

---

# **VSPK Documentation**

***Release 1***

**Nuage Networks**

**Dec 23, 2016**



## QUICKSTART



## QUICKSTART

### 1.1 Installation

The easiest way to install `vspk` is to use `pip`:

```
pip install vspk
```

### 1.2 Creating a session

```
# we'll use the 4.0 API, but to use the 3.2 API, replace v4_0 by v3_2
from vspk import v4_0 as vspk

# enable logging. it makes debugging much easier
import logging
from vspk.utils import set_log_level
set_log_level(logging.DEBUG, logging.Streamhandler())

# create a new session for csproot
session = vspk.NUVSDSession(
    username='csproot',
    password='csproot',
    enterprise='csp',
    api_url="https://my-vsd:8443")

# start the session.
try:
    session.start()
# we'll see later how to properly handle vspk exceptions.
# for now, just catch everything.
except:
    logging.error('Failed to start the session')
```

If the session started successfully, `session.user` should be an instance of `vspk.v4_0.num.NUme` corresponding to `csproot`. Let's wrap this in a function that we will reuse through this document:

```
from vspk import v4_0 as vspk

def setup_logging():
    import logging
    from vspk.utils import set_log_level
    set_log_level(logging.DEBUG, logging.Streamhandler())
```

```
def start_csproot_session():
    session = vspk.NUVSDSession(
        username='csproot',
        password='csproot',
        enterprise='csp',
        api_url="https://my-vsd:8443")
    try:
        session.start()
    except:
        logging.error('Failed to start the session')
    return session.user
```

## 1.3 CRUD operations

Each vspk class correspond to a VSD entity. All the classes provide the same API to perform CRUD operations:

- a `create_child()` method for creating children
- a `save()` method to update the current entity
- a `delete()` method to delete the current entity
- a `get()` and a `fetch()` method to retrieve the current entity
- multiple *fetchers* to retrieve children entities.

## 1.4 Creating objects

Here is an example of where we create an enterprise and a domain under csproot:

```
# we assume we have the setup_logging() and start_csproot_session() methods
# showed in the previous example

from vspk import v4_0 as vspk
setup_logging()
csproot = start_csproot_session()

# Create a new enterprise object. The only mandatory parameter is the name,
# so we give it directly to the constructor
new_enterprise = vspk.NUEnterprise(name="new-corp")

# Create the enterprise on VSD.
csproot.create_child(new_enterprise)

# Create a new domain object.
new_domain = vspk.NUDomain()
# The attributes can also be set on the object directly
new_domain.name = "new-dom"

# Create the domain on VSD.
new_enterprise.create_child(new_domain)
```

## 1.5 Updating objects

Let's change the name of the domain we just created. All we need to do to update an entity is change its attributes, and call `save()`:

```
new_domain.name = "better-named-domain"
new_domain.save()
```

That's it!

## 1.6 Deleting objects

Deleting objects is dead simple: just call `delete()`:

```
new_domain.delete()
```

## 1.7 Fetching objects

### 1.7.1 Fetching the current entity

Fetching the current entity is pretty simple:

```
new_enterprise.get()
```

There are two reasons why we would need to fetch the current entity:

- to get an up-to-date representation of the entity, in case it has been updated on the VSD by someone else
- to retrieve an entity from its UUID. For example, if we know the UUID of a subnet on VSD, we could do:

```
my_subnet = NUSubnet(id="123e4567-e89b-12d3-a456-426655440000")
my_subnet.get()

# Now, the attributes of the object are populated with data from VSD. We
# can for instance print the subnet's name:
logging.info("Fetched subnet %s!" % my_subnet.name)
```

### 1.7.2 Fetching child entities

Each child entity has a corresponding *fetcher*. Calling the fetcher will populate it. For instance, assuming we have vports under the subnet we just fetched, we could retrieve them like this:

```
# fetch the vports
my_subnet.vports.get()
for vport in my_subnet.vports:
    logging.info("vport: %s" % vport.name)

# fetch the host interfaces under the current vport
vport.host_interfaces.get()
for interface in vport.host_interfaces:
    logging.info("host ip: %s" % interface.ip_address)
```

Since `get` returns itself, we can make this shorter:

```
for vport in my_subnet.vports.get():
    logging.info("vport: %s" % vport.name)

    for interface in vport.host_interfaces.get():
        logging.info("host ip: %s" % interface.ip_address)
```

### 1.7.3 Filtering

By default, fetchers fetch all the child entities, which can lead to huge responses. Fortunately, the API offers filters on some attributes, and `vspk` provides a way to use them:

```
# get all the bridge vports in the current domain:
for vport in domain.vports.get(filter='type is "BRIDGE")':
    # do something
```

## 1.8 Assigning entities

Some entities do not follow the parent/children relationship. For example, users are not children of groups, they *belong* to one or multiple groups. Similarly, policy groups are *assigned* to vports. To assign entities to another entity, we use the `assign()` method:

```
entity.assign(assigned_entities_list, assigned_entities_class)
```

This method takes two arguments:

- the list of entities to be assigned
- the class of the assigned entities

For example, to add a user “bob” to a group “engineers”:

```
# Get the "engineers" group.
#
# get_first() is a convenient shortcut for get()[0], that returns None if
# no entity was fetched.
engineers = enterprise.groups.get_first(filter="name is 'engineers'")

# Get the user we want to add to the group
bob = enterprise.users.get_first(filter="userName is 'bob'")

# Fetch the users already assigned to this group
engineers.users.get()

engineers.assign(
    # We assign the list of *all* the users, not only "bob"
    [bob] + engineers.users,
    # We need to specify the class of the entities we are assigning
    vspk.NUUser
)
```

To un-assign resources, we just re-assign a list without these resources. To remove the user “bob” we just added, we could to this:



```
# Fetch the assigned users
assigned_users = engineers.users.get()

# Make new list of users without "bob"
new_assigned_users = [user if user.user_name != "bob" for user in assigned_users]

# Assign this new list
engineers.assign(new_assigned_users, vspk.NUUser)
```

To un-assign all the entities, assign an empty list:

```
engineers.assign([], vspk.NUUser)
```

## 1.9 Error handling

All of the previous methods raise a `bambou.exception.BambouHTTPError` when they fail, which contains some interesting information, like the HTTP status code. It can be useful to catch these exceptions:

```
from bambou.exceptions import BambouHTTPError

# We assume we have a parent trying to create a child.

try:
    parent_entity.create_child(child_entity)
except BambouHTTPError as exc:
    response = exc.connection.response
    if response.status_code == 409:
        # the entity probably already exists, so we just ignore this error:
        pass
    else:
        logging.Error("Failed to create entity: %s" % exc.message())
        # re-raise the exception
        raise
```



## NUADDRESSMAP

`nuaddressmap.NUAddressMap(bambou.nurest_object.NUMetaRESTObject, ) :`

Defines an address mapping between a private IP and a port with a public IP address and port.

### 2.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `private_ip` (**Mandatory**): Private IP address of the interface
- `private_port`: None
- `associated_patnat_pool_id` (**Mandatory**): Read Only - Indicates which PATNATPool this entry belongs to
- `public_ip` (**Mandatory**): Public IP address of the interface
- `public_port`: None
- `external_id`: External object ID. Used for integration with third party systems
- `type`: Identifies the type of address mapping

### 2.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nustatistics.NUStatistics</i>	statistics
<i>nustatisticspolicy.NUStatisticsPolicy</i>	statistics_policies

### 2.3 Parents

- *nupatnatpool.NUPATNATPool*



## NUADDRESSRANGE

`nuaddressrange.NUAddressRange(bambou.nurest_object.NUMetaRESTObject, ) :`

This is the definition of a Address Range associated with a Network.

### 3.1 Attributes

- `dhcp_pool_type`: DHCPPoolType is an enum that indicates if the DHCP Pool is for HOST/BRIDGE.
- `ip_type`: IPv4 or IPv6(only IPv4 is supported in R1.0) Possible values are IPV4, IPV6, DUALSTACK.
- `last_updated_by`: ID of the user who last updated the object.
- `max_address` (**Mandatory**): Highest address in the address range
- `min_address` (**Mandatory**): Lowest address in the address range
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 3.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>

### 3.3 Parents

- `nusharednetworkresource.NUSharedNetworkResource`
- `nusubnettemplate.NUSubnetTemplate`
- `nusubnet.NUSubnet`
- `nul2domaintemplate.NUL2DomainTemplate`
- `nul2domain.NUL2Domain`



## NUAGGREGATEMETADATA

`nuaggregatemetadata.NUAggregateMetadata(bambou.nurest_object.NUMetaRESTObject, ) :`

Metadata associated to a entity

### 4.1 Attributes

- `name`: Name of the Metadata.
- `description`: Description of the Metadata.
- `metadata_tag_ids`: Metadata tag IDs associated with this metadata. You can filter metadata based on this attribute for example X-Nuage-Filter: '2d6fb627-603b-421c-b63a-eb0a6d712761' IN metadataTagIDs
- `network_notification_disabled`: Specifies metadata changes need to be notified to controller, by default it is notified
- `blob` (**Mandatory**): Metadata that describes about the entity attached to it.
- `global_metadata`: Specifies whether the metadata is global or local
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 4.2 Parents

- `nuvport.NUVPort`





## NUALARM

`nualarm.NUAlarm(bambou.nurest_object.NUMetaRESTObject, ) :`

The alarm API allows the management of system alarms.

### 5.1 Attributes

- **name (Mandatory):** The alarm name. Each type of alarm will generate its own name
- **target\_object:** Identifies affected Entity. Example: Alarm generated by TCA for Domain domain1(Packets towards a VM, Breach)
- **last\_updated\_by:** ID of the user who last updated the object.
- **acknowledged:** Flag to indicate that alarm is already acknowledged or not
- **reason:** Provides a description of the alarm
- **description:** Description of the alarm
- **severity:** Severity of the alarm.
- **timestamp:** Indicates the time that the alarm was triggered
- **enterprise\_id:** Enterprise that this alarm belongs to
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **error\_condition:** Identifies the error condition
- **number\_of\_occurrences:** Number of times that the alarm was triggered
- **external\_id:** External object ID. Used for integration with third party systems

### 5.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas

### 5.3 Parents

- *nuredundancygroup.NURedundancyGroup*

- *nutca.NUTCA*
- *nuvsgredundantport.NUVsgRedundantPort*
- *nuwanservice.NUWANService*
- *nuvsd.NUVSD*
- *nuvport.NUVPort*
- *nuport.NUPort*
- *nucontainer.NUContainer*
- *nuvrs.NUVRS*
- *nuhsc.NUHSC*
- *nuvlan.NUVLAN*
- *nugateway.NUGateway*
- *nunsgateway.NUNSGateway*
- *nuvsc.NUVSC*
- *nuvm.NUVM*
- *nunsport.NUNSPort*
- *nunsredundantgatewaygroup.NUNSRedundantGatewayGroup*
- *nuenterprise.NUEnterprise*

## NUALLALARM

`nuallalarm.NUAllAlarm(bambou.nurest_object.NUMetaRESTObject, ) :`

The alarm API allows the management of system alarms.

### 6.1 Attributes

- **name (Mandatory):** The alarm name. Each type of alarm will generate its own name
- **target\_object:** Identifies affected Entity. Example: Alarm generated by TCA for Domain domain1(Packets towards a VM, Breach)
- **last\_updated\_by:** ID of the user who last updated the object.
- **acknowledged:** Flag to indicate that alarm is already acknowledged or not
- **reason:** Provides a description of the alarm
- **description:** Description of the alarm
- **severity:** Severity of the alarm.
- **timestamp:** Indicates the time that the alarm was triggered
- **enterprise\_id:** Enterprise that this alarm belongs to
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **error\_condition:** Identifies the error condition
- **number\_of\_occurrences:** Number of times that the alarm was triggered
- **external\_id:** External object ID. Used for integration with third party systems

### 6.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 6.3 Parents

- `nuenterprise.NUEnterprise`



## NUAPPLICATION

`nuapplication.NUApplication(bambou.nurest_object.NUMetaRESTObject, ) :`

represents a application with L4/L7 classification

### 7.1 Attributes

- `dscp`: DSCP match condition to be set in the rule
- `name` (**Mandatory**): name of the application
- `read_only`: determines whether this entity is read only. Read only objects cannot be modified or deleted.
- `performance_monitor_type`: Describes the trigger for the application.
- `description`: description of Application
- `destination_ip`: destination IP in CIDR format
- `destination_port`: value should be either \* or single port number
- `enable_pps`: Enable the performance probe for this application
- `one_way_delay`: one way Delay
- `one_way_jitter`: one way Jitter
- `one_way_loss`: one way loss
- `post_classification_path`: default set to any , possible values primary/secondary/any
- `source_ip`: source IP address
- `source_port`: source Port ,value should be either \* or single port number
- `optimize_path_selection`: with values being Latency, Jitter, PacketLoss
- `pre_classification_path`: default set to primary , possible values primary/secondary
- `protocol`: Protocol number that must be matched
- `associated_l7_application_signature_id` (**Mandatory**): associated Layer7 Application Type ID
- `ether_type`: Ether type of the packet to be matched. etherType can be \* or a valid hexadecimal value
- `symmetry`: Maintain path symmetry during SLA violation

## 7.2 Children

class	fetcher
<i>numonitorscope.NUMonitorscope</i>	monitorscopes
<i>nuapplicationbinding.NUApplicationBinding</i>	application_bindings

## 7.3 Parents

- *nul7applicationsignature.NUL7applicationsignature*
- *nuenterprise.NUEnterprise*

## NUAPPLICATIONBINDING

```
nuapplicationbinding.NUApplicationBinding(bambou.nurest_object.NUMetaRESTObject,):  
None
```

### 8.1 Attributes

- `read_only`: Determines whether this entity is read only.
- `priority`: Priority of the Application within an Application Group
- `associated_application_id`: Associated software application ID

### 8.2 Parents

- *nuapplication.NUApplication*
- *nuapplicationperformancemanagement.NUApplicationperformancemanagement*





## NUAPPLICATIONPERFORMANCEMANAGEMENT

`nuapplicationperformancemanagement.NUApplicationperformancemanagement (bambou.nurest_object`

Application Group is a container for group of applications

### 9.1 Attributes

- **name (Mandatory)**: name of the application group
- **read\_only**: Determines whether this entity is read only. Read only objects cannot be modified or deleted.
- **description**: Description of Application Group
- **application\_group\_type**: with values MONITOR\_PATH,DISCOVERY
- **associated\_performance\_monitor\_id**: associated Probe ID

### 9.2 Children

class	fetcher
<i>nuapplicationbinding.NUApplicationBinding</i>	application_bindings

### 9.3 Parents

- *nuvport.NUVPort*
- *nuperformancemonitor.NUPerformanceMonitor*
- *nuenterprise.NUEnterprise*



## NUAPPLICATIONPERFORMANCEMANAGEMENTBINDING

`nuapplicationperformancemanagementbinding.NUApplicationperformancemanagementbinding` (bamboo)  
Association object for maintaining the priority of AppliationGroup(s) associated to a Domain

### 10.1 Attributes

- `read_only`: Determines whether this entity is read only. Read only objects cannot be modified or deleted.
- `priority`: Priority of the associated Application Group
- `associated_application_performance_management_id`: Associated Application Group ID

### 10.2 Parents

- `nudomain.NUDomain`
- `nul2domain.NUL2Domain`



## NUAPPLICATIONSERVICE

`nuapplicationservice.NUApplicationService(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents a networking communication service.

### 11.1 Attributes

- `dscp`: DSCP match condition to be set in the rule. It is either \* or from 0-63. Required for etherType 0x0800
- `name` (**Mandatory**): Name of the application service.
- `last_updated_by`: ID of the user who last updated the object.
- `description`: Description of the application service.
- `destination_port`: The destination port to be matched if protocol is UDP or TCP. Value should be either \* or single port number or a port range.
- `direction` (**Mandatory**): Direction of the service. Default is UNIDIRECTIONAL.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `source_port`: Source port to be matched if protocol is UDP or TCP. Value can be either \* or single port number or a port range.
- `protocol` (**Mandatory**): Protocol that must be matched. Needs to be 6 (TCP) or 17 (UDP)
- `ether_type` (**Mandatory**): Ether type of the packet to be matched. Ether type can be \* or a valid hexadecimal value
- `external_id`: External object ID. Used for integration with third party systems

### 11.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>

### 11.3 Parents

- `nume.NUMe`

- *nuenterprise.NUEnterprise*

## NUAUTODISCOVERCLUSTER

`nuautodiscovercluster.NUAutoDiscoverCluster(bambou.nurest_object.NUMetaRESTObject,):`  
None

### 12.1 Attributes

- `name`: Name of the shared resource
- `managed_object_id`: VCenter Managed Object ID of the Datacenter
- `last_updated_by`: ID of the user who last updated the object.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `assoc_vcenter_data_center_id`: The ID of the vcenter to which this host is attached
- `external_id`: External object ID. Used for integration with third party systems

### 12.2 Parents

- *nuvcenterdatacenter.NUVCenterDataCenter*





## NUAUTODISCOVEREDDATACENTER

`nuautodiscovereddatacenter.NUAutodiscovereddatacenter(bambou.nurest_object.NUMetaRESTObject`  
None

### 13.1 Attributes

- `name`: Name of the shared resource. Valid characters are alphabets, numbers, space and hyphen( - ).
- `managed_object_id`: VCenter Managed Object ID of the Datacenter
- `last_updated_by`: ID of the user who last updated the object.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `associated_vcenter_id`: The ID of the vcenter to which this host is attached
- `external_id`: External object ID. Used for integration with third party systems

### 13.2 Parents

- *nuvcenter.NUVCenter*



## NUAUTODISCOVEREDGATEWAY

`nuautodiscoveredgateway.NUAutoDiscoveredGateway(bambou.nurest_object.NUMetaRESTObject, ) :`  
Represents Auto discovered Gateway.

### 14.1 Attributes

- **name (Mandatory):** Name of the Gateway
- **last\_updated\_by:** ID of the user who last updated the object.
- **gateway\_id:** The Gateway associated with this Auto Discovered Gateway . This is a read only attribute
- **peer:** The System ID of the peer gateway associated with this Gateway instance when it is discovered by the network manager (VSD) as being redundant.
- **personality (Mandatory):** Personality of the Gateway - VSG,VRSG,NONE,OTHER, cannot be changed after creation.
- **description:** A description of the Gateway
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **controllers:** Controllers to which this gateway instance is associated with.
- **use\_gateway\_vlanvnid:** When set, VLAN-VNID mapping must be unique for all the vports of the gateway
- **vtep:** Represent the system ID or the Virtual IP of a service used by a Gateway (VSG for now) to establish a tunnel with a remote VSG or hypervisor. The format of this field is consistent with an IP address.
- **external\_id:** External object ID. Used for integration with third party systems
- **system\_id:** Identifier of the Gateway

### 14.2 Children

class	fetcher
<i>nuwanservice.NUWANService</i>	wan_services
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuport.NUPort</i>	ports
<i>nunsport.NUNSPort</i>	ns_ports
<i>nueventlog.NUEventLog</i>	event_logs

## 14.3 Parents

- *nume.NUMe*

## NUAUTODISCOVERHYPERVISORFROMCLUSTER

`nuautodiscoverhypervisorfromcluster.NUAutoDiscoverHypervisorFromCluster (bambou.nurest_object_id)`  
None

### 15.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `network_list`: The available network list
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `assoc_entity_id`: The ID of the cluster to which this host is attached
- `external_id`: External object ID. Used for integration with third party systems
- `hypervisor_ip`: IP Address of the Hypervisor

### 15.2 Parents

- `nuvcentercluster.NUVCenterCluster`
- `nuvcenterdatacenter.NUVCenterDataCenter`



## NUAVATAR

`nuavatar.NUAvatar(bambou.nurest_object.NUMetaRESTObject, ) :`

Avatar

### 16.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems
- `type`: The image type

### 16.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 16.3 Parents

- `nuuser.NUUser`
- `nuenterprise.NUEnterprise`





## NUBGPNEIGHBOR

```
nubgpneighbor.NUBGPNeighbor(bambou.nurest_object.NUMetaRESTObject,):
```

None

### 17.1 Attributes

- **name**: Name of the peer
- **dampening\_enabled**: Enable/disable route flap damping.
- **peer\_as** (**Mandatory**): Local autonomous system to be used when establishing a session with the remote peer if it is different from the global BGP router autonomous system number.
- **peer\_ip**: IP Address of the neighbor. If the neighbor is attached to a host vPort this is optional or must be the same as the host's IP. For uplink or bridge vPort neighbors the IP address must be specified
- **description**: Short description for this peer
- **session**: neighbor session yang blob
- **entity\_scope**: Specify if scope of entity is Data center or Enterprise level
- **associated\_export\_routing\_policy\_id**: export policy ID
- **associated\_import\_routing\_policy\_id**: import routing policy ID
- **external\_id**: External object ID. Used for integration with third party systems

### 17.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas

### 17.3 Parents

- *nusubnet.NUSubnet*
- *nuvlan.NUVLAN*
- *nume.NUMe*



## NUBGPPEER

`nubgppeer.NUBGPPeer(bambou.nurest_object.NUMetaRESTObject, ) :`

Encapsulates the BGP peer information for system monitor entity.

### 18.1 Attributes

- `last_state_change`: Last state change timestamp.
- `address`: IP of the BGP peer.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `status`: Current connection status of the BGP peer.
- `external_id`: External object ID. Used for integration with third party systems

### 18.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 18.3 Parents

- `nuhsc.NUHSC`
- `nuvsc.NUVSC`



## NUBGPPROFILE

`nubgpprofile.NUBGPPProfile(bambou.nurest_object.NUMetaRESTObject, ) :`  
missing documentation.

### 19.1 Attributes

- **name (Mandatory):** Per enterprise unique name
- **dampening\_half\_life:** The time in minutes to wait before decrementing dampening penalty.
- **dampening\_max\_suppress:** The maximum duration in minutes that a route will be suppressed.
- **dampening\_name:** Name for the dampening profile. Unique per enterprise
- **dampening\_reuse:** This value is compared with penalty to determine route reusability, If the penalty is greater than the suppress limit, the route will be suppressed; if not, it will be reused.
- **dampening\_suppress:** Specifies the penalty that will be used if a route is suppressed.
- **description:** The description of the BGP Profile
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **associated\_export\_routing\_policy\_id:** export BGP policy ID
- **associated\_import\_routing\_policy\_id:** import BGP policy ID
- **external\_id:** External object ID. Used for integration with third party systems

### 19.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 19.3 Parents

- `nume.NUMe`
- `nuenterprise.NUEnterprise`



## NUBOOTSTRAP

`nubootstrap.NUBootstrap(bambou.nurest_object.NUMetaRESTObject,):`

Gateway bootstrap details.

### 20.1 Attributes

- `zfb_info`: Base64 Encoded JSON String of NSG ZFB Attribute Value Pairs
- `zfb_match_attribute`: Attribute to auto match on
- `zfb_match_value`: Attribute value to auto match on
- `last_updated_by`: ID of the user who last updated the object.
- `installer_id`: The Installer ID
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `status`: Bootstrap status.
- `external_id`: External object ID. Used for integration with third party systems

### 20.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 20.3 Parents

- `nunsgateway.NUNSGateway`





## NUBOOTSTRAPACTIVATION

`nubootstrapactivation.NUBootstrapActivation(bambou.nurest_object.NUMetaRESTObject, ) :`  
NSG Gateway initiated Bootstrap Activation

### 21.1 Attributes

- `cacert`: The CA Certificate Chain
- `hash`: The authentication hash of this request
- `last_updated_by`: ID of the user who last updated the object.
- `action`: The bootstrap action to perform.
- `seed`: The random seed for this request
- `cert`: The signed Certificate
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `config_url`: The configuration URL
- `tpm_owner_password`: TPM owner passphrase
- `tpm_state`: Gateway TPM Status reported by the device when generating CSR.
- `srk_password`: TPM SRK passphrase
- `vsd_time`: VSD Server time when an NSG is initiating a Bootstrapping request
- `csr`: The CSR of the request
- `status`: The agent status for the request
- `auto_bootstrap`: Indicates whether auto bootstrap is being used to bootstrap this NSG
- `external_id`: External object ID. Used for integration with third party systems

### 21.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

## 21.3 Parents

- *nunsgateway.NUNSGateway*

## NUBRCONNECTION

```
nubrconnection.NUBRConnection(bambou.nurest_object.NUMetaRESTObject,):
```

None

### 22.1 Attributes

- `dns_address`: DNS Address for the vlan
- `gateway`: IP address of the gateway bound to the VLAN.
- `address`: Static IP address for the VLAN
- `advertisement_criteria`: Advertisement Criteria for Traffic Flow
- `netmask`: network mask
- `mode`: Connection mode: Static.
- `uplink_id`: Internally generated ID in the range that identifies the uplink within the context of NSG

### 22.2 Parents

- *nuvlan.NUVLAN*
- *nuvlantemplate.NUVLANTemplate*



## NUBRIDGEINTERFACE

`nubridgeinterface.NUBridgeInterface(bambou.nurest_object.NUMetaRESTObject, ) :`

Provides information for each bridge interface.

### 23.1 Attributes

- `vport_id`: ID of the vport that the interface is attached to
- `vport_name`: Name of the vport that the VM is attached to
- `name`: Device name associated with this interface
- `last_updated_by`: ID of the user who last updated the object.
- `gateway`: Gateway of the subnet that the VM is connected to
- `netmask`: Netmask of the subnet that the VM is attached to
- `network_name`: Name of the network that the VM is attached to
- `tier_id`: ID of the tier that the interface is attached to.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `policy_decision_id`: The policy decision ID for this particular interface
- `domain_id`: ID of the domain that the VM is attached to
- `domain_name`: Name of the domain that the VM is attached to
- `zone_id`: ID of the zone that the interface is attached to
- `zone_name`: Name of the zone that the VM is attached to.
- `associated_floating_ip_address`: Floating Ip Address of this network interface eg: 10.1.2.1
- `attached_network_id`: ID of the l2 domain or Subnet that the VM is attached to
- `attached_network_type`: l2 domain or Subnet that the interface is attached to
- `external_id`: External object ID. Used for integration with third party systems

## 23.2 Children

<b>class</b>	<b>fetcher</b>
<i>nutca.NUTCA</i>	tcas
<i>nuredirectiontarget.NURedirectionTarget</i>	redirection_targets
<i>numetadata.NUMetadata</i>	metadatas
<i>nudhcpoption.NUDHCPOption</i>	dhcp_options
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nupolicydecision.NUPolicyDecision</i>	policy_decisions
<i>nupolicygroup.NUPolicyGroup</i>	policy_groups
<i>nuqos.NUQOS</i>	qoss
<i>nustatistics.NUStatistics</i>	statistics
<i>nueventlog.NUEventLog</i>	event_logs

## 23.3 Parents

- *nudomain.NUDomain*
- *nuvport.NUVPort*
- *nul2domain.NUL2Domain*

## NUBULKSTATISTICS

`nubulkstatistics.NUBulkStatistics(bambou.nurest_object.NUMetaRESTObject,):`

Retrieves the statistics for a particular Entity and its immediate child entity.

