></type>
    <name>My Name</name>
    <description>My Description</description>
    <revision>0</revision>
    <objectTypeName>VdnScope</objectTypeName>
    <extendedAttributes/>
    <id>vdnscope-2</id>
    <clusters>
      <cluster>
        <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/>
        </cluster>
      </cluster>
    ...
  </clusters>
  <virtualWireCount>10</virtualWireCount>
</vdnScope>
...
<vdnScope>...</vdnScope>
...

```

```
</vdmScopes>
```

---

## Query a Specific Network Scope

You can retrieve a specific network scope.

### Example 6-20. Get a network scope

---

Request:

```
GET https://<vsm-ip>/api/2.0/vdn/scopes/scopeID
```

Response Body:

```
<vdmScope>
<objectId>vdmnscope-2</objectId>
<type><typeName>VdmScope</typeName></type>
<name>My Name</name>
<description>My description</description>
<revision>0</revision>
<objectTypeName>VdmScope</objectTypeName>
<extendedAttributes/>
<id>vdmnscope-2</id>
<clusters>
<cluster>
<cluster>
<objectId>domain-c124</objectId>
<type><typeName>ClusterComputeResource</typeName></type>
<name>vmlan-cluster</name>
<scope><id>datacenter-2</id><objectTypeName>Datacenter</objectTypeName><name>dc1</name></scope>
<extendedAttributes/>
</cluster>
</cluster>
...
</clusters>
<virtualWireCount>10</virtualWireCount>
</vdmScope>
```

---

## Delete a Network Scope

You can delete a network scope.

### Example 6-21. Delete network scope

---

Request:

```
DELETE https://<vsm-ip>/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 network scope.

### Create a Logical Switch

You can create a new logical switch on the specified network scope. You must have defined a segment ID range and a multicast address range before creating a .

The default value of the controlPlaneMode parameter is the value specified for the transport zone.

**Example 6-22. Create a logical switch**

---

Request:

POST https://&lt;vsm-ip&gt;/api/2.0/vdn/scopes/&lt;scopeID&gt;/virtualwires

Request Body:

```

<virtualWireCreateSpec>
  <name>logical switch name</name>
  <description>logical switch description</description>
  <tenantId>logical switch tenant</tenantId>
  <controlPlaneMode>UNICAST_MODE</controlPlaneMode>...<!-- Optional. Default is the value specified for the transport
    zone. -->
</virtualWireCreateSpec>

```

---

**Query all Logical Switches on a Network Scope**

You can retrieve all logical switches on the specified network scope.

**Example 6-23. Get all logical switches**

---

Request:

GET https://&lt;vsm-ip&gt;/api/2.0/vdn/scopes/&lt;scopeID&gt;/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>
    </datapart>
    ....
    <datapart class="virtualWire">
      ....
    </datapart>
  </sortedDataPage>
  <pagingInfo>
    <pageSize>20</pageSize>
    <startIndex>0</startIndex>
    <totalCount>3</totalCount>
    <sortOrderAscending>false</sortOrderAscending>
  </pagingInfo>
</virtualWires>

```

---

**Query all Logical Switches on all Network Scopes**

You can retrieve all logical switches across all network scopes.

**Example 6-24. Get all logical switches on all network scopes**

---

Request:



GET https://<vsm-ip>/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>
    </datapart> ....
    <datapart class="virtualWire"> ....
  </datapart>
  <pagingInfo>
    <pageSize>20</pageSize>
    <startIndex>0</startIndex>
    <totalCount>3</totalCount>
    <sortOrderAscending>false</sortOrderAscending>
  </pagingInfo>
</sortedDataPage>
</virtualWires>
```

---

## Query a Specific Logical Switch

You can retrieve the definition for a logical switch.

**Example 6-25.** Get a logical switch definition

---

Request:

GET https://<vsm-ip>/api/2.0/vdn/virtualwires/*virtualWireID*

Response Body:

```
<virtualWire>
  <name>Test logical switch</name>
  <description>Test logical switch Description</description>
  <objectId>virtualwire-4</objectId>
  <vdnScopeId>vdnscope-3</vdnScopeId>
  <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>
</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 6-26. Delete logical switch

---

Request:

```
DELETE https://<vsm-ip>/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 6-27. Enable ARP suppression and MAC learning

---

Request:

```
PUT https://<vsm-ip>/api/2.0/xvs/networks/<dvpg moid>|<virtualWire-Id>/features
```

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<networkFeatureConfig>
  <ipDiscoveryConfig>
    <enabled>true</enabled>
  </ipDiscoveryConfig>
  <macLearningConfig>
    <enabled>false</enabled>
  </macLearningConfig>
</networkFeatureConfig>
```

---

### Example 6-28. Disable ARP suppression and MAC learning

---

Request:

```
PUT https://<vsm-ip>/api/2.0/xvs/networks/<dvpg moid>|<virtualWire-Id>/features
```

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<networkFeatureConfig>
  <ipDiscoveryConfig>
    <enabled>false</enabled>
  </ipDiscoveryConfig>
  <macLearningConfig>
    <enabled>false</enabled>
  </macLearningConfig>
</networkFeatureConfig>

```

---

#### Example 6-29. Query ARP suppression and MAC learning

---

Request:

GET https://<vsm-ip>/api/2.0/xvs/networks/<dvpg moid>|<virtualWire-Id>/features

Response Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<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 6-30. Get UDP port

---

Request:

Get https://<vsm-ip>/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 6-31. Change UDP port

---

Request:

PUT https://<vsm-ip>/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 6-32. Get resources

---

Get segment IDs allocated to s:

GET https://<vsm-ip>/api/2.0/vdn/config/resources/allocated?type=segmentId&pagesize={ pageSize }&startindex={ startIndex }

Get multicast address range allocated to s:

GET https://<vsm-ip>/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 network scope or .

### Test Multicast Group Connectivity in a Network Scope

**Example 6-33.** Test multicast group connectivity in network scope

---

Request:

PUT https://<vsm-ip>/api/2.0/vdn/scopes/ScopeID/conn-check/multicast

Request Body:

```
<testParameters>
  <gateway>172.23.233.1</gateway>
  <packetSize>1600</packetSize>
  <expectedResponse>5</expectedResponse>
  <returnHopCount>true</returnHopCount>
  <returnRecordIp>true</returnRecordIp>
  <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 6-34.** Test multicast group connectivity in logical switch

---

Request:

PUT https://<vsm-ip>/api/2.0/vdn/scopes/virtualWireID/conn-check/multicast

Request Body:

```
<testParameters>
  <gateway>172.23.233.1</gateway>
  <packetSize>1600</packetSize>
  <expectedResponse>5</expectedResponse>
  <returnHopCount>true</returnHopCount>
  <returnRecordIp>true</returnRecordIp>
  <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>
```

```

<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 spans.

### Example 6-35. Perform point to point test

---

Request:

PUT https://<vsm-ip>/api/2.0/vdn/virtualwires/*virtualWireID*/conn-check/p2p

Request Body:

```

<testParameters>
  <gateway>172.23.233.1</gateway>
  <packetSize>1600</packetSize>
  <expectedResponse>5</expectedResponse>
  <returnHopCount>true</returnHopCount>
  <returnRecordIp>true</returnRecordIp>
  <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 Installation and Management

# 7

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 [“Querying Object IDs”](#) on page 74.

This chapter includes the following topics:

- [“Installing a Logical Router”](#) on page 119
- [“Query a Logical Router”](#) on page 120
- [“Modify a Router”](#) on page 122
- [“Deleting a Router”](#) on page 122

---

**IMPORTANT** All NSX REST requests require authentication. See [“Using the NSX REST API”](#) on page 25 for details about basic authorization.

---

## 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 78.

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.

### Example 7-1. Install a logical router

---

Request:

POST https://<nsxmgr-ip>/api/4.0/edges

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. 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>
<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>

```

---

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

## Query a Logical Router

Retrieves information about the specified router.

### Example 7-2. Query a router

---

Request:

GET https://<nsxmgr-ip>/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>

```

---

## Modify a Router

Replaces the configuration of the specified router.

### Example 7-3. Modify router

---

Request:

PUT https://<nsxmgr-ip>/api/4.0/edges/{edgeId}

Request Body:

See [Example 7-1](#).

---

## Deleting a Router

You can delete a logical router instance. Appliances associated with the router instance are deleted as well.

### Example 7-4. Delete a router

---

Request:

DELETE https://<nsxmgr-ip>/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 7-5. Configure management interfaces**

---

Request:

PUT https://<nsxmgr-ip>/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 7-6. Query interfaces**

---

Request:

GET https://<nsxmgr-ip>/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 Edge instance, the specified configuration is applied to the appliance as well.

You should not define a 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 7-7. Add an interface**

---

Request:

POST https://<nsxmgr-ip>/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 7-8.** Retrieve all interfaces**Request:**

GET https://<nsxmgr-ip>/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>interface 10</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 7-9. Delete interface

---

Request:

```
DELETE https://<nsxmgr-ip>/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 7-10. Delete all interfaces

---

Request:

```
DELETE https://<nsxmgr-ip>/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 Edge.

#### Example 7-11. Get interface with specific index

---

Request:

GET https://<nsxmgr-ip>/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 7-12. Modify interface

---

Request:

PUT https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/interfaces/<index>

Response 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 7-13. Delete interface configuration**

Request:

DELETE https://&lt;nsxmgr-ip&gt;/api/4.0/edges/&lt;edgeId&gt;/interfaces/index

## Configure Routes

Configures globalConfig, staticRouting, OSPF, BGP, and IS-IS routes.

**Example 7-14. Configure routes**

Request:

PUT https://&lt;nsxmgr-ip&gt;/api/4.0/edges/&lt;edgeId&gt;/routing/config

Request Body:

```

<routing>
  <routingGlobalConfig>
    <routerId>1.1.1.1</routerId> <!-- Required when dynamic routing protocols like OSPF, BGP, IS-IS 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>3.1.1.0/22</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>
      <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 -->
  </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>
<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 -->
    <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, BGP, and ISIS configurations.

### Example 7-15. Retrieve routes

GET <https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/routing/config>

Response Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<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>
</routing>

```

```

    </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>
    </ospfInterface>
  </ospfInterfaces>
  <redistribution>
    <enabled>true</enabled>
  </redistribution>
  <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>
</ospf>

```

```

<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 7-16. Delete routing

---

Request

```
DELETE https://<nsxmgr-ip>/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 7-17. Configure global route

---

Request:

```
PUT https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/routing/config/global
```

Request Body:

```

<routingGlobalConfig>
  <routerId>1.1.1.1</routerId> <!-- Required when dynamic routing protocols like OSPF, BGP, IS-IS 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>

```

---

### 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 7-18. Query global route

---

```
PUT https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/routing/config/global
```

```
<routingGlobalConfig>
```

## Manage Static Routing

Add or query static and default routes for specified Edge.

### Configure Static Routes

Configures static and default routes.

#### Example 7-19. Configure static routes

---

Request:

PUT https://<nsxmgr-ip>/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>
      <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 7-20. Configure static routes

---

GET https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/routing/config/static

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 7-21. Delete static routes

---

Request

DELETE https://<nsxmgr-ip>/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 7-22. Configure OSPF

---

Request

PUT https://<nsxmgr-ip>/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>
      <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 7-23. Query OSPF

---

#### Request

GET https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/routing/config/ospf

#### Response Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<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>
    </ospfInterface>
  </ospfInterfaces>
</ospf>

```

```

    <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>
    <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 7-24. Delete OSPF

---

Request

```
DELETE https://<nsxmgr-ip>/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 7-25. Configure ISIS

---

Request



PUT https://<nsxmgr-ip>/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>mnr</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 7-26. Query ISIS

Request

GET https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/routing/config/isis

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 7-27. Delete ISIS

---

Request

DELETE https://<nsxmgr-ip>/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 7-28. Configure BGP

---

Request

PUT https://<nsxmgr-ip>/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 -->
        </from>
      </rule>
    </rules>
  </redistribution>
</bgp>
```

```

        <connected>true</connected> <!-- Optional. Defaults to false -->
    </from>
    <action>permit</action> <!-- Mandatory. Valid values are deny|permit -->
</rule>
</rules>
</redistribution>
</bgp>

```

---

## Query BGP

### Example 7-29. Query BGP

---

#### Request

GET https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/routing/config/bgp

#### Response Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<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 7-30. Delete BGP

---

Request

DELETE https://<nsxmgr-ip>/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 7-31. Configure bridge

---

Request

PUT https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/bridging/config

Request Body:

```

<bridges>
  <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>

```

---

## Query Bridge Configuration

Retrieves bridge configuration.

### Query BGP

#### Example 7-32. Query bridges

---

Request

GET https://<nsxmgr-ip>/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 7-33. Delete bridges

---

Request

DELETE https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/routing/bridging/config

---

# NSX Edge Services Gateway Installation, Upgrade, and Management

---

# 8

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 Edge Services gateway.

This chapter includes the following topics:

- [“Installing NSX Edge Services Gateway”](#) on page 145
- [“Upgrading vShield Edge 5.1.x or 5.5 to NSX Edge”](#) on page 147
- [“Query Installed Edges”](#) on page 147
- [“Modifying NSX Edge Configuration”](#) on page 151
- [“Deleting NSX Edge”](#) on page 155
- [“Configuring Edge Services in Async Mode”](#) on page 155
- [“Configuring Certificates”](#) on page 157
- [“Working with NSX Edge Firewall”](#) on page 159
- [“Working with NAT”](#) on page 170
- [“Working with Routing”](#) on page 173
- [“Working with Load Balancer”](#) on page 187
- [“Working with DHCP Service”](#) on page 207
- [“Working with DHCP Relay”](#) on page 211
- [“Working with High Availability \(HA\)”](#) on page 213
- [“Working with Syslog”](#) on page 214
- [“Managing SSL VPN”](#) on page 215
- [“Working with L2 VPN”](#) on page 244
- [“Working with IPSEC VPN”](#) on page 248

- [“Managing an NSX Edge”](#) on page 252

---

**IMPORTANT** All NSX REST requests require authentication. See [“Using the NSX REST API”](#) on page 25 for details about basic authorization.

---



## 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 NSXd 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 8-1. Install Services Gateway

#### Request

POST https://<nsxmgr-ip>/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 8GB 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>
<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 consecutively.-->
<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>

```

---

## Upgrading vShield Edge 5.1.x or 5.5 to NSX Edge

Upgrades vShield Edge 5.1.x or 5.5 to NSX Edge. The appliances are upgraded and feature configurations are retained and upgraded

### Example 8-2. Upgrade vShield Edge

---

Request:

POST https://<nsxmgr-ip>/api/4.0/edges/{edgeId}?action=upgrade

---

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

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

## Query Installed Edges

You can retrieve a list of NSX Edges in your inventory or filter the results by datacenter or port group.

### Example 8-3. Retrieve Edges

---

Retrieve all Edges Request:

GET https://<nsxmgr-ip>/api/4.0/edges/

Retrieve Edges by datacenter:

GET /api/4.0/edges/?datacenter=<datacenterMoid>

Retrieve Edges on specified tenant:

GET /api/4.0/edges/?tenant=<tenantId>

Retrieve Edges with one interface on specified port group:

GET /api/4.0/edges/?pg=<pgMoid>

Retrieve Edges with specified tenant and port group:

GET /api/4.0/edges/?tenant=<tenant>&pg=<pgMoid>

---

### Example 8-4. Retrieve Edge details

---

Request:

GET https://<nsxmgr-ip>/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>
        <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>
<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>
<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>
<vnid>1</vnid>
<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 8-5. Modify Edge configuration

---

Request:

PUT https://<nsxmgr-ip>/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>
<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>
<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>
</features>
<autoConfiguration>
<enabled>>true</enabled>
<rulePriority>high</rulePriority>
</autoConfiguration>
</edge>

```

---

where groupingObjectId can be cluster, network, etc.

## Deleting NSX Edge

Deletes specified Edge from database. Associated appliances are also deleted.

### Example 8-6. Delete Edge

---

Request

```
DELETE https://<nsxmgr-ip>/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 8-7. Query job status

---

Request:

```
GET https://<nsxmgr-ip>/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 8-8. Query all jobs

---

Request:

GET https://<nsxmgr-ip>/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 8-9. Query active jobs

---

Request:

GET https://<nsxmgr-ip>/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 8-10. Create self signed certificate

---

Request:

```
POST https://<vsm-ip>/api/2.0/services/truststore/certificate/<scopeId>
<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 8-11. Create certificate for CSR

---

Request:

```
POST https://<vsm-ip>/api/2.0/services/truststore/certificate?csrId=<csrId>
```

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 8-12. Query specific certificate

---

Request:

```
GET https://<vsm-ip>/api/2.0/services/truststore/certificate/<certificateId>
```

---

##### Example 8-13. Query all certificates for a scope

---

Request:

```
GET https://<vsm-ip>/api/2.0/services/truststore/certificate/scope/<scopeId>
```

---

#### Delete Certificate

Deletes the specified certificate.

**Example 8-14. Delete certificate**

---

Request:

DELETE https://&lt;vsm-ip&gt;/api/2.0/services/truststore/certificate/&lt;certificateId&gt;

**Working with Certificate Signing Requests (CSRs)**

Allows you to manage CSRs.

**Create CSR****Example 8-15. Create CSR**

---

Request:

POST https://&lt;vsm-ip&gt;/api/2.0/services/truststore/csr/&lt;scopeId&gt;

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 8-16. Create self signed certificate for CSR**

---

Request:

PUT https://&lt;vsm-ip&gt;/api/2.0/services/truststore/csr/&lt;csrId&gt;?noOfDays=&lt;value&gt;

**Query CSRs**

Retrieves specified CSR or all CSRs for specified scope.

**Example 8-17. Query specific CSR**

---

GET https://&lt;vsm-ip&gt;/api/2.0/services/truststore/csr/&lt;csrId&gt;

**Example 8-18. Query CSRs for specific scope**

---

GET https://&lt;vsm-ip&gt;/api/2.0/services/truststore/csr/scope/&lt;scopeId&gt;

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 8-19. Create CRL

---

Request:

POST https://<vsm-ip>/api/2.0/services/truststore/crl/<scopeId>

Request Body:

```
<trustObject>
  <pemEncoding></pemEncoding>
</trustObject>
```

---

### Query CRL

Retrieves all CRLs certificates for the specified certificate or scope.