### 24.1 Attributes

- `data`: Map<TCAMetric, Long[]> TCAMetric is an Enum. Possible values are `packets_in`, `bytes_in`, `packets_in_dropped`, `packets_in_errors`, `packets_out`, `bytes_out`, `packets_out_dropped`, `packets_out_errors`, `packets_out_rate_limit`
- `version`: Version of this Sequence number.
- `end_time`: End time for the statistics to be retrieved
- `start_time`: Start time for the statistics to be retrieved
- `number_of_data_points`: Number of data points between start time and end time

### 24.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 24.3 Parents

- `nupatnatpool.NUPATNATPool`





## NUCERTIFICATE

`nucertificate.NUCertificate(bambou.nurest_object.NUMetaRESTObject,)` :

This object represents a X509 Certificate Request

### 25.1 Attributes

- `pem_encoded`: The PEM encoded certificate.
- `serial_number`: The serial number of this certificate.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `issuer_dn`: The distinguished name of the authority that issued this certificate.
- `subject_dn`: The distinguished name of this certificate's end entity.
- `public_key`: The public key contained in this certificate.
- `external_id`: External object ID. Used for integration with third party systems

### 25.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 25.3 Parents

- `nume.NUMe`



## NUCLOUDMGMTSYSTEM

`nucloudmgmtsystem.NUCloudMgmtSystem(bambou.nurest_object.NUMetaRESTObject, ) :`

Object that identifies a cloud management system.

### 26.1 Attributes

- `name`: Name of the cloud management system
- `last_updated_by`: ID of the user who last updated the object.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 26.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 26.3 Parents

- `nume.NUMe`



## NUCONNECTIONENDPOINT

`nuconnectionendpoint.NUConnectionendpoint (bambou.nurest_object.NUMetaRESTObject, ) :`  
None

### 27.1 Attributes

- `ip_address` (**Mandatory**): IP Address of the end point.
- `ip_type`: IPv4 or IPv6 (only IPv4 supported for now).
- `name` (**Mandatory**): Name of the connection endpoint.
- `description`: A description of the connection endpoint.
- `end_point_type`: Indicates if this endpoint is the source/destination of a network connection.



## NUCONTAINER

`nucontainer.NUContainer(bambou.nurest_object.NUMetaRESTObject,):`

Read only API that can retrieve the containers associated with a domain, zone or subnet for mediation created containers for REST created containers you need to set the additional proxy user header in http request : X-Nuage-ProxyUservalue of the header has to be either :1) `enterpriseName@UserName` (example : Alcatel `Lucent@bob`), or 2) external ID of user in VSD, typically is UUID generally decided by the CMS tool in questionUser needs to have CMS privileges to use proxy user header.

### 28.1 Attributes

- `l2_domain_ids`: Array of IDs of the l2 domain that the container is connected to
- `vrsid`: Id of the VRS that this container is attached to.
- `uuid` (**Mandatory**): UUID of the container
- `name` (**Mandatory**): Name of the container
- `last_updated_by`: ID of the user who last updated the object.
- `reason_type`: Reason of the event associated with the container.
- `delete_expiry`: reflects the container Deletion expiry timer in secs , `deleteMode` needs to be non-null value for `deleteExpiry` to be taken in to effect. CMS created containers will always have `deleteMode` set to `TIMER`
- `delete_mode`: reflects the mode of container Deletion - `TIMER` Possible values are `TIMER`, .
- `resync_info`: Information of the status of the resync operation of a container
- `site_identifier`: This property specifies the site the container belongs to, for Geo-redundancy.
- `image_id`: Id of the container image
- `image_name`: Name of the container image
- `interfaces`: List of container interfaces associated with the container
- `enterprise_id`: ID of the enterprise that this container belongs to
- `enterprise_name`: Name of the enterprise that this container belongs to
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `domain_ids`: Array of IDs of the domain that the container is connected to
- `zone_ids`: Array of IDs of the zone that this container is attached to
- `orchestration_id` (**Mandatory**): Orchestration ID

- `user_id`: ID of the user that created this container
- `user_name`: Username of the user that created this container
- `status`: Status of the container.
- `subnet_ids`: Array of IDs of the subnets that the container is connected to
- `external_id`: External object ID. Used for integration with third party systems
- `hypervisor_ip`: IP address of the hypervisor that this container is currently running in

## 28.2 Children

<b>class</b>	<b>fetcher</b>
<i>numetadata.NUMetadata</i>	metadatas
<i>nualarm.NUAlarm</i>	alarms
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nucontainerinterface.NUContainerInterface</i>	container_interfaces
<i>nucontainerresync.NUContainerResync</i>	container_resyncs
<i>nuvrs.NUVRS</i>	vrss
<i>nueventlog.NUEventLog</i>	event_logs

## 28.3 Parents

- *nuqos.NUQOS*
- *nuzone.NUZone*
- *nudomain.NUDomain*
- *nuvport.NUVPort*
- *nusubnet.NUSubnet*
- *nuvrs.NUVRS*
- *nutier.NUTier*
- *nul2domain.NUL2Domain*
- *nume.NUMe*
- *nuegressacltemplate.NUEgressACLTemplate*
- *nuuser.NUUser*
- *nuenterprise.NUEnterprise*
- *nuingressacltemplate.NUIngressACLTemplate*



## NUCONTAINERINTERFACE

`nucontainerinterface.NUContainerInterface(bambou.nurest_object.NUMetaRESTObject, ) :`

Read only API that can retrieve the container interface associated with a domain, zone or subnet for mediation created container's for REST created container interfaces you need to set the additional proxy header in http request : X-Nuage-ProxyUservalue of the header has to be either :1) `enterpriseName@UserName` (example :`bob@Alcatel Lucent`), or 2) external ID of user in VSD, typically is UUID generally decided by the CMS tool in questionUser needs to have CMS privileges to use proxy user header.

### 29.1 Attributes

- `mac`: MAC address of the interface
- `ip_address`: IP address of the interface
- `vport_id`: ID of the vport that the interface is attached to
- `vport_name`: Name of the vport that the container is attached to
- `name`: Device name associated with this interface
- `last_updated_by`: ID of the user who last updated the object.
- `gateway`: Gateway of the subnet that the container is connected to
- `netmask`: Netmask of the subnet that the container is attached to
- `network_id`: Network ID of the container
- `network_name`: Name of the network that the container is attached to
- `tier_id`: ID of the tier that the interface is attached to.
- `endpoint_id`: End point ID of the container
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `policy_decision_id`: The policy decision ID for this particular interface
- `domain_id`: ID of the domain that the container is attached to
- `domain_name`: Name of the domain that the container is attached to
- `zone_id`: ID of the zone that the interface is attached to
- `zone_name`: Name of the zone that the container is attached to
- `container_uuid`: UUID of the associated container
- `associated_floating_ip_address`: Floating Ip Address of this network interface eg: 10.1.2.1

- `attached_network_id`: ID of the I2 domain or Subnet that the container is attached to
- `attached_network_type`: I2 domain or Subnet that the interface is attached to
- `multi_nic_vport_name`: Name of the Multi NIC VPort associated with this container Interface
- `external_id`: External object ID. Used for integration with third party systems

## 29.2 Children

class	fetcher
<i>nutca.NUTCA</i>	<code>tcas</code>
<i>nuredirectiontarget.NURedirectionTarget</i>	<code>redirection_targets</code>
<i>numetadata.NUMetadata</i>	<code>metadatas</code>
<i>nudhcpoption.NUDHCPOption</i>	<code>dhcp_options</code>
<i>nuglobalmetadata.NUGlobalMetadata</i>	<code>global_metadatas</code>
<i>nupolicydecision.NUPolicyDecision</i>	<code>policy_decisions</code>
<i>nupolicygroup.NUPolicyGroup</i>	<code>policy_groups</code>
<i>nustaticroute.NUStaticRoute</i>	<code>static_routes</code>
<i>nustatistics.NUStatistics</i>	<code>statistics</code>
<i>numulticastchannelmap.NUMultiCastChannelMap</i>	<code>multi_cast_channel_maps</code>
<i>nueventlog.NUEventLog</i>	<code>event_logs</code>

## 29.3 Parents

- *nuzone.NUZone*
- *nudomain.NUDomain*
- *nuvport.NUVPort*
- *nusubnet.NUSubnet*
- *nucontainer.NUContainer*
- *nul2domain.NUL2Domain*
- *nume.NUMe*

## NUCONTAINERRESYNC

`nucontainerresync.NUContainerResync(bambou.nurest_object.NUMetaRESTObject, ) :`

Provide information about the state of a container resync request.

### 30.1 Attributes

- `last_request_timestamp`: Time of the last timestamp received
- `last_time_resync_initiated`: Time that the resync was initiated
- `last_updated_by`: ID of the user who last updated the object.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `status`: Status of the resync
- `external_id`: External object ID. Used for integration with third party systems

### 30.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 30.3 Parents

- `nusubnet.NUSubnet`
- `nucontainer.NUContainer`



## NUDEMARCATIONSERVICE

```
nudemarcationservice.NUDemarcationService(bambou.nurest_object.NUMetaRESTObject,):  
None
```

### 31.1 Attributes

- `route_distinguisher`: The route distinguisher associated with the next hop. This is a read only property automatically created by VSD.
- `priority`: Next hop priority assigned by the user.
- `associated_gateway_id`: The ID of the NSGBR Gateway used as next hop in the untrusted domain.
- `associated_vlanid`: The VLAN ID of the BR VLAN used as next hop in the trusted domain.
- `type`: The type of next hop determines linking direction for a demarcation service, possible values: BR\_PORT, GATEWAY

### 31.2 Parents

- `nulink.NULink`



## NUDHCOPTION

`nudhcption.NUDHCOption(bambou.nurest_object.NUMetaRESTObject, ) :`

Allows the definition of one or more DHCP options that will be provided to all VMs that are associated with a given domain. DHCP options are provided as Type- Length-Value (TLV). There is no validation by VSD on whether these options are valid or not. It is up to the user to guarantee that the options make sense for their application.

### 32.1 Attributes

- `value`: DHCP option value. Value should be a hexadecimal value(ie. Hex value 0xac40 would be passed as 'ac40')
- `last_updated_by`: ID of the user who last updated the object.
- `actual_type`: This will be used to send actual type instead of the hexadecimal. Note: If `actualType` is set, it will override the entry set in the type attribute
- `actual_values`: This will be used to send actual values instead of the hexadecimal. Note: If `actualValues` are set, it will override entry set in the value attribute
- `length`: DHCP option length. Length should be a hexadecimal value(ie. Hex value 0x04 would be passed as '04')
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems
- `type` (**Mandatory**): DHCP option type. Type should be a hexadecimal value(ie. Hex value 0x06 would be passed as '06')

### 32.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>

### 32.3 Parents

- `nucontainerinterface.NUContainerInterface`

- *nusharednetworkresource.NUSharedNetworkResource*
- *nuzone.NUZone*
- *nudomain.NUDomain*
- *nuvport.NUVPort*
- *nusubnet.NUSubnet*
- *nuvminterface.NUVMInterface*
- *nubridgeinterface.NUBridgeInterface*
- *nul2domain.NUL2Domain*
- *nuhostinterface.NUHostInterface*



## NUDISKSTAT

`nudiskstat.NUDiskStat(bambou.nurest_object.NUMetaRESTObject, ) :`

Encapsulates the disk usage metrics for system monitor entity.

### 33.1 Attributes

- `name`: Name of the disk.
- `size`: Total disk space.
- `unit`: Storage unit type (example: bytes, KB, MB, etc.,).
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `used`: Disk space used.
- `available`: Available disk space.
- `external_id`: External object ID. Used for integration with third party systems



## NUDOMAIN

`nudomain.NUDomain(bambou.nurest_object.NUMetaRESTObject, ) :`

This object is used to manipulate domain state. A domain corresponds to a distributed Virtual Router and Switch.

### 34.1 Attributes

- `pat_enabled`: Indicates whether PAT is enabled for the subnets in this domain - ENABLED/DISABLED Possible values are INHERITED, ENABLED, DISABLED, .
- `ecmp_count`: Domain specific Equal-cost multi-path routing count, ECMPCount = 1 means no ECMP
- `bgp_enabled`: Read only flag to display if BGP is enabled for this domain
- `dhcp_behavior`: DHCPBehaviorType is an enum that indicates DHCP Behavior of VRS having VM's under this domain. Possible values are FLOOD, CONSUME ,RELAY Possible values are CONSUME, FLOOD, RELAY, .
- `dhcp_server_address`: when DHCPBehaviorType is RELAY, then DHCP Server IP Address needs to be set
- `dpi`: determines whether or not Deep packet inspection is enabled
- `label_id`: The label associated with the dVRS. This is a read only attribute
- `back_haul_route_distinguisher`: Route distinguisher associated with the BackHaul Service in dVRS. If not provided during creation, System generates this identifier automatically
- `back_haul_route_target`: Route target associated with the BackHaul Service in dVRS. If not provided during creation, System generates this identifier automatically
- `back_haul_subnet_ip_address`: IP Address of the backhaul subnet
- `back_haul_subnet_mask`: Network mask of the backhaul subnet
- `back_haul_vnid`: Current BackHaul Network's globally unique VXLAN network identifier generated by VSD
- `maintenance_mode`: maintenanceMode is an enum that indicates if the Domain is accepting VM activation requests. Possible values are DISABLED, ENABLED and ENABLED\_INHERITED Possible values are .
- `name (Mandatory)`: The name of the domain. Valid characters are alphabets, numbers, space and hyphen( - ).
- `last_updated_by`: ID of the user who last updated the object.
- `leaking_enabled`: Indicates if this domain is a leakable domain or not - boolean true/false
- `secondary_dhcp_server_address`: when DHCPBehaviorType is RELAY, then DHCP Server IP Address needs to be set

- `template_id` (**Mandatory**): The ID of the template that this domain was created from. This should be set when instantiating a domain
- `permitted_action`: The permitted action to USE/DEPLOY for the Domain Possible values are USE, READ, ALL, INSTANTIATE, EXTEND, DEPLOY, .
- `service_id`: The serviceID of the Virtual Router created in VSC and is associated with this object. This is auto-generated by VSD
- `description`: A description string of the domain that is provided by the user
- `dhcp_server_addresses`: when DHCPBehaviorType is RELAY, then DHCP Server IP Address needs to be set
- `global_routing_enabled`: Indicates if this domain is a globally routable domain or not - boolean true/false
- `import_route_target`: Route distinguisher associated with the dVRS. It is an optional parameter that can be provided by the user or auto-managed by VSD. System generates this identifier automatically, if not provided
- `encryption`: Determines whether IPSEC is enabled Possible values are ENABLED, DISABLED, .
- `underlay_enabled`: Indicates whether UNDERLAY is enabled for the subnets in this domain
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `policy_change_status`: None
- `domain_id`: A unique 20-bitID editable however could be auto-generated by VSD.
- `route_distinguisher`: Route distinguisher associated with the dVRS. It is an optional parameter that can be provided by the user or auto-managed by VSD. System generates this identifier automatically, if not provided
- `route_target`: Route target associated with the dVRS. It is an optional parameterthat can be provided by the user or auto-managed by VSDSystem generates this identifier automatically, if not provided
- `uplink_preference`: Indicates the preferential path selection for network traffic in this domain - Default is Primary 1 and Secondary 2. Possible values are PRIMARY\_SECONDARY, SECONDARY\_PRIMARY, PRIMARY, SECONDARY, SYMMETRIC, .
- `application_deployment_policy`: Application deployment policy.
- `associated_bgp_profile_id`: None
- `associated_multicast_channel_map_id`: The ID of the Multi Cast Channel Map this domain is associated with. This has to be set when enableMultiCast is set to ENABLED
- `associated_pat_mapper_id`: The ID of the PatMapper entity to which this l3-domain is associated to.
- `stretched`: Indicates whether this domain is stretched,if so remote VM resolutions will be allowed
- `multicast`: multicast is enum that indicates multicast policy on domain. Possible values are ENABLED ,DISABLED and INHERITED Possible values are INHERITED, ENABLED, DISABLED, .
- `tunnel_type`: Default Domain Tunnel Type
- `customer_id`: The customerID that is created in the VSC and identifies this dVRS. This is auto-generated by VSD
- `export_route_target`: Route target associated with the dVRS. It is an optional parameterthat can be provided by the user or auto-managed by VSDSystem generates this identifier automatically, if not provided
- `external_id`: External object ID. Used for integration with third party systems

## 34.2 Children

class	fetcher
<i>nutca.NUTCA</i>	tcas
<i>nuredirectiontarget.NURedirectionTarget</i>	redirection_targets
<i>nupermision.NUPermission</i>	permissions
<i>numetadata.NUMetadata</i>	metadatas
<i>nuegressaclentrytemplate.NUEgressACLEntryTemplate</i>	egress_acl_entry_templates
<i>nuegressacltemplate.NUEgressACLTemplate</i>	egress_acl_templates
<i>nudomainfipacltemplate.NUDomainFIPACLTemplate</i>	domain_fip_acl_templates
<i>nufloatingipacltemplate.NUFloatingIPACLTemplate</i>	floating_ipacl_templates
<i>nudhcpoption.NUDHCPOption</i>	dhcp_options
<i>nulink.NULink</i>	links
<i>nufirewallacl.NUFirewallAcl</i>	firewall_acls
<i>nufloatingip.NUFloatingIp</i>	floating_ips
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuvm.NUVM</i>	vms
<i>nuvminterface.NUVMInterface</i>	vm_interfaces
<i>ningressaclentrytemplate.NUIngressACLEntryTemplate</i>	ingress_acl_entry_templates
<i>ningressacltemplate.NUIngressACLTemplate</i>	ingress_acl_templates
<i>ningressadvfwdtemplate.NUIngressAdvFwdTemplate</i>	ingress_adv_fwd_templates
<i>ningressexternalservicetemplate.NUIngressExternalServiceTemplate</i>	ingress_external_service_templates
<i>nujob.NUJob</i>	jobs
<i>nupolicygroup.NUPolicyGroup</i>	policy_groups
<i>nudomain.NUDomain</i>	domains
<i>nudomaintemplate.NUDomainTemplate</i>	domain_templates
<i>nuzone.NUZone</i>	zones
<i>nucontainer.NUContainer</i>	containers
<i>nucontainerinterface.NUContainerInterface</i>	container_interfaces
<i>nuqos.NUQOS</i>	qoss
<i>nuhostinterface.NUHostInterface</i>	host_interfaces
<i>nuroutingpolicy.NURoutingPolicy</i>	routing_policies
<i>nuuplinkrd.NUUplinkRD</i>	uplink_rds
<i>nuvpnconnection.NUVPNConnection</i>	vpn_connections
<i>nuvport.NUVPort</i>	vports
<i>nuapplicationperformancemanagementbinding.NUApplicationperformancemanagementbinding</i>	applicationperformancema
<i>nubridgeinterface.NUBridgeInterface</i>	bridge_interfaces
<i>nugroup.NUGroup</i>	groups
<i>nustaticroute.NUStaticRoute</i>	static_routes
<i>nustatistics.NUStatistics</i>	statistics
<i>nustatisticspolicy.NUStatisticsPolicy</i>	statistics_policies
<i>nusubnet.NUSubnet</i>	subnets
<i>nueventlog.NUEventLog</i>	event_logs
<i>nuexternalappservice.NUExternalAppService</i>	external_app_services

## 34.3 Parents

- *nufirewallacl.NUFirewallAcl*
- *nudomain.NUDomain*

- *nume.NUme*
- *nudomaintemplate.NUDomainTemplate*
- *nuenterprise.NUEnterprise*

## NUDOMAINFIPACLTEMPLATE

`nudomainfipacltemplate.NUDomainFIPACLTemplate(bambou.nurest_object.NUMetaRESTObject,)` :

Defines the template for an Domain Floating IP ACL

### 35.1 Attributes

- `name`: The name of the entity
- `last_updated_by`: ID of the user who last updated the object.
- `active`: If enabled, it means that this ACL or QOS entry is active
- `default_allow_ip`: If enabled a default ACL of Allow All is added as the last entry in the list of ACL entries
- `default_allow_non_ip`: If enabled, non ip traffic will be dropped
- `description`: A description of the entity
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `entries`: List of Egress Domain ACL entries associated with this ACL
- `policy_state`: State of the policy
- `priority`: The priority of the ACL entry that determines the order of entries
- `priority_type`:
- `associated_live_entity_id`: ID of the associated live entity
- `external_id`: External object ID. Used for integration with third party systems

### 35.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nudomainfipacltemplateentry.NUDomainFIPACLTemplateEntry</code>	<code>domain_fip_acl_template_entries</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

## 35.3 Parents

- *nudomain.NUDomain*
- *nume.NUMe*
- *nudomaintemplate.NUDomainTemplate*



## NUDOMAINFIPACLTEMPLATEENTRY

`nudomainfipacltemplateentry.NUDomainFIPACLTemplateEntry(bambou.nurest_object.NUMetaRESTObj`

Defines the template of Egress Domain ACL Template entries

### 36.1 Attributes

- `acl_template_name` (**Mandatory**): The name of the parent Template for this acl entry
- `icmp_code`: The ICMP Code when protocol selected is ICMP
- `icmp_type`: The ICMP Type when protocol selected is ICMP
- `dscp`: DSCP match condition to be set in the rule. It is either \* or from 0-63
- `last_updated_by`: ID of the user who last updated the object.
- `action`: The action of the ACL entry DROP or FORWARD or REDIRECT. Action REDIRECT is allowed only for IngressAdvancedForwardingEntry
- `action_details`: Type of action to be performed when a ACL match criteria succeeds
- `address_override`: Overrides the source IP for Ingress and destination IP for Egress, macentries will use this adress as the match criteria.
- `reflexive`: True means that this ACL entry is reflexive, so there will be a corresponding rule that will be created by OVS in the network. False means that there is no corresponding rule created by OVS in the network.
- `description`: Description of the ACL entry
- `dest_pg_id`: In case of PG this will be its EVPNBGPCommunity String, incase of network it will be network cidr
- `dest_pg_type`: In case of PG this will be its EVPNBGPCommunity String, incase of network it will be network cidr
- `destination_port`: The destination port to be matched if protocol is UDP or TCP. Value should be either \* or single port number or a port range
- `destination_type`: Network Type - either PolicyGroup or Network
- `destination_value`: In case of PG this will be its EVPNBGPCommunity String, incase of network it will be network cidr
- `network_id`: The destination network entity that is referenced(subnet/zone/macro)
- `network_type`: Type of the source network - VM\_SUBNET or VM\_ZONE or VM\_DOMAIN or SUBNET or ZONE or ENTERPRISE\_NETWORK or PUBLIC\_NETWORK or ANY

- `mirror_destination_id`: This is the ID of the `mirrorDestination` entity associated with this entity
- `flow_logging_enabled`: Is flow logging enabled for this particular template
- `enterprise_name` (**Mandatory**): The name of the enterprise for the domains parent
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `location_id`: The ID of the location entity (Subnet/Zone/VportTag)
- `location_type`: Type of the location entity - ANY or SUBNET or ZONE or VPORTTAG
- `policy_state`: State of the policy.
- `domain_name` (**Mandatory**): The name of the domain/domain template for the `aclTemplateName` parent
- `source_pg_id`: In case of PG this will be its `EVPNBGPCommunity` String, incase of network it will be `network cidr`
- `source_pg_type`: In case of PG this will be its `EVPNBGPCommunity` String, incase of network it will be `network cidr`
- `source_port`: Source port to be matched if protocol is UDP or TCP. Value can be either \* or single port number or a port range
- `source_type`: Location Type - either `PolicyGroup` or `Network`
- `source_value`: In case of PG this will be its `EVPNBGPCommunity` String, incase of network it will be `network cidr`
- `priority`: The priority of the ACL entry that determines the order of entries
- `protocol`: Protocol number that must be matched
- `associated_application_id`: The associated application ID
- `associated_application_object_id`: The associated application object ID
- `associated_application_object_type`: The associated application object type
- `associated_live_entity_id`: ID of the associated live entity
- `stateful`: True means that this ACL entry is stateful, so there will be a corresponding rule that will be created by OVS in the network. False means that there is no corresponding rule created by OVS in the network.
- `stats_id`: The `statsID` that is created in the VSD and identifies this ACL Template Entry. This is auto-generated by VSD
- `stats_logging_enabled`: Is stats logging enabled for this particular template
- `ether_type`: Ether type of the packet to be matched. `etherType` can be \* or a valid hexadecimal value
- `external_id`: External object ID. Used for integration with third party systems

## 36.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

## 36.3 Parents

- *nudomainfipacltemplate.NUDomainFIPAcITemplate*



## NUDOMAINTEMPLATE

`nudomaintemplate.NUDomainTemplate(bambou.nurest_object.NUMetaRESTObject, ) :`

Domains in VSD are created from domain templates. This object provides the definition of the Domain Template.

### 37.1 Attributes

- `dpi`: determines whether or not Deep packet inspection is enabled
- `name` (**Mandatory**): The name of the domain template, that is unique within an enterprise. Valid characters are alphabets, numbers, space and hyphen ( - ).
- `last_updated_by`: ID of the user who last updated the object.
- `description`: Domain template description provided by the user
- `encryption`: Determines whether IPSEC is enabled. Possible values are ENABLED, DISABLED, .
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `policy_change_status`: None
- `associated_bgp_profile_id`: The ID of the associated BGP profile
- `associated_multicast_channel_map_id`: The ID of the Multi Cast Channel Map this domain template is associated with. This has to be set when `enableMultiCast` is set to ENABLED
- `associated_pat_mapper_id`: The ID of the PatMapper entity to which this domain-template is associated to.
- `multicast`: Indicates multicast policy on domain.
- `external_id`: External object ID. Used for integration with third party systems

## 37.2 Children

class	fetcher
<i>nuredirectiontargettemplate.NURedirectionTargetTemplate</i>	redirection_target_templates
<i>nupermision.NUPermission</i>	permissions
<i>numetadata.NUMetadata</i>	metadatas
<i>nuegressacltemplate.NUEgressACLTemplate</i>	egress_acl_templates
<i>nudomainfipacltemplate.NUDomainFIPACLTemplate</i>	domain_fip_acl_templates
<i>nufloatingipacltemplate.NUFloatingIPACLTemplate</i>	floating_ipacl_templates
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>ningressacltemplate.NUIngressACLTemplate</i>	ingress_acl_templates
<i>ningressadvfwdtemplate.NUIngressAdvFwdTemplate</i>	ingress_adv_fwd_templates
<i>ningressexternalservicetemplate.NUIngressExternalServiceTemplate</i>	ingress_external_service_templates
<i>nujob.NUJob</i>	jobs
<i>nupolicygrouptemplate.NUPolicyGroupTemplate</i>	policy_group_templates
<i>nudomain.NUDomain</i>	domains
<i>nuzonetemplate.NUZoneTemplate</i>	zone_templates
<i>nuqos.NUQOS</i>	qoss
<i>nugroup.NUGroup</i>	groups
<i>nusubnettemplate.NUSubnetTemplate</i>	subnet_templates
<i>nueventlog.NUEventLog</i>	event_logs

## 37.3 Parents

- *nudomain.NUDomain*
- *nuenterprise.NUEnterprise*

## NUDSCPFORWARDINGCLASSMAPPING

`nudscpforwardingclassmapping.NUDSCPForwardingClassMapping(bambou.nurest_object.NUMetaRESTO`

Provides the definition of a single DSCP to a Forwarding class mapping that is part of a Table used in QoS policies.