#### Example 8-20. Query CRL

---

Retrieve certificate object for the specified certificate ID:

GET https://<vsm-ip>/api/2.0/services/truststore/crl/<crId>

Retrieve all certificates for the specified scope:

GET https://<vsm-ip>/api/2.0/services/truststore/crl/scope/<scopeId>

---

### Delete CRL

Deletes the specified CRL.

#### Example 8-21. Delete CRL

---

Request:

DELETE https://<vsm-ip>/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.

NSX Edge firewall can be managed through the centralized firewall configuration. These are independent configurations and work together. You can define rules for an Edge through the centralized firewall configuration using the `appliedTo` parameter. These will be displayed at the top of the Edge firewall configuration. For more information on centralized firewall, see [Chapter 9, “Configuring Firewall,”](#) on page 277.

## 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, the user 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.

### Example 8-22. Configure firewall

Request:

PUT `https://<nsxmgr-ip>/api/4.0/edges/{edgeId}/firewall/config`

Request Body:

```
<?xml version="1.0"?>
<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>3600</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 -->
</globalConfig>
<rules>
  <rule>
    <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 -->
    <exclude>false</exclude> <!-- Optional. Setting this flag to "true" specifies the source to be everything except for the objects specified
      here.-->
    <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 -->
    <exclude>false</exclude> <!-- Optional. Setting this flag to "true" specifies the destination to be everything except for the objects
      specified here.-->
    <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 -->
    <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 -->
</rule>
<rule>
...
</rule>
.....
</rules>
</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 8-23. Query firewall

Request:

GET https://<nsxmgr-ip>/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>3600</tcpTimeoutEstablished>
    <tcpTimeoutClose>30</tcpTimeoutClose>
    <udpTimeout>60</udpTimeout>
    <icmpTimeout>10</icmpTimeout>
    <icmp6Timeout>10</icmp6Timeout>
    <ipGenericTimeout>120</ipGenericTimeout>
  </globalConfig>
  <rules>
    <rule>
      <id>131079</id>
      <ruleTag>131079</ruleTag>
      <name>firewall</name>
      <ruleType>internal_high</ruleType>
    </rule>
  </rules>
</firewall>

```

```

    <source>
<exclude>false<exclude>
<vnidGroupId>vse</vnidGroupId>
    </source>
    <action>accept</action>
    <enabled>true</enabled>
    <loggingEnabled>>false</loggingEnabled>
    <description>firewall</description>
</rule>
<rule>
    <id>131080</id>
    <ruleTag>131080</ruleTag>
    <name>ipsec</name>
    <ruleType>internal_high</ruleType>
    <source>
<exclude>false<exclude>
    <groupingObjectId>ipset-934</groupingObjectId>
    <groupingObjectId>ipset-933</groupingObjectId>
    </source>
    <destination>
<<exclude>false<exclude>
    <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>
</rule>
<rule>
    <id>131077</id>
    <ruleTag>131077</ruleTag>
    <name>name1</name>
    <ruleType>user</ruleType>
    <source>
<exclude>false<exclude>
    <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>
<exclude>false<exclude>
    <groupingObjectId>ipset-941</groupingObjectId>

    <vnidGroupId>vse</vnidGroupId>
    <vnidGroupId>external</vnidGroupId>
    </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>
</rule>
</rule>

```

```

<id>131078</id>
<ruleTag>131078</ruleTag>
<name>name2</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>
<rule>
  <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>
</rule>
</rules>
</firewall>

```

---

## Query Firewall Configuration with PreRules for Edge

Retrieves firewall configuration along with pre rules (rules created in the Distributed Firewall configuration for this Edge).

### Example 8-24. Query preRules applied to an Edge

---

Request:

GET https://<nsxmgr-ip>/api/4.0/edges/{edgeId}/firewall/config?showPreRules=true

Response Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<firewall>
  <version>3</version>
  <enabled>true</enabled>
  <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>
  <defaultPolicy>
    <action>deny</action>
    <loggingEnabled>false</loggingEnabled>
  </defaultPolicy>
  <firewallRules>

```

```

<firewallRule>
  <id>1028</id>
  <name />
  <ruleType>PRE_RULES</ruleType>
  <enabled>true</enabled>
  <loggingEnabled>>false</loggingEnabled>
  <description />
  <action>accept</action>
  <application>
    <applicationId>application-261</applicationId>
  </application>
</firewallRule>
<firewallRule>
  <id>1030</id>
  <name />
  <ruleType>PRE_RULES</ruleType>
  <enabled>true</enabled>
  <loggingEnabled>>false</loggingEnabled>
  <description />
  <action>reject</action>
</firewallRule>
<firewallRule>
  <id>131074</id>
  <ruleTag>131074</ruleTag>
  <name>firewall</name>
  <ruleType>internal_high</ruleType>
  <enabled>true</enabled>
  <loggingEnabled>>false</loggingEnabled>
  <description>firewall</description>
  <action>accept</action>
  <source>
    <exclude>>false</exclude>
    <vnicGroupId>vse</vnicGroupId>
  </source>
</firewallRule>
<firewallRule>
  <id>133123</id>
  <ruleTag>133123</ruleTag>
  <name>test1</name>
  <ruleType>user</ruleType>
  <enabled>true</enabled>
  <loggingEnabled>>false</loggingEnabled>
  <description />
  <matchTranslated>>false</matchTranslated>
  <action>reject</action>
  <source>
    <exclude>true</exclude>
    <ipAddress>192.168.10.2</ipAddress>
  </source>
  <application>
    <applicationId>applicationgroup-16</applicationId>
  </application>
</firewallRule>
<firewallRule>
  <id>131073</id>
  <ruleTag>131073</ruleTag>
  <name>default rule for ingress traffic</name>
  <ruleType>default_policy</ruleType>
  <enabled>true</enabled>
  <loggingEnabled>>false</loggingEnabled>
  <description>default rule for ingress traffic</description>
  <action>deny</action>
</firewallRule>
</firewallRules>
</firewall>

```

---

## Append Firewall Rules

Adds one or more rules below the existing rules in the rules table.

### Example 8-25. Add firewall rule

POST <https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/firewall/config/rules>

Request Body:

```
<rules>
<rule>
<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 -->
<exclude>false<exclude> <!-- Optional. Setting this flag to "true" specifies the source to be everything except for the objects specified
                             here.-->
<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 -->
<exclude>false<exclude> <!-- Optional. Setting this flag to "true" specifies the destination to be everything except for the objects
                             specified here.-->
<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/reject -->
<enabled>true</enabled>    <!-- Optional. Defaults to true -->
<loggingEnabled>true</loggingEnabled> <!-- Optional. Defaults to false -->
<description>comments</description> <!-- Optional -->
</rule>
<rule>
...
</rule>
```

### 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 8-26. Add a rule above a specific rule

Request:

POST <https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/firewall/config/rules?aboveRuleId=<ruleId>>

Request Body:

```
<rule>
<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 -->
<exclude>false<exclude> <!-- Optional. Setting this flag to "true" specifies the source to be everything except for the objects specified
                             here.-->
```

```

<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 -->
<exclude>false<exclude> <!-- Optional. Setting this flag to "true" specifies the destination to be everything except for the objects
specified here.-->
<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|reject -->
<enabled>true</enabled> <!-- Optional. Defaults to true -->
<loggingEnabled>true</loggingEnabled> <!-- Optional. Defaults to false -->
<description>comments</description> <!-- Optional -->
</rule>

```

---

## Query Specific Rule

### Example 8-27. Retrieve specific rule

GET https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/firewall/config/rules/<ruleId>

Response Body:

```

<rule>
<name>new rule</name>
<source>
<exclude>false<exclude>
<vnicGroupId>vnic-index-5</vnicGroupId>
</source>
<destination>
<exclude>false<exclude>
<groupingObjectId>ipset-127</groupingObjectId>
</destination>
<action>accept</action>
<enabled>true</enabled>
<loggingEnabled>true</loggingEnabled>
<description/>
</rule>

```

---

## Modify Firewall Rule

You can modify a rule by specifying its rule ID.

### Example 8-28. .Update specific rule

PUT https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/firewall/config/rules/<ruleId>

Response Body:

```

<rule>
<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 -->
<exclude>false<exclude> <!-- Optional. Setting this flag to "true" specifies the source to be everything except for the objects specified
here.-->

```

```

<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 -->
<exclude>false</exclude> <!-- Optional. Setting this flag to "true" specifies the destination to be everything except for the objects
specified here.-->
<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|reject -->
<enabled>true</enabled> <!-- Optional. Defaults to true -->
<loggingEnabled>true</loggingEnabled> <!-- Optional. Defaults to false -->
<description>comments</description> <!-- Optional -->
</rule>

```

---

## Delete a Firewall Rule

Deletes the rule with the specified rule ID.

### Example 8-29. Delete firewall rule

Request Body;

```
DELETE https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/firewall/config/rules/<ruleId>
```

---

## Delete Firewall Configuration

Deletes firewall configuration for Edge.

### Example 8-30. Delete firewall configuration

Request:

```
DELETE https://<nsxmgr-ip>/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 8-31. Query global firewall configuration

Request:

GET https://<nsxmgr-ip>/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 8-32. Modify global firewall configuration

---

Request:

PUT https://<nsxmgr-ip>/api/4.0/edges/{edgeId}/firewall/config/global

Response 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>3600</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 -->
</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 8-33. Query default firewall configuration

---

Request:

GET https://<nsxmgr-ip>/api/4.0/edges/{edgeId}/firewall/config/defaultpolicy

Response Body:



```
<?xml version="1.0" encoding="UTF-8"?>
<firewallDefaultPolicy>
<action>ACCEPT</action>
<loggingEnabled>true</loggingEnabled>
</firewallDefaultPolicy>
```

---

## Modify Default Firewall Policy

Configures default firewall policy for the specified Edge.

### Example 8-34. Modify default firewall configuration

---

Request:

PUT https://<nsxmgr-ip>/api/4.0/edges/{edgeId}/firewall/config/defaultpolicy

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<firewallDefaultPolicy>
<action>ACCEPT</action>
<loggingEnabled>true</loggingEnabled>
</firewallDefaultPolicy>
```

---

## Query Firewall Statistics

Retrieves number of ongoing connections for the firewall configuration.

### Example 8-35. Query firewall statistics

---

Request:

GET https://<nsxmgr-ip>/api/4.0/edges/{edgeId}/firewall/statistics/dashboard/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 8-36. Query statistics for a rule

---

Request:

GET https://<nsxmgr-ip>/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 8-37. Disable Firewall

---

Request:

PUT https://<nsxmgr-ip>/api/4.0/edges/{edgeId}/firewall/config

Request Body:

```
<firewall><enabled>>false</enabled></firewall>
```

---

## Working with NAT

### Configure NAT

NSX Edge provides network address translation (NAT) service to protect the IP addresses of internal (private) networks from the public network. 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 Edge in the Edge Manager user interface and in the vSphere Client plug-in.

### Example 8-38. Configure SNAT and DNAT rules for a Edge Edge

---

PUT https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/nat/config

```
<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>
      <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>
      <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 a Edge Edge

### Example 8-39. Query SNAT and DNAT rules for a Edge Edge

---

GET https://<nsmgr-ip>/api/4.0/edges/<edgeId>/nat/config

Response Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<nat>
  <natRules>
    <natRule>
      <ruleTag>196609</ruleTag>
      <ruleId>196609</ruleId>
      <action>dnat</action>
      <vnic>0</vnic>
      <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>
      <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>
    </natRule>
  </natRules>
</nat>

```

```

        <translatedPort>any</translatedPort>
        <ruleType>user</ruleType>

    </natRule>
</natRules>
</nat>

```

---

## Delete all NAT Rules

Deletes all SNAT and DNAT rules for a Edge Edge. The auto plumbed rules continue to exist.

### Example 8-40. Delete NAT rules

---

Request:

DELETE https://<nsxmgr-ip>/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 8-41. Add a NAT rule above a specific rule

---

POST https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/nat/config/rules?aboveRuleId=<ruleId>

Request Body:

```

<natRule>
  <action>dnat</action>
  <vnid>0</vnid>
  <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 8-42. Add NAT rules to the bottom of the rules table

---

POST https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/nat/config/rules

Response Body:

```

<natRules>
  <natRule>
    <action>dnat</action>
    <vnid>0</vnid>
    <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 Edge (0-9).

### Modify a NAT Rule

Replaces the NAT rule with the specified rule ID.

#### Example 8-43. Replaces a NAT rule

---

```
PUT https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/nat/config/rules/ruleID
```

Response Body:

```
<natRule>
  <action>dnat</action>
  <vnic>0</vnic>
  <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 Edge (0-9).

### Delete a NAT Rule

Deletes the rule with the specified rule ID.

#### Example 8-44. Delete NAT rule

---

Request:

```
DELETE https://<nsxmgr-ip>/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, BGP, and IS-IS.

#### Example 8-45. Configure routes

---

```
PUT https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/routing/config
```

```
<routing>
  <routingGlobalConfig>
    <ecmp>false</ecmp> <!-- Optional. Default is false -->
    <routerId>1.1.1.1</routerId> <!-- Required when dynamic routing protocols like OSPF, BGP, IS-IS 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>
      <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> <!-- Atleast 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>
  </rules>
</redistribution>

```

```

</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>
</rules>
<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 8-46. Retrieve routes

---

GET https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/routing/config

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 8-47. Delete routing

---

Request

DELETE https://<nsxmgr-ip>/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 8-48. Configure global route

---

Request:

PUT https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/routing/config/global

Request Body:

```

<routingGlobalConfig>
  <routerId>1.1.1.1</routerId> <!-- Required when dynamic routing protocols like OSPF, BGP, IS-IS 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>

```

---

## 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 8-49. Query global route

Request Body:

PUT https://<nsxmgr-ip>/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 8-50. Configure static routes

Request:

PUT https://<nsxmgr-ip>/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>
      <network>4.1.1.4/22</network>
      <nextHop>10.112.196.118</nextHop>
    </route>
  </staticRoutes>
</staticRouting>

```

```

    <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 8-51. Query static routes

---

Request:

GET https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/routing/config/static

Response Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<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 8-52. Delete static routes

---

Request

DELETE https://<nsxmgr-ip>/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 8-53. Configure OSPF

Request

PUT https://<nsxmgr-ip>/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>
      <vnid>0</vnid>
      <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>

```

---

## Query OSPF

### Example 8-54. Query OSPF

---

#### Request

GET https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/routing/config/ospf

#### Request Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<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>
      <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 8-55. Delete OSPF

---

Request

```
DELETE https://<nsxmgr-ip>/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 8-56. Configure ISIS

---

Request

```
PUT https://<nsxmgr-ip>/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> <!-- Atleast 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>mnr</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 8-57. Query ISIS

#### Request

GET https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/routing/config/isis

#### Request Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<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 8-58. Delete ISIS

---

Request

DELETE https://<nsxmgr-ip>/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 8-59. Configure BGP

---

Request

PUT https://<nsxmgr-ip>/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 -->
    </bgpNeighbour>
  </bgpNeighbours>
</bgp>

```



```

<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 8-60. Query BGP

#### Request

GET https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/routing/config/bgp

#### Request Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<bgp>
  <enabled>true</enabled>
  <localAS>65535</localAS>
  <bgpNeighbours>
    <bgpNeighbour>
      <ipAddress>192.168.1.10</ipAddress>
      <remoteAS>65500</remoteAS>
      <weight>60</weight>
      <holdDownTimer>180</holdDownTimer>
    </bgpNeighbour>
  </bgpNeighbours>
</bgp>

```

```

    <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 8-61. Delete BGP

---

Request

DELETE https://<nsxmgr-ip>/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 8-62. Configure load balancer

PUT https://<vsm-ip>/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 -->
    <!-- <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 enabledServiceInsertion 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 -->
    <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, msrpd -->
  </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>
<transparent>>false</transparent>
<algorithm>round-robin</algorithm>
    is round-robin -->
<monitorId>monitor-1</monitorId>
<member>
    <memberId>member-1</memberId>
        when it has reference -->
    <ipAddress>192.168.101.201</ipAddress>
        groupingObjectId is not specified -->
    <!-- <groupingObjectId>vm-24</groupingObjectId> -->
        dvportgroup-26 -->
    <weight>1</weight>
    <port>80</port>
    <minConn>10</minConn>
    <maxConn>100</maxConn>
    <name>m1</name>
</member>
<member>
    <memberId>member-2</memberId>
    <ipAddress>192.168.101.202</ipAddress>
    <weight>1</weight>
    <port>80</port>
    <minConn>10</minConn>
    <maxConn>100</maxConn>
    <name>m2</name>
    <condition>enabled</condition>
</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>

```

<!-- optional -->  
 <!-- optional, default is false -->  
 <!-- optional, valid values are round-robin, ip-hash, uri, leastconn, default  
 is round-robin -->  
 <!-- optional, it should be a valid monitorId, it is an array -->  
 <!-- 0-32 pool member items could be added -->  
 <!-- optional, it should match "member-X" pattern, this item is required  
 when it has reference -->  
 <!-- optional, a valid ip address(ipv4/ipv6), it is required when  
 groupingObjectId is not specified -->  
 <!-- optional, groupingObject id such as vm-24, network-25,  
 dvportgroup-26 -->  
 <!-- optional, default is 1 -->  
 <!-- required -->  
 <!-- optional, default is 0 -->  
 <!-- optional, default is 0 -->  
 <!-- optional, it is required when it is used in ACL rule -->  
 <!-- optional, default is enabled, valid values are enabled/disabled -->

```

    <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>
  when it is referenced -->
  <type>http</type>
  <interval>5</interval>
  <timeout>15</timeout>
  <maxRetries>3</maxRetries>
  <method>GET</method>
  OPTIONS/GET/HEAD/POST/PUT/DELETE/TRACE/CONNECT -->
  <url></url>
  <name>http-monitor</name>
  <!-- <expected>HTTP/1</expected> -->
  protocol -->
  <!-- <send>hello</send> -->
  <!-- <receive>ok</received> -->
  <!-- <extension>no-body
max-age=3h
content-type=Application/xml</extension> -->
parameters -->

  </monitor>
  <monitor>
    <monitorId>monitor-2</monitorId>
    <type>https</type>
    <interval>5</interval>

```

<!-- optional, this item should follow "monitor-X" pattern, it is required

<!-- required, valid values are http/https/tcp -->

<!-- optional, default is 5 -->

<!-- optional, default is 15 -->

<!-- optional, default is 3 -->

<!-- optional, valid value is

<!-- optional -->

<!-- required -->

<!-- optional, Expected response string. Default is "HTTP/1" for http(s)

<!-- optional, URL encoded http POST data for http(s) protocol -->

<!-- optional, String to expect in the content for http(s) protocol -->

<!-- optional, advanced setting for monitor to fill more customized

```

    <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 example2 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:!aNULL:!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 8-63. Retrieve load balancer configuration

---

GET https://<vsm-ip>/api/4.0/edges/<edgeId>/loadbalancer/config

Response Body:

See [Example 8-62](#).

---

## Delete Load Balancer Configuration

### Example 8-64. Delete load balancer configuration

---

Request:

DELETE https://<vsm-ip>/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.

#### Example 8-65. Append profile

---

Request:

POST https://<vsm-ip>/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 8-66. Modify profile

---

Request:

PUT https://<vsm-ip>/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 8-67. Query profile

---

Request:

GET https://<vsm-ip>/api/4.0/edges/<edgeId>/loadbalancer/config/applicationprofiles/{applicationProfileId}

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<applicationProfile>
  <applicationProfileId>applicationProfile-5</applicationProfileId>
  <persistence>
    <method>cookie</method>
    <cookieName>JSESSIONID</cookieName>
    <cookieMode>insert</cookieMode>
  </persistence>
  <name>http_application_profile_2</name>
  <insertXForwardedFor>true</insertXForwardedFor>
  <sslPassthrough>true</sslPassthrough>
</applicationProfile>

```

---

## Query all Application Profiles

Retrieves all application profiles on Edge.