### 38.1 Attributes

- `dscp`: DSCP value range from enumeration of 65 values : \*, 0, 1, ..., 63
- `last_updated_by`: ID of the user who last updated the object.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `forwarding_class`: Class of service to be used. Service classes in order of priority are A, B, C, D, E, F, G, and H.
- `external_id`: External object ID. Used for integration with third party systems

### 38.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 38.3 Parents

- `nudscpforwardingclasstable.NUDSCPForwardingClassTable`





## NUDSCPFORWARDINGCLASSTABLE

`nudscpforwardingclasstable.NUDSCPForwardingClassTable (bambou.nurest_object.NUMetaRESTObject`

Provides the definition of a table that holds multiple DSCP to Forwarding class mappings. Used in QoS policies.

### 39.1 Attributes

- **name (Mandatory):** A unique name of the dscp-fc mapping table.
- **last\_updated\_by:** ID of the user who last updated the object.
- **description:** A description of the dscp-fc mapping table.
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **external\_id:** External object ID. Used for integration with third party systems

### 39.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nudscpforwardingclassmapping.NUDSCPForwardingClassMapping</i>	dscp_forwarding_class_mappings

### 39.3 Parents

- *numenterprise.NUEnterprise*



## NUDUCGROUP

`nuducgroup.NUDUCGroup(bambou.nurest_object.NUMetaRESTObject, ) :`

None

### 40.1 Attributes

- `name`: Name given to the UBR Group.
- `description`: Description of the UBR Group.
- `associated_ducs`: List of NSG UBRs (formely named DUCs) that belong to this Disjoint Underlay Connector Group.
- `associated_performance_monitor_id`: Identification of the Performance Monitoring Probe that is associated with this instance of a UBR Group.

### 40.2 Children

class	fetcher
<code>nunsgateway.NUNSGateway</code>	<code>ns_gateways</code>

### 40.3 Parents

- `nume.NUMe`



## NUDUCGROUPBINDING

```
nuducgroupbinding.NUDUCGroupBinding(bambou.nurest_object.NUMetaRESTObject,):  
None
```

### 41.1 Attributes

- `id`: VSD UUID of the UBR Group Binding instance.
- `one_way_delay`: SLA delay value in milliseconds that is tolerated between NSG instances and NSG-UBR (DUC) instances being bound through this binding instance. If delay is to be ignored, then the value of -1 is to be entered. Value 0 is not permitted.
- `priority`: The priority for NSG Group to UBR Group relationship.
- `associated_duc_group_id`: Identification of the UBR Group associated to this group binding instance.

### 41.2 Parents

- *nunsggroup.NUNSGGroup*



## NUEGRESSACLENTRYTEMPLATE

`nuegressaclentrytemplate.NUEgressACLEntryTemplate(bambou.nurest_object.NUMetaRESTObject,)` :

Defines the template of Egress ACL Template entries

### 42.1 Attributes

- `acl_template_name` (**Mandatory**): The name of the parent Template for this acl entry
- `icmp_code`: The ICMP Code when selected protocol is ICMP
- `icmp_type`: The ICMP Type when selected protocol is ICMP
- `ipv6_address_override`: Overrides the source IPv6 for Ingress and destination IP for Egress, macentries will use this adress as the match criteria.
- `dscp` (**Mandatory**): DSCP match condition to be set in the rule. It is either \* or from 0-63
- `last_updated_by`: ID of the user who last updated the object.
- `action` (**Mandatory**): The action of the ACL entry.
- `address_override`: Overrides the source IP for Ingress and destination IP for Egress, macentries will use this adress as the match criteria.
- `reflexive`: True means that this ACL entry is reflexive, so there will be a corresponding rule that will be created by OVS in the network. False means that there is no corresponding rule created by OVS in the network.
- `description`: Description of the ACL entry
- `destination_port`: The destination port to be matched if protocol is UDP or TCP. Value should be either \* or single port number or a port range
- `network_id`: The destination network entity that is referenced(subnet/zone/macro)
- `network_type`: Type of the source network.
- `mirror_destination_id`: Destination ID of the mirror destination object.
- `flow_logging_enabled`: Is flow logging enabled for this particular template
- `enterprise_name` (**Mandatory**): The name of the enterprise for the domains parent
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `location_id`: The ID of the location entity (Subnet/Zone/VportTag)
- `location_type` (**Mandatory**): Type of the location entity.
- `policy_state`: State of the policy. Possible values are DRAFT, LIVE, .

- `domain_name` (**Mandatory**): The name of the domain/domain template for the `aclTemplateNames` parent
- `source_port`: Source port to be matched if protocol is UDP or TCP. Value can be either \* or single port number or a port range
- `priority`: The priority of the ACL entry that determines the order of entries
- `protocol`: Protocol number that must be matched
- `associated_application_id`: The associated application ID
- `associated_application_object_id`: The associated application object ID
- `associated_application_object_type`: The associated application object type Refer to API section for supported types.
- `associated_live_entity_id`: In the draft mode, the ACL entry refers to this LiveEntity. In non-drafted mode, this is null.
- `stateful`: True means that this ACL entry is stateful, so there will be a corresponding rule that will be created by OVS in the network. False means that there is no corresponding rule created by OVS in the network.
- `stats_id`: The statsID that is created in the VSD and identifies this ACL Template Entry. This is auto-generated by VSD
- `stats_logging_enabled`: Is stats logging enabled for this particular template
- `ether_type` (**Mandatory**): Ether type of the packet to be matched. `etherType` can be \* or a valid hexadecimal value
- `external_id`: External object ID. Used for integration with third party systems

## 42.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nujob.NUJob</i>	jobs
<i>nustatistics.NUStatistics</i>	statistics

## 42.3 Parents

- *nudomain.NUDomain*
- *numirrordestination.NUMirrorDestination*
- *nul2domain.NUL2Domain*
- *nume.NUMe*
- *nuegressacltemplate.NUEgressACLTemplate*



## NUEGRESSACLTEMPLATE

`nuegressacltemplate.NUEgressACLTemplate(bambou.nurest_object.NUMetaRESTObject,):`

Defines the template for an Egress ACL.

### 43.1 Attributes

- `name` (**Mandatory**): The name of the entity
- `last_updated_by`: ID of the user who last updated the object.
- `active`: If enabled, it means that this ACL or QOS entry is active
- `default_allow_ip`: If enabled a default ACL of Allow All is added as the last entry in the list of ACL entries
- `default_allow_non_ip`: If enabled, non ip traffic will be dropped
- `default_install_acl_implicit_rules`: If enabled, implicit rule will allow intra domain traffic by default
- `description`: A description of the entity
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `policy_state`:
- `priority`: The priority of the ACL entry that determines the order of entries
- `priority_type`:
- `associated_live_entity_id`: In the draft mode, the ACL entry refers to this LiveEntity. In non-drafted mode, this is null.
- `external_id`: External object ID. Used for integration with third party systems

## 43.2 Children

<b>class</b>	<b>fetcher</b>
<i>numetadata.NUMetadata</i>	metadatas
<i>nuegressaclentrytemplate.NUEgressACLEntryTemplate</i>	egress_acl_entry_templates
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuvm.NUVM</i>	vms
<i>nujob.NUJob</i>	jobs
<i>nucontainer.NUContainer</i>	containers
<i>nueventlog.NUEventLog</i>	event_logs

## 43.3 Parents

- *nudomain.NUDomain*
- *nul2domaintemplate.NUL2DomainTemplate*
- *nul2domain.NUL2Domain*
- *nume.NUMe*
- *nudomaintemplate.NUDomainTemplate*

## NUEGRESSQOSPOLICY

`nuegressqospolicy.NUEgressQOSPolicy(bambou.nurest_object.NUMetaRESTObject,)` :

The object manipulates Egress QoS parameters attached to a Access Port / VLAN or Network port.

### 44.1 Attributes

- **name (Mandatory):** A unique name of the QoS object
- **parent\_queue\_associated\_rate\_limiter\_id:** ID of the parent rate limiter associated with this Egress QOS policy.
- **last\_updated\_by:** ID of the user who last updated the object.
- **description:** A description of the QoS object
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **assoc\_egress\_qos\_id:** ID of object associated with this QoS object
- **queue1\_associated\_rate\_limiter\_id:** ID of the queue1 rate limiter associated with this Egress QOS policy.
- **queue1\_forwarding\_classes:** Queue1 Forwarding Classes for this Egress QOS Policy Possible values are NONE, A, B, C, D, E, F, G, H, .
- **queue2\_associated\_rate\_limiter\_id:** ID of the queue2 rate limiter associated with this Egress QOS policy.
- **queue2\_forwarding\_classes:** Queue2 Forwarding Classes for this Egress QOS Policy Possible values are NONE, A, B, C, D, E, F, G, H, .
- **queue3\_associated\_rate\_limiter\_id:** ID of the queue3 rate limiter associated with this Egress QOS policy.
- **queue3\_forwarding\_classes:** Queue3 Forwarding Classes for this Egress QOS Policy Possible values are NONE, A, B, C, D, E, F, G, H, .
- **queue4\_associated\_rate\_limiter\_id:** ID of the queue4 rate limiter associated with this Egress QOS policy.
- **queue4\_forwarding\_classes:** Queue4 Forwarding Classes for this Egress QOS Policy Possible values are NONE, A, B, C, D, E, F, G, H, .
- **external\_id:** External object ID. Used for integration with third party systems

## 44.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas

## 44.3 Parents

- *nume.NUMe*
- *nuenterprise.NUEnterprise*

## NUENDPOINT

`nuendpoint.NUEndPoint(bambou.nurest_object.NUMetaRESTObject,):`

Representation of End Point

### 45.1 Attributes

- `name`: unique name of the External Service.
- `last_updated_by`: ID of the user who last updated the object.
- `description`: Description of the External Service.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 45.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>

### 45.3 Parents

- `nuexternalservice.NUExternalService`



## NUENTERPRISE

`nenterprise.NUEnterprise(bambou.nurest_object.NUMetaRESTObject,):`

Definition of the enterprise object. This is the top level object that represents an enterprise.

### 46.1 Attributes

- `ldap_authorization_enabled`: Read-only flag - indicates if LDAP is used for authorization for the enterprise. For detailed explanation, see definition in `LDAPConfiguration` class
- `ldap_enabled`: Read-only flag - indicates if LDAP is used for authentication for the enterprise. For detailed explanation, see definition in `LDAPConfiguration` class
- `bgp_enabled`: Read only flag to display if BGP is enabled for this enterprise
- `dhcp_lease_interval`: DHCP Lease Interval (in hrs) to be used by an enterprise.
- `name` (**Mandatory**): The unique name of the enterprise. Valid characters are alphabets, numbers, space and hyphen( - ).
- `last_updated_by`: ID of the user who last updated the object.
- `receive_multi_cast_list_id`: Readonly Id of the auto generated receive multicast list associated with this enterprise profile
- `send_multi_cast_list_id`: Readonly Id of the auto generated send multicast list associated with this enterprise profile
- `description`: A description of the enterprise
- `dictionary_version`: L7 Application Type version
- `allow_advanced_qos_configuration`: Controls whether this enterprise has access to advanced QoS settings
- `allow_gateway_management`: This flag indicates if the enterprise/organization can manage gateways. If yes then it can create gateway templates, instantiate them etc.
- `allow_trusted_forwarding_class`: Controls whether QoS policies and templates created under this enterprise set the trusted flag to true
- `allowed_forwarding_classes`: Allowed Forwarding Classes for this enterprise. Possible values are NONE, A, B, C, D, E, F, G, H, .
- `floating_ips_quota`: Quota set for the number of floating IPs to be used by an enterprise.
- `floating_ips_used`: Number of floating IPs used by the enterprise from the assigned `floatingIPsQuota`

- `enable_application_performance_management`: This flag indicates if the DPI can be enabled for this enterprise/enterprise/organization.
- `encryption_management_mode`: Readonly encryption management mode of the associated profile
- `enterprise_profile_id`: Enterprise profile id for this enterprise
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `local_as`: Local autonomous system for the enterprise
- `associated_enterprise_security_id`: Readonly Id of the associated group key encryption profile
- `associated_group_key_encryption_profile_id`: Readonly Id of the associated group key encryption profile
- `associated_key_server_monitor_id`: Readonly Id of the associated keyserver monitor
- `customer_id`: CustomerID that is used by VSC to identify this enterprise. This is a read only attribute.
- `avatar_data`: URL to the avatar data associated with the enterprise. If the avatarType is URL then value of avatarData should an URL of the image. If the avatarType BASE64 then avatarData should be BASE64 encoded value of the image
- `avatar_type`: Avatar type - URL or BASE64 Possible values are URL, BASE64, COMPUTEDURL, .
- `external_id`: External object ID. Used for integration with third party systems

## 46.2 Children

class	fetcher
<code>nul2domain.NUL2Domain</code>	<code>l2_domains</code>
<code>nul2domaintemplate.NUL2DomainTemplate</code>	<code>l2_domain_templates</code>
<code>nul7applicationsignature.NUL7applicationsignature</code>	<code>l7applicationsignatures</code>
<code>nuratelimiter.NURateLimiter</code>	<code>rate_limiters</code>
<code>nugateway.NUGateway</code>	<code>gateways</code>
<code>nugatewaytemplate.NUGatewayTemplate</code>	<code>gateway_templates</code>
<code>nupatnatpool.NUPATNATPool</code>	<code>patnat_pools</code>
<code>nuldapconfiguration.NULDAConfiguration</code>	<code>ldap_configurations</code>
<code>nuredundancygroup.NURedundancyGroup</code>	<code>redundancy_groups</code>
<code>nuperformancemonitor.NUPerformanceMonitor</code>	<code>performance_monitors</code>
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>numetadatatag.NUMetadataTag</code>	<code>metadata_tags</code>
<code>nunetworkmacrogroup.NUNetworkMacroGroup</code>	<code>network_macro_groups</code>
<code>nunetworkperformancemeasurement.NUNetworkPerformanceMeasurement</code>	<code>network_performance_measurements</code>
<code>nukeyservermonitor.NUKeyServerMonitor</code>	<code>key_server_monitors</code>
<code>nuzfbrequest.NUZFBRequest</code>	<code>zfb_requests</code>
<code>nubgppprofile.NUBGPPProfile</code>	<code>bgp_profiles</code>
<code>nuegressqospolicy.NUEgressQOSPolicy</code>	<code>egress_qos_policies</code>
<code>nusharednetworkresource.NUSharedNetworkResource</code>	<code>shared_network_resources</code>
<code>nufirewallacl.NUFirewallAcl</code>	<code>firewall_acls</code>
<code>nufirewallrule.NUFirewallRule</code>	<code>firewall_rules</code>
<code>nuikecertificate.NUIKECertificate</code>	<code>ike_certificates</code>
<code>nuikeencryptionprofile.NUIKEEncryptionprofile</code>	<code>ike_encryptionprofiles</code>
<code>nuikegateway.NUIKEGateway</code>	<code>ike_gateways</code>

Continued on next page



Table 46.1 – continued from previous page

<i>nuikegatewayprofile.NUIKEGatewayProfile</i>	ike_gateway_profiles
<i>nuikepsk.NUIKEPSK</i>	ikepsks
<i>nualarm.NUAlarm</i>	alarms
<i>nuallalarm.NUAllAlarm</i>	all_alarms
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuvm.NUVM</i>	vms
<i>nuenterprisenetwork.NUEnterpriseNetwork</i>	enterprise_networks
<i>nuenterprisesecurity.NUEnterpriseSecurity</i>	enterprise_securities
<i>nujob.NUJob</i>	jobs
<i>nudomain.NUDomain</i>	domains
<i>nudomaintemplate.NUDomainTemplate</i>	domain_templates
<i>nucontainer.NUContainer</i>	containers
<i>nuroutingpolicy.NURoutingPolicy</i>	routing_policies
<i>nuapplication.NUApplication</i>	applications
<i>nuapplicationperformancemanagement.NUApplicationperformancemanagement</i>	applicationperformancemanagements
<i>nuapplicationservice.NUApplicationService</i>	application_services
<i>nugroup.NUGroup</i>	groups
<i>nugroupkeyencryptionprofile.NUGroupKeyEncryptionProfile</i>	group_key_encryption_profiles
<i>nudscpforwardingclasstable.NUDSCPForwardingClassTable</i>	dscp_forwarding_class_tables
<i>nuuser.NUUser</i>	users
<i>nunsgateway.NUNSGateway</i>	ns_gateways
<i>nunsgatewaytemplate.NUNSGatewayTemplate</i>	ns_gateway_templates
<i>nunsggroup.NUNSGGroup</i>	nsg_groups
<i>nunsredundantgatewaygroup.NUNSRedundantGatewayGroup</i>	ns_redundant_gateway_groups
<i>nupublicnetworkmacro.NUPublicNetworkMacro</i>	public_network_macros
<i>numulticastlist.NUMultiCastList</i>	multi_cast_lists
<i>nuavatar.NUAvatar</i>	avatars
<i>nueventlog.NUEventLog</i>	event_logs
<i>nuexternalappservice.NUExternalAppService</i>	external_app_services
<i>nuexternalservice.NUExternalService</i>	external_services

## 46.3 Parents

- *nuenterpriseprofile.NUEnterpriseProfile*
- *nume.NUMe*



## NUENTERPRISENETWORK

`numenterprisenetwork.NUEnterpriseNetwork(bambou.nurest_object.NUMetaRESTObject, ) :`

Administrators of an enterprise can define macros that are set of IP addresses that identify enterprise networks. These macros can be used in the ACL definitions by network designers and other users to identify access restrictions towards specific enterprise networks.

### 47.1 Attributes

- `ip_type`: IPv4 or IPv6(only IPv4 is supported in R1.0) Possible values are IPV4, IPV6, .
- `ipv6_address`: IPv6 address of the subnet defined. In case of zone, this is an optional field for and allows users to allocate an IP address range to a zone. The VSD will auto-assign IP addresses to subnets from this range if a specific IP address is not defined for the subnet
- `name` (**Mandatory**): Name of the current entity(Zone or zone template or subnet etc..) Valid characters are alphabets, numbers, space and hyphen( - ).
- `last_updated_by`: ID of the user who last updated the object.
- `address` (**Mandatory**): IP address of the subnet defined. In case of zone, this is an optional field for and allows users to allocate an IP address range to a zone. The VSD will auto-assign IP addresses to subnets from this range if a specific IP address is not defined for the subnet
- `netmask` (**Mandatory**): Netmask of the subnet defined
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 47.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nunetworkmacrogroup.NUNetworkMacroGroup</code>	<code>network_macro_groups</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>

### 47.3 Parents

- `nunetworkmacrogroup.NUNetworkMacroGroup`

- *numenterprise.NUEnterprise*

## NUENTERPRISEPERMISSION

`numenterprisepermission.NUEnterprisePermission(bambou.nurest_object.NUMetaRESTObject,)` :  
Represents Enterprise Permission for a CSP entity.

### 48.1 Attributes

- `name`: Name of the Permission
- `last_updated_by`: ID of the user who last updated the object.
- `permitted_action` (**Mandatory**): The permitted action.
- `permitted_entity_description`: Description for the permittedEntity
- `permitted_entity_id`: The enterprise permitted to use/extend this Gateway
- `permitted_entity_name`: Name of the entity for which we have given permission.
- `permitted_entity_type`: Type of the entity for which we have given permission.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 48.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 48.3 Parents

- `nusharednetworkresource.NUSharedNetworkResource`
- `nuredundancygroup.NURedundancyGroup`
- `nuvsgredundantport.NUVsgRedundantPort`
- `nuwanservice.NUWANService`
- `nuport.NUPort`
- `nuvlan.NUVLAN`

- *nugateway.NUGateway*
- *nunsgateway.NUNSGateway*
- *nunsport.NUNSPort*
- *nupatnatpool.NUPATNATPool*

## NUENTERPRISEPROFILE

`nenterpriseprofile.NUEnterpriseProfile(bambou.nurest_object.NUMetaRESTObject,):`

Enterprise profile, used to store an enterprise's policies, quota etc.

### 49.1 Attributes

- `bgp_enabled`: Enable BGP for this enterprise profile
- `dhcp_lease_interval`: DHCP Lease Interval (in hours) to be used by an enterprise.
- `dpi_enabled`: Enable DPI for this Enterprise Profile
- `name` (**Mandatory**): The unique name of the enterprise. Valid characters are alphabets, numbers, space and hyphen( - ).
- `last_updated_by`: ID of the user who last updated the object.
- `receive_multi_cast_list_id`: Readonly ID of the auto generated receive multicast list associated with this enterprise profile
- `send_multi_cast_list_id`: Readonly ID of the auto generated send multicast list associated with this enterprise profile
- `description`: A description of the enterprise/organisation profile.
- `allow_advanced_qos_configuration`: Controls whether this enterprise has access to advanced QoS settings.
- `allow_gateway_management`: If set to true lets the enterprise admin create gateway templates and instances.
- `allow_trusted_forwarding_class`: Controls whether QoS policies and templates created under this enterprise set the trusted flag to true
- `allowed_forwarding_classes`: Allowed Forwarding Classes for this enterprise. Possible values are NONE, A, B, C, D, E, F, G, H, .
- `floating_ips_quota`: Quota set for the number of floating IPs to be used by an enterprise.
- `enable_application_performance_management`: Enable DPI for this enterprise
- `encryption_management_mode`: encryption management mode for this enterprise Possible values are DISABLED, MANAGED, .
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

## 49.2 Children

<b>class</b>	<b>fetcher</b>
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuenterprise.NUEnterprise</i>	enterprises
<i>numulticastlist.NUMultiCastList</i>	multi_cast_lists
<i>nueventlog.NUEventLog</i>	event_logs
<i>nuexternalservice.NUExternalService</i>	external_services

## 49.3 Parents

- *nume.NUMe*



## NUENTERPRISESECUREDATA

`numenterprisesecreddata.NUEnterpriseSecuredData(bambou.nurest_object.NUMetaRESTObject,)` :

This object represents the secured data object under the enterprise

### 50.1 Attributes

- `hash`: authentication hash
- `last_updated_by`: ID of the user who last updated the object.
- `data`: encrypted data
- `sek_id`: Seed Encryption Key id that encrypted this data
- `keyserver_cert_serial_number`: Serial Number of the certificate needed to verify the encrypted data
- `signed_hash`: private key signed hash
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 50.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 50.3 Parents

- `numenterprisesecurity.NUEnterpriseSecurity`



## NUENTERPRISESECURITY

`numenterprisesecurity.NUEnterpriseSecurity(bambou.nurest_object.NUMetaRESTObject, ) :`

This object represents the enterprise security

### 51.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `gateway_security_revision`: change revision number for the gateway security data
- `revision`: revision number for the enterprise security data
- `enterprise_id`: The enterprise associated with this object. This is a read only attribute
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 51.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>numenterprisesecureddata.NUEnterpriseSecuredData</code>	<code>enterprise_secured_datas</code>

### 51.3 Parents

- `numenterprise.NUEnterprise`



## NUEVENTLOG

`nueventlog.NUEventLog(bambou.nurest_object.NUMetaRESTObject, ) :`

The API retrieves the events related to a particular entity

### 52.1 Attributes

- `diff`: Holds the results of diff between two objects of same type.
- `enterprise`: The enterprise name of the user who triggered this event.
- `entities`: List of entities associated with the event.
- `entity_id`: The entity id associated with this event.
- `entity_parent_id`: The entity parent id associated with this event. It can be null.
- `entity_parent_type`: Event parent entity type. Generally reported against enterprise.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `entity_type`: The entity type of this event. It may be Domain, VirtualMachine, etc.,
- `user`: The authenticated user who triggered this event.
- `event_received_time`: The time that event was received.
- `external_id`: External object ID. Used for integration with third party systems
- `type`: The event type (CREATE, UPDATE or DELETE).

### 52.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 52.3 Parents

- `nucontainerinterface.NUContainerInterface`
- `nuqos.NUQOS`
- `nuvirtualip.NUVirtualIP`

- *numulticastchannelmap.NUMultiCastChannelMap*
- *nuredundancygroup.NURedundancyGroup*
- *nutca.NUTCA*
- *nugroup.NUGroup*
- *nuzone.NUZone*
- *nupolicygrouptemplate.NUPolicyGroupTemplate*
- *nuflowsecuritypolicy.NUFlowSecurityPolicy*
- *nuenterprisenetwork.NUEnterpriseNetwork*
- *nupermision.NUPermission*
- *nuireservation.NUIPReservation*
- *nuredirectiontargettemplate.NURedirectionTargetTemplate*
- *nusubnettemplate.NUSubnetTemplate*
- *numetadatatag.NUMetadataTag*
- *nuredirectiontarget.NURedirectionTarget*
- *numetadata.NUMetadata*
- *nuvsp.NUVSP*
- *nudomain.NUDomain*
- *nuwanservice.NUWANService*
- *nuvsd.NUVSD*
- *nuvport.NUVPort*
- *nuflowforwardingpolicy.NUFlowForwardingPolicy*
- *nuport.NUPort*
- *nusubnet.NUSubnet*
- *nucontainer.NUContainer*
- *nuzonetemplate.NUZoneTemplate*
- *nuvrs.NUVRS*
- *nuhsc.NUHSC*
- *nuvlan.NUVLAN*
- *nustaticroute.NUStaticRoute*
- *nuvminterface.NUVMInterface*
- *nulicense.NULicense*
- *nuenterpriseprofile.NUEnterpriseProfile*
- *nubridgeinterface.NUBridgeInterface*
- *numulticastrange.NUMultiCastRange*
- *nuautodiscoveredgateway.NUAutoDiscoveredGateway*
- *nutier.NUTier*

- *nugateway.NUGateway*
- *nul2domaintemplate.NUL2DomainTemplate*
- *nuapplicationservice.NUApplicationService*
- *nul2domain.NUL2Domain*
- *nuhostinterface.NUHostInterface*
- *nuendpoint.NUEndPoint*
- *nudhcption.NUDHCPOption*
- *nunsgateway.NUNSGateway*
- *nuvsc.NUVSC*
- *nuvm.NUVM*
- *nunsport.NUNSPort*
- *nupublicnetworkmacro.NUPublicNetworkMacro*
- *nuaddressrange.NUAddressRange*
- *nudomaintemplate.NUDomainTemplate*
- *nufloatingip.NUFloatingIp*
- *nuegressacltemplate.NUEgressACLTemplate*
- *nunsredundantgatewaygroup.NUNSRedundantGatewayGroup*
- *nuexternalservice.NUExternalService*
- *nuuser.NUUser*
- *nupolicygroup.NUPolicyGroup*
- *nuflow.NUFlow*
- *nuenterprise.NUEnterprise*
- *nuingressacltemplate.NUIngressACLTemplate*





## NUEXTERNALAPPSERVICE

`nuexternalappservice.NUExternalAppService(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents an External Service in the Application Designer.

### 53.1 Attributes

- `name` (**Mandatory**): Name of the flow.
- `last_updated_by`: ID of the user who last updated the object.
- `description`: Description of the flow.
- `destination_nat_address`: Destination NAT Address
- `destination_nat_enabled`: Boolean flag to indicate whether source NAT is enabled
- `destination_nat_mask`: netmask of the Destination NAT
- `metadata`: metadata
- `egress_type`: Egress type.
- `virtual_ip`: Virtual IP Address
- `virtual_ip_required`: Boolean flag to indicate whether we require a VIP
- `ingress_type`: Ingress type.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `source_nat_address`: Source NAT Address
- `source_nat_enabled`: Boolean flag to indicate whether source NAT is enabled
- `associated_service_egress_group_id`: ID of service port group identifying the output ports
- `associated_service_egress_redirect_id`: the redirect target ID that identifies the output ports
- `associated_service_ingress_group_id`: ID of service port group identifying the input ports
- `associated_service_ingress_redirect_id`: the redirect target ID that identifies the input ports
- `external_id`: External object ID. Used for integration with third party systems

## 53.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas

## 53.3 Parents

- *nudomain.NUDomain*
- *nume.NUMe*
- *nuenterprise.NUEnterprise*

## NUEXTERNALSERVICE

`nuexternalservice.NUExternalService(bambou.nurest_object.NUMetaRESTObject,)` :

Representation of External Service.