### Example 8-68. Query profiles

---

Request:

GET https://<vsm-ip>/api/4.0/edges/<edgeId>/loadbalancer/config/applicationprofiles/

Response Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<loadBalancer>
  <applicationProfile>
    <applicationProfileId>applicationProfile-1</applicationProfileId>
    <persistence>
      <method>cookie</method>
      <cookieName>JSESSIONID</cookieName>
      <cookieMode>insert</cookieMode>
    </persistence>
    <name>http_application_profile</name>
    <insertXForwardedFor>true</insertXForwardedFor>
    <sslPassthrough>true</sslPassthrough>
  </applicationProfile>
  <applicationProfile>
    <applicationProfileId>applicationProfile-2</applicationProfileId>
    <persistence>
      <method>ssl_sessionid</method>
    </persistence>
    <name>https_application_profile</name>
    <insertXForwardedFor>true</insertXForwardedFor>
    <sslPassthrough>true</sslPassthrough>
  </applicationProfile>
  <applicationProfile>
    <applicationProfileId>applicationProfile-3</applicationProfileId>
    <name>tcp_application_profile</name>
    <insertXForwardedFor>false</insertXForwardedFor>
    <sslPassthrough>true</sslPassthrough>
  </applicationProfile>
</loadBalancer>

```

---

## Delete Application Profile

Deletes an application profile.

### Example 8-69. Delete profile

---

Request:

DELETE https://<vsm-ip>/api/4.0/edges/<edgeId>/loadbalancer/config/applicationprofiles/{applicationProfileId}

---

**Delete all Application Profiles**

Deletes all application profile.

**Example 8-70.** Delete profiles

Request:

DELETE https://<vsm-ip>/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 8-71.** Append rule

Request:

POST https://<vsm-ip>/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 8-72.** Modify rule

Request:

PUT https://<vsm-ip>/api/4.0/edges/<edgeId>/loadbalancer/config/applicationrules/{applicationruleId}

Request Body:

See [Example 8-71](#).

**Query Application Rule**

Retrieves an application rule.

**Example 8-73.** Query rule

Request:

GET https://<vsm-ip>/api/4.0/edges/<edgeId>/loadbalancer/config/applicationrules/{applicationruleId}

Response Body:

See [Example 8-71](#).

**Query all Application Rules**

Retrieves all application rules on Edge.

**Example 8-74. Query rules**

---

Request:

GET https://<vsm-ip>/api/4.0/edges/<edgeId>/loadbalancer/config/applicationrules/{applicationruleId}

---

**Delete Application Rule**

Deletes an application rule.

**Example 8-75. Delete rule**

---

Request:

DELETE https://<vsm-ip>/api/4.0/edges/<edgeId>/loadbalancer/config/applicationrules/{applicationruleId}

---

**Delete all Application Rules**

Deletes all application rules.

**Example 8-76. Delete rules**

---

Request:

DELETE https://<vsm-ip>/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 8-77. Append monitor**

---

Request:

POST https://<vsm-ip>/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 8-78. Modify monitor**

---

Request:

PUT https://<vsm-ip>/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 8-79. Query monitor

---

Request:

GET https://<vsm-ip>/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 8-80. Query monitors

---

Request:

POST https://<vsm-ip>/api/4.0/edges/<edgeId>/loadbalancer/config/monitors

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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>

```

---

## Delete Monitor

Deletes a load balancer monitor.

### Example 8-81. Delete monitor

---

Request:

```
DELETE https://<vsm-ip>/api/4.0/edges/<edgeId>/loadbalancer/config/monitors/{monitorId}
```

---

## Delete all Monitors

Deletes all load balancer monitors.

### Example 8-82. Delete monitors

---

Request:

```
DELETE https://<vsm-ip>/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 8-83. Append virtual server

---

Request:

```
POST https://<vsm-ip>/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>82</port>
  <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 8-84. Query virtual server

---

Request:

GET https://<vsm-ip>/api/4.0/edges/<edgeId>/loadbalancer/config/virtualservers/*virtualserverID*

Response Body:

See [Example 8-83](#).

---

## Query all Virtual Servers

Retrieves all virtual servers.

### Example 8-85. Query virtual servers

---

Request:

GET https://<vsm-ip>/api/4.0/edges/<edgeId>/loadbalancer/config/virtualservers

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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>80</port>
    <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</port>
    <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>1234</port>
    <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>1235</port>
<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 8-86. Delete virtual server

---

Request:

```
DELETE https://<vsm-ip>/api/4.0/edges/<edgeId>/loadbalancer/config/virtualservers/virtualserverID
```

---

### Delete all Virtual Server

Deletes all virtual servers.

#### Example 8-87. Delete all virtual server

---

Request:

```
DELETE https://<vsm-ip>/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 8-88. Append backend pool

---

Request:

```
POST https://<vsm-ip>/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>1234</port>
  <minConn>10</minConn>
  <maxConn>100</maxConn>
  <name>m5</name>
  <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>
  <monitorPort>80</monitorPort>
</member>
</pool>

```

---

## Modify a Backend Pool

Updates the specified pool.

### Example 8-89. Modify backend pool

---

Request:

PUT https://<vsm-ip>/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>
    <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>
    <monitorPort>80</monitorPort>
  </member>
</pool>

```

---

## Query Backend Pool Details

Retrieves information about the specified pool.

**Example 8-90. Get backend pool details**

---

Request:

GET https://<vsm-ip>/api/4.0/edges/<edgeId>/loadbalancer/config/pools/*poolId*

Response Body:

See Example [Example 8-89](#).

---

**Query all Backend Pools**

Gets all backend pools configured for the specified NSX Edge.

**Example 8-91. Query all backend pools**

---

Request:

GET https://<vsm-ip>/api/4.0/edges/<edgeId>/loadbalancer/config/pools

Response Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<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 8-92. Delete backend pool

---

Request:

```
DELETE https://<vsm-ip>/api/4.0/edges/<edgeId>/loadbalancer/config/pools/poolID
```

---

## Delete all Backend Pools

Deletes all backend pools configured for the specified NSX Edge.

### Example 8-93. Delete backend pool

---

Request:

```
DELETE https://<vsm-ip>/api/4.0/edges/<edgeId>/loadbalancer/config/pools
```

---

## Query Statistics

Retrieves load balancer statistics.

### Example 8-94. Retrieve load balancer statistics

---

Request:

```
GET https://<vsm-ip>/api/4.0/edges/<edgeId>/loadbalancer/statistics
```

Response Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<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>
  </pool>

```

```

    <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>
  <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>
  </member>

```

```

    <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 LoadBalancer Acceleration Mode

**Example 8-95.** Update acceleration mode

---

Request:

POST https://<nsxmgr-ip>/api/4.0/edges/{edgeId}/loadbalancer/acceleration?enable=true/false

---

## Update Load Balancer Member Condition

**Example 8-96.** Update member condition

---

Request:

POST https://<nsxmgr-ip>/api/4.0/edges/{edgeId}/loadbalancer/config/members/{memberId}?enable=true/false

---

## 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 vnicId 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 8-97. Configure DHCP service

Request:

PUT https://<vsm-ip>/api/4.0/<edgeId>/dhcp/config

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<dhcp>
  <enabled>true</enabled> <!-- optional, default is "true". -->
  <staticBindings>
    <staticBinding>
      <!-- NOTE: user can either specify macAddress directly, or specify vmId and vnicId.
            In case both are specified, only macAddress will be used; vmId and vnicId
            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. -->
      <vnicId>1</vnicId> <!-- optional. possible values 0 to 9 -->
      <hostname>abcd</hostname> <!-- optional. disallow duplicate. the -->
      <ipAddress>192.168.4.2</ipAddress> <!-- required. the IP must belongs to one subnet of edge vNics,
            but must NOT overlap any primary/secondary ips of defined explicitly in vNic. -->
      <defaultGateway>192.168.4.1</defaultGateway> <!-- optional. default is the primary ip of the belonging vNic.-->
      <domainName>eng.vmware.com</domainName> <!-- optional. -->
      <primaryNameServer>192.168.4.1</primaryNameServer> <!-- optional. if autoConfigDNS=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 second, default is "86400". valid leaseTime
            is a valid digit, or "infinite". -->
      <autoConfigDNS>true</autoConfigDNS> <!-- optional. default is true. -->
    </staticBinding>
  </staticBindings>
  <ipPools>
    <ipPool>
      <ipRange>192.168.4.192-192.168.4.220</ipRange> <!-- required. the ipRange must belongs to one of
            a subnet of Edge vNics. And can NOT contains any ip that defined explicitly as vNic
            primary ip or secondary ip. -->
      <defaultGateway>192.168.4.1</defaultGateway> <!-- optional. default is the primary ip of the belonging vNic.-->
      <domainName>eng.vmware.com</domainName> <!-- optional. -->
      <primaryNameServer>192.168.4.1</primaryNameServer> <!-- optional. if autoConfigDNS=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 second, default is "86400". valid leaseTime is a valid
            digit, or "infinite". -->
      <autoConfigDNS>true</autoConfigDNS> <!-- optional. default is true. -->
    </ipPool>
  </ipPools>
</dhcp>
```



```

<logging> <!-- optional. logging is disable 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 8-98. Get DHCP configuration

---

Request:

GET https://<vsm-ip>/api/4.0/<edgeId>/dhcp/config

Response Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<dhcp>
  <enabled>true</enabled>
  <staticBindings>
    <staticBinding>
      <vmId>vm-111</vmId>
      <vnicId>1</vnicId>
      <hostname>abcd</hostname>
      <ipAddress>192.168.4.2</ipAddress>
      <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>
      <autoConfigureDNS>true</autoConfigureDNS>
    </staticBinding>
  </staticBindings>
  <ipPools>
    <ipPool>
      <ipRange>192.168.4.192-192.168.4.220</ipRange>
      <defaultGateway>192.168.4.1</defaultGateway>
      <domainName>eng.vmware.com</domainName>
      <primaryNameServer>192.168.4.1</primaryNameServer>
      <secondaryNameServer>4.2.2.4</secondaryNameServer>
      <leaseTime>3600</leaseTime>
      <autoConfigureDNS>true</autoConfigureDNS>
    </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 8-99. Delete DHCP configuration

---

Request:

DELETE https://<vsm-ip>/api/4.0/<edgeId>/dhcp/config

---

## Retrieve DHCP Lease Information

### Example 8-100. Get DHCP lease information

---

Request:

GET https://<vsm-ip>/api/4.0/<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 8-101. Add IP pool

---

Request:

POST https://<vsm-ip>/api/4.0/<edgeId>/dhcp/config/ippools

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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>
  <autoConfigureDNS>true</autoConfigureDNS>
</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 8-102. Add static binding

---

Request:

POST https://<vsm-ip>/api/4.0/<edgeId>/dhcp/config/bindings

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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>
  <autoConfigureDNS>true</autoConfigureDNS>
</staticBinding>
```

---

### Delete DHCP Pool

Deletes a pool specified by pool-id.

**Example 8-103.** Delete DHCP pool

---

Request:

```
DELETE https://<vsm-ip>/api/4.0/edges/<edgeId>/dhcp/config/ippools/<poolId>
```

---

### Delete DHCP Static Binding

Deletes the static-binding specified by binding-id.

**Example 8-104.** Delete DHCP static binding

---

Request:

```
DELETE https://<vsm-ip>/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 8-105.** Configure DHCP relay

---

```
PUT https://<vsm-ip>/api/4.0/<edgeId>/dhcp/config/relay
```

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<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 maxium 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 8-106. Query DHP Relay

Request:

GET https://<vsm-ip>/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>
      </giAddress>
    </relayAgent>
    <relayAgent>
      <vnicIndex>3</vnicIndex>
      <giAddress>192.168.3.254</giAddress>
    </giAddress>
    </relayAgent>
  </relayAgents>
</relay>

```

---

## Delete DHCP Relay Configuration

### Example 8-107. Delete DHP relay configuration

Request:

DELETE https://<vsm-ip>/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 8-108. Configure high availability

---

Request:

PUT `https://<vsm-ip>/api/4.0/edges/<edgeId>/highavailability/config`

Request Body:

```
<highAvailability>
  <vnic>1</vnic> <!-- Optional. User can provide the vNic Index. If not provided, the first internal-connected vnic will be used as
    the vnic -->
  <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 8-109. Get high availability configuration

---

Request: `api/`

GET `https://<vsm-ip>/4.0/edges/<edgeId>/highavailability/config`

Request 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 8-110.** Delete high availability configuration

---

Request:

DELETE https://<vsm-ip>/api/4.0/edges/<edgeId>/highavailability/config

---

## 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 8-111.** Configure syslog servers

---

Request:

PUT https://&lt;vsm-ip&gt;/api/4.0/edges/&lt;edgeId&gt;/syslog/config

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 8-112.** Query syslog servers

---

Request:

GET https://&lt;vsm-ip&gt;/api/4.0/edges/&lt;edgeId&gt;/syslog/config

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 8-113.** Delete syslog servers

---

Request:

---

DELETE https://<vsm-ip>/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 8-114. Enable or disable SSL VPN

---

Request:

POST https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/?enableService=true|False

---

### Query SSL VPN Details

Retrieves SSL VPN details.

#### Example 8-115. Get SSL VPN details

---

Request:

GET https://<vsm-ip>/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 8-116. Apply server settings

---

Request:

PUT https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/server/

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 -->
  <!-- Certificate has to be generated using certificate REST API and id returned should be mentioned here-->
  <!-- <certificateId>certificate-1</certificateId> --> <!-- optional. -->
  <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 8-117. Apply server settings

---

Request:

```
GET https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/server/
```

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 8-118. Add private network

---

Request:

```
POST https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/client/networkextension/privatenetworks/
```

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 8-119. Add private network

---

Request:

```
PUT https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/client/networkextension/privatenetworks/<privateNetworkID>
```

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
```



```

<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 8-120. Query private network

---

Request:

GET https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/client/networkextension/privatenetworks/*privateNetworkID*

Request Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<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 8-121. Query private network

---

Request:

GET https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/client/networkextension/privatenetworks/

Request Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<privateNetwork>
  <privateNetwork>
    <onjectId>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 8-122. Delete private network**

---

Request:

```
DELETE https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/client/networkextension/
        privatenetworks/privatenetworkID
```

---

**Delete all Private Networks**

Deletes all dynamic IP address configurations from the SSL VPN instance on NSX Edge.

**Example 8-123. Delete private network**

---

Request:

```
DELETE https://<vsm-ip>/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 8-124. Apply all private networks**

---

Request:

```
PUT https://<vsm-ip>/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 8-125. Add portal web resource**

---

Request:

```
POST https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/webresources/
```

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 8-126. Modify portal web resource**

---

Request:

PUT https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/webresources/*ID*

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 8-127. Get specific portal web resource**

---

Request:

GET https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/webresources/*ID*

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 8-128. Get portal web resource**

---

Request:

GET https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/webresources/

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 8-129.** Delete specific portal web resource

---

Request:

```
DELETE https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/webresources/ID
```

---

### Deletes all Web Resources

Deletes all web resources on the SSL VPN instance.

**Example 8-130.** Deletes all portal web resources

---

Request:

```
DELETE https://<vsm-ip>/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 deleted.

**Example 8-131.** Apply all private networks

---

Request:

```
PUT https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/client/networkextension/  
privatenetworks/
```

---

## Configure Users

### Add User

Adds a new portal user.

**Example 8-132.** Add a user

---

Request:

```
POST https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/auth/localserver/users/
```

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
  <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 8-133. Modify user

---

Request:

PUT https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/auth/localserver/users/

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
  <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 8-134. Query user

---

Request:

GET https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/auth/localserver/users/*userID*

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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>
```

---

## Delete User

Deletes specified user.

### Example 8-135. Delete user

---

Request:

DELETE https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/auth/localserver/users/*userID*

---

## Delete all Users

Deletes all users on the specified SSL VPN instance.

### Example 8-136. Delete all user

---

Request:

DELETE https://<vsm-ip>/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 deleted.

### Example 8-137. Apply all users

---

Request:

PUT https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/auth/localusers/users

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 8-138. Add IP pool

---

Request:

POST https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/client/networkextension/ippools/

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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>
```

```
</ipAddressPool>
```

---

## Modify IP Pool

Modifies the specified IP pool.

### Example 8-139. Modify IP pool

---

Request:

PUT https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/client/networkextension/ippools/*ippoolID*

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 8-140. Get IP pool

---

Request:

GET https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/client/networkextension/ippools/*ippoolID*

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 8-141. Gets all IP pools

---

Request:

GET https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/client/networkextension/ippools/

Response Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<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 8-142. Delete IP pool

Request:

```
DELETE https://<vsm-ip>/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 8-143. Deletes all IP pools

Request:

```
DELETE https://<vsm-ip>/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 deleted.

### Example 8-144. Apply IP pools

Request:

```
PUT https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/client/networkextension/ippools/
```

Request Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<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 8-145. Apply IP pools

---

Request:

PUT https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/client/networkextension/clientconfig/

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 8-146. Get client configuration

---

Request:

GET https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/client/networkextension/clientconfig/

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 8-147. Add installation package**

---

Request:

POST https://&lt;vsm-ip&gt;/api/4.0/edges/&lt;edgeId&gt;/sslvpn/config/client/networkextension/installpackages/

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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> <!--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 8-148. Modify installation package**

---

Request:

PUT https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/client/networkextension/  
installpackages/*ID*

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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> <!--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 8-149. Query installation package

---

Request:

```
GET https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/client/networkextension/
installpackages/<ID>
```

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 8-150. Query all installation package

---

Request:

```
GET https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/client/networkextension/
installpackages/
```

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 8-151. Delete installation package

---

Request:

```
DELETE https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/client/networkextension/
installpackages/ID
```

---

## Delete all Client Installation Packages

Deletes all installation packages.

### Example 8-152. Delete all installation packages

---

Request:

```
DELETE https://<vsm-ip>/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 8-153. Apply installation packages

---

Request:

```
PUT https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/client/networkextension/
installpackages/
```

Request 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>
  </clientInstallPackage>
</clientInstallPackages>

```

```

    <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>

```

---

## 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 8-154.** Upload portal logo

Request:

```
POST https://<vsm-ip>/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 8-155.** Upload phat banner

Request:

```
POST https://<vsm-ip>/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 8-156.** Upload client connected icon

Request:

```
POST https://<vsm-ip>/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 ico.

**Example 8-157.** Upload client disconnected icon

Request:

```
POST https://<vsm-ip>/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 8-158.** Upload client desktop icon

---

Request:

POST https://<vsm-ip>/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 8-159.** Upload client desktop icon

---

Request:

POST https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/layout/images/erroricon/

---

**Apply Layout Configuration**

Sets the portal layout.

**Example 8-160.** Apply layout configuration

---

Request:

PUT https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/layout/images/

---

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
  <layout>
    <!-- portal layout configuration-->
    <portalTitle>Pepsi Remote Access</portalTitle><!--optional. Default value is VMware -->
    <companyName>pepsi, Inc.</companyName><!--optional. Default value is VMware -->
    <!-- Portal Color Configuration-->
    <logoBackgroundColor>FFFFFF</logoBackgroundColor><!--optional. 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 8-161.** Query layout configuration

---

Request:

GET https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/layout/

---

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
  <layout>
    <!-- portal layout configuration-->
    <portalTitle>Pepsi Remote Access</portalTitle><!--optional. Default value is VMware -->
    <companyName>pepsi, Inc.</companyName><!--optional. Default value is VMware -->
    <!-- Portal Color Configuration-->
    <logoBackgroundColor>FFFFFF</logoBackgroundColor><!--optional. 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>

```

---

## 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.

#### Example 8-162. Upload RSA config file

---

Request:

POST https://<vsm-ip>/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 8-163. Apply Authentication Configuration

---

Request:edgeId

PUT https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/auth/settings/

Request Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<authenticationConfig>
  <passwordAuthentication>
    <authenticationTimeout>1</authenticationTimeout>    <!--optional. Default value is 1 mins-->
    <!-- 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 -->
  <primaryAuthServers>
    <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 -->
  </primaryAuthServers>
</authenticationConfig>

```

```

<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> <!-- 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>
<!-- 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-->
<com.vmware.vshield.edge.sslvpn.dto.RsaAuthServerDto>
    <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> <!--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 8-164. Query Authentication Configuration

Request:

GET https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/auth/settings/

Response Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<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 8-165. Apply advanced configuration

---

Request:

PUT https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/advancedconfig/

Request Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<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)-->
        <sessionIdleTimeout>10</sessionIdleTimeout>      <!--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 8-166. Query advanced configuration

---

Request:

GET https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/advancedconfig/

Response Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<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>
    <forcedTimeout>16</forcedTimeout>                 <!--optional. Value is in minute(s)-->
    <sessionIdleTimeout>10</sessionIdleTimeout>        <!--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 8-167. Query active clients

---

Request:

GET https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/activesessions/

Response Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<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 8-168. Disconnect active client

---

Request:

DELETE https://<vsm-ip>/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 8-169. Upload script

---

Request:

POST https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/script/file/

---

### Configure Script Parameters

Configures parameters associated with the uploaded script file.

#### Example 8-170. Add script parameters

---

Request:

POST https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/script/

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 8-171. Modify script parameters

---

Request:

PUT https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/script/*scriptId*

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<logonLogoffScript>
  <scriptId>logonlogoffscriptfile-12</scriptId>
  <type>BOTH</type>
  <description>Testing modify script</description>
  <enabled>>false</enabled>
</logonLogoffScript>
```

---

### Query Script Configuration

Retrieves parameters associated with the specified script file ID.

**Example 8-172. Get script parameters**

---

Request:

GET https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/script/*scriptId*

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<logonLogoffScript>
  <objectId>logonlogoffscript-1</objectId>
  <scriptId>logonlogoffscriptfile-12</scriptId>
  <type>BOTH</type>
  <description>Testing modify script</description>
  <scriptIdUri>https://vsm-ip/api/4.0/edges/edge-id/sslvpn/config/script/file/scriptId/scriptId</scriptIdUri>
  <enabled>>false</enabled>
</logonLogoffScript>
```

---

**Query All Script Configurations**

Retrieves all script configurations for the specified NSX Edge.

**Example 8-173. Get all script parameters**

---

Request:

GET https://&lt;vsm-ip&gt;/api/4.0/edges/&lt;edgeId&gt;/sslvpn/config/script/

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 8-174. Delete script parameters**

---

Request:

DELETE https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/script/*scriptId***Delete All Script Configuragtions**

Deletes all script configurations for the specified NSX Edge.

**Example 8-175. Delete script parameters**

---

Request:

DELETE https://&lt;vsm-ip&gt;/api/4.0/edges/&lt;edgeId&gt;/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 8-176. Apply script configurations

Request:

PUT https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/script/

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 8-177. Reconfigure SSL VPN

Request:

PUT https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 -->
    <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>
    <!-- Optional Parameters-->
    <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>
    <!--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-->
      <!-- 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 -->
    <primaryAuthServers>
      <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>
    <!-- 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 -->

```

```

<!--<com.vmware.vshield.edge.sslvpn.dto.RsaAuthServerDto>
<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>
                                                    <!--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 8-178. Query SSL VPN Configuration

Request:

GET https://vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/

Response Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<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>
    </privateNetwork>
  </privateNetworks>
</sslvpnConfig>

```



```

        <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>
<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</enforceServerSecurityCertValidation>
        <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>
    <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>
        <!--<com.vmware.vshield.edge.sslvpn.dto.RsaAuthServerDto>
        <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 8-179. Delete SSL VPN Configuration

---

Request:

DELETE https://<vsm-ip>/api/4.0/edges/<edgeId>/sslvpn/config/

---

## Query SSL VPN Statistics

Retrieves SSL VPN statistics on the specified NSX Edge.

### Example 8-180. Get SSL VPN statistics

---

Request:

GET https://<vsm-ip>/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:

```

<?xml version="1.0" encoding="UTF-8"?>
<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>

```

---

## 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 8-181. Configure L2VPN for Server

---

Request:

PUT https://<nsxmgr-ip>/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 /> <!-- 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 8-182. Configure L2VPN for Client

---

Request:

PUT <https://<nsxmgr-ip>/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 -->
          <vnics>10</vnics> <!-- Required. Traffic from this internal vnic interfaces will be forwarded to L2VPN tunnel -->
          <vnics>11</vnics>
          <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>
        <proxySetting> <!-- Optional, List of proxy configurations -->
          <type>https</type>
          <address>10.112.243.202</address>
          <port>443</port>
          <userName>root</userName>
          <password>java123</password>
        </proxySetting>
      </client>
    </l2VpnSite>
  </l2VpnSites>
</l2Vpn>

```

```

    <l2VpnUser> <!-- Required. these credentials will be used to get authenticated by server-->
      <userId>apple</userId>
      <password>apple</password>
    </l2VpnUser>
  </client>
</l2VpnSite>
</l2VpnSites>
</l2Vpn>

```

---

## Query L2VPN

Retrieves the current L2VPN configuration for NSX Edge.

### Example 8-183. Query L2VPN

---

Request:

GET https://<nsxmgr-ip>/api/4.0/edges/{edgeId}/l2vpn/config/

Response Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<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 , recieved bytes etc. for the given edge.

### Example 8-184. Query L2VPN statistics

---

Request:

GET https://<nsxmgr-ip>/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 8-185. Enable L2VPN

---

Request:

POST https://<nsxmgr-ip>/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 8-186. Delete L2VPN

---

Request:

DELETE https://<nsxmgr-ip>/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 an 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 8-187. Configure IPSEC VPN

Request:

PUT https://<vsm-ip>/api/4.0/edges/<edgeId>/ipsec/config

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<ipsec>
  <enabled>true</enabled> <!-- Optional, true by default -->
  <logging> <!-- optional. logging is disabled 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>
  </sites>
  <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 8-188. Get IPSec Configuration

---

Request:

GET https://<vsm-ip>/api/4.0/edges/<edgeId>/ipsec/config

Response Body when IPSec is not configured:

```

<?xml version="1.0" encoding="UTF-8"?>
<ipsec>
  <enabled>true</enabled>
  <logging>
    <enable>true</enable>
    <logLevel>debug</logLevel>
  </logging>
  <sites/> <!-- No site to site config present -->
</ipsec>

```

Response Body when IPSec is configured for site-to-site:

```

<?xml version="1.0" encoding="UTF-8"?>
<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>
    </site>
  </sites>
</ipsec>

```

```

    <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 8-189. Get IPSEC statistics

---

Request:

GET https://<vsm-ip>/api/4.0/edges/<edgeId>/ipsec/statistics

Response Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<?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 8-190. Get tunnel traffic statistics

---

Request:

GET https://<vsm-ip>/api/4.0/edges/<edgeId>/statistics/dashboard/ipsec?interval=<range>

Response Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<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>

```

```

        <value>0.0</value>
      </dashboardStatistic>
    </ipsecBytesOut>
  </ipsec>
</data>
</dashboardStatistics>

```

---

### Delete IPSec Configuration

Deletes the IPSEC configuration for the specified NSX Edge.

#### Example 8-191. Delete IPSec

---

Request:

DELETE https://<vsm-ip>/api/4.0/edges/<edgeId>/ipsec/config/

---

## Managing an NSX Edge

### Force Sync Edge

Re-synchronizes the NSX Edge virtual machines.

#### Example 8-192. Force sync Edge

---

Request:

POST https://<nsxmgr-ip>/api/4.0/edges/{edgeId}?action=forcesync

---

### Redeploy Edge

Redeploys NSX Edge virtual machines.

#### Example 8-193. Redeploy Edge

---

Request:

POST https://<nsxmgr-ip>/api/4.0/edges/{edgeId}?action=redeploy

---

### Update DNS Settings

Update dns settings (primary/secondary and search domain) of an Edge.

#### Example 8-194. Update DNS

---

Request:

PUT https://<nsxmgr-ip>/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 8-195. Modify AESNI

---

Request:

```
POST https://<nsxmgr-ip>/api/4.0/edges/{edgeId}/aesni?enable=true|false
```

---

## Modify Edge Appliance Core Dump Setting

Enabling Edge appliance core-dump feature results in deployment of an inbuilt extra disk to save the core-dump files. The extra disk consumes 1GB for compact edge and 8GB for other edge types. Disabling this feature detaches the disk.

### Example 8-196. Modify core dump setting

---

Request:

```
POST https://<nsxmgr-ip>/api/4.0/edges/{edgeId}/coredump?enable=true|false
```

---

## Modify FIPs Setting

### Example 8-197. Modify FIPs

---

Request:

```
POST https://<nsxmgr-ip>/api/4.0/edges/{edgeId}/fips?enable=true|false
```

---

## Modify Log Setting

### Example 8-198. Modify log setting

---

Request:

```
POST https://<nsxmgr-ip>/api/4.0/edges/{edgeId}/logging?level=<logLevel>
```

---

## Query Edge Summary

Retrieves details about the specified Edge.

### Example 8-199. Retrieve Edge details

---

Request:

```
GET https://<nsxmgr-ip>/api/4.0/edges/{edgeId}/summary
```

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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>
  <featureCapability>
    <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>

```

&lt;/edgeSummary&gt;

## Query Edge Status

Retrieves the status of the specified Edge.

### Example 8-200. Query status

Request:

GET https://<nsxmgr-ip>/api/4.0/edges/{edgeId}/status

Request 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>
    <featureStatus>
      <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</lastPublishedPreRulesGenerationNumber>

```

Sample calls include:

```

GET https://<nsxmgr-ip>/api/4.0/edges/{edgeId}/status?getlatest=false&detailed=true
GET https://<nsxmgr-ip>/api/4.0/edges/{edgeId}/status?getlatest=true&detailed=true
GET https://<nsxmgr-ip>/api/4.0/edges/{edgeId}/status?getlatest=false&detailed=false
GET https://<nsxmgr-ip>/api/4.0/edges/{edgeId}/status?detailed=true
GET https://<nsxmgr-ip>/api/4.0/edges/{edgeId}/status?getlatest=false
GET https://<nsxmgr-ip>/api/4.0/edges/{edgeId}/status?preRulesStatus=true

```

## Query Edge Tech Support Logs

Retrieves the tech support logs for the specified Edge.

### Example 8-201. Query tech support logs

---

Request:

```
GET https://<nsxmgr-ip>/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 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 8-202. Modify CLI credentials

---

Request:

PUT https://<nsxmgr-ip>/api/4.0/edges/{edgeId}/clisettings

Request Body:

```
<cliSettings> <!-- optional. Default user/pass is admin/default, 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>          <!-- optional. pre-formatted text. default to system default banner text. -->
  Hello, VshieldEdge Administrator
  </sshLoginBannerText>
</cliSettings>
```

---

### Change CLI Remote Access

Enables or disables the SSH service on the specified Edge Edge.

#### Example 8-203. Change CLI remote access

---

Request:

POST https://<nsxmgr-ip>/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 8-204. Modify auto configuration settings

---

PUT https://<nsxmgr-ip>/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 8-205. Retrieve auto configuration settings

---

GET https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/autoconfiguration

Response Body:

```

<autoConfiguration>
  <enabled>true</enabled>
  <rulePriority>high</rulePriority>
</autoConfiguration>

```

---

## Working with Appliances

You can manage the Edge 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.

### Example 8-206. Get appliance configuration

---

Request:

GET https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/appliances

Response Body:

```

<appliances>
  <applianceSize>large</applianceSize>
  <appliance>
    <highAvailabilityIndex>0</highAvailabilityIndex>
    <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>
  <appliance>
    <highAvailabilityIndex>1</highAvailabilityIndex>
    <resourcePoolId>resgroup-53</resourcePoolId>
    <datastoreId>datastore-29</datastoreId>
    <hostId>host-28</hostId>
  </appliance>
</appliances>

```

```

    <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>
</appliances>

```

---

## Modify Appliance Configuration

You can retrieve the configuration of both appliances by using the GET call in [Example 8-206](#) 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 8-207. Modify appliance configuration

---

Request:

PUT https://<nsxmgr-ip>/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 8-208. Change appliance size

---

Request:

POST https://<nsxmgr-ip>/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 8-209.** Get configuration of appliance with specified haIndex

---

Request:

GET https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/appliances/haIndex

Response 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>

```

---

**Modify Appliance**

Modifies the configuration of the appliance with the specified haIndex.

**Example 8-210.** Modify configuration of appliance with specified haIndex

---

Request:

PUT https://<nsxmgr-ip>/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>

```

---

**Delete Appliance**

Deletes the appliance with the specified haIndex.

**Example 8-211. Delete appliance configuration**

Request:

DELETE https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/appliances/*haIndex***Working with Interfaces**

You can add up to ten internal or uplink interfaces to each Edge Edge instance. A Edge 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 Edge instance, the specified configuration is applied to the appliance as well.