### 54.1 Attributes

- **name (Mandatory)**: unique name of the External Service.
- **last\_updated\_by**: ID of the user who last updated the object.
- **service\_type (Mandatory)**: Type of the service.
- **description**: Description of the External Service.
- **direction**: Direction
- **entity\_scope**: Specify if scope of entity is Data center or Enterprise level
- **stage**: Stage - START,END Possible values are START, .
- **external\_id**: External object ID. Used for integration with third party systems

### 54.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>numetadatatag.NUMetadataTag</i>	metadata_tags
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuendpoint.NUEndPoint</i>	end_points
<i>nueventlog.NUEventLog</i>	event_logs

### 54.3 Parents

- *nuenterpriseprofile.NUEnterpriseProfile*
- *nume.NUMe*
- *nuenterprise.NUEnterprise*



## NUFIREWALLACL

```
nufirewallacl.NUFirewallAcl(bambou.nurest_object.NUMetaRESTObject,):
```

None

### 55.1 Attributes

- **name**: The name of the entity
- **active**: If enabled, it means that this ACL or QOS entry is active
- **default\_allow\_ip**: If enabled a default ACL of Allow All is added as the last entry in the list of ACL entries
- **default\_allow\_non\_ip**: If enabled, non ip traffic will be dropped
- **description**: A description of the entity
- **rule\_ids**: Firewall rules associated with this firewall acl.

### 55.2 Children

class	fetcher
<i>nufirewallrule.NUFirewallRule</i>	firewall_rules
<i>nudomain.NUDomain</i>	domains

### 55.3 Parents

- *nudomain.NUDomain*
- *numenterprise.NUEnterprise*



## NUFIREWALLRULE

`nufirewallrule.NUFirewallRule(bambou.nurest_object.NUMetaRESTObject,):`

None

### 56.1 Attributes

- `acl_template_name`: The name of the parent Template for this acl entry
- `icmp_code`: The ICMP Code when protocol selected is ICMP
- `icmp_type`: The ICMP Type when protocol selected is ICMP
- `ipv6_address_override`: Overrides the source IPV6 for Ingress and destination IPV6 for Egress, macentries will use this address as the match criteria.
- `dscp`: DSCP match condition to be set in the rule. It is either \* or from 0-63
- `action`: The action of the ACL entry DROP or FORWARD or REDIRECT.
- `address_override`: Overrides the source IP for Ingress and destination IP for Egress, macentries will use this adress as the match criteria.
- `description`: Description of the ACL entry
- `dest_network`: Destination network - available in version 1.0 api
- `dest_pg_id`: In case of PG this will be its EVPNBGPCommunity String, incase of network it will be network cidr
- `dest_pg_type`: In case of PG this will be its EVPNBGPCommunity String, incase of network it will be network cidr
- `destination_ipv6_value`: destination IPV6 address
- `destination_port`: The destination port to be matched if protocol is UDP or TCP. Value should be either \* or single port number or a port range
- `destination_type`: Network Type - either PolicyGroup or Network
- `destination_value`: In case of PG this will be its EVPNBGPCommunity String, incase of network it will be network cidr
- `network_id`: The destination network entity that is referenced(subnet/zone/macro)
- `network_type`: Type of the source network - VM\_SUBNET or VM\_ZONE or VM\_DOMAIN or SUBNET or ZONE or ENTERPRISE\_NETWORK or PUBLIC\_NETWORK or ANY
- `mirror_destination_id`: This is the ID of the mirrorDestrination entity associated with this entity

- `flow_logging_enabled`: Is flow logging enabled for this particular template
- `enterprise_name`: The name of the enterprise for the domains parent
- `location_id`: The ID of the location entity (Subnet/Zone/VportTag)
- `location_type`: Type of the location entity - ANY or SUBNET or ZONE or VPORTTAG
- `domain_name`: The name of the domain/domain template for the `aclTemplateNames` parent
- `source_ipv6_value`: source IPV6 address
- `source_network`: Source network - available in version 1.0 api
- `source_pg_id`: In case of PG this will be its `EVPNBGPCommunity` String, incase of network it will be `network cidr`
- `source_pg_type`: in case of PG this will be its `EVPNBGPCommunity` String, incase of network itdomainfip will be `network cidr`
- `source_port`: Source port to be matched if protocol is UDP or TCP. Value can be either \* or single port number or a port range
- `source_type`: Location Type - either `PolicyGroup` or `Network`
- `source_value`: In case of PG this will be its `EVPNBGPCommunity` String, incase of network it will be `network cidr`
- `priority`: The priority of the ACL entry that determines the order of entries
- `associated_application_id`: The associated application ID
- `associated_application_object_id`: The associated application object ID
- `associatedfirewall_aclid`: Associated Firewall Acl ID
- `stateful`: true means that this ACL entry is stateful, so there will be a corresponding rule that will be created by OVS in the network. false means that there is no correspondingrule created by OVS in the network
- `stats_id`: The statsID that is created in the VSD and identifies this ACL Template Entry.. This is auto-generated by VSD
- `stats_logging_enabled`: Is stats logging enabled for this particular template
- `ether_type`: Ether type of the packet to be matched. `etherType` can be \* or a valid hexadecimal value

## 56.2 Parents

- *nufirewallacl.NUFirewallAcl*
- *numenterprise.NUEnterprise*



## NUFLOATINGIP

`nufloatingip.NUFloatingIp(bambou.nurest_object.NUMetaRESTObject, ) :`

Floating IP that is associated to a Domain. This floating IP could be used in the VM interface for NAT functionality.

### 57.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `access_control`: If access control is enabled this FIP is part of the Internet PG.
- `address`: Floating IP address assigned to the Domain
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `assigned`: True if this floating IP is assigned to a network interface else the value is false
- `assigned_to_object_type`: The object type to which this floating ip is assigned. Eg. vport or virtualip
- `associated_shared_network_resource_id` (**Mandatory**): Id of the shared network resource subnet which was used to get this floating IP address
- `external_id`: External object ID. Used for integration with third party systems

### 57.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nuvport.NUVPort</code>	<code>vports</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>

### 57.3 Parents

- `nudomain.NUDomain`
- `nume.NUMe`



## NUFLOATINGIPACLTEMPLATE

`nufloatingipacltemplate.NUFloatingIPACLTemplate(bambou.nurest_object.NUMetaRESTObject, ) :`

Defines the template for an Floating IP ACL

### 58.1 Attributes

- `name`: The name of the entity
- `last_updated_by`: ID of the user who last updated the object.
- `active`: If enabled, it means that this ACL or QOS entry is active
- `default_allow_ip`: If enabled a default ACL of Allow All is added as the last entry in the list of ACL entries
- `default_allow_non_ip`: If enabled, non ip traffic will be dropped
- `description`: A description of the entity
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `policy_state`: State of the policy
- `priority`: The priority of the ACL entry that determines the order of entries
- `priority_type`: Priority type
- `associated_live_entity_id`: ID of the associated live entity
- `external_id`: External object ID. Used for integration with third party systems

### 58.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nufloatingipacltemplateentry.NUFloatingIPACLTemplateEntry</code>	<code>floating_ipacl_template_entries</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 58.3 Parents

- `nudomain.NUDomain`

- *nume.NUMe*
- *nudomaintemplate.NUDomainTemplate*

## NUFLOATINGIPACLTEMPLATEENTRY

`nufloatingipacltemplateentry.NUFloatingIPACLTemplateEntry(bambou.nurest_object.NUMetaRESTO`

Defines the template of Egress ACL Template entries

### 59.1 Attributes

- `acl_template_name` (**Mandatory**): The name of the parent Template for this acl entry
- `icmp_code`: The ICMP Code when protocol selected is ICMP
- `icmp_type`: The ICMP Type when protocol selected is ICMP
- `dscp`: DSCP match condition to be set in the rule. It is either \* or from 0-63
- `last_updated_by`: ID of the user who last updated the object.
- `action`: The action of the ACL entry DROP or FORWARD or REDIRECT. Action REDIRECT is allowed only for IngressAdvancedForwardingEntry
- `address_override`: Overrides the source IP for Ingress and destination IP for Egress, macentries will use this adress as the match criteria.
- `reflexive`: True means that this ACL entry is reflexive, so there will be a corresponding rule that will be created by OVS in the network. False means that there is no corresponding rule created by OVS in the network.
- `description`: Description of the ACL entry
- `destination_port`: The destination port to be matched if protocol is UDP or TCP. Value should be either \* or single port number or a port range
- `network_id`: The destination network entity that is referenced(subnet/zone/macro)
- `network_type`: Type of the source network - VM\_SUBNET or VM\_ZONE or VM\_DOMAIN or SUBNET or ZONE or ENTERPRISE\_NETWORK or PUBLIC\_NETWORK or ANY
- `mirror_destination_id`: This is the ID of the mirrorDestination entity associated with this entity
- `flow_logging_enabled`: Is flow logging enabled for this particular template
- `enterprise_name` (**Mandatory**): The name of the enterprise for the domains parent
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `location_id`: The ID of the location entity (Subnet/Zone/VportTag)
- `location_type`: Type of the location entity - ANY or SUBNET or ZONE or VPORTTAG
- `policy_state`: State of the policy.

- `domain_name` (**Mandatory**): The name of the domain/domain template for the `aclTemplateNames` parent
- `source_port`: Source port to be matched if protocol is UDP or TCP. Value can be either \* or single port number or a port range
- `priority`: The priority of the ACL entry that determines the order of entries
- `protocol`: Protocol number that must be matched
- `associated_application_id`: The associated application ID
- `associated_application_object_id`: The associated application object ID
- `associated_application_object_type`: The associated application object type
- `associated_live_entity_id`: ID of the associated live entity
- `stateful`: True means that this ACL entry is stateful, so there will be a corresponding rule that will be created by OVS in the network. False means that there is no corresponding rule created by OVS in the network.
- `stats_id`: The statsID that is created in the VSD and identifies this ACL Template Entry. This is auto-generated by VSD
- `stats_logging_enabled`: Is stats logging enabled for this particular template
- `ether_type`: Ether type of the packet to be matched. `etherType` can be \* or a valid hexadecimal value
- `external_id`: External object ID. Used for integration with third party systems

## 59.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas

## 59.3 Parents

- *nufloatingipacltemplate.NUFloatingIPACLTemplate*

## NUFLOW

`nuflow.NUFlow(bambou.nurest_object.NUMetaRESTObject, ) :`

Flow represents the traffic between two different application tiers.

### 60.1 Attributes

- **name (Mandatory):** Name of the flow.
- **last\_updated\_by:** ID of the user who last updated the object.
- **description:** Description of the flow.
- **destination\_tier\_id:** Flow destination tier id.
- **metadata:** Metadata field to store flow related data.
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **origin\_tier\_id:** Flow origin tier id.
- **external\_id:** External object ID. Used for integration with third party systems

### 60.2 Children

<b>class</b>	<b>fetcher</b>
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nuflowforwardingpolicy.NUFlowForwardingPolicy</code>	<code>flow_forwarding_policies</code>
<code>nuflowsecuritypolicy.NUFlowSecurityPolicy</code>	<code>flow_security_policies</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>





## NUFLOWFORWARDINGPOLICY

`nuflowforwardingpolicy.NUFlowForwardingPolicy(bambou.nurest_object.NUMetaRESTObject,)` :

The redirect policy on the flow.

### 61.1 Attributes

- `redirect_target_id`: The associated service id.
- `destination_address_rewrite`: The destination address rewrite. Needs to be in CIDR format `x.x.x.x/n`
- `flow_id`: The associated service id.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `source_address_rewrite`: The source address rewrite. Needs to be in CIDR format `x.x.x.x/n`
- `associated_application_service_id`: The associated service id.
- `associated_network_object_id`: The associated network object id.
- `associated_network_object_type`: The associated network object type. Refer to API section for supported types.
- `external_id`: External object ID. Used for integration with third party systems
- `type`: The redirect type.

### 61.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>

### 61.3 Parents

- `nuflow.NUFlow`



## NUFLOWSECURITYPOLICY

`nuflowsecuritypolicy.NUFlowSecurityPolicy(bambou.nurest_object.NUMetaRESTObject, ) :`

The security policy on the flow.

### 62.1 Attributes

- `action`: The flow action. The action can be either FORWARD or DROP.
- `destination_address_rewrite`: The destination address rewrite. Needs to be in CIDR format x.x.x.x/n
- `flow_id`: The associated service id.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `source_address_rewrite`: The source address rewrite. Needs to be in CIDR format x.x.x.x/n
- `priority`: The priority of the flow security policy that determines the order of entries.
- `associated_application_service_id`: The associated service id.
- `associated_network_object_id`: The associated network object id.
- `associated_network_object_type`: The associated network object type. Refer to API section for supported types.
- `external_id`: External object ID. Used for integration with third party systems

### 62.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>

### 62.3 Parents

- `nuflow.NUFlow`



## NUGATEWAY

`nugateway.NUGateway(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents Gateway object.

### 63.1 Attributes

- **name (Mandatory):** Name of the Gateway
- **last\_updated\_by:** ID of the user who last updated the object.
- **redundancy\_group\_id:** The Redundancy Gateway Group associated with this Gateway Instance. This is a read only attribute
- **peer:** The System ID of the peer gateway associated with this Gateway instance when it is discovered by the network manager (VSD) as being redundant.
- **template\_id:** The ID of the template that this Gateway was created from. This should be set when instantiating a Gateway
- **pending:** Indicates that this gateway is pending state or state. When in pending state it cannot be modified from REST.
- **permitted\_action:** The permitted action to USE/EXTEND this Gateway.
- **personality (Mandatory):** Personality of the Gateway, cannot be changed after creation.
- **description:** A description of the Gateway
- **enterprise\_id:** The enterprise associated with this Gateway. This is a read only attribute
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **use\_gateway\_vlanvnid:** When set, VLAN-VNID mapping must be unique for all the vports of the gateway
- **vtep:** Represent the system ID or the Virtual IP of a service used by a Gateway (VSG for now) to establish a tunnel with a remote VSG or hypervisor. The format of this field is consistent with an IP address.
- **auto\_disc\_gateway\_id:** The Auto Discovered Gateway associated with this Gateway Instance
- **external\_id:** External object ID. Used for integration with third party systems
- **system\_id:** Identifier of the Gateway, cannot be modified after creation

## 63.2 Children

<b>class</b>	<b>fetcher</b>
<i>nupatnatpool.NUPATNATPool</i>	patnat_pools
<i>nupermision.NUPermission</i>	permissions
<i>nuwanservice.NUWANService</i>	wan_services
<i>numetadata.NUMetadata</i>	metadatas
<i>nualarm.NUAlarm</i>	alarms
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>numenterprisepermission.NUEnterprisePermission</i>	enterprise_permissions
<i>nujob.NUJob</i>	jobs
<i>nuport.NUPort</i>	ports
<i>nueventlog.NUEventLog</i>	event_logs

## 63.3 Parents

- *nuredundancygroup.NURedundancyGroup*
- *nume.NUMe*
- *numenterprise.NUEnterprise*

## NUGATEWAYSECUREDDATA

`nugatewaysecureddata.NUGatewaySecuredData (bambou.nurest_object.NUMetaRESTObject, ) :`

This object represents the secured data object under the gateway

### 64.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `data`: encrypted data
- `gateway_cert_serial_number`: Serial Number of the certificate of the public key that encrypted this data
- `keyserver_cert_serial_number`: Serial Number of the certificate needed to verify the encrypted data
- `signed_data`: private key signed data
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 64.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 64.3 Parents

- `nugatewaysecurity.NUGatewaySecurity`





## NUGATEWAYSECURITY

`nugatewaysecurity.NUGatewaySecurity(bambou.nurest_object.NUMetaRESTObject,)` :

This object represents the gateway security object

### 65.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `gateway_id`: The gateway associated with this object. This is a read only attribute
- `revision`: change revision number for the gateway security data
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 65.2 Children

class	fetcher
<code>nugatewaysecureddata.NUGatewaySecuredData</code>	<code>gateway_secured_datas</code>
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 65.3 Parents

- `nunsgateway.NUNSGateway`



## NUGATEWAYTEMPLATE

`nugatewaytemplate.NUGatewayTemplate(bambou.nurest_object.NUMetaRESTObject,)` :

Represents Gateway Template object.

### 66.1 Attributes

- **name (Mandatory)**: Name of the Gateway
- **last\_updated\_by**: ID of the user who last updated the object.
- **personality (Mandatory)**: Personality of the Gateway, cannot be changed after creation.
- **description**: A description of the Gateway
- **enterprise\_id**: The enterprise associated with this Gateway. This is a read only attribute
- **entity\_scope**: Specify if scope of entity is Data center or Enterprise level
- **external\_id**: External object ID. Used for integration with third party systems

### 66.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuporttemplate.NUPortTemplate</i>	port_templates

### 66.3 Parents

- *nume.NUMe*
- *nuenterprise.NUEnterprise*



## NUGLOBALMETADATA

`nuglobalmetadata.NUGlobalMetadata(bambou.nurest_object.NUMetaRESTObject, ) :`

Metadata associated to a entity.

### 67.1 Attributes

- `name`: name of the Metadata.
- `last_updated_by`: ID of the user who last updated the object.
- `description`: Description of the Metadata.
- `metadata_tag_ids`: metadata tag IDs associated with this metadata you can filter metadata based on this attribute for example X-Nuage-Filter: '2d6fb627-603b-421c-b63a-eb0a6d712761' IN metadataTagIDs
- `network_notification_disabled`: specifies metadata changes need to be notified to controller,by default it is notified
- `blob` (**Mandatory**): Metadata that describes about the entity attached to it.
- `global_metadata`: specifies metadata is global or local
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 67.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>numetadatatag.NUMetadataTag</code>	<code>metadata_tags</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 67.3 Parents

- `nucontainerinterface.NUContainerInterface`
- `nugroupkeyencryptionprofile.NUGroupKeyEncryptionProfile`
- `nuqos.NUQOS`
- `nuingressadvfwdentrytemplate.NUIngressAdvFwdEntryTemplate`

- *nubgppeer.NUBGPPeer*
- *nusharednetworkresource.NUSharedNetworkResource*
- *nuvirtualip.NUVirtualIP*
- *nudscpforwardingclasstable.NUDSCPForwardingClassTable*
- *numulticastchannelmap.NUMultiCastChannelMap*
- *nuredundancygroup.NURedundancyGroup*
- *nutca.NUTCA*
- *nugroup.NUGroup*
- *nuvsgredundantport.NUVsgRedundantPort*
- *nuzone.NUZone*
- *nuikegatewayprofile.NUIKEGatewayProfile*
- *nuikesubnet.NUIKESubnet*
- *nuinfrastructuregatewayprofile.NUInfrastructureGatewayProfile*
- *nupolicygrouptemplate.NUPolicyGroupTemplate*
- *nuflowsecuritypolicy.NUFlowSecurityPolicy*
- *nuvcentereamconfig.NUVCenterEAMConfig*
- *nulocation.NULocation*
- *nuenterprisesecurity.NUEnterpriseSecurity*
- *nuvcentervrsconfig.NUVCenterVRSCfg*
- *nuenterprisenetwork.NUEnterpriseNetwork*
- *nuinfrastructurevscprofile.NUInfrastructureVscProfile*
- *nupermision.NUPermission*
- *nuireservation.NUIPReservation*
- *nuredirectiontargettemplate.NURedirectionTargetTemplate*
- *nusubnettemplate.NUSubnetTemplate*
- *numetadatetag.NUMetadataTag*
- *nufloatingipacltemplate.NUFloatingIPACLTemplate*
- *nuikegatewayconnection.NUIKEGatewayConnection*
- *nuredirectiontarget.NURedirectionTarget*
- *nuegressaclentrytemplate.NUEgressACLEntryTemplate*
- *nuingressexternalservicetemplateentry.NUIngressExternalServiceTemplateEntry*
- *nuavatar.NUAvatar*
- *nubootstrapactivation.NUBootstrapActivation*
- *nucloudmgmtsystem.NUCloudMgmtSystem*
- *nuvsp.NUVSP*
- *nudomain.NUDomain*

- *nuredundantport.NURedundantPort*
- *nudscpforwardingclassmapping.NUDSCPForwardingClassMapping*
- *nunetworklayout.NUNetworkLayout*
- *nuwanservice.NUWANService*
- *nuvsd.NUVSD*
- *nuvport.NUVPort*
- *nuflowforwardingpolicy.NUFlowForwardingPolicy*
- *nuport.NUPort*
- *nustatisticspolicy.NUStatisticsPolicy*
- *nusubnet.NUSubnet*
- *nucontainer.NUContainer*
- *nuratelimiter.NURateLimiter*
- *nukeyservermonitorencryptedseed.NUKeyServerMonitorEncryptedSeed*
- *nuzonetemplate.NUZoneTemplate*
- *nukeyservermonitorseed.NUKeyServerMonitorSeed*
- *nugatewaytemplate.NUGatewayTemplate*
- *nuvrs.NUVRS*
- *nuhsc.NUHSC*
- *nuvlan.NUVLAN*
- *nuldapconfiguration.NULDAPConfiguration*
- *nuvsdcomponent.NUVSDComponent*
- *nuzfbrequest.NUZFBRequest*
- *nuenterprisepermission.NUEnterprisePermission*
- *nustaticroute.NUStaticRoute*
- *nujob.NUJob*
- *nuvminterface.NUVMInterface*
- *nugatewaysecureddata.NUGatewaySecuredData*
- *nuvcenterhypervisor.NUVCenterHypervisor*
- *nukeyservermonitor.NUKeyServerMonitor*
- *nueventlog.NUEventLog*
- *nulicense.NULicense*
- *nuenterpriseprofile.NUEnterpriseProfile*
- *nubridgeinterface.NUBridgeInterface*
- *nuvcentercluster.NUVCenterCluster*
- *numulticastrange.NUMultiCastRange*
- *nunetworkmacrogroup.NUNetworkMacroGroup*

- *nuinfrastructureaccessprofile.NUInfrastructureaccessprofile*
- *nuautodiscoveredgateway.NUAutoDiscoveredGateway*
- *numulticastlist.NUMultiCastList*
- *nunexthop.NUNextHop*
- *nukeyservernotification.NUKeyServerNotification*
- *numirrordestination.NUMirrorDestination*
- *nutier.NUTier*
- *nudomainfipacltemplate.NUDomainFIPACLTemplate*
- *nuaddressmap.NUAddressMap*
- *nugateway.NUGateway*
- *numultinicvport.NUMultiNICVPort*
- *nustatistics.NUStatistics*
- *nunsporttemplate.NUNSPortTemplate*
- *nucertificate.NUCertificate*
- *nuvcenterdatacenter.NUVCenterDataCenter*
- *nul2domaintemplate.NUL2DomainTemplate*
- *nuapplicationservice.NUApplicationService*
- *nuikegateway.NUIKEGateway*
- *nustatscollectorinfo.NUStatsCollectorInfo*
- *nuvcenter.NUVCenter*
- *nubulkstatistics.NUBulkStatistics*
- *nuingressaclentrytemplate.NUIngressACLEntryTemplate*
- *nuroutingpolicy.NURoutingPolicy*
- *nul2domain.NUL2Domain*
- *nuikegatewayconfig.NUIKEGatewayConfig*
- *nuhostinterface.NUHostInterface*
- *nuenterprisesecureddata.NUEnterpriseSecuredData*
- *nuikecertificate.NUIKECertificate*
- *nuingressexternalservicetemplate.NUIngressExternalServiceTemplate*
- *nuporttemplate.NUPortTemplate*
- *nume.NUMe*
- *nuendpoint.NUEndPoint*
- *nudhcption.NUDHCPOption*
- *nukeyservermember.NUKeyServerMember*
- *nunsgateway.NUNSGateway*
- *nunsgatewaytemplate.NUNSGatewayTemplate*



- *nuvsc.NUVSC*
- *nuuplinkrd.NUUplinkRD*
- *nuvm.NUVM*
- *nunsport.NUNSPort*
- *nuvrsaddressrange.NUVRAddressRange*
- *nubgppprofile.NUBGPPProfile*
- *nuegressqospolicy.NUEgressQOSPolicy*
- *nupublicnetworkmacro.NUPublicNetworkMacro*
- *nudomainfipacltemplateentry.NUDomainFIPACLTemplateEntry*
- *nuaddressrange.NUAddressRange*
- *nudomaintemplate.NUDomainTemplate*
- *nusiteinfo.NUSiteInfo*
- *nuvmresync.NUVMResync*
- *nupolicydecision.NUPolicyDecision*
- *nufloatingip.NUFloatingIp*
- *nuegressacltemplate.NUEgressACLTemplate*
- *numonitoringport.NUMonitoringPort*
- *nuvpnconnection.NUVPNConnection*
- *nunsredundantgatewaygroup.NUNSRedundantGatewayGroup*
- *nuexternalservice.NUExternalService*
- *nukeyservermonitorek.NUKeyServerMonitorSEK*
- *nuingressadvfwdtemplate.NUIngressAdvFwdTemplate*
- *nuvportmirror.NUVPortMirror*
- *nupatnatpool.NUPATNATPool*
- *nubgpneighbor.NUBGPNeighbor*
- *nucontainerresync.NUContainerResync*
- *nuallalarm.NUAllAlarm*
- *nuikepsk.NUIKEPSK*
- *nusystemconfig.NUSystemConfig*
- *nuikeencryptionprofile.NUIKEEncryptionprofile*
- *nufloatingipacltemplateentry.NUFloatingIPACLTemplateEntry*
- *nuinfrastructureconfig.NUInfrastructureConfig*
- *nuuser.NUUser*
- *nunatmapentry.NUNATMapEntry*
- *nupolicygroup.NUPolicyGroup*
- *nuexternalappservice.NUExternalAppService*

- *nualarm.NUAlarm*
- *nubootstrap.NUBootstrap*
- *nuflow.NUFlow*
- *nuvlantemplate.NUVLANTemplate*
- *nugatewaysecurity.NUGatewaySecurity*
- *nuglobalmetadata.NUGlobalMetadata*
- *nuenterprise.NUEnterprise*
- *nulink.NULink*
- *nuingressacltemplate.NUIngressACLTemplate*

## NUGROUP

`nugroup.NUGroup(bambou.nurest_object.NUMetaRESTObject, ) :`

Identifies a group within an enterprise

### 68.1 Attributes

- **name (Mandatory):** A unique name of the group
- **management\_mode:** Management mode of the user object - allows for override of external authorization and syncup
- **last\_updated\_by:** ID of the user who last updated the object.
- **account\_restrictions:** Determines whether group is disabled or not.
- **description:** Description of the group
- **restriction\_date:** When the group was disabled.
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **role:** The role associated with this group.
- **private:** A private group is visible only by the owner of the group. Public groups are visible by all users in the enterprise
- **external\_id:** External object ID. Used for integration with third party systems

### 68.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuuser.NUUser</i>	users
<i>nueventlog.NUEventLog</i>	event_logs

### 68.3 Parents

- *nuzone.NUZone*
- *nudomain.NUDomain*

- *nul2domaintemplate.NUL2DomainTemplate*
- *nul2domain.NUL2Domain*
- *nudomaintemplate.NUDomainTemplate*
- *nuuser.NUUser*
- *numenterprise.NUEnterprise*

## NUGROUPKEYENCRYPTIONPROFILE

`nugroupkeyencryptionprofile.NUGroupKeyEncryptionProfile(bambou.nurest_object.NUMetaRESTObj`

Represents a Group Key Profile

### 69.1 Attributes

- `sek_generation_interval`: Group Key SEK Generation Interval in Seconds. Min=1, Max=86400
- `sek_lifetime`: Group Key SEK Lifetime in Seconds. Min=1, Max=86400
- `sek_payload_encryption_algorithm`: Group Key SEK Payload Encryption Algorithm.
- `sek_payload_encryption_bc_algorithm`: Group Key Sek Payload Encryption BC Algorithm (read only)
- `sek_payload_encryption_key_length`: Group Key Sek Payload Encryption Key Length (read only)
- `sek_payload_signing_algorithm`: Group Key SEK Payload Signature Algorithm.
- `name`: Name of the Encryption Profile
- `last_updated_by`: ID of the user who last updated the object.
- `seed_generation_interval`: Group Key SEED Generation Interval in Seconds.
- `seed_lifetime`: Group Key SEED Lifetime in Seconds. Min=1, Max=86400
- `seed_payload_authentication_algorithm`: Group Key SEK Payload Signature Algorithm.
- `seed_payload_authentication_bc_algorithm`: Group Key Seed Payload Authentication Algorithm (read only)
- `seed_payload_authentication_key_length`: Group Key Seed Payload Authentication Key Length (read only)
- `seed_payload_encryption_algorithm`: Group Key SEED Payload Encryption Algorithm.
- `seed_payload_encryption_bc_algorithm`: Group Key Seed Payload Encryption Algorithm (read only)
- `seed_payload_encryption_key_length`: Group Key Seed Payload Encryption Key Length (read only)
- `seed_payload_signing_algorithm`: Group Key Seed Payload Signature Algorithm.
- `description`: A description of the Profile instance created.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level

- `traffic_authentication_algorithm`: Group Key traffic Authentication Algorithm. Possible values are HMAC\_SHA1, HMAC\_SHA256, HMAC\_SHA384, HMAC\_SHA512, HMAC\_MD5, .
- `traffic_encryption_algorithm`: Group Key traffic Encryption Algorithm. Possible values are AES\_128\_CBC, AES\_192\_CBC, AES\_256\_CBC, TRIPLE\_DES\_CBC, .
- `traffic_encryption_key_lifetime`: Group Key Traffic Encryption Key Lifetime in Seconds. Min=1, Max=86400
- `associated_enterprise_id`: The ID of the associated Enterprise
- `external_id`: External object ID. Used for integration with third party systems

## 69.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas

## 69.3 Parents

- *numenterprise.NUEnterprise*

## NUHOSTINTERFACE

`nuhostinterface.NUHostInterface(bambou.nurest_object.NUMetaRESTObject, ) :`

Provides information for each host interface.

### 70.1 Attributes

- `mac`: MAC address of the interface, cannot be modified after creation.
- `ip_address`: IP address of the interface
- `vport_id`: ID of the vport that the interface is attached to
- `vport_name`: Name of the vport that the VM is attached to
- `name`: Device name associated with this interface
- `last_updated_by`: ID of the user who last updated the object.
- `gateway`: Gateway of the subnet that the VM is connected to
- `netmask`: Netmask of the subnet that the VM is attached to
- `network_name`: Name of the network that the VM is attached to
- `tier_id`: ID of the tier that the interface is attached to.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `policy_decision_id`: The policy decision ID for this particular interface
- `domain_id`: ID of the domain that the VM is attached to
- `domain_name`: Name of the domain that the VM is attached to
- `zone_id`: ID of the zone that the interface is attached to
- `zone_name`: Name of the zone that the VM is attached to
- `associated_floating_ip_address`: Floating Ip Address of this network interface eg: 10.1.2.1
- `attached_network_id`: ID of the l2 domain or Subnet that the VM is attached to
- `attached_network_type`: l2 domain or Subnet that the interface is attached to
- `external_id`: External object ID. Used for integration with third party systems

## 70.2 Children

<b>class</b>	<b>fetcher</b>
<i>nutca.NUTCA</i>	tcas
<i>nuredirectiontarget.NURedirectionTarget</i>	redirection_targets
<i>numetadata.NUMetadata</i>	metadatas
<i>nudhcpoption.NUDHCPOption</i>	dhcp_options
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nupolicydecision.NUPolicyDecision</i>	policy_decisions
<i>nupolicygroup.NUPolicyGroup</i>	policy_groups
<i>nuqos.NUQOS</i>	qoss
<i>nustaticroute.NUStaticRoute</i>	static_routes
<i>nustatistics.NUStatistics</i>	statistics
<i>numulticastchannelmap.NUMultiCastChannelMap</i>	multi_cast_channel_maps
<i>nueventlog.NUEventLog</i>	event_logs

## 70.3 Parents

- *nudomain.NUDomain*
- *nuvport.NUVPort*
- *nul2domain.NUL2Domain*
- *nume.NUMe*



## NUHSC

`nuhsc.NUHSC (bambou.nurest_object.NUMetaRESTObject, ) :`

System Monitoring details for hardware service controllers.

### 71.1 Attributes

- `name`: Identifies the entity with a name.
- `management_ip`: The management IP of the VSC/HSC entity
- `last_state_change`: Last state change timestamp (in millis).
- `last_updated_by`: ID of the user who last updated the object.
- `address`: The IP of the VRS entity
- `peak_cpuusage`: Peek CPU usage percentage.
- `peak_memory_usage`: Peek memory usage percentage.
- `description`: Description of the entity.
- `messages`: An array of degraded messages.
- `disks`: Set of disk usage details.
- `already_marked_for_unavailable`: Flag to indicate that it is already marked a unavailable.
- `unavailable_timestamp`: The duration the controller is unavailable (in millis).
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `location`: Identifies the entity to be associated with a location.
- `model`: The model of the hardware service controller
- `product_version`: Product version supported by this entity.
- `vsds`: A collection of VSD id(s) which are identified by this controller.
- `status`: Computed status of the entity. Possible values are UP, DOWN, ADMIN\_DOWN, .
- `current_cpuusage`: Current CPU usage percentage.
- `current_memory_usage`: Current memory usage percentage.
- `average_cpuusage`: Average CPU usage percentage.
- `average_memory_usage`: Average memory usage percentage.
- `external_id`: External object ID. Used for integration with third party systems

- `type`: The type of the hardware service controller.

## 71.2 Children

<b>class</b>	<b>fetcher</b>
<i>numetadata.NUMetadata</i>	metadatas
<i>nubgppeer.NUBGPPeer</i>	bgp_peers
<i>nualarm.NUAlarm</i>	alarms
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nujob.NUJob</i>	jobs
<i>numonitoringport.NUMonitoringPort</i>	monitoring_ports
<i>nuvrs.NUVRS</i>	vrss
<i>nueventlog.NUEventLog</i>	event_logs

## 71.3 Parents

- *nuvsp.NUVSP*
- *nuvrs.NUVRS*

## NUIKECERTIFICATE

`nuikecertificate.NUIKECertificate(bambou.nurest_object.NUMetaRESTObject,)` :

Represents an IKE Trusted Certificate

### 72.1 Attributes

- `pem_encoded`: PEM Encoded Certificate
- `name`: Name of the Encryption Profile
- `last_updated_by`: ID of the user who last updated the object.
- `serial_number`: Serial Number of the Certificate - Read Only Attribute
- `description`: Description of the IKEv2 Authentication
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `not_after`: Date this Certificate is valid to - Read Only Attribute
- `not_before`: Date this Certificate is valid from - Read Only Attribute
- `associated_enterprise_id`: The ID of the associated Enterprise
- `issuer_dn`: Issuer Distinguished Name of the Certificate - Read Only Attribute
- `subject_dn`: Subject Distinguished Name of the Certificate - Read Only Attribute
- `external_id`: External object ID. Used for integration with third party systems

### 72.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 72.3 Parents

- `nuenterprise.NUEnterprise`



## NUIKEENCRIPTIONPROFILE

`nuikeencryptionprofile.NUIKEEncryptionprofile(bambou.nurest_object.NUMetaRESTObject,):`

Represents an IKE Profile

### 73.1 Attributes

- `dpd_interval`: ISAKMP Keep Alive Interval.
- `dpd_mode`: DPD Mode.
- `dpd_timeout`: DPD Timeout.
- `ipsec_authentication_algorithm`: IPsec Authentication Algorithm.
- `ipsec_dont_fragment`: IPsec Don't Fragment
- `ipsec_enable_pfs`: IPsec Enable PFS
- `ipsec_encryption_algorithm`: IPsec Encryption Algorithm.
- `ipsec_pre_fragment`: IPsec PreFragment
- `ipsec_sa_lifetime`: IPsec SA Lifetime in Seconds.
- `ipsec_sa_replay_window_size`: IPsec Replay Window Size in Packets.
- `isakmp_authentication_mode`: ISAKMP Authentication Algorithm.
- `isakmp_diffie_helman_group_identifier`: ISAKMP Diffie-Helman Group Identifier.
- `isakmp_encryption_algorithm`: ISAKMP Encryption Algorithm.
- `isakmp_encryption_key_lifetime`: ISAKMP Encryption Key Lifetime in Seconds
- `isakmp_hash_algorithm`: ISAKMP Hash Algorithm.
- `name`: Name of the Encryption Profile
- `last_updated_by`: ID of the user who last updated the object.
- `sequence`: None
- `description`: A description of the Profile instance created.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `associated_enterprise_id`: The ID of the associated Enterprise
- `external_id`: External object ID. Used for integration with third party systems

## 73.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas

## 73.3 Parents

- *numenterprise.NUEnterprise*

## NUIKEGATEWAY

`nuikegateway.NUIKEGateway(bambou.nurest_object.NUMetaRESTObject,)` :

Represents an IKE Gateway

### 74.1 Attributes

- `ike_version`: The IKE Version
- `ik_ev1_mode`: Mode for IKEv1
- `ip_address`: IP Address of the IKEv2 Gateway
- `name`: Name of the IKEv2 Gateway
- `last_updated_by`: ID of the user who last updated the object.
- `description`: Description of the IKEv2 Gateway
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `associated_enterprise_id`: The ID of the associated Enterprise
- `external_id`: External object ID. Used for integration with third party systems

### 74.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuikegatewayconfig.NUIKEGatewayConfig</code>	<code>ike_gateway_configs</code>
<code>nuikesubnet.NUIKESubnet</code>	<code>ike_subnets</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 74.3 Parents

- `numenterprise.NUEnterprise`





## NUIKEGATEWAYCONFIG

`nuikegatewayconfig.NUIKEGatewayConfig(bambou.nurest_object.NUMetaRESTObject,)` :

Represents an IKE Gateway Configuration Object

### 75.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `config`: Configuration Object
- `external_id`: External object ID. Used for integration with third party systems

### 75.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 75.3 Parents

- `nuikegateway.NUIKEGateway`



## NUIKEGATEWAYCONNECTION

`nuikegatewayconnection.NUIKEGatewayConnection(bambou.nurest_object.NUMetaRESTObject,)` :

Represents an IKE Gateway Connection object

### 76.1 Attributes

- `nsg_identifier`: NSG Identifier. Null to take on the default 'uuid'
- `nsg_identifier_type`: NSG Identifier Type.
- `nsg_role`: NSG role
- `name`: Optional Name of the connection
- `last_updated_by`: ID of the user who last updated the object.
- `sequence`:
- `allow_any_subnet`: Allow any local subnets to be used
- `unencrypted_psk`: Unencrypted PSK
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `port_vlan_name`: The Name of the Port and Vlan the IKEv2 Connection is on
- `priority`: Priority of the IKEv2 Gateway Connection
- `associated_ike_authentication_id`: Associated Authentication ID
- `associated_ike_authentication_type`: Associated Authentication Type
- `associated_ike_encryption_profile_id`: The ID of the associated IKEEncryptionProfile
- `associated_ike_gateway_profile_id`: The ID of the associated IKEGatewayProfile
- `associated_vlanid`: The ID of the associated Vlan
- `external_id`: External object ID. Used for integration with third party systems

### 76.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nusubnet.NUSubnet</code>	<code>subnets</code>

## 76.3 Parents

- *nusubnet.NUSubnet*
- *nuvlan.NUVLAN*

## NUIKEGATEWAYPROFILE

`nuikegatewayprofile.NUIKEGatewayProfile(bambou.nurest_object.NUMetaRESTObject,):`

Represents an IKE Gateway

### 77.1 Attributes

- `ike_gateway_identifier`: IKE Gateway Identifier. Null to take on the default 'ipaddress'
- `ike_gateway_identifier_type`: IKE Gateway Identifier Type.
- `name`: Name of the IKEv2 Gateway Profile
- `last_updated_by`: ID of the user who last updated the object.
- `service_class`: Class of service to be used. Service classes in order of priority are A, B, C, D, E, F, G, and H.
- `description`: Description of the IKEv2 Gateway Profile
- `anti_replay_check`: Allow any local subnets to be used
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `associated_enterprise_id`: The ID of the associated Enterprise
- `associated_ike_authentication_id`: Associated IKE Authentication ID
- `associated_ike_authentication_type`: Associated IKE Authentication Type
- `associated_ike_encryption_profile_id`: The ID of the associated IKE Encryption Profile
- `associated_ike_gateway_id`: The IKE Gateway associated with this Profile. This is a read only attribute
- `external_id`: External object ID. Used for integration with third party systems

### 77.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

## 77.3 Parents

- *numenterprise.NUEnterprise*

## NUIKEPSK

`nuikepsk.NUIKEPSK(bambou.nurest_object.NUMetaRESTObject,)` :

Represents an IKE Pre Shared Key

### 78.1 Attributes

- `name`: Name of the Encryption Profile
- `last_updated_by`: ID of the user who last updated the object.
- `description`: Description of the IKEv2 Authentication
- `signature`: Base64 Encoded private key signature
- `signing_certificate_serial_number`: Serial Number of the certificate needed to verify the encrypted data
- `encrypted_psk`: Base64 Encoded Encrypted PSK
- `encrypting_certificate_serial_number`: Serial Number of the certificate of the public key that encrypted this data
- `unencrypted_psk`: Unencrypted PSK
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `associated_enterprise_id`: The ID of the associated Enterprise
- `auto_created`: Was this object autocreated from the connection
- `external_id`: External object ID. Used for integration with third party systems

### 78.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 78.3 Parents

- `nuenterprise.NUEnterprise`





## NUIKESUBNET

`nuiquesubnet.NUIKESubnet(bambou.nurest_object.NUMetaRESTObject,)` :

Represents an IKE Subnet (remote side)

### 79.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `prefix`: The subnet prefix (eg: 10.0.0.0/24)
- `associated_ike_gateway_id`: The ID of the associated IKEGateway
- `external_id`: External object ID. Used for integration with third party systems

### 79.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 79.3 Parents

- `nuikegateway.NUIKEGateway`



## NUINFRASTRUCTUREACCESSPROFILE

`nuinfrastructureaccessprofile.NUInfrastructureaccessprofile(bambou.nurest_object.NUMetaRES`

Represents an Infrastructure Access Profile

### 80.1 Attributes

- `ssh_auth_mode`: Indicates the Authentication method used during a SSH session.
- `name` (**Mandatory**): Name of the Infrastructure Profile
- `password` (**Mandatory**): Password of the default user associated to the access profile.
- `last_updated_by`: ID of the user who last updated the object.
- `description`: A description of the Profile instance created.
- `enterprise_id`: Enterprise/Organisation associated with this Profile instance.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `source_ip_filter`: Indicates if source based IP filtering is enabled for this access profile.
- `user_name`: Default user name which is associated to the access profile.
- `external_id`: External object ID. Used for integration with third party systems

### 80.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>



## NUINFRASTRUCTURECONFIG

`nuinfrastructureconfig.NUInfrastructureConfig(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents Infrastructure Config

### 81.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `config`: Infrastructure Config
- `config_status`: Status of the configuration application
- `external_id`: External object ID. Used for integration with third party systems

### 81.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 81.3 Parents

- `nunsgateway.NUNSGateway`



## NUINFRASTRUCTUREGATEWAYPROFILE

`nuinfrastructuregatewayprofile.NUInfrastructureGatewayProfile (bambou.nurest_object.NUMetaR`

Represents Infrastructure Gateway Profile

### 82.1 Attributes

- `ntp_server_key`: If set, this represents the security key for the Gateway to communicate with the NTP server (a VSC).
- `ntp_server_key_id`: Correspond to the key ID on the NTP server that matches the `ntpServerKey` value. Valid values are from 1 to 255 as specified by SR-OS and 0 to specify unused (VSD/NSG only).
- `name` (**Mandatory**): Name of the Infrastructure Profile
- `last_updated_by`: ID of the user who last updated the object.
- `datapath_sync_timeout`: Datapath flows sync-time-interval specified in milliseconds
- `dead_timer`: ISO 8601 format duration: **P nYnMnD T nHnMnS**. **P** represents the period field and **T** the time field. Period field: **Y** = year, **M** = month, **D** = day. Time field: **H** = hours, **M** = minutes, **S** = seconds. **n** is the value of each field. Because the years and month are units that vary in length, for the time being those are not supported yet.
- `dead_timer_enabled`: Flag to enable automatic deactivation.
- `remote_log_mode`: Type of Log Server for system logs generated by Gateways associated with this Infrastructure Profile.
- `remote_log_server_address`: Primary Log Server for system logs generated by Gateways associated with this Infrastructure Profile. Can be an IP address or a URL. This field is optional.
- `remote_log_server_port`: Port to be used to access the Remote Syslog server. By default, this is port 514.
- `description`: A description of the Profile instance created.
- `metadata_upgrade_path`: Path/URL to retrieve the NSG Upgrade information meta data files. From that meta data, the NSG will be able to retrieve the upgrade package files and perform some validations. It is expected that the meta data file is in JSON format. RFC 2616 states that there are no 'official' maximum length for a URL but different browsers and servers have limits. Our friendly Internet Explorer has a maximum of 'around' 2048 characters, we shall use this as a limit here.
- `enterprise_id`: Enterprise/Organisation associated with this Profile instance.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `controller_less_duration`: Duration for a controller-less local operation (in ISO-duration format).

- `controller_less_enabled`: Flag to enable controller-less operations.
- `controller_less_forwarding_mode`: The forwarding mode to use for controllerLess operations
- `controller_less_remote_duration`: Duration for a controller-less remote operation (in ISO-duration format).
- `force_immediate_system_sync`: Usually the synchronization will span across 1 hour window after the defined synchronization time. Forcing an immediate synchronization can overload the system and can have a negative impact on the system.
- `upgrade_action`: Upgrade action for NSG associated with this Infrastructure Gateway Profile instance.
- `proxy_dns_name` (**Mandatory**): Proxy DNS Name : DNS Name of the system acting as a proxy between the NSG instances and the VSD.
- `use_two_factor`: Use Two Factor : When set to true, the use of two independent authentication factors will be used to secure the installed NSG. When set to false, there is an assumption that the NSG is being installed in a secure environment and the installer is also trusted. The default value is true, using 2-factor.
- `stats_collector_port`: The port to open by the proxy for stats collector to use
- `external_id`: External object ID. Used for integration with third party systems
- `system_sync_scheduler`: Time in a Cron format when configuration update are being applied on the Gateway (NSG). This property is linked to `systemSyncWindow`. Default value is every midnight (0 0 \* \* \*). Format: Minutes Hours DayOfMonth Month DayOfWeek

## 82.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

## 82.3 Parents

- `nume.NUMe`



## NUINFRASTRUCTUREVSCPROFILE

`nuinfrastructurevscprofile.NUInfrastructureVscProfile (bambou.nurest_object.NUMetaRESTObject`

Represents an Infrastructure VSC Profile.

### 83.1 Attributes

- **name (Mandatory):** Name of the Infrastructure Profile
- **last\_updated\_by:** ID of the user who last updated the object.
- **second\_controller:** Second VSC Controller : IP Address of the secondary VSC system NSG instances associated to this profile will be reaching for.
- **description:** A description of the Profile instance created.
- **first\_controller:** First VSC Controller : IP Address of the first VSC system NSG instances associated to this profile will be reaching for.
- **enterprise\_id:** Enterprise/Organisation associated with this Profile instance.
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **probe\_interval:** Openflow echo timer in millisecond
- **external\_id:** External object ID. Used for integration with third party systems

### 83.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas

### 83.3 Parents

- *nume.NUMe*



## NUINGRESSACLENTRYTEMPLATE

`nuingressaclentrytemplate.NUIngressACLEntryTemplate(bambou.nurest_object.NUMetaRESTObject, )`

Defines the template of Ingress ACL entries

### 84.1 Attributes

- `acl_template_name` (**Mandatory**): The name of the parent Template for this acl entry
- `icmp_code`: The ICMP Code when protocol selected is ICMP.
- `icmp_type`: The ICMP Type when protocol selected is ICMP.
- `ipv6_address_override`: Overrides the source IPv6 for Ingress and destination IPv6 for Egress, macentries will use this address as the match criteria.
- `dscp` (**Mandatory**): DSCP match condition to be set in the rule. It is either \* or from 0-63
- `last_updated_by`: ID of the user who last updated the object.
- `action` (**Mandatory**): The action of the ACL entry DROP or FORWARD or REDIRECT. Action REDIRECT is allowed only for IngressAdvancedForwardingEntry Possible values are DROP, FORWARD, REDIRECT, .
- `address_override`: Overrides the source IP for Ingress and destination IP for Egress, macentries will use this address as the match criteria.
- `reflexive`: True means that this ACL entry is reflexive, so there will be a corresponding rule that will be created by OVS in the network. False means that there is no corresponding rule created by OVS in the network.
- `description`: Description of the ACL entry
- `destination_port`: The destination port to be matched if protocol is UDP or TCP. Value should be either \* or single port number or a port range
- `network_id`: The destination network entity that is referenced(subnet/zone/macro)
- `network_type`: Type of the source network.
- `mirror_destination_id`: Destination ID of the mirror destination object.
- `flow_logging_enabled`: Is flow logging enabled for this particular template
- `enterprise_name` (**Mandatory**): The name of the enterprise for the domains parent
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `location_id`: The ID of the location entity (Subnet/Zone/VportTag)
- `location_type` (**Mandatory**): Type of the location entity.

- `policy_state`: State of the policy.
- `domain_name` (**Mandatory**): The name of the domain/domain template for the `aclTemplateNames` parent
- `source_port`: Source port to be matched if protocol is UDP or TCP. Value can be either \* or single port number or a port range
- `priority`: The priority of the ACL entry that determines the order of entries
- `protocol`: Protocol number that must be matched
- `associated_application_id`: The associated application ID
- `associated_application_object_id`: The associated application object ID
- `associated_application_object_type`: The associated application object type Refer to API section for supported types.
- `associated_live_entity_id`: In the draft mode, the ACL entry refers to this LiveEntity. In non-drafted mode, this is null.
- `stateful`: True means that this ACL entry is stateful, so there will be a corresponding rule that will be created by OVS in the network. False means that there is no corresponding rule created by OVS in the network.
- `stats_id`: The statsID that is created in the VSD and identifies this ACL Template Entry. This is auto-generated by VSD
- `stats_logging_enabled`: Is stats logging enabled for this particular template
- `ether_type` (**Mandatory**): Ether type of the packet to be matched. `etherType` can be \* or a valid hexadecimal value
- `external_id`: External object ID. Used for integration with third party systems

## 84.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	<code>metadatas</code>
<i>nuglobalmetadata.NUGlobalMetadata</i>	<code>global_metadatas</code>
<i>nujob.NUJob</i>	<code>jobs</code>
<i>nustatistics.NUStatistics</i>	<code>statistics</code>

## 84.3 Parents

- *nudomain.NUDomain*
- *numirrordestination.NUMirrorDestination*
- *nul2domain.NUL2Domain*
- *nume.NUMe*
- *nuingressacltemplate.NUIngressACLTemplate*

## NUINGRESSACLTEMPLATE

`nuingressacltemplate.NUIngressACLTemplate (bambou.nurest_object.NUMetaRESTObject, ) :`

Defines the template for an Ingress ACL.

### 85.1 Attributes

- **name (Mandatory):** The name of the entity
- **last\_updated\_by:** ID of the user who last updated the object.
- **active:** If enabled, it means that this ACL or QOS entry is active
- **default\_allow\_ip:** If enabled a default ACL of Allow All is added as the last entry in the list of ACL entries
- **default\_allow\_non\_ip:** If enabled, non ip traffic will be dropped
- **description:** A description of the entity
- **allow\_address\_spoof:** If enabled, it will disable the default anti-spoof ACL for this domain that essentially prevents any VM to send packets that do not originate from that particular VM
- **allow\_l2\_address\_spoof:** If enabled, it will disable the default anti-spoof ACL for this domain that essentially prevents any VM to send packets that do not originate from that particular VM
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **policy\_state:** None
- **priority:** The priority of the ACL entry that determines the order of entries
- **priority\_type:** None
- **assoc\_acl\_template\_id:** ID of the ACL template associated with this ACL template
- **associated\_live\_entity\_id:** In the draft mode, the ACL entry refers to this LiveEntity. In non-drafted mode, this is null.
- **external\_id:** External object ID. Used for integration with third party systems

## 85.2 Children

<b>class</b>	<b>fetcher</b>
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuvm.NUVM</i>	vms
<i>nuingressaclentrytemplate.NUIngressACLEntryTemplate</i>	ingress_acl_entry_templates
<i>nul2job.NUL2Job</i>	jobs
<i>nucontainer.NUContainer</i>	containers
<i>nueventlog.NUEventLog</i>	event_logs

## 85.3 Parents

- *nudomain.NUDomain*
- *nul2domaintemplate.NUL2DomainTemplate*
- *nul2domain.NUL2Domain*
- *nume.NUMe*
- *nudomaintemplate.NUDomainTemplate*

## NUINGRESSADVFWDENTRYTEMPLATE

`nuingressadvfwdentrytemplate.NUIngressAdvFwdEntryTemplate(bambou.nurest_object.NUMetaRESTO`

Defines the template of Ingress Advanced Forwarding entries

### 86.1 Attributes

- `acl_template_name` (**Mandatory**): The name of the parent Template for this acl entry
- `icmp_code`: The ICMP Code when protocol selected is ICMP.
- `icmp_type`: The ICMP Type when protocol selected is ICMP.
- `fc_override`: Value of the Service Class to be overridden in the packet when the match conditions are satisfied Possible values are NONE, A, B, C, D, E, F, G, H, .
- `ipv6_address_override`: Overrides the source IPv6 for Ingress and destination IPv6 for Egress, macentries will use this adress as the match criteria.
- `dscp` (**Mandatory**): DSCP match condition to be set in the rule. It is either \* or from 0-63
- `name` (**Mandatory**): Name of the entity.
- `last_updated_by`: ID of the user who last updated the object.
- `action` (**Mandatory**): The action of the ACL entry DROP or FORWARD or REDIRECT. Action REDIRECT is allowed only for IngressAdvancedForwardingEntry Possible values are DROP, FORWARD, REDIRECT, .
- `address_override`: Overrides the source IP for Ingress and destination IP for Egress, macentries will use this adress as the match criteria.
- `redirect_vport_tag_id`: VPort tag to which traffic will be redirected to, when ACL entry match criteria succeeds
- `description`: Description of the ACL entry
- `destination_port`: The destination port to be matched if protocol is UDP or TCP. Value should be either \* or single port number or a port range
- `network_id`: The destination network entity that is referenced(subnet/zone/macro)
- `network_type`: Type of the source network.
- `mirror_destination_id`: Destination ID of the mirror destination object.
- `flow_logging_enabled`: Is flow logging enabled for this particular template
- `enterprise_name` (**Mandatory**): The name of the enterprise for the domains parent
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level

- `location_id`: The ID of the location entity (Subnet/Zone/VportTag)
- `location_type` (**Mandatory**): Type of the location entity.
- `policy_state`: State of the policy. Possible values are DRAFT, LIVE, .
- `domain_name` (**Mandatory**): The name of the domain/domain template for the `aclTemplateNames` parent
- `source_port`: Source port to be matched if protocol is UDP or TCP. Value can be either \* or single port number or a port range
- `uplink_preference`: Indicates the preferential path selection for network traffic for this ACL - default is DEFAULT when the attribute is applicable.
- `priority`: The priority of the ACL entry that determines the order of entries
- `protocol`: Protocol number that must be matched
- `associated_application_id`: The associated application ID
- `associated_application_object_id`: The associated application object ID
- `associated_application_object_type`: The associated application object type Refer to API section for supported types.
- `associated_live_entity_id`: In the draft mode, the ACL entry refers to this LiveEntity. In non-drafted mode, this is null.
- `stats_id`: The statsID that is created in the VSD and identifies this ACL Template Entry. This is auto-generated by VSD
- `stats_logging_enabled`: Is stats logging enabled for this particular template
- `ether_type` (**Mandatory**): Ether type of the packet to be matched. `etherType` can be \* or a valid hexadecimal value
- `external_id`: External object ID. Used for integration with third party systems

## 86.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nujob.NUJob</i>	jobs
<i>nustatistics.NUStatistics</i>	statistics

## 86.3 Parents

- *numirrordestination.NUMirrorDestination*
- *nume.NUMe*
- *nuingressadvfwdtemplate.NUIngressAdvFwdTemplate*



## NUINGRESSADVFDTEMPLATE

`nuingressadvfdtemplate.NUIngressAdvFwdTemplate(bambou.nurest_object.NUMetaRESTObject, ) :`

Defines the template for an Ingress Advanced Forwarding.