**Example 8-212. Add an interface or sub interface**

Request:

POST https://&lt;nsxmgr-ip&gt;/api/4.0/edges/&lt;edgeId&gt;/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>400000000</averageBandwidth>
  <peakBandwidth>400000000</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>
  </subInterfaces>
</vnic>

```

```

    <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 8-213. Retrieve all interfaces

---

Request:

GET https://<nsxmgr-ip>/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 8-214. Retrieve interface

---

Request:

GET https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/vnics/<index>

Response Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<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 8-215. Modify interface

Request:

PUT https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/vnics/<index>

Response Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<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> <!--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 -->
    </subInterface>
  </subInterfaces>
</vnic>

```

```

<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 8-216. Delete interface

---

Request:

```
DELETE https://<nsxmgr-ip>/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 Edge.

### Example 8-217. Get interface with specific index

---

Request:

```
GET https://<nsxmgr-ip>/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>
      <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 8-218. Modify interface

---

Request:

PUT https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/vnics/<index>

Response 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 virtualWire-id. 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 8-219.** Delete interface configuration

Request:

DELETE https://&lt;nsxmgr-ip&gt;/api/4.0/edges/&lt;edgeId&gt;/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 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 8-220.** Get interface statistics

Request:

GET https://&lt;nsxmgr-ip&gt;/api/4.0/edges/&lt;edgeId&gt;/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 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 8-221.** Get uplink interface statistics

Request:

GET https://&lt;nsxmgr-ip&gt;/api/4.0/edges/&lt;edgeId&gt;/statistics/interfaces/uplink

Response Body:

&lt;statistics&gt;

```

    <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 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 8-222. Get internal interface statistics

---

Request:

GET https://<nsxmgr-ip>/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 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 8-223. Get interface statistics

---

Request:

GET https://<nsxmgr-ip>/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>
</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 9-1.** 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 9-1.** 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
IPv4 addresses	Ipv4Address	source/destination
IPv6 addresses	Ipv6Address	source/destination
logical switch	VirtualWire	source/destination appliedTo
network	Network	appliedTo
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 60. For information on creating a security group, see [“Working with Security Groups”](#) on page 53.

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

This chapter includes the following topics:

- [“Configuring Firewall”](#) on page 277
- [“Working with Firewall Sections”](#) on page 281
- [“Working with Firewall Rules”](#) on page 286
- [“Working with Layer3 Redirect Sections and Rules”](#) on page 289
- [“Synchronizing and Enabling Firewall”](#) on page 300
- [“Importing and Exporting Firewall Configurations”](#) on page 301
- [“Working with Local Rules”](#) on page 305
- [“Firewall Migration Switch”](#) on page 317
- [“Configuring Fail-Safe Mode for Distributed Firewall”](#) on page 318
- [“Working with SpoofGuard”](#) on page 318
- [“Getting Flow Statistic Details”](#) on page 321
- [“Flow Exclusion”](#) on page 327
- [“Working with IPFix”](#) on page 328
- [“Excluding Virtual Machines from Firewall Protection”](#) on page 329

---

**IMPORTANT** All NSX REST requests require authentication. See [“Using the NSX REST API”](#) on page 25 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 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.

### Example 9-1. Get firewall configuration for NSX Manager

---

Request:

GET https://<nsxmgr-ip>/api/4.0/firewall/globalroot-0/config

Response Body:

```
HTTP/1.1 200 OK
Cache-Control: private
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Cache-Control: no-cache
Set-Cookie: JSESSIONID=4CAE025C868939C35245B2553079807A; Path=/
ETag: 1395341576368
Date: Wed, 02 Oct 2013 20:58:39 GMT
Server: vShield Manager
Content-Type: application/xml
Transfer-Encoding: chunked
<?xml version="1.0" encoding="UTF-8"?>
<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>
      </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>
      </rule>
    </section>
  </layer2Sections>
</firewallConfiguration>
```

```

    </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 9-2. Filter firewall configuration

---

Request:

GET

```
https://<nsxmgr-ip>/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 ruletypes. This parameter is mandatory if any of the other parameters is specified.
- source/destination can contain IPv4/v6 address or vm-id.
- enable can be true/false/any. Specifying any is equivalent to ignoring the field.
- logging can be true/false/any. Specifying any is equivalent to ignoring the field.
- action can be allow/deny/reject/any for L3 rules and allow/redirect/any for L3 redirect rules.
- 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 9-3. Filter rules for an Edge

---

Request:

```
GET https://<nsxmgr-ip>/api/4.0/firewall/globalroot-0/config?edgeId=<edgeId>
```

---

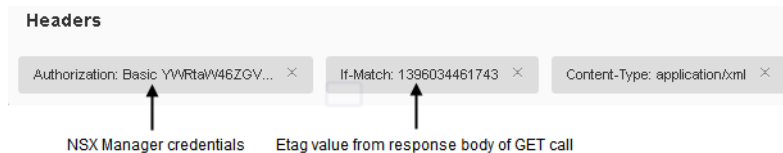
## Modify Firewall Configuration

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.

Response Headers	Response Body (Raw)	Response Body (Highlight)	Response Body (Preview)
<ol style="list-style-type: none"> <li>1. Status Code : 200 OK</li> <li>2. Cache-Control : private, no-cache</li> <li>3. Content-Type : application/xhtml+xml</li> <li>4. Date : Tue, 01 Apr 2014 20:04:58 GMT</li> <li>5. Etag : 1396034461743</li> <li>6. Expires : Thu, 01 Jan 1970 00:00:00 GMT</li> <li>7. Server : NSX Manager</li> <li>8. Set-Cookie : JSESSIONID=AD4EA9CF06190B80B113D79B83C67535; Path=/ 9. Transfer-Encoding : chunked</li> </ol>			

- 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 9-4. Modify firewall configuration

Request:

```
PUT https://<nsxmgr-ip>/api/4.0/firewall/globalroot-0/config
--header 'Content-Type:text/xml' --header 'if-match:"1396034461743"
```

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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>ClusterComputeResource</type>
          </source>
          <source>
            <value>10.112.1.1</value>
            <type>Ipv4Address</type>
          </source>
        </sources>
        <services>
          <service>
            <destinationPort>80</destinationPort>
          </service>
        </services>
      </rule>
    </section>
  </layer3Sections>
</firewallConfiguration>
```

```

        <protocol>6</protocol>
        <subProtocol>6</subProtocol>
    </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>
        </rule>
    </section>
</layer2Sections>
</firewallConfiguration>

```

---

where:

- appliedTo can be DISTRIBUTED\_FIREWALL, ALL\_EDGES, ANY, edgeIds, cluster, vm, vnic, dvportGroups.
- 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 9-5. Delete firewall configuration

Request:

DELETE https://<nsxmgr-ip>/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. There are two kinds of sections

- 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 289.

## Query Firewall Sections

Retrieves section configuration either by section ID or section name.

### Example 9-6. Get section configuration

Request:

GET https://<nsxmgr-ip>/api/4.0/firewall/globalroot-0/config/layer3sections|layer2sections/<sectionId> |<sectionName>

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 9-7. Add section

---

Request:

POST https://<nsxmgr-ip>/api/4.0/firewall/globalroot-0/config/layer3sections|layer2sections

Request Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<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>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>
</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.

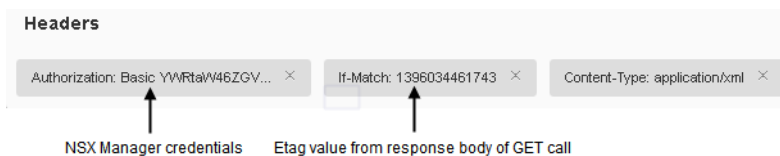
## Modify Firewall Section

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.

Response Headers	Response Body (Raw)	Response Body (Highlight)	Response Body (Preview)
<pre> 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 8. Set-Cookie       : JSESSIONID=AD4EA9CF06190B80B113D79B83C67535; Path=/ 9. Transfer-Encoding : chunked </pre>			

- 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 9-8. Modify section

---

Request:

```
PUT https://<nsxmgr-ip>/api/4.0/firewall/globalroot-0/config/layer3sections|layer2sections/<sectionId>|<sectionName>
--header 'Content-Type:text/xml' --header 'if-match:"1396034461743"
```

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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>
      <service>
        <name>VMware-VDM2.x-Ephemeral</name>
        <value>application-161</value>
```

```

        <isValid>true</isValid>
    </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 9-9. Delete section

---

Request:

```
DELETE https://<nsxmgr-ip>/api/4.0/firewall/globalroot-0/config/layer3sections|layer2sections/<sectionId> |<sectionName>
```

---

## 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.

## Query Firewall Rule

Retrieves rule details from either a Layer3 or Layer2 section.

### Example 9-10. Get firewall rule

---

Request:

```
GET https://<nsxmgr-ip>/api/4.0/firewall/globalroot-0/config/layer3sections|layer2sections/<sectionId>/rules/<ruleId>
```

Response Body:

```

HTTP/1.1 200 OK
Cache-Control: private
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Cache-Control: no-cache
Set-Cookie: JSESSIONID=FED4857DF7A2A5CCD7F818A87F463629; Path=/
ETag: 1380747467905
Date: Wed, 02 Oct 2013 21:04:29 GMT
Server: vShield Manager
Content-Type: application/xml
Transfer-Encoding: chunked
<?xml version="1.0" encoding="UTF-8"?>
<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.

### Example 9-11. Add firewall rule

Request:

POST https://<nsxmgr-ip>/api/4.0/firewall/globalroot-0/config/layer3sections|layer2sections/<sectionId>/rules

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<rule disabled="false" logged="false">
  <name>AddRuleTest</name>
  <action>allow</action>
  <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.

## Modify Firewall Rule

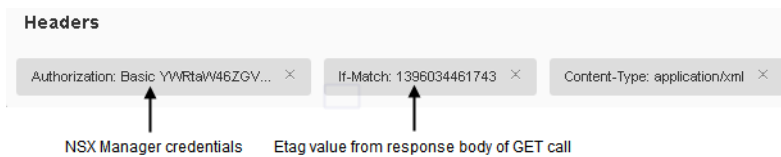
Modifies a rule in the Layer2 or Layer3 section.

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.

Response Headers	Response Body (Raw)	Response Body (Highlight)	Response Body (Preview)
<pre> 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 8. Set-Cookie      : JSESSIONID=AD4EA9CF06190B80B113D79B83C67535; Path=/ 9. Transfer-Encoding : chunked </pre>			

- 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 9-12. Modify firewall rule

---

Request:

```
PUT https://<nsxmgr-ip>/api/4.0/firewall/globalroot-0/config/layer3sections/layer3sections/<sectionId>/rules
--header 'Content-Type:text/xml' --header 'if-match:"1380747467905"'
```

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<rule id="23" disabled="enabled" logged="true">
  <name>AddRuleTestUpdated</name>
  <action>allow</action>
  <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.

## 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 Section

Retrieves layer3 redirect section configuration either by section ID or section name.

#### Example 9-13. Get layer3 redirect section configuration

---

Request:

```
GET https://<nsxmgr-ip>/api/4.0/firewall/globalroot-0/config/layer3redirectsections?<sectionId> |<name=sectionName>
```

Response Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<section id="1007" name="TestSectionTraffSteerPUT" generationNumber="1393905701256" timestamp="1393905701256">
  <name>test-TSrule</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>1007</sectionId>
</rule>
</section>

```

---

## Add Layer3 Redirect Section

Adds layer3 redirect section configuration.

### Example 9-14. Add layer3 redirect section configuration

---

Request:

POST https://<nsxmgr-ip>/api/4.0/firewall/globalroot-0/config/layer3redirectsections

Request Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<name>test-TSrule</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>1007</sectionId>
</rule>
</section>

```

---

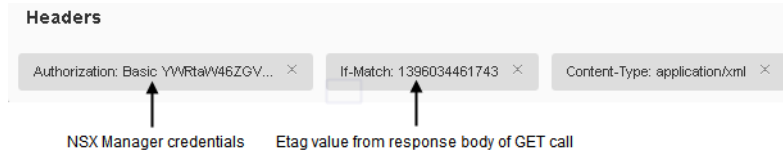
## Modify Layer3 Redirect Section

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.

Response Headers	Response Body (Raw)	Response Body (Highlight)	Response Body (Preview)
<pre> 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 8. Set-Cookie      : JSESSIONID=AD4EA9CF06190B80B113D79B83C67535; Path=/ 9. Transfer-Encoding : chunked </pre>			

- 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 9-15. Modify layer3 redirect section

Request:

```
PUT https://<nsxmgr-ip>/api/4.0/firewall/globalroot-0/config/layer3redirectsections/<sectionId> |<sectionName>
--header 'Content-Type:text/xml' --header 'if-match:"1396034461743"
```

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<section id="1007" name="TestSectionTraffSteerPUT" generationNumber="1393905701256" timestamp="1393905701256">
  <rule id="1008" disabled="false" logged="true">
    <name>test-TSrule</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>1007</sectionId>
  </rule>
</section>
```

## Delete Layer3 Redirect Section

Deletes the specified section.

#### Example 9-16. Delete layer3 redirect section

Request:

```
DELETE https://<nsxmgr-ip>/api/4.0/firewall/globalroot-0/config/layer3redirectsections/<sectionId> |<sectionName>
```

## Query Layer3 Redirect Rules

Retrieves layer3 redirect rules for the specified section.

**Example 9-17. Get layer3 redirect rules**

---

Request:

GET https://<nsxmgr-ip>/api/4.0/firewall/globalroot-0/config/layer3redirectsections/<sectionId>/rules/<ruleID>

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<rule id="1008" disabled="false" logged="true">
  <name>test-TSrule</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>
</rule>
```

---

**Add Layer3 Redirect Rule**

Adds layer3 redirect rule.

**Example 9-18. Add layer3 redirect rule**

---

Request:

POST https://<nsxmgr-ip>/api/4.0/firewall/globalroot-0/config/layer3redirectsections/<sectionId>/rules

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<section id="1004" name="TestSectionTS" generationNumber="1394650573090" timestamp="1394650573090">
  <name>test-TS</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>1004</sectionId>
</rule>
</section>
```

---

**Modify Layer3 Redirect Rule**

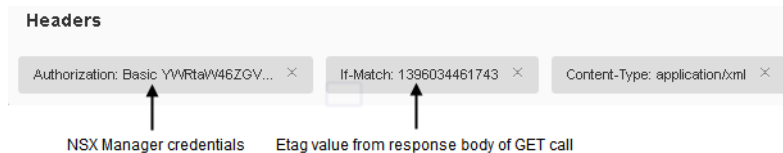
Modifies a layer3 redirect rule.

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.

Response Headers	Response Body (Raw)	Response Body (Highlight)	Response Body (Preview)
<ol style="list-style-type: none"> <li>1. Status Code : 200 OK</li> <li>2. Cache-Control : private, no-cache</li> <li>3. Content-Type : application/xhtml+xml</li> <li>4. Date : Tue, 01 Apr 2014 20:04:58 GMT</li> <li>5. Etag : 1396034461743</li> <li>6. Expires : Thu, 01 Jan 1970 00:00:00 GMT</li> <li>7. Server : NSX Manager</li> <li>8. Set-Cookie : JSESSIONID=AD4EA9CF06190B80B113D79B83C67535; Path=/</li> <li>9. Transfer-Encoding : chunked</li> </ol>			

- 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 9-19. Modify layer3 redirect rule

Request:

```
PUT https://<nsxmgr-ip>/api/4.0/firewall/globalroot-0/config/layer3redirectsections/<sectionId>/rules/<ruleId>'
--header 'Content-Type:text/xml' --header 'if-match:"1396034461743"'
```

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<rule id="1008" disabled="true" logged="true">
  <name>test-TSrulePUTedited</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>
</rule>
```

## Delete Layer3 Redirect Rule

Deletes the specified layer3 redirect rule.

#### Example 9-20. Delete layer3 redirect rule

Request:

DELETE https://<nsxmgr-ip>/api/4.0/firewall/globalroot-0/config/layer3redirectsections/<sectionId>/rules/<ruleId>

---

## Query Service Insertion Profiles

Retrieves the Service Insertion profiles that can be applied to layer3 redirect rules.

### Example 9-21. Query Service Insertion profiles

---

Request:

GET https://<nsxmgr-ip>/api/4.0/firewall/globalroot-0/config/layer3redirect/profiles

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<serviceProfiles>
  <serviceProfile>
    <objectId>serviceprofile-1</objectId>
    <objectTypeName>ServiceProfile</objectTypeName>
    <vsmUuid>4231191D-91DA-B6E9-6555-B4E59C9DE677</vsmUuid>
    <revision>0</revision>
    <type>
      <typeName>ServiceProfile</typeName>
    </type>
    <name>asd</name>
    <description />
    <clientHandle />
    <extendedAttributes />
    <profileAttributes>
      <id>290</id>
      <revision>1</revision>
      <attribute>
        <id>291</id>
        <revision>0</revision>
        <key>failOpen</key>
        <name>Failure Policy</name>
        <value>true</value>
      </attribute>
    </profileAttributes>
  </service>
  <service>
    <objectId>service-7</objectId>
    <objectTypeName>Service</objectTypeName>
    <vsmUuid>4231191D-91DA-B6E9-6555-B4E59C9DE677</vsmUuid>
    <revision>1</revision>
    <type>
      <typeName>Service</typeName>
    </type>
    <name>Sitest</name>
    <clientHandle />
    <extendedAttributes />
  </service>
  <serviceInstance>
    <objectId>serviceinstance-1</objectId>
    <objectTypeName>ServiceInstance</objectTypeName>
    <vsmUuid>4231191D-91DA-B6E9-6555-B4E59C9DE677</vsmUuid>
    <revision>1</revision>
    <type>
      <typeName>ServiceInstance</typeName>
    </type>
    <name>Sitest-GlobalInstance</name>
    <clientHandle />
    <extendedAttributes />
  </serviceInstance>
  <status>IN_SERVICE</status>
  <runtime>
    <nonCompliantDvpg />
    <nonCompliantVwire />
  </runtime>
</serviceProfiles>
```

```

</runtime>
<serviceProfileBinding>
  <distributedVirtualPortGroups />
  <virtualWires />
  <excludedVnics />
  <virtualServers />
</serviceProfileBinding>
<vendorTables />
<vendorSections />
</serviceProfile>
<serviceProfile>
  <objectId>serviceprofile-2</objectId>
  <objectTypeName>ServiceProfile</objectTypeName>
  <vsmUuid>4231191D-91DA-B6E9-6555-B4E59C9DE677</vsmUuid>
  <revision>0</revision>
  <type>
    <typeName>ServiceProfile</typeName>
  </type>
  <name>jkl</name>
  <description />
  <clientHandle />
  <extendedAttributes />
  <profileAttributes>
    <id>293</id>
    <revision>1</revision>
    <attribute>
      <id>294</id>
      <revision>0</revision>
      <key>failOpen</key>
      <name>Failure Policy</name>
      <value>true</value>
    </attribute>
  </profileAttributes>
  <service>
    <objectId>service-7</objectId>
    <objectTypeName>Service</objectTypeName>
    <vsmUuid>4231191D-91DA-B6E9-6555-B4E59C9DE677</vsmUuid>
    <revision>1</revision>
    <type>
      <typeName>Service</typeName>
    </type>
    <name>Sitest</name>
    <clientHandle />
    <extendedAttributes />
  </service>
  <serviceInstance>
    <objectId>serviceinstance-1</objectId>
    <objectTypeName>ServiceInstance</objectTypeName>
    <vsmUuid>4231191D-91DA-B6E9-6555-B4E59C9DE677</vsmUuid>
    <revision>1</revision>
    <type>
      <typeName>ServiceInstance</typeName>
    </type>
    <name>Sitest-GlobalInstance</name>
    <clientHandle />
    <extendedAttributes />
  </serviceInstance>
  <status>IN_SERVICE</status>
</runtime>
<nonCompliantDvpg />
<nonCompliantVwire />
</runtime>
<serviceProfileBinding>
  <distributedVirtualPortGroups />
  <virtualWires />
  <excludedVnics />
  <virtualServers />
</serviceProfileBinding>

```

```

    <vendorTables />
    <vendorSections />
  </serviceProfile>
</serviceProfiles>

```

---

## Query Status

Retrieves status of the entire firewall configuration or individual sections.

### Query Firewall Configuration Status

#### Example 9-22. Get firewall configuration status

---

Request:

GET https://<nsxmgr-ip>/api/4.0/firewall/globalroot-0/status

Response Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<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

### Example 9-23. Get Layer3 status

---

Request:

GET https://<nsxmgr-ip>/api/4.0/firewall/globalroot-0/status/layer3sections/<sectionId>

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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

### Example 9-24. Get layer2 status

---

Request:

GET https://<nsxmgr-ip>/api/4.0/firewall/globalroot-0/status/layer2sections/<sectionId>

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 9-25. Configure thresholds

---

Request:

PUT https://<nsxmgr-ip>/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 9-26. Query thresholds

---

Request:

GET https://<nsxmgr-ip>/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 9-27. Set RuleOptimize and TCPStrict flags

---

Request:

PUT https://<nsxmgr-ip>/api/4.0/firewall/config/globalconfiguration

Request Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<globalConfiguration>
  <layer3RuleOptimize>true</layer3RuleOptimize>
  <layer2RuleOptimize>>false</layer2RuleOptimize>
  <tcpStrictOption>true</tcpStrictOption>
</globalConfiguration>

```

---

### Example 9-28. Query RuleOptimize and TCPStrict flags

---

Request:

GET https://<nsxmgr-ip>/api/4.0/firewall/config/globalconfiguration

Response Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<globalConfiguration>
  <layer3RuleOptimize>true</layer3RuleOptimize>
  <layer2RuleOptimize>>false</layer2RuleOptimize>
  <tcpStrictOption>true</tcpStrictOption>

```

&lt;/globalConfiguration&gt;

## 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 9-29. Force sync host

Request:

POST https://<nsxmgr-ip>/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 9-30. Force sync cluster

Request:

POST https://<nsxmgr-ip>/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 9-31. Enable or disable API

Request:

POST https://<nsxmgr-ip>/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 9-32. Save a firewall configuration

---

Request:

POST https://<nsxmgr-ip>/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
<?xml version="1.0" encoding="UTF-8"?>
<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 9-33. Get all saved firewall configurations

---

Request:

GET https://<nsxmgr-ip>/api/4.0/firewall/globalroot-0/drafts/

**Request 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 301.

**Example 9-34. Get a saved firewall configuration****Request:**

GET https://<nsxmgr-ip>/api/4.0/firewall/globalroot-0/drafts/<draftID>

**Request Body:**

```

<?xml version="1.0" encoding="UTF-8"?>
<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 301.

**Example 9-35.** Update a saved firewall configuration

---

Request:

PUT https://<nsxmgr-ip>/api/4.0/firewall/globalroot-0/drafts/<draftID>

Request Body:

```

<f<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 301.

**Example 9-36. Delete a saved firewall configuration**

---

Request:

DELETE https://&lt;nsxmgr-ip&gt;/api/4.0/firewall/globalroot-0/drafts/&lt;draftID&gt;

**Export a Saved Configuration**Retrieve the draftID of the configuration. See [“Get all saved firewall configurations”](#) on page 301.**Example 9-37. Export a saved firewall configuration**

---

Request:

GET https://&lt;nsxmgr-ip&gt;/api/4.0/firewall/globalroot-0/drafts/&lt;draftID&gt;/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 301.Use the response body of the export command as the request body in this command. See [“Export a saved firewall configuration”](#) on page 304.**Example 9-38. Import a saved firewall configuration**

---

Request:

POST https://&lt;nsxmgr-ip&gt;/api/4.0/firewall/globalroot-0/drafts/&lt;draftID&gt;/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 Local Rules

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.

### Example 9-39. Configure firewall

Request:

PUT https://<nsxmgr-ip>/api/4.0/edges/{edgeId}/firewall/config

Request Body:

```

<?xml version="1.0"?>
<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>3600</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 -->
</globalConfig>
<rules>
  <rule>
    <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 -->
      <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 -->
  </rule>
  <rule>
    ...
  </rule>
  ....
</rules>
</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 9-40. Query firewall**

Request:

GET https://<nsxmgr-ip>/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>3600</tcpTimeoutEstablished>
    <tcpTimeoutClose>30</tcpTimeoutClose>
    <udpTimeout>60</udpTimeout>
    <icmpTimeout>10</icmpTimeout>
    <icmp6Timeout>10</icmp6Timeout>
    <ipGenericTimeout>120</ipGenericTimeout>
  </globalConfig>
  <rules>
    <rule>
      <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>
    </rule>
    <rule>
      <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>
    </rule>
    <rule>
      <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>
  <vnidGroupId>vse</vnidGroupId>
  <vnidGroupId>external</vnidGroupId>
</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>
</rule>
<rule>
  <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>
</rule>
<rule>
  <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>
</rule>
</rules>
</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 9-41. Query firewall

Request:

GET https://<nsxmgr-ip>/api/4.0/edges/{edgeId}/firewall/config

---

## Append Firewall Rules

Adds one or more rules below the existing rules in the rules table.

### Example 9-42. Add firewall rule

---

POST https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/firewall/config/rules

Request Body:

```
<rules>
<rule>
  <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 -->
</rule>
<rule>
...
</rule>
```

---

### 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 9-43. Add a rule above a specific rule

---

Request:

POST https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/firewall/config/rules?aboveRuleId=<ruleId>

Request Body:

```
<rule>
  <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 -->
</rule>

```

---

## Query Specific Rule

### Example 9-44. Retrieve specific rule

GET https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/firewall/config/rules/<ruleId>

Response Body:

```

<rule>
  <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/>
</rule>

```

---

## Modify Firewall Rule

You can modify a rule by specifying its rule ID. Note that only local rules can be modified at the NSX Edge level.

### Example 9-45. Update specific rule

PUT https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/firewall/config/rules/<ruleId>

Response Body:

```

<rule>
  <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 -->
<vnidGroupId>vnid-index-5</vnidGroupId> <!-- Possible values are "vnid-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 -->
</rule>

```

---

## 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 9-46. Delete firewall rule

Request Body:

```
DELETE https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/firewall/config/rules/<ruleId>
```

---

## Delete Firewall Configuration

Deletes firewall configuration for Edge.

### Example 9-47. Delete firewall configuration

Request:

```
DELETE https://<nsxmgr-ip>/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 9-48. Query global firewall configuration

Request:

```
GET https://<nsxmgr-ip>/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 9-49. Modify global firewall configuration

---

Request:

PUT https://<nsxmgr-ip>/api/4.0/edges/{edgeId}/firewall/config/global

Response 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>3600</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 -->
</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 9-50. Query default firewall configuration

---

Request:

GET https://<nsxmgr-ip>/api/4.0/edges/{edgeId}/firewall/config/defaultpolicy

Response Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<firewallDefaultPolicy>
  <action>ACCEPT</action>
  <loggingEnabled>true</loggingEnabled>
</firewallDefaultPolicy>

```

---



## Modify Default Firewall Policy

Configures default firewall policy for the specified Edge.

### Example 9-51. Modify default firewall configuration

---

Request:

PUT https://<nsxmgr-ip>/api/4.0/edges/{edgeId}/firewall/config/defaultpolicy

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<firewallDefaultPolicy>
  <action>ACCEPT</action>
  <loggingEnabled>true</loggingEnabled>
</firewallDefaultPolicy>
```

---

## Query Firewall Statistics

Retrieves number of ongoing connections for the firewall configuration.

### Example 9-52. Query firewall statistics

---

Request:

GET https://<nsxmgr-ip>/api/4.0/edges/{edgeId}/firewall/statistics/dashboard/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 9-53. Query statistics for a rule

---

Request:

GET https://<nsxmgr-ip>/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 9-54. Disable Firewall

---

Request:

```
PUT https://<nsxmgr-ip>/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 Edge in the Edge Manager user interface and in the vSphere Client plug-in.

### Example 9-55. Configure SNAT and DNAT rules for a Edge Edge

---

```
PUT https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/nat/config
```

```
<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>
      <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>
      <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>
```

</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 a Edge Edge

### Example 9-56. Query SNAT and DNAT rules for a Edge Edge

GET https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/nat/config

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<nat>
  <natRules>
    <natRule>
      <ruleTag>196609</ruleTag>
      <ruleId>196609</ruleId>
      <action>dnat</action>
      <vnic>0</vnic>
      <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>
      <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>
```

---

```
</nat>
```

---

### Delete all NAT Rules

Deletes all SNAT and DNAT rules for a Edge Edge. The auto plumbed rules continue to exist.

#### Example 9-57. Delete NAT rules

---

Request:

```
DELETE https://<nsxmgr-ip>/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 9-58. Add a NAT rule above a specific rule

---

```
POST https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/nat/config/rules?aboveRuleId=<ruleId>
```

Request Body:

```
<natRule>
  <action>dnat</action>
  <vnic>0</vnic>
  <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 9-59. Add NAT rules to the bottom of the rules table

---

```
POST https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/nat/config/rules
```

Response Body:

```
<natRules>
  <natRule>
    <action>dnat</action>
    <vnic>0</vnic>
    <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 Edge (0-9).

## Modify a NAT Rule

Replaces the NAT rule with the specified rule ID.

### Example 9-60. Replaces a NAT rule

---

PUT https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/nat/config/rules/*ruleID*

Response Body:

```
<natRule>
  <action>dnat</action>
  <vnic>0</vnic>
  <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 Edge (0-9).

## Delete a NAT Rule

Deletes the rule with the specified rule ID.

### Example 9-61. Delete NAT rule

---

Request:

DELETE https://<nsxmgr-ip>/api/4.0/edges/<edgeId>/nat/config/rules/*ruleID*

---

# Firewall Migration Switch

### Example 9-62. Firewall migration

---

Request:

GET https://<nsxmgr-ip>/api/4.0/firewall/globalroot-0/state

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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>
</rule>

```

---

## Configuring Fail-Safe Mode for Distributed Firewall

By default, failure or unavailability of the Firewall results in traffic being blocked (fail close). You can change this to allow traffic (fail open).

### Configure Fail-Safe Mode for Firewall

**Example 9-63.** Configure fail-safe mode

---

Example:

PUT https://<nsxmgr-ip>/api/2.1/failsafemode

Request Body

```

<VshieldAppConfiguration>
  <failsafeConfiguration>
    <failsafemode>FAIL_OPEN</failsafemode>
  </failsafeConfiguration>
</VshieldAppConfiguration>

```

---

### Query Fail-Safe Mode Configuration for Firewall

**Example 9-64.** Get fail-safe mode configuration

---

Example:

GET https://<nsxmgr-ip>/api/2.1/failsafemode

---

## 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 9-65. Create SpoofGuard policy

---

Request:

POST https://<nsxmgr-ip>/api/4.0/services/spoofguard/policies/

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 9-66. Create SpoofGuard policy

---

Request:

PUT https://<nsxmgr-ip>/api/4.0/services/spoofguard/policies/<policy-id>

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 9-67. Query SpoofGuard policy

---

Request:

GET https://<nsxmgr-ip>/api/4.0/services/spoofguard/policies/<policy-id>

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>
</spoofguardPolicy>
```

---

## Query all SpoofGuard Policies

Retrieves all SpoofGuard policies.

### Example 9-68. Query SpoofGuard policies

---

Request:

GET https://<nsxmgr-ip>/api/4.0/services/spoofguard/policies/

Request Body:

```
<spoofguardPolicies>
  <spoofguardPolicy>
    <policyId>spoofguardpolicy-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>
  </spoofguardPolicy>
  <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>
</spoofguardPolicy>
</spoofguardPolicies>

```

---

## Delete SpoofGuard Policy

Deletes a SpoofGuard policy.

### Example 9-69. Delete SpoofGuard policy

---

Request:

```
DELETE https://<nsxmgr-ip>/api/4.0/services/spoofguard/policies/<policy-id>
```

---

## 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 9-70. Retrieve flow statistics

---

Request:

```
GET https://<nsxmgr-ip>/api/2.1/flow/flowstats?contextId=datacenter-21&flowType=TCP_UDP
&startTime=0&endTime=1320917094000&startIndex=0&pageSize=2
```

Request 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>
  </flowStatsTcpUdp>
</FlowStatsPage>

```

```

    <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 9-2.** 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 9-3.** Response values for retrieving flow statistics call

Value	Description
startTime	Start time for current flow.
endTime	End time for current flow.

**Table 9-3.** Response values for retrieving flow statistics call

Value	Description
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 9-71. Get flow meta-data for flow type

Request:

```
GET https://<nsxmgr-ip>/api/2.1/flow/flowstats?contextId=datacenter-2538\&flowType=TCP_UDP\
&startTime=1327405883000\&endTime=1327482600000\&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 Flow Summary

Retrieves flow summary for given context.

### Example 9-72. Get flow summary

---

Request:

GET

```
https://<nsxmgr-ip>/api/2.1/internal/flow/flowssummary?contextId=datacenter-2538&&startTime=1327405883000&endTime=1327482600000
```

---

where

- contextId: vc-moref-id of the datacenter, port group, virtual machine, or uuid in case vNIC for which the traffic flow is to be retrieved.
- startTime: Flows with start time greater than this will be fetched.
- endTime: Flows with end time lesser than this will be fetched.

## Query Flow Table

Retrieves top rows for given context and table type.

### Example 9-73. Get flow table

Request:

GET

```
https://<nsxmgr-ip>/api/2.1/internal/flow/flowtable?contextId=datacenter-2538&&startTime=1327405883000&endTime=1327482600000&tableType=source
```

where

- contextId: vc-moref-id of the datacenter, port group, virtual machine, or uuid in case vNIC for which the traffic flow is to be retrieved.
- startTime: Flows with start time greater than this will be fetched.
- endTime: Flows with end time lesser than this will be fetched.
- tableType: This parameter indicates the type of the flow to be fetched. Possible values are: Source, Application, Destination.
- maxRows: (optional) Maximum number of rows to be returned (default value : 5).

## Query Flow Details

Retrieves flow details for given context.

### Example 9-74. Get flow details

Request:

GET

```
https://<nsxmgr-ip>/api/2.1/internal/flow/flowdetail?contextId=datacenter-2538&&&flowType=Allowed&startTime=0&endTime=1327482600000
```

where

- contextId: vc-moref-id of the datacenter, port group, virtual machine, or uuid in case vNIC for which the traffic flow is to be retrieved.
- flowType: This parameter indicates the type of the flow to be fetched. Possible values for flowType parameter are: Allowed or Blocked.
- startTime: Flows with start time greater than this will be fetched.
- endTime: Flows with end time lesser than this will be fetched.

## Query Paged Flow Details

Retrieves a page of flow details for given context.

### Example 9-75. Get flow details

Request:

```
GET https://<nsxmgr-ip>/api/2.1/internal/flow/pagedflowdetails?contextId=datacenter-2538&&&flowType=Allowed&startTime=0&endTime=1327482600000
```

where

- **contextId:** vc-moref-id of the datacenter, port group, virtual machine, or uuid in case vNIC for which the traffic flow is to be retrieved.
- **flowType:** This parameter indicates the type of the flow to be fetched. Possible values for flowType parameter are: Allowed or Blocked.
- **startTime:** Flows with start time greater than this will be fetched.
- **endTime:** Flows with end time lesser than this will be fetched.
- **startIndex:** (optional) This is the start index of the flows to be returned (default value : 0).
- **pageSize:** (optional) This is the maximum number of flows to be returned in a single get call (default value is 256).

## Query Flow Details Application

Retrieves flow details for given context by application. If available, the source and destination names are returned.

### Example 9-76. Get flow details by application

Request:

GET

```
https://<nsxmgr-ip>/api/2.1/internal/flow/flowdetail/application?contextId=datacenter-2538&&&flowType=Allowed&startTime=0&endTime=1327482600000&serviceId=application-211
```

where

- **contextId:** vc-moref-id of the datacenter, port group, virtual machine, or uuid in case vNIC for which the traffic flow is to be retrieved.
- **flowType:** This parameter indicates the type of the flow to be fetched. Possible values for flowType parameter are: Allowed or Blocked.
- **startTime:** Flows with start time greater than this will be fetched.
- **endTime:** Flows with end time lesser than this will be fetched.
- **serviceId:** The service identifier of the application to be queried.

## Query Paged Flow Details Application

Retrieves a page of flow details for given context by application. If available, the source and destination names are returned.

### Example 9-77. Get paged flow details by application

Request:

GET

```
https://<nsxmgr-ip>/api/2.1/internal/flow/pagedflowdetails/application?contextId=datacenter-2538&&&flowType=Allowed&startTime=0&endTime=1327482600000&serviceId=application-211
```

where

- **contextId:** vc-moref-id of the datacenter, port group, virtual machine, or uuid in case vNIC for which the traffic flow is to be retrieved.
- **flowType:** This parameter indicates the type of the flow to be fetched. Possible values for flowType parameter are: Allowed or Blocked.
- **startTime:** Flows with start time greater than this will be fetched.

- `endTime`: Flows with end time lesser than this will be fetched.
- `serviceId`: The service identifier of the application to be queried.
- `startIndex`: (optional) This is the start index of the flows to be returned (default value : 0).
- `pageSize`: (optional) This is the maximum number of flows to be returned in a single get call (default value is 256).