### 87.1 Attributes

- `name`: The name of the entity
- `last_updated_by`: ID of the user who last updated the object.
- `active`: If enabled, it means that this ACL or QOS entry is active
- `description`: A description of the entity
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `policy_state`:
- `priority`: The priority of the ACL entry that determines the order of entries
- `priority_type`:
- `associated_live_entity_id`: In the draft mode, the ACL entry refers to this LiveEntity. In non-drafted mode, this is null.
- `external_id`: External object ID. Used for integration with third party systems

### 87.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nuingressadvfdentrytemplate.NUIngressAdvFwdEntryTemplate</code>	<code>ingress_adv_fwd_entry_templates</code>
<code>nul2domain.NUL2DomainTemplate</code>	<code>jobs</code>

### 87.3 Parents

- `nudomain.NUDomain`
- `nul2domaintemplate.NUL2DomainTemplate`

- *nul2domain.NUL2Domain*
- *nudomaintemplate.NUDomainTemplate*

## NUINGRESSEXTERNALSERVICETEMPLATE

`nuingressexternalservicetemplate.NUIngressExternalServiceTemplate (bambou.nurest_object.NUMetadata)`

Defines the template for an Ingress External Service Acls.

### 88.1 Attributes

- `name`: The name of the entity
- `active`: If enabled, it means that this ACL or QOS entry is active
- `description`: A description of the entity
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `policy_state`:
- `priority`: The priority of the ACL entry that determines the order of entries
- `priority_type`:
- `associated_live_entity_id`: In the draft mode, the ACL entry refers to this LiveEntity. In non-drafted mode, this is null.
- `external_id`: External object ID. Used for integration with third party systems

### 88.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nuingressexternalservicetemplateentry.NUIngressExternalServiceTemplateEntry</code>	<code>ingress_external_service_template_entries</code>
<code>nujob.NUJob</code>	<code>jobs</code>

### 88.3 Parents

- `nudomain.NUDomain`
- `nul2domaintemplate.NUL2DomainTemplate`
- `nul2domain.NUL2Domain`

- *nudomaintemplate.NUDomainTemplate*

## NUINGRESSEXTERNALSERVICETEMPLATEENTRY

`nuingressexternalservicetemplateentry.NUIngressExternalServiceTemplateEntry(bambou.nurest_`

Defines the template of Ingress External Service ACL entries

### 89.1 Attributes

- `acl_template_name` (**Mandatory**): The name of the parent Template for this acl entry
- `icmp_code`: The ICMP Code when protocol selected is ICMP.
- `icmp_type`: The ICMP Type when protocol selected is ICMP.
- `ipv6_address_override`: Overrides the source IPv6 for Ingress and destination IPv6 for Egress, macentries will use this address as the match criteria.
- `dscp` (**Mandatory**): DSCP match condition to be set in the rule. It is either \* or from 0-63
- `name` (**Mandatory**): Name of the entity.
- `last_updated_by`: ID of the user who last updated the object.
- `action` (**Mandatory**): The action of the ACL entry DROP or FORWARD or REDIRECT. Action REDIRECT is allowed only for IngressAdvancedForwardingEntry
- `address_override`: Overrides the source IP for Ingress and destination IP for Egress, macentries will use this address as the match criteria.
- `redirect_external_service_end_point_id`: VPort tag to which traffic will be redirected to, when ACL entry match criteria succeeds
- `description`: Description of the ACL entry
- `destination_port`: The destination port to be matched if protocol is UDP or TCP. Value should be either \* or single port number or a port range
- `network_id`: The destination network entity that is referenced(subnet/zone/macro)
- `network_type`: Type of the source network.
- `mirror_destination_id`: Destination ID of the mirror destination object.
- `flow_logging_enabled`: Is flow logging enabled for this particular template
- `enterprise_name` (**Mandatory**): The name of the enterprise for the domains parent
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `location_id`: The ID of the location entity (Subnet/Zone/VportTag)

- `location_type` (**Mandatory**): Type of the location entity.
- `policy_state`: State of the policy.
- `domain_name` (**Mandatory**): The name of the domain/domain template for the `aclTemplateNames` parent
- `source_port`: Source port to be matched if protocol is UDP or TCP. Value can be either \* or single port number or a port range
- `priority`: The priority of the ACL entry that determines the order of entries
- `protocol`: Protocol number that must be matched
- `associated_application_id`: The associated application ID
- `associated_application_object_id`: The associated application object ID
- `associated_application_object_type`: The associated application object type Refer to API section for supported types.
- `associated_live_entity_id`: In the draft mode, the ACL entry refers to this LiveEntity. In non-drafted mode, this is null.
- `stats_id`: The statsID that is created in the VSD and identifies this ACL Template Entry. This is auto-generated by VSD
- `stats_logging_enabled`: Is stats logging enabled for this particular template
- `ether_type` (**Mandatory**): Ether type of the packet to be matched. `etherType` can be \* or a valid hexadecimal value
- `external_id`: External object ID. Used for integration with third party systems

## 89.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	<code>metadatas</code>
<i>nuglobalmetadata.NUGlobalMetadata</i>	<code>global_metadatas</code>
<i>nujob.NUJob</i>	<code>jobs</code>
<i>nustatistics.NUStatistics</i>	<code>statistics</code>

## 89.3 Parents

- *nuingressexternalservicetemplate.NUIngressExternalServiceTemplate*

## NUIPRESERVATION

`nuipreservation.NUIPReservation(bambou.nurest_object.NUMetaRESTObject,):`

This is the definition of a IP Bindings associated with in a Network.

### 90.1 Attributes

- `mac` (**Mandatory**): MAC Address
- `ip_address` (**Mandatory**): Static IP Address
- `last_updated_by`: ID of the user who last updated the object.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems
- `dynamic_allocation_enabled`: Binding is static or dynamic

### 90.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>

### 90.3 Parents

- `nusubnet.NUSubnet`





## NUJOB

`nujob.NUJob(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents JOB entity. The job API accepts a command and parameters and executes the job and returns the results. Jobs API are typically used for long running tasks.

### 91.1 Attributes

- `parameters`: Additional arguments required for the specific command. Differs based on types of command.
- `last_updated_by`: ID of the user who last updated the object.
- `result`: Results from the execution of the job
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `command` (**Mandatory**): Name of the command.
- `progress`: Indicates the progress of the job as a faction. eg : 0.5 means 50% done.
- `assoc_entity_type`: Entity with which this job is associated Refer to API section for supported types.
- `status`: Current status of the job. Possible values are RUNNING, FAILED, SUCCESS, .
- `external_id`: External object ID. Used for integration with third party systems

### 91.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 91.3 Parents

- `nuingressadvfwdentrytemplate.NUIngressAdvFwdEntryTemplate`
- `nupolicygrouptemplate.NUPolicyGroupTemplate`
- `nuredirectiontargettemplate.NURedirectionTargetTemplate`
- `nuredirectiontarget.NURedirectionTarget`
- `nuegressaclentrytemplate.NUEgressACLEntryTemplate`

- *nuingressexternalservicetemplateentry.NUIngressExternalServiceTemplateEntry*
- *nudomain.NUDomain*
- *nuvsd.NUVSD*
- *nuvrs.NUVRS*
- *nuhsc.NUHSC*
- *nuzfbrequest.NUZFBRequest*
- *nuvcenterhypervisor.NUVCenterHypervisor*
- *nuvcentercluster.NUVCenterCluster*
- *nugateway.NUGateway*
- *nul2domaintemplate.NUL2DomainTemplate*
- *nuvcenter.NUVCenter*
- *nuingressaclentrytemplate.NUIngressACLEntryTemplate*
- *nul2domain.NUL2Domain*
- *nuingressexternalservicetemplate.NUIngressExternalServiceTemplate*
- *nume.NUMe*
- *nunsgateway.NUNSGateway*
- *nuvsc.NUVSC*
- *nudomaintemplate.NUDomainTemplate*
- *nuegressacltemplate.NUEgressACLTemplate*
- *nuingressadvfwdtemplate.NUIngressAdvFwdTemplate*
- *nupolicygroup.NUPolicyGroup*
- *nuenterprise.NUEnterprise*
- *nuingressacltemplate.NUIngressACLTemplate*

## NUKEYSERVERMEMBER

`nukeyservermember.NUKeyServerMember(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents a KeyServer

### 92.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `pem_encoded`: PEM Encoded Certificate
- `certificate_serial_number`: Certificate serial number associated to the keyserver private key which it is currently signing with
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `fqdn`: FQDN of the keyserver member
- `issuer_dn`: Issuer DN
- `subject_dn`: Subject DN
- `public_key`: Public Key
- `external_id`: External object ID. Used for integration with third party systems

### 92.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 92.3 Parents

- `nume.NUMe`



## NUKEYSERVERMONITOR

`nukeyservermonitor.NUKeyServerMonitor(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents a Keyserver Monitor Snapshot.

### 93.1 Attributes

- `last_update_time`: The time the latest SEK or Seed was created/removed (milliseconds since epoch)
- `last_updated_by`: ID of the user who last updated the object.
- `gateway_secured_data_record_count`: Total number of Gateway Secured Data records
- `keyserver_monitor_encrypted_sek_count`: Total number of Keyserver Monitor Encrypted SEK records
- `keyserver_monitor_encrypted_seed_count`: Total number of Keyserver Monitor Encrypted Seed records
- `keyserver_monitor_sek_count`: Total number of Keyserver Monitor SEK records
- `keyserver_monitor_seed_count`: Total number of Keyserver Monitor Seed records
- `enterprise_secured_data_record_count`: Total number of Enterprise Secured Data records
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 93.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nukeyservermonitorencrypted-seed.NUKeyServerMonitorEncryptedSeed</code>	<code>key_server_monitor_encrypted_seeds</code>
<code>nukeyservermonitorseed.NUKeyServerMonitorSeed</code>	<code>key_server_monitor_seeds</code>
<code>nukeyservermonitorek.NUKeyServerMonitorSEK</code>	<code>key_server_monitor_seks</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 93.3 Parents

- `nuenterprise.NUEnterprise`



## NUKEYSERVERMONITORENCRYPTEDSEED

`nukeyservermonitorencryptedseed.NUKeyServerMonitorEncryptedSeed(bambou.nurest_object.NUMeta`

Represents a Keyserver Monitor Encrypted Seed Snapshot.

### 94.1 Attributes

- `sek_creation_time`: SEK Creation Time
- `last_updated_by`: ID of the user who last updated the object.
- `key_server_certificate_serial_number`: KeyServer Certificate Serial Number
- `enterprise_secured_data_id`: Enterprise Secured ID record this monitor represents
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `associated_key_server_monitor_sek_creation_time`: The creation time of the associated KeyServer Monitor Seed ID
- `associated_key_server_monitor_sekid`: The ID of the associated KeyServer Monitor SEK ID
- `associated_key_server_monitor_seed_creation_time`: The creation time of the associated KeyServer Monitor Seed ID
- `associated_key_server_monitor_seed_id`: The ID of the associated KeyServer Monitor Seed ID
- `external_id`: External object ID. Used for integration with third party systems

### 94.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 94.3 Parents

- `nukeyservermonitorseed.NUKeyServerMonitorSeed`
- `nukeyservermonitor.NUKeyServerMonitor`





## NUKEYSERVERMONITORSEED

**nukeyservermonitorseed.NUKeyServerMonitorSeed(bambou.nurest\_object.NUMetaRESTObject, ) :**

Represents a Keyserver Monitor Seed Snapshot.

### 95.1 Attributes

- **last\_updated\_by:** ID of the user who last updated the object.
- **seed\_traffic\_authentication\_algorithm:** Seed traffic Authentication Algorithm.
- **seed\_traffic\_encryption\_algorithm:** Seed traffic Encryption Algorithm.
- **seed\_traffic\_encryption\_key\_lifetime:** Seed Traffic Encryption Key Lifetime in Seconds
- **lifetime:** The lifetime of this entry (seconds)
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **creation\_time:** The time this entry was created (milliseconds since epoch)
- **start\_time:** The time this entry was activated (milliseconds since epoch)
- **external\_id:** External object ID. Used for integration with third party systems

### 95.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nukeyservermonitorencrypted-seed.NUKeyServerMonitorEncryptedSeed</i>	key_server_monitor_encrypted_seeds
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas

### 95.3 Parents

- *nukeyservermonitor.NUKeyServerMonitor*



## NUKEYSERVERMONITORSEK

`nukeyservermonitorsek.NUKeyServerMonitorSEK(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents a Keyserver Monitor SEK Snapshot.

### 96.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `seed_payload_authentication_algorithm`: SEK Payload Signature Algorithm Possible values are HMAC\_SHA1, HMAC\_SHA256, HMAC\_SHA512, .
- `seed_payload_encryption_algorithm`: SEK Payload Encryption Algorithm Possible values are AES\_128\_CBC, AES\_256\_CBC, TRIPLE\_DES\_CBC, .
- `lifetime`: The lifetime of this entry (seconds)
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `creation_time`: The time this entry was created (milliseconds since epoch)
- `start_time`: The time this entry was activated (milliseconds since epoch)
- `external_id`: External object ID. Used for integration with third party systems

### 96.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 96.3 Parents

- `nukeyservermonitor.NUKeyServerMonitor`



## NUKEYSERVERNOTIFICATION

`nukeyservernotification.NUKeyServerNotification(bambou.nurest_object.NUMetaRESTObject, ) :`

KeyServer Notification - Create one of these transient objects to push an event to the KeyServer

### 97.1 Attributes

- `base64_json_string`: The base 64 encoded JSON String of the message object
- `message`: The message to send
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `notification_type`: The notification type to trigger
- `external_id`: External object ID. Used for integration with third party systems

### 97.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas



## NUL2DOMAIN

`nul2domain.NUL2Domain(bambou.nurest_object.NUMetaRESTObject,):`

This is the definition of a l2 domain associated with a Enterprise.

### 98.1 Attributes

- `dhcp_managed`: decides whether L2Domain / L2Domain template DHCP is managed by VSD
- `dpi`: determines whether or not Deep packet inspection is enabled
- `ip_type`: IPv4 or IPv6
- `maintenance_mode`: `maintenanceMode` is an enum that indicates if the L2Domain is accepting VM activation requests. Possible values are DISABLED, ENABLED and ENABLED\_INHERITED Possible values are .
- `name` (**Mandatory**): Name of the L2Domain / L2Domain template,has to be unique within a Enterprise. Valid characters are alphabets, numbers, space and hyphen( - ).
- `last_updated_by`: ID of the user who last updated the object.
- `gateway`: The IP address of the gateway of this l2 domain
- `gateway_mac_address`: The MAC address of the Gateway.
- `address`: Network address of the L2Domain / L2Domain template defined.
- `template_id`: The ID of the L2 Domain template that this L2 Domain object was derived from
- `service_id`: The service ID used by the VSCs to identify this subnet
- `description`: A description field provided by the user that identifies the L2Domain / L2Domain template.
- `netmask`: Netmask of the L2Domain / L2Domain template defined
- `vn_id`: Current Network's globally unique VXLAN network identifier generated by VSD
- `encryption`: Determines whether IPSEC is enabled Possible values are ENABLED, DISABLED, .
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `policy_change_status`: None
- `route_distinguisher`: The Route Distinguisher value assigned by VSD for this subnet that is used by the BGP-EVPN protocol in VSC
- `route_target`: The Route Target value assigned by VSD for this subnet that is used by the BGP-EVPN protocol in VSC

- `uplink_preference`: Indicates the preferential path selection for network traffic in this domain - Default is Primary 1 and Secondary 2. Possible values are PRIMARY\_SECONDARY, SECONDARY\_PRIMARY, PRIMARY, SECONDARY, SYMMETRIC, .
- `associated_multicast_channel_map_id`: The ID of the Multi Cast Channel Map this L2Domain / L2Domain template template is associated with. This has to be set when `enableMultiCast` is set to ENABLED
- `associated_shared_network_resource_id`: The ID of the L2 Domain that this L2 Domain object is pointing to
- `stretched`: Indicates whether this domain is stretched, if so remote VM resolutions will be allowed
- `multicast`: Indicates multicast policy on L2Domain.
- `external_id`: External object ID. Used for integration with third party systems

## 98.2 Children

class	fetcher
<i>nutca.NUTCA</i>	tcas
<i>nuaddressrange.NUAddressRange</i>	address_ranges
<i>nuredirectiontarget.NURedirectionTarget</i>	redirection_targets
<i>nupermision.NUPermission</i>	permissions
<i>numetadata.NUMetadata</i>	metadatas
<i>nuegressaclentrytemplate.NUEgressACLEntryTemplate</i>	egress_acl_entry_templates
<i>nuegressacltemplate.NUEgressACLTemplate</i>	egress_acl_templates
<i>nudhcpoption.NUDHCPOption</i>	dhcp_options
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuvm.NUVM</i>	vms
<i>nuvminterface.NUVMInterface</i>	vm_interfaces
<i>nuingressaclentrytemplate.NUIngressACLEntryTemplate</i>	ingress_acl_entry_templates
<i>nuingressacltemplate.NUIngressACLTemplate</i>	ingress_acl_templates
<i>nuingressadvfwdtemplate.NUIngressAdvFwdTemplate</i>	ingress_adv_fwd_templates
<i>nuingressexternalservicetemplate.NUIngressExternalServiceTemplate</i>	ingress_external_service_templates
<i>nujob.NUJob</i>	jobs
<i>nupolicygroup.NUPolicyGroup</i>	policy_groups
<i>nucontainer.NUContainer</i>	containers
<i>nucontainerinterface.NUContainerInterface</i>	container_interfaces
<i>nuqos.NUQOS</i>	qoss
<i>nuhostinterface.NUHostInterface</i>	host_interfaces
<i>nuuplinkrd.NUUplinkRD</i>	uplink_rds
<i>nuvpnconnection.NUVPNConnection</i>	vpn_connections
<i>nuvport.NUVPort</i>	vports
<i>nuapplicationperformancemanagementbinding.NUApplicationperformancemanagementbinding</i>	applicationperformancema
<i>nubridgeinterface.NUBridgeInterface</i>	bridge_interfaces
<i>nugroup.NUGroup</i>	groups
<i>nustaticroute.NUStaticRoute</i>	static_routes
<i>nustatistics.NUStatistics</i>	statistics
<i>nustatisticspolicy.NUStatisticsPolicy</i>	statistics_policies
<i>nueventlog.NUEventLog</i>	event_logs



## 98.3 Parents

- *nul2domaintemplate.NUL2DomainTemplate*
- *nume.NUMe*
- *nuenterprise.NUEnterprise*



## NUL2DOMAINTEMPLATE

`nul2domaintemplate.NUL2DomainTemplate(bambou.nurest_object.NUMetaRESTObject,):`

L2 Domain in VSD as derived by templates. This object describes the L2 Domain template.

### 99.1 Attributes

- `dhcp_managed`: decides whether L2Domain / L2Domain template DHCP is managed by VSD
- `dpi`: determines whether or not Deep packet inspection is enabled
- `ip_type`: IPv4 or IPv6
- `name` (**Mandatory**): Name of the L2Domain / L2Domain template, has to be unique within a Enterprise. Valid characters are alphabets, numbers, space and hyphen( - ).
- `last_updated_by`: ID of the user who last updated the object.
- `gateway`: The IP address of the gateway of this l2 domain
- `gateway_mac_address`: gateway MAC address for a managed l2 domain
- `address`: Network address of the L2Domain / L2Domain template defined.
- `description`: A description field provided by the user that identifies the L2Domain / L2Domain template.
- `netmask`: Netmask of the L2Domain / L2Domain template defined
- `encryption`: Determines whether IPSEC is enabled Possible values are ENABLED, DISABLED, .
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `policy_change_status`: None
- `use_global_mac`: Enable this flag to use system configured globalMACAddress as the gateway mac address for managed l2 domains
- `associated_multicast_channel_map_id`: The ID of the Multi Cast Channel Map this L2Domain / L2Domain template template is associated with. This has to be set when enableMultiCast is set to ENABLED
- `multicast`: Indicates multicast policy on L2Domain template.
- `external_id`: External object ID. Used for integration with third party systems

## 99.2 Children

class	fetcher
<i>nul2domain.NUL2Domain</i>	l2_domains
<i>nuaddressrange.NUAddressRange</i>	address_ranges
<i>nuredirectiontargettemplate.NURedirectionTargetTemplate</i>	redirection_target_templates
<i>nupermission.NUPermission</i>	permissions
<i>numetadata.NUMetadata</i>	metadatas
<i>nuegressacltemplate.NUEgressACLTemplate</i>	egress_acl_templates
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>ningressacltemplate.NUIngressACLTemplate</i>	ingress_acl_templates
<i>ningressadvfwdtemplate.NUIngressAdvFwdTemplate</i>	ingress_adv_fwd_templates
<i>ningressexternalservicetemplate.NUIngressExternalServiceTemplate</i>	ingress_external_service_templates
<i>nujob.NUJob</i>	jobs
<i>nupolicygrouptemplate.NUPolicyGroupTemplate</i>	policy_group_templates
<i>nuqos.NUQOS</i>	qoss
<i>nugroup.NUGroup</i>	groups
<i>nueventlog.NUEventLog</i>	event_logs

## 99.3 Parents

- *nuenterprise.NUEnterprise*

## NUL7APPLICATIONSIGNATURE

`nul7applicationsignature.NUL7applicationsignature(bambou.nurest_object.NUMetaRESTObject,):`

Layer 7 ApplicationType , these are auto created as part of VSD bringup

### 100.1 Attributes

- `guid`: GUID of the Application
- `name` (**Mandatory**): name of the L7 App
- `category`: Category of this application
- `description`: description for L7 App
- `dictionary_version`: Version of the L7 Application Type

### 100.2 Children

class	fetcher
<i>nuapplication.NUApplication</i>	applications

### 100.3 Parents

- *nuenterprise.NUEnterprise*



## NULDAPCONFIGURATION

```
nuldapconfiguration.NULDAPConfiguration(bambou.nurest_object.NUMetaRESTObject,):
```

Configuration of LDAP parameters associated with an enterprise. This will enable authentication through an external LDAP server for this enterprise.

### 101.1 Attributes

- `ssl_enabled`: Enable SSL for communication with the LDAP server
- `password`: This attribute is a mandatory field for LDAP authorization. Password that will be used to verify the integrity of groups and users in LDAP server for the enterprise.
- `last_updated_by`: ID of the user who last updated the object.
- `accept_all_certificates`: Accept all certificates from the LDAP server
- `certificate`: The certificate to authenticate with the LDAP server
- `server` (**Mandatory**): The LDAP server IP or FQDN
- `enabled`: To enable LDAP authentication for an enterprise, set this attribute to true. If enabled is set to false, `authorizationEnabled` attribute is ignored and LDAP is not used for authentication as well as authorization. The relationship between `enabled` and `authorizationEnabled` attributes is as follows, `enabled = true, authorizationEnabled = false`, LDAP is used only for Authentication `enabled = true, authorizationEnabled = true`, LDAP is used for both authentication and authorization. `enabled = false, authorizationEnabled = true`, LDAP is not used. `enabled = false, authorizationEnabled = false`, LDAP is not used.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `port` (**Mandatory**): Port to be used for the LDAP server
- `group_dn` (**Mandatory**): This attribute is a mandatory field for LDAP authorization. When LDAP is used for authorization for an enterprise, the group DN will be used to get the list of VSD specific groups in LDAP server for the enterprise. For example, `OU=VSDGroups,DC=company,DC=com`
- `user_dn_template` (**Mandatory**): The DN template to be used for authentication. The template needs to have a string `_USERID_` in it. This will be replaced by the `userId` of the user who makes the REST API call. For example, template `UID=_USERID_,OU=company,DC=com` will be converted to `UID=admin,OU=company,DC=com` and this will be used as DN for LDAP authentication.
- `authorization_enabled`: To enable LDAP authorization for an enterprise, both `authorizationEnabled` and `enabled` attributes must be set to true. If `enabled` attribute is not set, this attribute is ignored. The relationship between `enabled` and `authorizationEnabled` attributes is as follows, `enabled = true, authorizationEnabled = false`, LDAP is used only for Authentication. `enabled = true, authorizationEnabled = true`, LDAP is used for both

authentication and authorization. enabled = false, authorizationEnabled = true, LDAP is not used. enabled = false, authorizationEnabled = false, LDAP is not used.

- `authorizing_user_dn` (**Mandatory**): This attribute is a mandatory field for LDAP authorization. When LDAP is used for authorization for an enterprise, the user DN that will be used to verify the integrity of groups and users in LDAP server for the enterprise. For example, CN=groupAdmin,OU=VSD\_USERS,OU=Personal,OU=Domain Users,DC=company,DC=com
- `external_id`: External object ID. Used for integration with third party systems

## 101.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

## 101.3 Parents

- `numenterprise.NUEnterprise`



## NULICENSE

`nulicense.NULicense(bambou.nurest_object.NUMetaRESTObject,):`

Enables retrieval/modification and creation of license files. Most of the attributes are retrieved from the encrypted license. The create API simply provides the encrypted license that is in base64 format.

### 102.1 Attributes

- `major_release`: Major software release associated with this license
- `last_updated_by`: ID of the user who last updated the object.
- `additional_supported_versions`: Indicates additional versions supported by the license.
- `phone`: Phone number of the owner associated with the license file
- `license` (**Mandatory**): Base 64 value of the license
- `license_encryption`: License encryption
- `license_entities`: Indicates non enforceable entities associated with the license.
- `license_id`: Unique identifier of the license file
- `license_type`: None
- `minor_release`: Minor software release for which this license has been issued
- `zip`: Zipcode of the owner associated with the license file
- `city`: City of the owner associated with the license file
- `allowed_cpes_count`: Maximum number of CPEs enabled with this license. A value of -1 indicates an unlimited number of CPEs
- `allowed_nics_count`: Maximum number of NICs allowed. A value of -1 indicates unlimited number of NICs
- `allowed_vms_count`: Maximum number of VMs enabled with this license. A value of -1 indicates an unlimited number of VMs
- `allowed_vrsgs_count`: Maximum number of VRSGs enabled with this license. A value of -1 indicates an unlimited number of VRSGs
- `allowed_vrss_count`: Maximum number of VRSSs enabled with this license. A value of -1 indicates an unlimited number of VRSSs
- `email`: Email of the owner associated with the license file
- `encryption_mode`: Indicates if the system is associated with a license that allows encryption or not

- `unique_license_identifier`: Indicates combined string of first 16 and last 16 characters of the license string to be shown in the API
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `company`: Company of the owner associated with the license file
- `country`: Country of the owner associated with the license file
- `product_version`: Version of the product that this license applies to
- `provider`: Provider of the license file
- `is_cluster_license`: Indicates if the license is associated with standalone or cluster setup of VSD
- `user_name`: The name of the user associated with the license
- `state`: State of the owner associated with the license file
- `street`: Address of the owner associated with the license file
- `customer_key`: Customer key associated with the license
- `expiration_date`: Expiration date of this license
- `external_id`: External object ID. Used for integration with third party systems

## 102.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nueventlog.NUEventLog</i>	event_logs

## 102.3 Parents

- *nume.NUme*

## NULICENSESTATUS

`nulicensestatus.NULicenseStatus(bambou.nurest_object.NUMetaRESTObject,):`

None

### 103.1 Attributes

- `accumulate_licenses_enabled`: Whether the various VRS license flavours be merged in one pool
- `total_licensed_gateways_count`: Indicates total VRS+VRSG+VRSB licenses licensed in the system
- `total_licensed_nics_count`: Indicates total NIC count for all the licenses in the system
- `total_licensed_nsgs_count`: Indicates total NSG count for all the licenses in the system
- `total_licensed_used_nics_count`: Indicates total used NIC count for all the licenses in the system
- `total_licensed_used_nsgs_count`: Indicates total used NSG count for all the licenses in the system
- `total_licensed_used_vms_count`: Indicates total used VM count for all the licenses in the system
- `total_licensed_used_vrsgs_count`: Indicates total used VRSG count for all the licenses in the system
- `total_licensed_used_vrss_count`: Indicates total used VRS count for all the licenses in the system
- `total_licensed_vms_count`: Indicates total VM count for all the licenses in the system
- `total_licensed_vrsgs_count`: Indicates total VRSG count for all the licenses in the system
- `total_licensed_vrss_count`: Indicates total VRS count for all the licenses in the system
- `total_used_gateways_count`: Indicates total VRS+VRSG+VRSB licenses used in the system

### 103.2 Parents

- `nume.NUMe`



## NULINK

**nulink.NULink(bambou.nurest\_object.NUMetaRESTObject,):**

This object represents the link between a source and destination domain in service chaining

### 104.1 Attributes

- **last\_updated\_by:** ID of the user who last updated the object.
- **acceptance\_criteria:** A route filtering criteria enum. Defaults to ALL.
- **read\_only:** This is set to true if a link has been created in the opposite direction
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **associated\_destination\_id:** This is the ID of the domain receiving the routes from the source. This can only be set for links of type OVERLAY\_ADDRESS\_TRANSLATION.
- **associated\_destination\_name:** None
- **associated\_destination\_type:** Type of the entity type for the source
- **associated\_source\_id:** The ID of the domain receiving the routes from another domain
- **associated\_source\_name:** None
- **associated\_source\_type:** This is the source object type for the associatedSourceID
- **external\_id:** External object ID. Used for integration with third party systems
- **type:** This is used to distinguish between different type of links: hub and spoke, ip address, VNS border router links.

### 104.2 Children

class	fetcher
<i>nudemarcationservice.NUDemarcationService</i>	demarcation_services
<i>numetadata.NUMetadata</i>	metadatas
<i>nunexthopaddress.NUNextHopAddress</i>	next_hop_address
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuoverlayaddresspool.NUOverlayAddressPool</i>	overlay_address_pools

## 104.3 Parents

- *nudomain.NUDomain*

## **NULOCATION**

**nulocation.NULocation(bambou.nurest\_object.NUMetaRESTObject,):**

Gateway location details.

### **105.1 Attributes**

- **last\_updated\_by**: ID of the user who last updated the object.
- **latitude**: Latitude in decimal format.
- **address**: Formatted address including property number, street name, suite or office number, ...
- **ignore\_geocode**: Request BSS to perform a geocode on the address - If no value passed, requestGeocode will be set to true
- **time\_zone\_id**: Time zone in which the Gateway is located. This can be in the form of a UTC/GMT offset, continent/city location, or country/region. The available time zones can be found in /usr/share/zoneinfo on a Linux machine or retrieved with `TimeZone.getAvailableIDs()` in Java. Refer to the IANA (Internet Assigned Numbers Authority) for a list of time zones. URL : <http://www.iana.org/time-zones> Default value is UTC (translating to Etc/Zulu)
- **entity\_scope**: Specify if scope of entity is Data center or Enterprise level
- **locality**: Locality/City/County
- **longitude**: Longitude in decimal format.
- **country**: Country
- **state**: State/Province/Region
- **external\_id**: External object ID. Used for integration with third party systems

### **105.2 Children**

<b>class</b>	<b>fetcher</b>
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas

## 105.3 Parents

- *nunsgateway.NUNSGateway*



`nume.NUME (bambou.nurest_object.NUMetaRESTObject, ) :`

Object that identifies the user functions

## 106.1 Attributes

- `password` (**Mandatory**): User password stored as a hash (SHA-1 encrypted)
- `last_name` (**Mandatory**): Last name of the user
- `last_updated_by`: ID of the user who last updated the object.
- `first_name` (**Mandatory**): First name of the user
- `disabled`: Status of the user account; true=disabled, false=not disabled; default value = false
- `elastic_search_ui_address`: elastic search UI address
- `flow_collection_enabled`: Enables flow statistics collection. It is needed for the VSS feature, and requires a valid VSS license. This option requires “statisticsEnabled”.
- `email` (**Mandatory**): Email address of the user
- `enterprise_id`: Identifier of the enterprise.
- `enterprise_name`: Name of the enterprise.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `mobile_number`: Mobile Number of the user
- `role`: Role of the user.
- `user_name` (**Mandatory**): Unique Username of the user. Valid characters are alphabets, numbers and hyphen(-).
- `statistics_enabled`: This flag is used to indicate if statistics is enabled in the system. CSProot is expected to activate this through the enable statistics script.
- `avatar_data`: URL to the avatar data associated with the enterprise. If the avatarType is URL then value of avatarData should an URL of the image. If the avatarType BASE64 then avatarData should be BASE64 encoded value of the image
- `avatar_type`: Avatar type.
- `external_id`: External object ID. Used for integration with third party systems

## 106.2 Children

class	fetcher
<i>nul2domain.NUL2Domain</i>	l2_domains
<i>nuvcentereamconfig.NUVCenterEAMConfig</i>	vcenter_eam_configs
<i>nuratelimiter.NURateLimiter</i>	rate_limiters
<i>nugateway.NUGateway</i>	gateways
<i>nugatewaytemplate.NUGatewayTemplate</i>	gateway_templates
<i>nupatmapper.NUPATMapper</i>	pat_mappers
<i>nupatnatpool.NUPATNATPool</i>	patnat_pools
<i>nutca.NUTCA</i>	tcas
<i>nuvcenter.NUVCenter</i>	vcenters
<i>nuvcenterhypervisor.NUVCenterHypervisor</i>	vcenter_hypervisors
<i>nuredirectiontarget.NURedirectionTarget</i>	redirection_targets
<i>nuredundancygroup.NURedundancyGroup</i>	redundancy_groups
<i>nuperformancemonitor.NUPerformanceMonitor</i>	performance_monitors
<i>nucertificate.NUCertificate</i>	certificates
<i>numetadata.NUMetadata</i>	metadatas
<i>numetadatatag.NUMetadataTag</i>	metadata_tags
<i>nunetworklayout.NUNetworkLayout</i>	network_layouts
<i>nukeyservermember.NUKeyServerMember</i>	key_server_members
<i>nuzfbautoassignment.NUZFBAutoAssignment</i>	zfb_auto_assignments
<i>nuzfbrequest.NUZFBRequest</i>	zfb_requests
<i>nubgpneighbor.NUBGPNeighbor</i>	bgp_neighbors
<i>nubgpprofile.NUBGPProfile</i>	bgp_profiles
<i>negressaclentrytemplate.NUEgressACLEntryTemplate</i>	egress_acl_entry_templates
<i>negressacltemplate.NUEgressACLTemplate</i>	egress_acl_templates
<i>nudomainfipacltemplate.NUDomainFIPACLTemplate</i>	domain_fip_acl_templates
<i>nufloatingipacltemplate.NUFloatingIPACLTemplate</i>	floating_ipacl_templates
<i>negressqospolicy.NUEgressQOSPolicy</i>	egress_qos_policies
<i>nusharednetworkresource.NUSharedNetworkResource</i>	shared_network_resources
<i>nulicense.NULicense</i>	licenses
<i>nulicensestatus.NULicenseStatus</i>	license_status
<i>numirrordestination.NUMirrorDestination</i>	mirror_destinations
<i>nusiteinfo.NUSiteInfo</i>	site_infos
<i>nufloatingip.NUFloatingIp</i>	floating_ips
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuvm.NUVM</i>	vms
<i>nuvminterface.NUVMInterface</i>	vm_interfaces
<i>nucloudmgmtsystem.NUCloudMgmtSystem</i>	cloud_mgmt_systems
<i>nuunderlay.NUUnderlay</i>	underlays
<i>nuinfrastructuregatewayprofile.NUInfrastructureGatewayProfile</i>	infrastructure_gateway_profiles
<i>nuinfrastructurevscprofile.NUInfrastructureVscProfile</i>	infrastructure_vsc_profiles
<i>ningressaclentrytemplate.NUIngressACLEntryTemplate</i>	ingress_acl_entry_templates
<i>ningressacltemplate.NUIngressACLTemplate</i>	ingress_acl_templates
<i>ningressadvfwdentrytemplate.NUIngressAdvFwdEntryTemplate</i>	ingress_adv_fwd_entry_templates
<i>nuenterprise.NUEnterprise</i>	enterprises
<i>nuenterpriseprofile.NUEnterpriseProfile</i>	enterprise_profiles
<i>nujob.NUJob</i>	jobs
<i>nupolicygroup.NUPolicyGroup</i>	policy_groups

Continued on next page

Table 106.1 – continued from previous page

<i>nudomain.NUDomain</i>	domains
<i>nuzone.NUZone</i>	zones
<i>nucontainer.NUContainer</i>	containers
<i>nucontainerinterface.NUContainerInterface</i>	container_interfaces
<i>nuhostinterface.NUHostInterface</i>	host_interfaces
<i>nuroutingpolicy.NURoutingPolicy</i>	routing_policies
<i>nuuplinkrd.NUUplinkRD</i>	uplink_rds
<i>nuapplicationsevice.NUApplicationService</i>	application_services
<i>nuvcentervrconfig.NUVCenterVRSCfg</i>	vcenter_vrs_configs
<i>nuuser.NUUser</i>	users
<i>nunsgateway.NUNSGateway</i>	ns_gateways
<i>nunsgatewaytemplate.NUNSGatewayTemplate</i>	ns_gateway_templates
<i>nunsggroup.NUNSGGroup</i>	nsg_groups
<i>nunsredundantgatewaygroup.NUNSRedundantGatewayGroup</i>	ns_redundant_gateway_groups
<i>nuvsp.NUVSP</i>	vsps
<i>nustaticroute.NUStaticRoute</i>	static_routes
<i>nustatscollectorinfo.NUStatsCollectorInfo</i>	stats_collector_infos
<i>nusubnet.NUSubnet</i>	subnets
<i>nuducgroup.NUDUCGroup</i>	duc_groups
<i>numulticastchannelmap.NUMultiCastChannelMap</i>	multi_cast_channel_maps
<i>nuautodiscoveredgateway.NUAutoDiscoveredGateway</i>	auto_discovered_gateways
<i>nuexternalappservice.NUExternalAppService</i>	external_app_services
<i>nuexternalsevice.NUExternalService</i>	external_services
<i>nusystemconfig.NUSystemConfig</i>	system_configs



## NUMETADATA

`numetadata.NUMetadata (bambou.nurest_object.NUMetaRESTObject, ) :`

Metadata associated to a entity.

### 107.1 Attributes

- `name`: name of the Metadata.
- `description`: Description of the Metadata.
- `metadata_tag_ids`: metadata tag IDs associated with this metadata you can filter metadata based on this attribute for example X-Nuage-Filter: '2d6fb627-603b-421c-b63a-eb0a6d712761' IN metadataTagIDs
- `network_notification_disabled`: specifies metadata changes need to be notified to controller,by default it is notified
- `blob` (**Mandatory**): Metadata that describes about the entity attached to it.
- `global_metadata`: specifies metadata is global or local
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 107.2 Children

class	fetcher
<code>numetadatatag.NUMetadataTag</code>	<code>metadata_tags</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>

### 107.3 Parents

- `nucontainerinterface.NUContainerInterface`
- `nugroupkeyencryptionprofile.NUGroupKeyEncryptionProfile`
- `nuqos.NUQOS`
- `nuingressadvfwdentrytemplate.NUIngressAdvFwdEntryTemplate`
- `nubgppeer.NUBGPpeer`

- *nusharednetworkresource.NUSharedNetworkResource*
- *nuvirtualip.NUVirtualIP*
- *nudscpforwardingclasstable.NUDSCPForwardingClassTable*
- *numulticastchannelmap.NUMultiCastChannelMap*
- *nuredundancygroup.NURedundancyGroup*
- *nutca.NUTCA*
- *nugroup.NUGroup*
- *nuvsgredundantport.NUVsgRedundantPort*
- *nuzone.NUZone*
- *nuikegatewayprofile.NUIKEGatewayProfile*
- *nuikesubnet.NUIKESubnet*
- *nuinfrastructuregatewayprofile.NUInfrastructureGatewayProfile*
- *nupolicygrouptemplate.NUPolicyGroupTemplate*
- *nuflowsecuritypolicy.NUFlowSecurityPolicy*
- *nuvcentereamconfig.NUVCenterEAMConfig*
- *nulocation.NULocation*
- *nuenterprisesecurity.NUEnterpriseSecurity*
- *nuvcentervrsconfig.NUVCenterVRSCfg*
- *nuenterprisenetwork.NUEnterpriseNetwork*
- *nuinfrastructurevscprofile.NUInfrastructureVscProfile*
- *nupermision.NUPermission*
- *nuipreservation.NUIPReservation*
- *nuredirectiontargettemplate.NURedirectionTargetTemplate*
- *nusubnettemplate.NUSubnetTemplate*
- *numetadatetag.NUMetadataTag*
- *nufloatingipacltemplate.NUFloatingIPACLTemplate*
- *nuikegatewayconnection.NUIKEGatewayConnection*
- *nuredirectiontarget.NURedirectionTarget*
- *nuegressaclentrytemplate.NUEgressACLEntryTemplate*
- *nuingressexternalservicetemplateentry.NUIngressExternalServiceTemplateEntry*
- *nuavatar.NUAvatar*
- *nubootstrapactivation.NUBootstrapActivation*
- *nucloudmgmtsystem.NUCloudMgmtSystem*
- *nuvsp.NUVSP*
- *nudomain.NUDomain*
- *nuredundantport.NURedundantPort*

- *nudscpf forwardingclassmapping.NUDSCPForwardingClassMapping*
- *nunetworklayout.NUNetworkLayout*
- *nuwanservice.NUWANService*
- *nuvsd.NUVSD*
- *nuvport.NUVPort*
- *nuflowforwardingpolicy.NUFlowForwardingPolicy*
- *nuport.NUPort*
- *nustatisticspolicy.NUStatisticsPolicy*
- *nusubnet.NUSubnet*
- *nucontainer.NUContainer*
- *nuratelimiter.NURateLimiter*
- *nukeyservermonitorencryptedseed.NUKeyServerMonitorEncryptedSeed*
- *nuzonetemplate.NUZoneTemplate*
- *nukeyservermonitorseed.NUKeyServerMonitorSeed*
- *nugatewaytemplate.NUGatewayTemplate*
- *nuvrs.NUVRS*
- *nuhsc.NUHSC*
- *nuvlan.NUVLAN*
- *nuldapconfiguration.NULDAPConfiguration*
- *nuvsdcomponent.NUVSDComponent*
- *nuzfbrequest.NUZFBRequest*
- *nuenterprisepermission.NUEnterprisePermission*
- *nustaticroute.NUStaticRoute*
- *nujob.NUJob*
- *nuvminterface.NUVMInterface*
- *nugatewaysecureddata.NUGatewaySecuredData*
- *nuvcenterhypervisor.NUVCenterHypervisor*
- *nukeyservermonitor.NUKeyServerMonitor*
- *nueventlog.NUEventLog*
- *nulicense.NULicense*
- *nuenterpriseprofile.NUEnterpriseProfile*
- *nubridgeinterface.NUBridgeInterface*
- *nuvcentercluster.NUVCenterCluster*
- *numulticastrange.NUMultiCastRange*
- *nunetworkmacrogroup.NUNetworkMacroGroup*
- *nuinfrastructureaccessprofile.NUInfrastructureaccessprofile*

- *nuautodiscoveredgateway.NUAutoDiscoveredGateway*
- *numulticastlist.NUMultiCastList*
- *nunexthop.NUNextHop*
- *nukeyservernotification.NUKeyServerNotification*
- *numirrordestination.NUMirrorDestination*
- *nutier.NUTier*
- *nudomainfipacltemplate.NUDomainFIPAcITemplate*
- *nuaddressmap.NUAddressMap*
- *nugateway.NUGateway*
- *numultinicyport.NUMultiNICVPort*
- *nustatistics.NUStatistics*
- *nunsporttemplate.NUNSPortTemplate*
- *nucertificate.NUCertificate*
- *nuvcenterdatacenter.NUVCenterDataCenter*
- *nul2domaintemplate.NUL2DomainTemplate*
- *nuapplicationservice.NUApplicationService*
- *nuikegateway.NUIKEGateway*
- *nustatscollectorinfo.NUStatsCollectorInfo*
- *nuvcenter.NUVCenter*
- *nubulkstatistics.NUBulkStatistics*
- *nuingressaclentrytemplate.NUIngressACLEntryTemplate*
- *nuroutingpolicy.NURoutingPolicy*
- *nul2domain.NUL2Domain*
- *nuikegatewayconfig.NUIKEGatewayConfig*
- *nuhostinterface.NUHostInterface*
- *nuenterprisesecureddata.NUEnterpriseSecuredData*
- *nuikecertificate.NUIKECertificate*
- *nuingressexternalservicetemplate.NUIngressExternalServiceTemplate*
- *nuporttemplate.NUPortTemplate*
- *nume.NUMe*
- *nuendpoint.NUEndPoint*
- *nudhcption.NUDHCPOption*
- *nukeyservermember.NUKeyServerMember*
- *nunsgateway.NUNSGateway*
- *nunsgatewaytemplate.NUNSGatewayTemplate*
- *nuvsc.NUVSC*



- *nuuplinkrd.NUUplinkRD*
- *nuvm.NUVM*
- *nunsport.NUNSPort*
- *nuvrsaddressrange.NUVRSAAddressRange*
- *nubgppprofile.NUBGPPProfile*
- *nuegressqospolicy.NUEgressQOSPolicy*
- *nupublicnetworkmacro.NUPublicNetworkMacro*
- *nudomainfipacltemplateentry.NUDomainFIPACLTemplateEntry*
- *nuaddressrange.NUAddressRange*
- *nudomaintemplate.NUDomainTemplate*
- *nusiteinfo.NUSiteInfo*
- *nuvmresync.NUVMResync*
- *nupolicydecision.NUPolicyDecision*
- *nufloatingip.NUFloatingIp*
- *nuegressacltemplate.NUEgressACLTemplate*
- *numonitoringport.NUMonitoringPort*
- *nuvpnconnection.NUVPNConnection*
- *nunsredundantgatewaygroup.NUNSRedundantGatewayGroup*
- *nuexternalservice.NUExternalService*
- *nukeyservermonitorsek.NUKeyServerMonitorSEK*
- *nuingressadvfwdtemplate.NUIngressAdvFwdTemplate*
- *nuvportmirror.NUVPortMirror*
- *nupatnatpool.NUPATNATPool*
- *nubgpneighbor.NUBGPNeighbor*
- *nucontainerresync.NUContainerResync*
- *nuallalarm.NUAllAlarm*
- *nuikepsk.NUIKEPSK*
- *nusystemconfig.NUSystemConfig*
- *nuikeencryptionprofile.NUIKEEncryptionprofile*
- *nufloatingipacltemplateentry.NUFloatingIPACLTemplateEntry*
- *nuinfrastructureconfig.NUInfrastructureConfig*
- *nuuser.NUUser*
- *nunatmapentry.NUNATMapEntry*
- *nupolicygroup.NUPolicyGroup*
- *nuexternalappservice.NUExternalAppService*
- *nualarm.NUAlarm*

- *nubootstrap.NUBootstrap*
- *nuflow.NUFlow*
- *nuvlantemplate.NUVLANTemplate*
- *nugatewaysecurity.NUGatewaySecurity*
- *nuglobalmetadata.NUGlobalMetadata*
- *nuenterprise.NUEnterprise*
- *nulink.NULink*
- *nuingressacltemplate.NUIngressACLTemplate*

## NUMETADATATAG

`numetadatatag.NUMMetadataTag(bambou.nurest_object.NUMetaRESTObject, ) :`

Metadata tag associated to a metadata.

### 108.1 Attributes

- **name (Mandatory):** name of the Metadata tag.
- **last\_updated\_by:** ID of the user who last updated the object.
- **description:** Description of the Metadata tag.
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **associated\_external\_service\_id:** ID of the entity to which the Metadata tag is associated to
- **auto\_created:** set to true if it is the default metadata tag created as part of external service creation
- **external\_id:** External object ID. Used for integration with third party systems

### 108.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nueventlog.NUEventLog</i>	event_logs

### 108.3 Parents

- *numetadata.NUMetadata*
- *nume.NUMe*
- *nuexternalservice.NUExternalService*
- *nuglobalmetadata.NUGlobalMetadata*
- *nuenterprise.NUEnterprise*



## NUMIRRORDESTINATION

`numirrordestination.NUMirrorDestination(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents a mirror destination.

### 109.1 Attributes

- `name`: Name of the mirror destination. Valid characters are alphabets, numbers, space and hyphen ( - ).
- `last_updated_by`: ID of the user who last updated the object.
- `service_id`: Service ID of the mirror destination.
- `destination_ip`: IP address of the destination server where you want your traffic to be mirrored.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 109.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuegressaclentrytemplate.NUEgressACLEntryTemplate</code>	<code>egress_acl_entry_templates</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nuingressaclentrytemplate.NUIngressACLEntryTemplate</code>	<code>ingress_acl_entry_templates</code>
<code>nuingressadvfwdentrytemplate.NUIngressAdvFwdEntryTemplate</code>	<code>ingress_adv_fwd_entry_templates</code>
<code>nuvportmirror.NUVPortMirror</code>	<code>vport_mirrors</code>

### 109.3 Parents

- `nume.NUMe`



## NUMONITORINGPORT

`numonitoringport.NUMonitoringPort (bambou.nurest_object.NUMetaRESTObject, ) :`

Encapsulates the port information for system monitoring entity.

### 110.1 Attributes

- `name`: Name for the port.
- `last_state_change`: Last port state change timestamp.
- `access`: Flag to indicate that it is a access port or network port.
- `description`: Optional port description.
- `resiliency_state`:
- `resilient`: Flag to indicate if an ACCESS port is resilient or not.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `uplink`: Flag to indicate that is an uplink or downlink port.
- `state`: The current state of the port.
- `external_id`: External object ID. Used for integration with third party systems

### 110.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 110.3 Parents

- `nuvrs.NUVRS`
- `nuhsc.NUHSC`
- `nuvsc.NUVSC`





## NUMONITORSCOPE

`numonitorscope.NUMonitorscope(bambou.nurest_object.NUMetaRESTObject, ) :`

This class tries to define the scope of probe (the NSGs between which the probe needs) to run.

### 111.1 Attributes

- **name (Mandatory):** Name for the given scope
- **read\_only:** Determines whether this entity is read only. Read only objects cannot be modified or deleted.
- **destination\_nsgs:** List of destination NSGs to which the probe needs to run
- **allow\_all\_destination\_nsgs:** When set true, allows all destination NSGs
- **allow\_all\_source\_nsgs:** When set true, allows all Source NSGs
- **source\_nsgs:** List of source NSGs from which the probe needs to be started.

### 111.2 Parents

- *nuapplication.NUApplication*
- *nunsgateway.NUNSGateway*
- *nunetworkperformancemeasurement.UNNetworkPerformanceMeasurement*



## NUMULTICASTCHANNELMAP

```
nummulticastchannelmap.NUMMultiCastChannelMap(bambou.nurest_object.NUMetaRESTObject,):
```

This is the definition of a MultiCast Channel Map.

### 112.1 Attributes

- **name (Mandatory):** Name of the current entity
- **last\_updated\_by:** ID of the user who last updated the object.
- **description:** A description field provided by the user that identifies the MultiCast Channel Map
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **external\_id:** External object ID. Used for integration with third party systems

### 112.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>numulticastrange.NUMultiCastRange</i>	multi_cast_ranges
<i>nueventlog.NUEventLog</i>	event_logs

### 112.3 Parents

- *nucontainerinterface.NUContainerInterface*
- *nuvminterface.NUVMInterface*
- *numulticastlist.NUMultiCastList*
- *nuhostinterface.NUHostInterface*
- *nume.NUMe*



## NUMULTICASTLIST

```
nummulticastlist.NUMMultiCastList(bambou.nurest_object.NUMetaRESTObject,):
```

This is the definition of a MultiCast Channel List.