## 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 9-78. Exclude flows

---

Request:

POST https://<nsxmgr-ip>/api/2.1/flow/config

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 9-79. Get excluded flows

---

Request:

GET https://<nsxmgr-ip>/api/2.1/flow/config

Response Body:

```

<?xml version="1.0" encoding="UTF-8"?>
<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 9-80. Configure IPFix

---

Request:



PUT https://<nsxmgr-ip>/api/4.0/firewall/{contextId}/config/ipfix

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 9-81. Query IPFix Configuration

---

Request:

GET https://<nsxmgr-ip>/api/4.0/firewall/{contextId}/config/ipfix

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 9-82. Delete IPFix Configuration

---

Request:

DELETE https://<nsxmgr-ip>/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 9-83. Add a virtual machine to exclusion list

---

Example:

```
PUT https://<nsxmgr-ip>/api/2.1/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 9-84. Get exclusion list

---

Example:

```
GET https://<nsxmgr-ip>/api/2.1/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 9-85.** Delete virtual machine from exclusion list

---

Example:

```
DELETE https://<nsxmgr-ip>/api/2.1/excludelist/<memberID>
```

---

Where memberId is the vc-moref-id of a virtual machine.



# 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 10-1.** 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

## Mapping Security Policy to Security Group

You map 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.

authenticationService 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](#)
- [Working with Security Actions](#)
- [Query Security Policies Mapped to a Security Group](#)
- [Query Service Provider Data](#)
- [Query Security Group Effective Membership](#)
- [Query Security Groups to which a VM Belongs](#)

---

**IMPORTANT** All NSX vSphere REST requests require authentication. See [“Using the NSX REST API”](#) on page 25 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 53.

### 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 10-1. Create security policy

---

Request:

POST https://<nsxmgr-ip>/api/2.0/services/policy/securitypolicy

Request Body:

```
<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>
<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>
<description></description>
<category></category>
<actionType></actionType>
<isActionEnforced></isActionEnforced>
<isActive></isActive>
<isEnabled></isEnabled>
<serviceId></serviceId>
<vendorTemplateId></vendorTemplateId>
<serviceProfile>
<objectId>serviceprofile-1</objectId>
...
</serviceProfile>

```

```

<invalidServiceProfile>false</invalidServiceProfile>
    </action>
  </actionsByCategory>
</actionsByCategory>
<category>traffic_steering</category>
<action class="trafficSteeringSecurityAction">
  <name></name>
  <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`

### TrafficSteering/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 10-2. Query security policies

Request:

GET `https://<nsxmgr-ip>/api/2.0/services/policy/securitypolicy/securitypolicyID | all`

Response Body:

```
<securityPolicy><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>
```

```

</description>
</category>
</actionType>
</isActionEnforced>
</isActive>
</isEnabled>
<secondarySecurityGroup>
  <objectId>
</objectId>
</secondarySecurityGroup>
<secondarySecurityGroup>
  ...
</secondarySecurityGroup>
...
...
<secondarySecurityGroup>
  ...
</secondarySecurityGroup>
<applications>
  <application>
    <objectId>
</objectId>
</application>
  <applicationGroup>
    <objectId>
</objectId>
</applicationGroup>
  ...
  ...
</applications>
</logged>
</action>
</direction>
</outsideSecondaryContainer>
</action>
<action>
  ...
</action>
  ...
  ...
  ...
<action>
  ...
</action>
</actionsByCategory>
<actionsByCategory>
  <category>endpoint</category>
  <action class="endpointSecurityAction">
    <name>
</name>
    <description>
</description>

  <category>
</category>
  <actionType>
</actionType>
  </isActionEnforced>
  </isActive>
  </isEnabled>

  <serviceId>
</serviceId>
  <vendorTemplateId>
</vendorTemplateId>
</action>
</actionsByCategory>
<actionsByCategory>
  <category>traffic_steering</category>
  <action class="trafficSteeringSecurityAction">
    <name>
</name>
    <description>
</description>

  <category>
</category>
  <actionType>
</actionType>
  </isActionEnforced>
  </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>
<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>
<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>
<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 10-3. Edit a security policy

---

Request:

PUT https://<nsxmgr-ip>/api/2.0/services/policy/securitypolicy/securitypolicyID

Response Body:

See [Example 10-2](#).

---

## 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 10-4. Delete a security policy

---

Request:

DELETE <https://<nsxmgr-ip>/api/2.0/services/policy/securitypolicy/securitypolicyID?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 10-5. Export a security policy

---

Request for selective export:

GET

[https://<nsxmgr-ip>/api/2.0/services/policy/securitypolicy/hierarchy?policyIds=comma\\_separated\\_securitypolicy\\_ids&prefix=optional\\_some\\_prefix\\_before\\_names](https://<nsxmgr-ip>/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://<nsxmgr-ip>/api/2.0/services/policy/hierarchy?prefix=optional\\_some\\_prefix\\_before\\_names](https://<nsxmgr-ip>/api/2.0/services/policy/hierarchy?prefix=optional_some_prefix_before_names)

Response Body:

```
<securityPolicyHierarchy>
  <name></name>
  <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 10-6. Import a security policy

---

Request for selective export:

POST [https://<nsxmgr-ip>/api/2.0/services/policy/securitypolicy/hierarchy?suffix=optional\\_suffix\\_to\\_be\\_added\\_after\\_names](https://<nsxmgr-ip>/api/2.0/services/policy/securitypolicy/hierarchy?suffix=optional_suffix_to_be_added_after_names)

Request Body:

See [Example 10-5](#).

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 10-7. Query security actions for a security policy

Request:

GET <https://<nsxmgr-ip>/api/2.0/services/policy/securitypolicy/securitypolicyId/securityactions>

Response Body:

```
<securityPolicies>
  <securityPolicy></securityPolicy>
  <securityPolicy></securityPolicy>
  ...
  ...
  <securityPolicy></securityPolicy>
</securityPolicies>
```

## 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 10-8. Query virtual machines for security action

Request:

GET [https://<nsxmgr-ip>/api/2.0/services/policy/securityaction/category/virtualmachines?attributeKey=attribute\\_name&attributeValue=attribute\\_value](https://<nsxmgr-ip>/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 security action will be applied (by the enforcement engine) on the security group.

### Example 10-9. Query security actions for security group

---

Request:

GET https://<nsxmgr-ip>/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>
      <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>
        <description></description>
        <scope>
          <id></id>
          <objectTypeName></objectTypeName>
          <name></name>
          <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>
<description></description>
<scope>
<id></id>
<objectTypeName></objectTypeName>
<name></name>
<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/>
<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>
<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>
    <description></description>
  </scope>
</securityPolicy>
  <serviceName></serviceName>
  <serviceId></serviceId>
  <vendorTemplateId></vendorTemplateId>
  <invalidServiceId></invalidServiceId>
  <vendorTemplateName></vendorTemplateName>
  <invalidVendorTemplateId></invalidVendorTemplateId>
</ServiceProfile>
<objectId>serviceprofile-1</objectId>
...
</ServiceProfile>
<invalidServiceProfile>>false</invalidServiceProfile>
</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>
  <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>
<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/>
<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>
<clientHandle></clientHandle>
<extendedAttributes/>
</service>
<category></category>
<vendorTemplate>
<id></id>
<revision></revision>
<name></name>
<idFromVendor></idFromVendor>
<vendorAttributes>
<id></id>
<revision></revision>
</vendorAttributes>
</vendorTemplate>
<status></status>
<vendorAttributes>
<id></id>
<revision></revision>
</vendorAttributes>
<runtime>

```

```

<nonCompliantDvpg/>
<nonCompliantVwire/>
</runtime>
<serviceProfileBinding>
<distributedVirtualPortGroups/>
<virtualWires/>
<excludedVnics/>
<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 10-10. Query security actions on a virtual machine

---

Request:

GET https://<nsxmgr-ip>/api/2.0/services/policy/virtualmachine/VM\_ID//securityactions

Response Body:

```

<securityPolicies>
  <securityPolicy></securityPolicy>
  <securityPolicy></securityPolicy>
  ...
  ...
  <securityPolicy></securityPolicy>
</securityPolicies>

```

---

## 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 10-11. Query security policies mapped to a security group

---

Request:

GET https://<nsxmgr-ip>/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 10-12. Query service provider data

---

Request:

GET https://<nsxmgr-ip>/api/2.0/services/policy/serviceprovider/category

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 10-13. Query virtual machines in a security group

---

Request:

GET https://<nsxmgr-ip>/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 10-14. Query security groups to which a virtual machine belongs

---

Request:

GET https://<nsxmgr-ip>/api/2.0/services/securitygroup/lookup/virtualmachine/<virtualMachineId>

---



# 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 351
- [“Defining a Data Security Policy”](#) on page 352
- [“Saving and Publishing Policies”](#) on page 357
- [“Data Security Scanning”](#) on page 358
- [“Querying Scan Results”](#) on page 359
- [“Querying Violation Details”](#) on page 363

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 11-1.** 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 11-1. Get all SDD policy regulations

---

Request:

GET https://<nsxmgr-ip>/api/2.0/dlp/regulation

Response:

```
<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)...
  <classifications>
    <Classification>
      <id>10</id>
      <name>Credit Card Track Data</name>
      <providerName>Credit Card Track Data</providerName>
      <description>Credit Card Track Data</description>
      <customizable>>false</customizable>
    </Classification>
    ...
```

---

## 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 11-1](#)). In the example request body, regulation 66 is California AB-1298, and regulations 67 and 68 originate elsewhere.

### Example 11-2. Enable a regulation

---

Request:



PUT https://<nsxmgr-ip>/api/2.0/dlp/policy/regulations

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 11-3.** Get all classification values associated with customizable classifications

---

Request:

GET https://<nsxmgr-ip>/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 11-4.** Configure a customized regex as a classification value

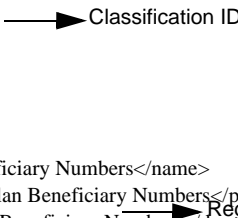
---

Request:

PUT https://<nsxmgr-ip>/api/2.0/dlp/policy/classificationvalues

Authorization: Basic YWRtaW46ZGVmYXVsdA==

```
<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 11-5.** View the list of excludable areas

---

Request:

GET https://<nsxmgr-ip>/api/2.0/dlp/excludableareas

Response:

```
<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 11-8, “Exclude a security group from the scan,”](#) on page 355.

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 11-5](#)).

### Example 11-6. Exclude areas from policy inspection

---

Request:

PUT https://<nsxmgr-ip>/api/2.0/dlp/policy/excludedareas

Authorization: Basic YWRtaW46ZGVmYXVsdA==

```
<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 11-7. Include a security group in the scan

---

Request:

PUT https://<nsxmgr-ip>/api/2.0/dlp/policy/includedsecuritygroups/

Request Body:

```
<set>
  <string>securitygroup-id-1</string>
  <string>securitygroup-id-1</string>
</set>
```

---

### Example 11-8. Exclude a security group from the scan

---

Request:

PUT https://<nsxmgr-ip>/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 11-9. Get included security groups

---

Request:

GET https://<nsxmgr-ip>/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 11-10. Get excluded security groups

---

Request:

GET https://<nsxmgr-ip>/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 11-1](#).

**Table 11-2.** 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 11-11. Scan only PDF and XLSX files modified after 10/19/2011

Request:

```
PUT https://<nsxmgr-ip>/api/2.0/dlp/policy/FileFilters
<FileFilters>
  <scanAllFiles>false</scanAllFiles>
  <lastModifiedAfter>2011-10-19 15:16:04.0 EST</lastModifiedAfter>
  <extensionsIncluded>true</extensionsIncluded>
  <extensions>pdf,xlsx</extensions>
</FileFilters>
```

### Example 11-12. Scan all files except PDF and XLSX files

Request:

```
PUT https://<nsxmgr-ip>/api/2.0/dlp/policy/FileFilters
<FileFilters>
```

```

    <scanAllFiles>false</scanAllFiles>
    <extensionsIncluded>false</extensionsIncluded>
    <extensions>pdf,xlsx</extensions>
  </FileFilters>

```

---

**Example 11-13.** Scan PDF and XLXS files that are less than 100 MB in size
 

---

Request:

```

PUT https://<nsxmgr-ip>/api/2.0/dlp/policy/FileFilters
<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 11-14.** Get saved SDD policy
 

---

Request:

```

GET https://<nsxmgr-ip>/api/2.0/dlp/policy/saved
Authorization: Basic YWRtaW46ZGVmYXVsdA==

```

Response: 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>
      </categories>
    </Regulation>
  </regulations>
</DlpPolicy>

```

```

        <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 11-15. Get published SDD policy

---

Request:

```
GET https://<nsxmgr-ip>/api/2.0/dlp/policy/published
Authorization: Basic YWRtaW46ZGVmYXVsZA==
```

---

## 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 11-16. Publish the updated policy

---

Request:

```
PUT https://<nsxmgr-ip>/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 11-17. Start, pause, resume, or stop a scan operation

---

Request:

```
PUT https://<nsxmgr-ip>/api/2.0/dlp/scanop
```

```
<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 11-18. Get scan status

---

Request:

```
GET https://<nsxmgr-ip>/api/2.0/dlp/scanstatus
```

Response:

```
<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 11-19. Get list of virtual machines being scanned

---

Request:

```
GET https://<nsxmgr-ip>/api/2.0/dlp/scan/current/vms/<id>
?scanstatus=COMPLETED&pagesize=10&startindex=1
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 11-20.** Get number of virtual machines being scanned

---

Request:

```
GET https://<nsxmgr-ip>/api/2.0/dlp/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 11-21.** Get summary information about last five scans

---

Request:

GET https://&lt;nsxmgr-ip&gt;/api/2.0/dlp/completedscansummaries

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<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>
```

---

—————▶ Scan ID

## 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 11-22.** Get Information for virtual machines scanned during last scan

---

Request:

GET https://&lt;nsxmgr-ip&gt;/api/2.0/dlp/scan/&lt;scan\_ID&gt;/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 11-23.** Retrieves Information for virtual machines scanned during last scan

---

Request:

GET https://&lt;nsxmgr-ip&gt;/api/2.0/dlp/scan/&lt;scan\_ID&gt;/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 11-24.** Get XML representation of policy used in previous scan

---

Request:

GET https://&lt;nsxmgr-ip&gt;/api/2.0/dlp/scan/&lt;scan\_ID&gt;/policyasxml

Response:

```
<DlpPolicy>
  <objectId>dlppolicy-2</objectId>
  <type>
    <typeName>DlpPolicy</typeName>
  </type>
</DlpPolicy>
```

```

</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"/>
    <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,xlsx,xltx,xltm,xlsb,xlsb,xlsm,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,C
ATFCT,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,jrnl,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>

```

```
</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 11-25. Get violation count for entire inventory

---

Request:

```
GET https://<nsxmgr-ip>/api/2.0/dlp/violations/
```

---

### Example 11-26. Get violation count for specific resource

---

Request:

```
GET https://<nsxmgr-ip>/api/2.0/dlp/violations/<context_ID>
```

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>
        ...
      </regions>
```

```

        <string>NA</string>
      </regions>
      <categories>
        <string>PHI</string>
        <string>PCI</string>
        <string>PII</string>
      </categories>
    </regulation>
    <violationCount>1</violationCount>
  </Violations>
</Violations>
</list>

```

---

Where context\_ID 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 11-27. Get violating files for entire inventory

---

Request:

GET https://<nsxmgr-ip>/api/2.0/dlp/violatingfiles?pagesize=<i>&startindex=<j>

---

Where:

- pagesize is the number of results to view.
- startindex is the page number from which the results should be displayed.

### Example 11-28. Get violating files for a resource

---

Request:

GET https://<nsxmgr-ip>/api/2.0/dlp/violatingfiles/<context\_ID>?pagesize=<i>&startindex=<j>

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>
  </dataPage>
</ViolatingFiles>

```

```

<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:

- context\_ID 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 11-29.** Get list of violating files in CSV format

---

Request:

GET https://<nsxmgr-ip>/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 11-30.** Get list of violated regulations

---

Request:

GET https://<nsxmgr-ip>/api/2.0/dlp/violatingfilescsv/<context\_ID>

---

Where context\_ID 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 367
- [“Query Resources”](#) on page 370
- [“Query User Details”](#) on page 373
- [“Query Discovered User Details”](#) on page 377
- [“Working with Domains”](#) on page 378
- [“Working with Activity Monitoring Syslog Support”](#) on page 381

---

**IMPORTANT** All NSX REST requests require authentication. See [“Using the NSX REST API”](#) on page 25 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 378.

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 queying the vCenter mob structure (https:<vc-ip>/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 12-1. Enable data collection on a virtual machine

---

Request:

POST https://<nsxmgr\_ip>/api/1.0/eventcontrol/vm/<vmID>/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 12-2. Disable data collection on a virtual machine

---

Request:

POST https://<nsxmgr\_ip>/api/1.0/eventcontrol/vm/<vmID>/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 12-3. Turn on kill switch

---

Request:

POST https://<nsxmgr\_ip>/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 12-4. Turn off kill switch

---

Request:

POST https://<nsxmgr\_ip>/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 12-5.** 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://<nsxmgr\_ip>/api/1.0/eventcontrol/eventcontrol/config/vm/<vm-id>

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<perVmConfig>
  <actions>
    <action>
      <type>global_switch</type>
      <value>disabled</value>
    </action>
    <action>
      <type>per_vm_config</type>
      <value>enabled</value>
    </action>
  </actions>
</perVmConfig>
```

---

**Example 12-6.** 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://<nsxmgr\_ip>/api/1.0/eventcontrol/eventcontrol/config/vm/<vm-id>

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 12-1.** Parameters for GET <https://<nsxmgr-ip>/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 12-7.** View user activities to VM id 1 originating from application id 1

---

Request:

GET

`https://<nsxmgr_ip>/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 12-8.** View user activities to VM id 1 originating from application id 1

---

Request:

```
GET https://<nsxmgr_ip>/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 12-9. View interaction between inventory containers

Request:

```
GET https://<nsxmgr_ip>/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 12-10. View interaction between inventory containers

Request:

```
GET https://<nsxmgr-ip>/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 12-2.** Parameters for GET https://<nsxmgr-ip>/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 12-11.** View user activities to VM id1 originating from application id1

---

Request:

GET

[https://<nsxmgr\\_ip>/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](https://<nsxmgr_ip>/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 12-12.** View user activities to VM id 1 originating from application id 1

---

Request:

GET

[https://<nsxmgr\\_ip>/api/3.0/ai/userdetails?query=sam&interval=60m&param=app:DEST\\_APP:1:EXCLUDE&param=dest:IP:1:EXCLUDE&param=src:SECURITY\\_GROUP:1:EXCLUDE](https://<nsxmgr_ip>/api/3.0/ai/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 12-13. View interaction between inventory containers

---

Request:

GET

[https://<nsxmgr\\_ip>/api/3.0/ai/userdetails?query=containers&interval=60m&param=dest:SECURITY\\_GROUP:1:EXCLUDE&param=src:SECURITY\\_GROUP:1](https://<nsxmgr_ip>/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 12-14. View interaction between inventory containers

---

Request:

GET

[https://<nsxmgr\\_ip>/api/3.0/ai/userdetails?query=adg&interval=24h&param=adg:USER:1:INCLUDE&param=src:SECURITY\\_GROUP:1:EXCLUDE](https://<nsxmgr_ip>/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 12-15.** View inbound vm activities to a VM id1 for a specific service used (app=16)

---

Request:

GET

`https://<nsxmgr_ip>/api/3.0/ai/userdetails?query=vma&interval=60m&param=dest:VIRTUAL_MACHINE:1&param=app:DEST_APP:16`

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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$</destHost>
    <destIP>172.16.4.70</destIP>
    <id>0</id>
    <srcContainer>HOKUIFLVPC</srcContainer>
  </aiActionRecord>
  <aiActionRecord>
    <application>PNA</application>
    <connectionCount>3</connectionCount>
    <destHost>IDC-DEV-1$</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 12-16. Retrieve user details

---

Retrieve user details for a specific user:

```
GET https://<nsxmgr_ip>/api/3.0/ai/user/<userID>
```

Retrieve app details:

```
GET https://<nsxmgr_ip>/api/3.0/ai/app
```

Retrieve application details for a specific application:

```
GET https://<nsxmgr_ip>/api/3.0/ai/app/<appID>
```

Retrieve list of all discovered hosts (both by agent introspection and LDAP Sync) and their detail:

```
GET https://<nsxmgr_ip>/api/3.0/ai/host
```

Retrieve host details:

```
GET https://<nsxmgr_ip>/api/3.0/ai/host/<hostID>
```

Retrieve list of all discovered desktop pools by agent introspection:

```
GET https://<nsxmgr_ip>/api/3.0/ai/desktoppool
```

Retrieve details specific desktop pool:

```
GET https://<nsxmgr_ip>/api/3.0/ai/desktoppool/<desktoppoolID>
```

Retrieve list of all discovered virtual machines:

```
GET https://<nsxmgr_ip>/api/3.0/ai/vm
```

Retrieve details about a specific virtual machine:

```
GET https://<nsxmgr_ip>/api/3.0/ai/vm/<vmID>
```

Retrieve list of all the discovered (and configured) LDAP directory groups:

```
GET https://<nsxmgr_ip>/api/3.0/ai/directorygroup
```

Retrieve details about a specific directory groups:

```
GET https://<nsxmgr_ip>/api/3.0/ai/directorygroup/<directorygroupID>
```

Retrieve list of AD groups a user belongs to:

GET https://<nsxmgr\_ip>/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 be reported as observed in SAM database:

GET https://<nsxmgr\_ip>/api/3.0/ai/securitygroup

Retrieve details about specific security group:

GET https://<nsxmgr\_ip>/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 12-17. Register or update domain

---

Request:

POST https://<nsxmgr\_ip>/api/3.0/directory/updateDomain

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml version="1.0" encoding="UTF-8"?>
<DirectoryDomain>
  <name>vs4.net</name>
  <type>ActiveDirectory</type>
  <netbiosName>VS4</netbiosName>
  <username>Administrator</username>
  <password>xxx</password>
</DirectoryDomain>
```

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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
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 12-18. Query domains

Request:

GET https://<nsxmgr\_ip>/api/1.0/directory/listDomains

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 12-19. Delete domain

---

Request:

```
DELETE https://<nsxmgr_ip>/api/1.0/directory/deleteDomain/<Domain Id>
```

---

## Working with LDAP Servers

### Example 12-20. Create LDAP server

---

Request:

```
POST https://<nsxmgr_ip>/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 12-21. LDAP server calls

---

Query LDAP servers for a domain:

```
GET https://<nsxmgr_ip>/api/1.0/directory/listLdapServersForDomain/<domain id>
```

Start LDAP full sync:

```
PUT https://<nsxmgr_ip>/api/1.0/directory/fullSync/<domain id>
```

Start LDAP delta sync:

```
PUT https://<nsxmgr_ip>/api/1.0/directory/deltaSync/<domain id>
```

Delete LDAP server:

```
DELETE https://<nsxmgr_ip>/api/1.0/directory/deleteLdapServer/<LdapServerID>
```

---

## Working with EventLog Servers

### Example 12-22. Create EventLog server

---

Request:

```
POST https://<nsxmgr_ip>/api/1.0/directory/updateEventLogServer
```

Request Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<EventlogServer>
  <id>1</id>
```

```
<domainId>4</domainId>
<hostName>10.142.72.70</hostName>
<enabled>false</enabled>
</EventlogServer>
```

---

### Example 12-23. EventLog server calls

---

Query EventLog servers for a domain:

```
GET https://<nsxmgr_ip>/api/1.0/directory/listEventLogServersForDomain/<EventLogServer id>
```

Delete EventLog server:

```
DELETE https://<nsxmgr_ip>/api/1.0/directory/deleteEventLogServer/<EventLogServerID>
```

---

## Working with Mapping Lists

### Example 12-24. Query mapping lists

---

Query user-to-ip mapping list from database:

```
GET https://<nsxmgr_ip>/api/1.0/identity/userIpMapping
```

Query host-to-ip mapping list from database:

```
GET https://<nsxmgr_ip>/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://<nsxmgr_ip>/api/1.0/identity/ipToUserMapping
```

Query set of Windows Domain Groups (AD Groups) to which the specified user belongs:

```
GET https://<nsxmgr_ip>/api/1.0/identity/directoryGroupsForUser
```

Create static user IP mapping:

```
POST https://<nsxmgr_ip>/api/1.0/identity/staticUserMapping/<userID>/<IP>
```

Query static user IP mapping list:

```
GET https://<nsxmgr_ip>/api/1.0/identity/staticUserMappings
```

Query static user IP mapping for specified user:

```
GET https://<nsxmgr_ip>/api/1.0/identity/staticUserMappingsbyUser/<userID>
```

Query static user IP mapping for specified IP:

```
GET https://<nsxmgr_ip>/api/1.0/identity/staticUserMappingsbyIP/<userID>
```

Delete static user IP mapping for specified user:

```
DELETE https://<nsxmgr_ip>/api/1.0/identity/staticUserMappingsbyUser/<userID>
```

Delete static user IP mapping for specified IP:

```
DELETE https://<nsxmgr_ip>/api/1.0/identity/staticUserMappingsbyIP/<userID>
```

---

## Working with Activity Monitoring Syslog Support

### Example 12-25. Enable Activity Monitoring syslog support

---

Request:

POST https://<vsm\_ip>/api/1.0/sam/syslog/enable

---

**Example 12-26.** Disable Activity Monitoring syslog support

---

Request:

POST https://<vsm\_ip>/api/1.0/sam/syslog/disable

---

# 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 383
- [“Query Job Instances for Job ID”](#) on page 384
- [“Query Latest Job Instances for Job ID”](#) on page 385
- [“Block REST Thread”](#) on page 385
- [“Query Job Instances by Criterion”](#) on page 385

---

**IMPORTANT** All REST requests require authentication. See [“Using the NSX REST API”](#) on page 25 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 13-1. Query job instances

---

Request Body:

GET https://<nsxmgr-ip>/api/2.0/services/taskservice/job/<jobID>

Response Body:

```
<?xml version="1.0" encoding="UTF-8"?>
<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>
      <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>
    </task>
  </taskList>
</jobInstances>
```



```

<systemTask>true</systemTask>
<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 13-2. Query job instances

---

Request Body:

GET https://<nsxmgr-ip>/api/2.0/services/taskservice/job/<jobID>

Response Body:

See [Example 13-1](#)

---

## 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 13-3. Query job instances

---

Request Body:

GET https://<nsxmgr-ip>/api/2.0/services/taskservice/job/<jobID>

Response Body:

See [Example 13-1](#).

---

## Query Job Instances by Criterion

You can specify filtering criteria and paging information and query the task framework.

### Example 13-4. Query job instances by criterion

---

Request Body:

GET

https://<nsxmgr-ip>/api/2.0/services/taskservice/job/startIndex=<0>&pageSize=<10>&sortBy=startTime&sortOrderAscending=false|true

Response Body:

See [Example 13-1](#).

---

## 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 387
- [“Query Datacenter ID”](#) on page 387
- [“Query Host ID”](#) on page 387 authentication
- [“Query Portgroup ID”](#) on page 388
- [“Query VMID”](#) on page 388

---

**IMPORTANT** All NSX REST requests require authentication. See [“Using the NSX REST API”](#) on page 25 for details about basic authorization.

---

### Query Datacenter MOID

- 1 In a web browser, type the following:  
`http://<vCenter-IP>/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>/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>/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>/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>/mob`

The VMID is listed under host structure.

# vShield Endpoint Management

---



Always check with the NSX administrator to ensure that the vShield Endpoint solution that was developed with ESXi Partner Program 3.0 or earlier – (or vShield 5.5 or earlier). These partner solutions are also supported on NSX 6.0 and need the API listed below. These API 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 389
- [“Registering a Solution with vShield Endpoint Service”](#) on page 390
- [“Querying Registration Status of vShield Endpoint”](#) on page 391
- [“Querying Activated Security Virtual Machines for a Solution”](#) on page 392
- [“Unregistering a Solution with vShield Endpoint”](#) on page 393
- [“Status Codes and Error Schema”](#) on page 394

---

**IMPORTANT** All vShield REST requests require authentication. See [“Using the NSX REST API”](#) on page 25 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:

- 5 Deactivate solutions per host.
- 6 Unset a solution’s IP address and port.
- 7 Unregister solutions.
- 8 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 394.

### Register a Vendor

You can register the vendor of an antivirus solution.

#### Example 15-1. Register a vendor

---

Request:

POST https://<nsxmgr-ip>/api/2.0/endpointsecurity/registration

Request Body:

```
<VendorInfo>
  <id>vendor_id</id>
  <title>vendor_title</title>
  <description>vendor_description</description>
</VendorInfo>
```

---

In the request body, vendor\_id 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 15-2. Register a solution

---

Request:

POST https://<nsxmgr-ip>/api/2.0/endpointsecurity/registration/<vendor\_id>

Request Body:

```
<SolutionInfo>
  <altitude>solution_altitude</altitude>
  <title>solution_title</title>
  <description>solution_description</description>
</SolutionInfo>
```

---

In the request, <vendor\_id> 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 390.

#### 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 15-3. Set IP address and port

---

Request:

POST https://<nsxmgr-ip>/api/2.0/endpointsecurity/registration/<vendor\_id>/<altitude>/location

Request Body:

```
<LocationInfo>
  <ip>solution_ip_address</ip>
  <port>solution_port</port>
</LocationInfo>
```

---

In the request, <vendor\_id> 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 15-4. Activate solution

---

Request:

```
POST https://<nsxmgr-ip>/api/2.0/endpointsecurity/activation/<vendor_id>/<altitude>
```

Request Body:

```
<ActivationInfo>
  <moid>svm_moid</moid>
</ActivationInfo>
```

---

In the request, <vendor\_id> 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 15-5. Get list of all registered vendors

---

Request:

```
GET https://<nsxmgr-ip>/api/2.0/endpointsecurity/registration/vendors
```

---

#### Example 15-6. Get vendor registration information

---

Request:

```
GET https://<nsxmgr-ip>/api/2.0/endpointsecurity/registration/<vendor_id>
```

---

### Get Solution Registration

You can retrieve solution registration information.

**Example 15-7. Get all registered solutions for a vendor**

---

Request:

GET https://<nsxmgr-ip>/api/2.0/endpointsecurity/registration/<vendor\_id>/solutions

---

**Example 15-8. Get solution registration information**

---

Request:

GET https://<nsxmgr-ip>/api/2.0/endpointsecurity/registration/<vendor\_id>/<altitude>

---

**Get IP Address of a Solution**

This call retrieves the IP address and port associated with a solution.

**Example 15-9. Get IP address and port of a solution**

---

Request:

GET https://<nsxmgr-ip>/api/2.0/endpointsecurity/registration/<vendor\_id>/<altitude>/location

---

**Get Activation Status of a Solution**

This call retrieves solution activation status, given the managed object reference &lt;moid&gt; of its virtual machine.

**Example 15-10. Get activation status of a solution**

---

Request:

GET https://<nsxmgr-ip>/api/2.0/endpointsecurity/activation/<vendor\_id>/<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 15-11. Get activated security virtual machines**

---

Request:

GET https://&lt;nsxmgr-ip&gt;/api/2.0/endpointsecurity/activation/&lt;vendor\_id&gt;/&lt;solution\_id&gt;

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, `vendor_id` is the VMware-assigned ID for the vendor, while `solution_id` is the solution ID.

## Query Activation Information

You can retrieve activation information for all activated security virtual machines on the specified host.

### Example 15-12. Get activation information

---

Request:

```
GET https://<nsxmgr-ip>/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 15-13. Unregister a vendor

---

Request:

```
DELETE https://<nsxmgr-ip>/api/2.0/endpointsecurity/registration/<vendor_id>
```

---

### Unregister a Solution

This call unregisters a solution.

#### Example 15-14. Unregister a vendor

---

Request:

```
DELETE https://<nsxmgr-ip>/api/2.0/endpointsecurity/registration/<vendor_id>/<altitude>
```

---

## Unset IP Address

This call unsets a solution's IP address and port.

### Example 15-15. Unset IP address and port

---

Request:

```
DELETE https://<nsxmgr-ip>/api/2.0/endpointsecurity/registration/<vendor_id>/<altitude>/location
```

---

## Deactivate a Solution

This call deactivates a solution on a host.

### Example 15-16. Deactivate a solution

---

Request:

```
DELETE https://<nsxmgr-ip>/api/2.0/endpointsecurity/activation/<vendor_id>/<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.

```
<?xml version="1.0" encoding="UTF-8"?><xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified">
  <xs:element name="Error">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="code" type="xs:unsignedInt"/>
        <xs:element name="description" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
```



## Deprecated APIs

The following APIs have been deprecated in the NSX 6.0 release.

**Table 16-1.** 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/ <i>clusterID</i>	
/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 399
- [“Deprecated: vShield Manager Global Configuration Schema”](#) on page 401
- [“Deprecated: ESX Host Preparation and Uninstallation Schema”](#) on page 406
- [“Deprecated: vShield App Schemas”](#) on page 407
- [“Error Message Schema”](#) on page 413

## 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:IpssecVpnTunnels" /> <!-- 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:element>

  <xs:complexType name="ReleaseInfoType">
    <!-- can be re-used for release information of vshield, vShield
      Manager, or vShield Edge-->
    <xs:sequence>
      <xs:element name="buildNumber" type="xs:NMTOKEN" /> <!-- add fields as required -->
```

```
<xs:element minOccurs="0" name="vseLocationOnnsmgr" type="xs:string" />
</xs:sequence>
</xs:complexType>

<xs:complexType name="SSOInfoType">
<xs:sequence>
<xs:element minOccurs="0" name="nsxmgrSolutionName">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:minLength value="1"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="lookupServiceUrl">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:minLength value="1"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="ssoAdminUserName">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:minLength value="1"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="ssoAdminPassword">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:minLength value="1"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element minOccurs="0" name="certificateThumbprint">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:pattern
value="[a-fA-F0-9]{2}:[a-fA-F0-9]{2}:[a-fA-F0-9]{2}:[a-fA-F0-9]{2}:[a-fA-F0-9]{2}:[a-fA-F0-9]{2}:[a-fA-F0-9]{2}:[a-fA-F0-9]{2}:[a-fA-F0-9]{2}:[a-fA-F0-9]{2}:[a-fA-F0-9]{2}:[a-fA-F0-9]{2}:[a-fA-F0-9]{2}:[a-fA-F0-9]{2}:[a-fA-F0-9]{2}:[a-fA-F0-9]{2}:[a-fA-F0-9]{2}:[a-fA-F0-9]{2}:[a-fA-F0-9]{2}"></xs:pattern>
</xs:restriction>
</xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>

<xs:complexType name="VcInfoType">
<xs:sequence>
<xs:element name="ipAddress">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:minLength value="1"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="userName">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:minLength value="1"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="password">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:minLength value="1"/>
```

```
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element minOccurs="0" name="token">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:minLength value="1"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element minOccurs="0" name="certificateThumbprint">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:pattern value="[a-fA-F0-9]{2}:[a-fA-F0-9]{2}:[a-fA-F0-9]{2}:[a-fA-F0-9]{2}:[a-fA-F0-9]{2}:[a-fA-F0-9]{2}:[a-fA-F0-9]{2}:[a-fA-F0-9]{2}:[a-fA-F0-9]{2}:[a-fA-F0-9]{2}:[a-fA-F0-9]{2}:[a-fA-F0-9]{2}:[a-fA-F0-9]{2}:[a-fA-F0-9]{2}:[a-fA-F0-9]{2}:[a-fA-F0-9]{2}:[a-fA-F0-9]{2}:[a-fA-F0-9]{2}:[a-fA-F0-9]{2}"></xs:pattern>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element minOccurs="0" name="pluginDownloadServer">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:minLength value="1"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element minOccurs="0" name="pluginDownloadPort">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:minLength value="1"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>

<xs:complexType name="HostInfoType">
<xs:sequence>
<xs:element name="hostId" type="xs:string" />
<xs:element name="ipAddress" type="xs:string" />
<xs:element name="userName" type="xs:string" />
<xs:element name="password" type="xs:string" />
</xs:sequence>
</xs:complexType>

<xs:complexType name="SecurityGroups">
<xs:choice>
<xs:element name="securityGroup" type="vse:SecurityGroup" maxOccurs="unbounded" />
<xs:element name="securityGroupIdList" type="vse:SecurityGroupIdList" />
</xs:choice>
</xs:complexType>

<xs:complexType name="SecurityGroup">
<xs:sequence>
<xs:element name="securityGroupBaseNode" type="xs:string"/>
<xs:element name="securityGroupName" type="xs:string"/>
<xs:element name="securityGroupId" type="xs:string" minOccurs="0" />
<xs:element name="securityGroupNodeList" type="vse:NodeList" minOccurs="0"/>
<xs:element name="securityGroupIpList" type="vse:IpList" minOccurs="0" />
</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: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:IpssecVpnTunnelStatus" />
    </xs:sequence>
</xs:complexType>

<xs:complexType name="IpssecVpnTunnelStatus">
    <xs:sequence>
        <xs:element name="networkId" type="xs:string" />
        <xs:element name="ipsecVpnTunnelConfig" type="vse:IpssecVpnTunnelConfigType" />
    </xs:sequence>
</xs:complexType>

<xs:complexType name="IpssecVpnTunnelConfigType"> <!--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="EpsecInstallParams" 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="EpsecInstalled" 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 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:schema>
```

```

<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="ProtocolsTypeEnum" maxOccurs="4" />
      </xs:choice>
    </xs:complexType>
  </xs:element>

</xs:schema>

```

```

        </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: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" />
    </xs:sequence>

```



```

        <!-- Only in response, not considered in request -->
    </xs:sequence>
</xs:complexType>

<xs:complexType name="Layer2DestinationDto">
    <xs:complexContent>
        <xs:extension base="DestinationDto">
            </xs:extension>
            <xs:element name="application" type="ApplicationDto" minOccurs="0" />
        </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: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: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="namespacesType" type="vsns:NamespacesTypeEnum" />
      </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:NamespacesTypeEnum" />
    <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="NamespacesTypeEnum">
    <xs:restriction base="xs:NCName">
      <xs:enumeration value="DEFAULT" />
    </xs:restriction>
  </xs:simpleType>

```

```

<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 17-1.** Error Message Status Codes

Code	Description
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).