### 113.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `mcast_type`: Type of multicast list.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 113.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>numulticastchannelmap.NUMultiCastChannelMap</i>	multi_cast_channel_maps

### 113.3 Parents

- *nuenterpriseprofile.NUEnterpriseProfile*
- *nuenterprise.NUEnterprise*



## NUMULTICASTRANGE

`nummulticasterange.NUMMultiCastRange(bambou.nurest_object.NUMetaRESTObject,):`

This is the definition of a MultiCast Range associated with a MultiCast Channel Map.

### 114.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `max_address` (**Mandatory**): Highest address in the MultiCast range
- `min_address` (**Mandatory**): Lowest address in the MultiCast range
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 114.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nueventlog.NUEventLog</i>	event_logs

### 114.3 Parents

- *numulticastchannelmap.NUMultiCastChannelMap*





## NUMULTINICVPORT

`numultinicvport.NUMMultiNICVPort(bambou.nurest_object.NUMetaRESTObject,)` :

Encapsulates the Multi NIC VPort information for system monitoring entity.

### 115.1 Attributes

- `name`: Name for the Multi NIC VPort.
- `last_updated_by`: ID of the user who last updated the object.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 115.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nuvport.NUVPort</code>	<code>vports</code>

### 115.3 Parents

- `nuvrs.NUVRS`



## NUNATMAPENTRY

`nunatmapentry.NUNATMapEntry(bambou.nurest_object.NUMetaRESTObject, ) :`

Defines a MAP between the private ip and public ip.

### 116.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `private_ip` (**Mandatory**): Private IP address of the interface
- `associated_patnat_pool_id`: Indicates which PATNATPool this entry belongs to
- `public_ip` (**Mandatory**): Public IP address of the interface
- `external_id`: External object ID. Used for integration with third party systems
- `type` (**Mandatory**): The type of address mapping this instance is.

### 116.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 116.3 Parents

- `nupatnatpool.NUPATNATPool`



## NUNETWORKLAYOUT

`nunetworklayout.NUNetworkLayout (bambou.nurest_object.NUMetaRESTObject, ) :`

This API defines the AS number that should be used in the data center as well as the IP address of the route reflector.

### 117.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `service_type`: Identifies whether L3 or L2 services are supported.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `route_reflector_ip`: The IP address of the route reflector that can be used by the VSCs
- `autonomous_system_num`: The AS number associated with this data center
- `external_id`: External object ID. Used for integration with third party systems

### 117.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 117.3 Parents

- `nume.NUMe`



## NUNETWORKMACROGROUP

`nunetworkmacrogroup.NUNetworkMacroGroup(bambou.nurest_object.NUMetaRESTObject, ) :`

Administrators of an enterprise can define macros that are set of IP addresses that identify enterprise networks. These macros can be used in the ACL definitions by network designers and other users to identify access restrictions towards specific enterprise networks.

### 118.1 Attributes

- **name (Mandatory):** Name of the macro group
- **last\_updated\_by:** ID of the user who last updated the object.
- **description:** Description of the macro group
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **external\_id:** External object ID. Used for integration with third party systems

### 118.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nunetwork.NUNetwork</i>	enterprise_networks

### 118.3 Parents

- *nunetwork.NUNetwork*
- *nunetwork.NUNetwork*





## **NUNETWORKPERFORMANCEBINDING**

`nunetworkperformancebinding.NUNetworkPerformanceBinding(bambou.nurest_object.NUMetaRESTObj`  
None

### **119.1 Attributes**

- `read_only`: Determines whether this entity is read only. Read only objects cannot be modified or deleted.
- `priority`: Priority of the associated Network Performance Measurement
- `associated_network_measurement_id`: Associated Network Performance Measurement UD

### **119.2 Parents**

- `nunetworkperformancemeasurement.NUNetworkPerformanceMeasurement`



## NUNETWORKPERFORMANCEMEASUREMENT

`nunetworkperformancemeasurement.NUNetworkPerformanceMeasurement` (`bambou.nurest_object.NUMeta`

Network Performance Measurement is a container for group of applications and monitor scopes

### 120.1 Attributes

- **name (Mandatory):** name of the network performance measurement
- **read\_only:** Determines whether this entity is read only. Read only objects cannot be modified or deleted.
- **description:** description of network performance measurement
- **associated\_performance\_monitor\_id:** associated Performance Monitor ID

### 120.2 Children

class	fetcher
<code>nunetworkperformancebinding.NUNetworkPerformanceBinding</code>	<code>network_performance_bindings</code>
<code>numonitorscope.NUMonitorscope</code>	<code>monitorscopes</code>

### 120.3 Parents

- `nuenterprise.NUEnterprise`



## NUNEXTHOP

`nunexthop.NUNextHop(bambou.nurest_object.NUMetaRESTObject, ) :`

This represents a /32 IPv4 address as the next-hop. In the future can be a /128 IPv6 address.

### 121.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `route_distinguisher` (**Mandatory**): The next-hop's route distinguisher. A unique 8 byte long. If not provided one will be generated.
- `ip`: This is the /32 or /128 next-hop IP address. Currently we support only IPv4 address family.
- `external_id`: External object ID. Used for integration with third party systems

### 121.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>



## NUNEXTHOPADDRESS

```
nunexthopaddress.NUNextHopAddress(bambou.nurest_object.NUMetaRESTObject,):
```

None

### 122.1 Attributes

- `address`: IP address for the next hop.
- `route_distinguisher`: A unique route distinguisher associated with the nexthop. If one is not provided the system generated one automatically.
- `type`: Next hop type: IP only supported for service chaining

### 122.2 Parents

- `nulink.NULink`





## NUNSGATEWAY

`nunsgateway.NUNSGateway(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents a Network Service Gateway.

### 123.1 Attributes

- `mac_address`: MAC Address of the NSG
- `nat_traversal_enabled`: This attribute is deprecated in version 4.0.
- `sku`: The part number of the NSG
- `tpm_status`: TPM Status of the NSG based on the information received by the device during bootstrapping or upgrade.
- `cpu_type`: The NSG Processor Type
- `nsg_version`: The NSG Version
- `ssh_service`: Indicates if SSH Service is enabled/disabled on a NSG. The value configured for this attribute is used only when `instanceSSHOverride` is allowed on the associated Gateway Template.
- `uuid`: The Redhat UUID of the NSG
- `name` (**Mandatory**): Name of the Gateway
- `family`: The NSG Type
- `last_configuration_reload_timestamp`: Time stamp of the last known configuration update of the NSG. This timestamp gets updated when a bootstrap is successful or when a configuration reload request triggered by VSD is successful.
- `last_updated_by`: ID of the user who last updated the object.
- `datapath_id`: Identifier of the Gateway, based on the `systemId`
- `redundancy_group_id`: The Redundancy Gateway Group associated with this Gateway Instance. This is a read only attribute
- `template_id` (**Mandatory**): The ID of the template that this Gateway was created from. This should be set when instantiating a Gateway
- `pending`: Indicates that this gateway is pending state or state. When in pending state it cannot be modified from REST.
- `serial_number`: The NSG's serial number

- `derived_ssh_service_state`: Indicates the SSH Service state on a NSG. This value is derived based on the SSHService configuration on the NSG and the associated Gateway Template.
- `permitted_action`: The permitted action to USE/EXTEND this Gateway.
- `personality`: Personality of the Gateway - NSG, cannot be changed after creation.
- `description`: A description of the Gateway
- `libraries`: Transient representation of the same property on NSGInfo.
- `inherited_ssh_service_state`: Indicates the SSH Service state which is configured on the associated template instance.
- `enterprise_id`: The enterprise associated with this Gateway. This is a read only attribute
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `location_id`: The NSGateway's Location. NOTE: this is a read only property, it can only be set through the location object
- `configuration_reload_state`: None
- `configuration_status`: None
- `bootstrap_id`: The bootstrap details associated with this NSGateway. NOTE: this is a read only property, it can only be set during creation of an NSG
- `bootstrap_status`: The bootstrap status of this NSGateway. NOTE: this is a read only property
- `associated_gateway_security_id`: Readonly Id of the associated gateway security object
- `associated_gateway_security_profile_id`: Readonly Id of the associated gateway security profile object
- `associated_nsg_info_id`: Readonly Id of the associated nsg info object
- `auto_disc_gateway_id`: The Auto Discovered Gateway associated with this Gateway Instance
- `external_id`: External object ID. Used for integration with third party systems
- `system_id`: Identifier of the Gateway, cannot be modified after creation

## 123.2 Children

<b>class</b>	<b>fetcher</b>
<i>nugatewaysecurity.NUGatewaySecurity</i>	gateway_securities
<i>nupatnatpool.NUPATNATPool</i>	patnat_pools
<i>nupermision.NUPermission</i>	permissions
<i>numetadata.NUMetadata</i>	metadatas
<i>nualarm.NUAlarm</i>	alarms
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuinfrastructureconfig.NUInfrastructureConfig</i>	infrastructure_configs
<i>nueenterprisepermission.NUEnterprisePermission</i>	enterprise_permissions
<i>nujob.NUJob</i>	jobs
<i>nulocation.NULocation</i>	locations
<i>numonitorscope.NUMonitorscope</i>	monitorscopes
<i>nubootstrap.NUBootstrap</i>	bootstraps
<i>nubootstrapactivation.NUBootstrapActivation</i>	bootstrap_activations
<i>nunsginfo.NUNSGInfo</i>	nsg_infos
<i>nunsport.NUNSPort</i>	ns_ports
<i>nusubnet.NUSubnet</i>	subnets
<i>nueventlog.NUEventLog</i>	event_logs

## 123.3 Parents

- *nuducgroup.NUDUCGroup*
- *nume.NUMe*
- *nunsggroup.NUNSGGroup*
- *nunsredundantgatewaygroup.NUNSRedundantGatewayGroup*
- *nueenterprise.NUEnterprise*



## NUNSGATEWAYTEMPLATE

`nunsgatewaytemplate.NUNSGatewayTemplate(bambou.nurest_object.NUMetaRESTObject,)` :

Represents a Network Service Gateway Template.

### 124.1 Attributes

- `ssh_service`: Enable/Disable SSH Service on all the Gateway instances which inherit from this template.
- `name` (**Mandatory**): Name of the Gateway
- `last_updated_by`: ID of the user who last updated the object.
- `personality`: Personality of the Gateway - NSG, NSGBR, cannot be changed after creation.
- `description`: A description of the Gateway
- `infrastructure_access_profile_id`: The ID of the infrastructure access profile associated to this Gateway Template.
- `infrastructure_profile_id` (**Mandatory**): The ID of the infrastructure gateway profile this instance of a Gateway is associated with.
- `instance_ssh_override`: Indicates if this template instance allows the gateway instance(s) which inherit from it to independently enable/disable SSH service.
- `enterprise_id`: The enterprise associated with this Gateway. This is a read only attribute
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 124.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nunsporttemplate.NUNSPortTemplate</code>	<code>ns_port_templates</code>

### 124.3 Parents

- `nume.NUMe`

- *numenterprise.NUEnterprise*

## NUNSGGROUP

```
nunsggroup.NUNSGGroup(bambou.nurest_object.NUMetaRESTObject,):
```

None

### 125.1 Attributes

- `name`: Name of the NSG Group
- `description`: Description of the NSG Group
- `associated_nsgs`: List of NSGs that belong to NSG Group

### 125.2 Children

class	fetcher
<i>nunsgateway.NUNSGateway</i>	<code>ns_gateways</code>
<i>nuducgroupbinding.NUDUCGroupBinding</i>	<code>duc_group_bindings</code>

### 125.3 Parents

- *nume.NUMe*
- *nuenterprise.NUEnterprise*





## NUNSGINFO

`nunsginfo.NUNSGInfo(bambou.nurest_object.NUMetaRESTObject, ) :`

Device information coming from the NSG

### 126.1 Attributes

- `mac_address`: MAC Address of the NSG
- `sku`: The part number of the NSG
- `tpm_status`: TPM status
- `cpu_type`: The NSG Processor Type
- `nsg_version`: The NSG Version
- `uuid`: The Redhat UUID of the NSG
- `family`: The NSG Type
- `serial_number`: The NSG's serial number
- `libraries`: Tracks RPM package installed for some libraries installed on the NSG.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `associated_ns_gateway_id`: Associated NS Gateway ID
- `external_id`: External object ID. Used for integration with third party systems

### 126.2 Parents

- `nunsgateway.NUNSGateway`



## NUNSPORT

`nunsport.NUNSPORT (bambou.nurest_object.NUMetaRESTObject, ) :`

Represents a Port of a particular NS Gateway object.

### 127.1 Attributes

- `nat_traversal`: Enum value that states the type of NAT Traversal the NSG instance will use to talk to other NSGs and the Internet.
- `vlan_range`: VLAN Range of the Port. Format must conform to a-b,c,d-f where a,b,c,d,f are integers between 0 and 4095.
- `name` (**Mandatory**): Name of the Port
- `last_updated_by`: ID of the user who last updated the object.
- `template_id`: The ID of the template that this Port was created from
- `permitted_action`: The permitted action to USE/EXTEND this Gateway.
- `description`: A description of the Port
- `physical_name` (**Mandatory**): Identifier of the Port
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `port_type` (**Mandatory**): Type of the Port.
- `speed`: Port Speed in Mb/s : Supported Ethernet speeds are 10 (10Base-T), 100 (Fast-ethernet 100Base-TX), 1000 (Gigabit Ethernet 1000Base-T), 10 000 (10 Gigabit Ethernet 10GBase-X), and Auto-Negotiate.
- `use_user_mnemonic`: determines whether to use user mnemonic of the Port
- `user_mnemonic` (**Mandatory**): user mnemonic of the Port
- `associated_egress_qos_policy_id`: ID of the Egress QOS Policy associated with this Vlan.
- `associated_redundant_port_id`: ID of the redundant port to which the Port is associated to.
- `status`: Status of the port.
- `mtu`: Port MTU (Maximum Transmission Unit) : The size in octets of the largest protocol data unit (PDU) that the layer can pass on. The default value is normally 1500 octets for Ethernet v2 and can go up to 9198 for Jumbo Frames.
- `external_id`: External object ID. Used for integration with third party systems

## 127.2 Children

<b>class</b>	<b>fetcher</b>
<i>nupermision.NUPermission</i>	permissions
<i>numetadata.NUMetadata</i>	metadatas
<i>nuvlan.NUVLAN</i>	vlangs
<i>nualarm.NUAlarm</i>	alarms
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuenterpriasepermission.NUEnterprisePermission</i>	enterprise_permissions
<i>nustatistics.NUStatistics</i>	statistics
<i>nustatisticspolicy.NUStatisticsPolicy</i>	statistics_policies
<i>nueventlog.NUEventLog</i>	event_logs

## 127.3 Parents

- *nuredundantport.NURedundantPort*
- *nuautodiscoveredgateway.NUAutoDiscoveredGateway*
- *nunsgateway.NUNSGateway*

## NUNSPORTTEMPLATE

`nunsporttemplate.NUNSPORTTemplate(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents Port Template object under a given gateway template object.

### 128.1 Attributes

- `vlan_range`: VLAN Range of the Port. Format must conform to a-b,c,d-f where a,b,c,d,f are integers between 0 and 4095.
- `name` (**Mandatory**): Name of the Port
- `last_updated_by`: ID of the user who last updated the object.
- `description`: A description of the Port
- `physical_name` (**Mandatory**): Identifier of the Port
- `infrastructure_profile_id`: The ID of the infrastructure profile this instance is associated with.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `port_type` (**Mandatory**): Type of the Port.
- `speed`: Port Speed in Mb/s : Supported Ethernet speeds are 10 (10Base-T), 100 (Fast-ethernet 100Base-TX), 1000 (Gigabit Ethernet 1000Base-T), 10 000 (10 Gigabit Ethernet 10GBase-X), and Auto-Negotiate
- `associated_egress_qos_policy_id`: ID of the Egress QOS Policy associated with this Vlan.
- `mtu`: Port MTU (Maximum Transmission Unit) : The size in octets of the largest protocol data unit hat the layer can pass on.
- `external_id`: External object ID. Used for integration with third party systems

### 128.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuvlantemplate.NUVLANTemplate</code>	<code>vlan_templates</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

## 128.3 Parents

- *nunsgatewaytemplate.NUNSGatewayTemplate*

## NUNSREDUNDANTGATEWAYGROUP

`nunsredundantgatewaygroup.NUNSRedundantGatewayGroup(bambou.nurest_object.NUMetaRESTObject, )`

Represents Redundant Group formed by two VNS Gateways.

### 129.1 Attributes

- `name` (**Mandatory**): Name of the Redundancy Group
- `last_updated_by`: ID of the user who last updated the object.
- `gateway_peer1_autodiscovered_gateway_id`: The Auto Discovered Gateway configuration owner in this Redundant Group.
- `gateway_peer1_id`: The gateway configuration owner in this Redundant Group. when Redundant Group is deleted this gateway will receive vport associations
- `gateway_peer1_name`: The gateway configuration owner name in this Redundant Group
- `gateway_peer2_autodiscovered_gateway_id`: The Auto Discovered Gateway peer in this Redundant Group
- `gateway_peer2_id`: The gateway peer in this Redundant Group. when Redundant Group is deleted this gateway will not receive vport associations
- `gateway_peer2_name`: The gateway peer name in this Redundant Group
- `heartbeat_interval`: Heartbeat interval in milliseconds to declare the neighbor dead.
- `heartbeat_vlanid`: Heartbeat VLAN used for BFD.
- `redundancy_port_ids`: Collections resilient port ids associated with this redundant group.
- `redundant_gateway_status`: The status of Redundant Group.
- `permitted_action`: The permitted action to USE/EXTEND this Gateway.
- `personality`: Derived personality of the Redundancy Group.
- `description`: Description of the Redundancy Group
- `enterprise_id`: The enterprise associated with this Redundant Group. This is a read only attribute
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `consecutive_failures_count`: Consecutive failure count.
- `external_id`: External object ID. Used for integration with third party systems

## 129.2 Children

<b>class</b>	<b>fetcher</b>
<i>numetadata.NUMetadata</i>	metadatas
<i>nualarm.NUAlarm</i>	alarms
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nunsgateway.NUNSGateway</i>	ns_gateways
<i>nuredundantport.NURedundantPort</i>	redundant_ports
<i>nueventlog.NUEventLog</i>	event_logs

## 129.3 Parents

- *nume.NUMe*
- *numenterprise.NUEnterprise*



## NUOVERLAYADDRESSPOOL

`nuoverlayaddresspool.NUOverlayAddressPool (bambou.nurest_object.NUMetaRESTObject, ) :`

The address pool the public IP of the PAT/NAT entries belong too.

### 130.1 Attributes

- `name`: Name for the PAT NAT pool
- `description`: addresspool description
- `end_address_range`: The end address for the pool range.
- `associated_domain_id`: The ID of the associated l3-domain.
- `start_address_range`: Start address for the pool range

### 130.2 Children

class	fetcher
<code>nuoverlaypatnatentry.NUOverlayPATNATEntry</code>	<code>overlay_patnat_entries</code>

### 130.3 Parents

- `nulink.NULink`



## NUOVERLAYPATNATENTRY

```
nuoverlaypatnatentry.NUOverlayPATNATEntry(bambou.nurest_object.NUMetaRESTObject,):  
None
```

### 131.1 Attributes

- `nat_enabled`: This flag will determine whether the entry is NAT or PAT.
- `private_ip`: Private IP address for the interface
- `associated_domain_id`: The ID of the associated l3-domain.
- `associated_link_id`: The ID of the associated domain-link.
- `public_ip`: Public IP address of the interface

### 131.2 Parents

- *nuoverlayaddresspool.NUOverlayAddressPool*



## NUPATIPENTRY

`nupatipentry.NUPATIPEntry(bambou.nurest_object.NUMetaRESTObject,):`  
missing documentation.

### 132.1 Attributes

- `pat_centralized`: This flag will determine whether we can expect anchor point or not.
- `ip_address`: Its own IPAddress.
- `ip_type`: IPv4 or IPv6 (only IPv4 supported in R1.0)
- `last_updated_by`: ID of the user who last updated the object.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `associated_domain_id`: The ID of the associated l3-domain.
- `external_id`: External object ID. Used for integration with third party systems
- `hypervisor_id`: The ID of the PatMapper entity to which this domain is associated to.

### 132.2 Parents

- *nusharednetworkresource.NUSharedNetworkResource*



## NUPATMAPPER

`nupatmapper.NUPATMapper(bambou.nurest_object.NUMetaRESTObject,):`  
missing documentation.

### 133.1 Attributes

- `name` (**Mandatory**): None
- `last_updated_by`: ID of the user who last updated the object.
- `description`: None
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 133.2 Children

class	fetcher
<i>nusharednetworkresource.NUSharedNetworkResource</i>	<code>shared_network_resources</code>

### 133.3 Parents

- *nume.NUMe*





## NUPATNATPOOL

`nupatnatpool.NUPATNATPool(bambou.nurest_object.NUMetaRESTObject,):`

Represents a PAT NAT Pool object.

### 134.1 Attributes

- **name (Mandatory):** Name of the PATNATPool
- **last\_updated\_by:** ID of the user who last updated the object.
- **address\_range:** Default PAT IP Address, must belong to the pool above
- **default\_patip:** Default PAT IP Address, must belong to the pool above
- **permitted\_action:** The permitted action to USE/EXTEND this Gateway.
- **description:** A description of the PATNATPool
- **end\_address\_range:** Ending IP Address for the pool of available addresses for use
- **end\_source\_address:** Ending Source IP Address for the pool. (Dynamic Source NAT)
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **associated\_gateway\_id:** UUID of the NSG instance this Pool is associated with. This attribute may be auto-populated when the pool is assigned to a Network VLAN instance.
- **associated\_gateway\_type:** None
- **associated\_subnet\_id:** ID of the Subnet for which the information will be used to populate Source Address Range (Dynamic Source NAT).
- **associated\_vlan\_id:** ID of the network port VLAN on which the pool is associated.
- **start\_address\_range:** Starting IP Address for the pool of available addresses for use
- **start\_source\_address:** Starting Source IP Address for the pool. (Dynamic Source NAT)
- **external\_id:** External object ID. Used for integration with third party systems
- **dynamic\_source\_enabled:** Set to True if the address translation pool at the address translation pool definition level

## 134.2 Children

<b>class</b>	<b>fetcher</b>
<i>nunatmapentry.NUNATMapEntry</i>	nat_map_entries
<i>nuaddressmap.NUAddressMap</i>	address_maps
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuenterprisepermission.NUEnterprisePermission</i>	enterprise_permissions
<i>nustatistics.NUStatistics</i>	statistics
<i>nustatisticspolicy.NUStatisticsPolicy</i>	statistics_policies
<i>nubulkstatistics.NUBulkStatistics</i>	bulk_statistics

## 134.3 Parents

- *nuvlan.NUVLAN*
- *nugateway.NUGateway*
- *nume.NUMe*
- *nunsgateway.NUNSGateway*
- *nuenterprise.NUEnterprise*

## NUPERFORMANCEMONITOR

```
nupersormancemonitor.NUPersormanceMonitor (bambou.nurest_object.NUMetaRESTObject, ) :  
None
```

### 135.1 Attributes

- **name (Mandatory):** Name of the application group probe
- **payload\_size (Mandatory):** Payload size
- **read\_only:** Determines whether this entity is read only. Read only objects cannot be modified or deleted.
- **service\_class:** Class of service to be used. Service classes in order of priority are A, B, C, D, E, F, G, and H.
- **description:** Description of application group probe
- **interval (Mandatory):** interval in seconds
- **number\_of\_packets (Mandatory):** number of packets

### 135.2 Children

class	fetcher
<i>nuapplicationperformancemanagement.NUApplicationperformancemanagement</i>	applicationperformancemanagements

### 135.3 Parents

- *nume.NUMe*
- *nuenterprise.NUEnterprise*



## NUPERMISSION

`nupermision.NUPermission(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents Permitted action on an entity for a group.

### 136.1 Attributes

- `name`: Name of the Permission
- `last_updated_by`: ID of the user who last updated the object.
- `permitted_action` (**Mandatory**): The permitted action to USE/EXTEND/READ/INstantiate an entity.
- `permitted_entity_description`: Description for the permittedEntity
- `permitted_entity_id` (**Mandatory**): The entity ID for which this permission action is associated against.
- `permitted_entity_name`: Name of the entity for which we have given permission.
- `permitted_entity_type`: Type of the entity for which we have given permission.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 136.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nueventlog.NUEventLog</i>	event_logs

### 136.3 Parents

- *nuredundancygroup.NURedundancyGroup*
- *nuvsgredundantport.NUVsgRedundantPort*
- *nuzone.NUZone*
- *nudomain.NUDomain*

- *nuwanservice.NUWANService*
- *nuport.NUPort*
- *nuvlan.NUVLAN*
- *nugateway.NUGateway*
- *nul2domaintemplate.NUL2DomainTemplate*
- *nul2domain.NUL2Domain*
- *nunsgateway.NUNSGateway*
- *nunsport.NUNSPort*
- *nudomaintemplate.NUDomainTemplate*

## NUPOLICYDECISION

`nupolicydecision.NUPolicyDecision(bambou.nurest_object.NUMetaRESTObject,)` :

This object is a read only object that provides the policy decisions for a particular VM interface.

### 137.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `egress_acls`: List of actual Egress ACLs that will be applied on the interface of this VM
- `egress_qos`: Egress QoS primitive that was selected
- `fip_acls`: List of actual Egress ACLs that will be applied on the interface of this VM
- `ingress_acls`: List of actual Ingress ACLs that will be applied on the interface of this VM
- `ingress_adv_fwd`: List of actual Ingress Redirect ACLs that will be applied on the interface of this VM
- `ingress_external_service_acls`: List of actual Ingress External Service ACLs that will be applied on the interface of this VM
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `qos`: QoS primitive that was selected based on inheritance policies
- `stats`: Stats primitive that was selected based on inheritance policies
- `external_id`: External object ID. Used for integration with third party systems

### 137.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nuqos.NUQOS</code>	<code>qoss</code>

### 137.3 Parents

- `nucontainerinterface.NUContainerInterface`
- `nuvminterface.NUVMInterface`

- *nubridgeinterface.NUBridgeInterface*
- *nuhostinterface.NUHostInterface*



## NUPOLICYGROUP

`nupolicygroup.NUPolicyGroup(bambou.nurest_object.NUMetaRESTObject, ) :`

PolicyGroup is group of polycys on which a user can policies like ACL, QoS, etc.

### 138.1 Attributes

- `evpn_community_tag`: Assigned by VSD. An extended community or other similar BGP attribute to the specific EVPN / IP-VPN NLRI where the VM or network macro is being advertised.
- `name` (**Mandatory**): Name of the policy group
- `last_updated_by`: ID of the user who last updated the object.
- `template_id`: Determines which template ID this policy group belongs to.
- `description`: Describes this policy group
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `policy_group_id`: PG ID for the subnet. This is unique per domain and will be in the range 1-4095
- `external`: Indicates whether this PG is internal to VSP or not.
- `external_id`: External object ID. Used for integration with third party systems
- `type` (**Mandatory**): Type of policy group.

### 138.2 Children

<b>class</b>	<b>fetcher</b>
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nujob.NUJob</code>	<code>jobs</code>
<code>nuvport.NUVPort</code>	<code>vports</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>

### 138.3 Parents

- `nucontainerinterface.NUContainerInterface`
- `nudomain.NUDomain`

- *nuvport.NUVPort*
- *nuvminterface.NUVMInterface*
- *nubridgeinterface.NUBridgeInterface*
- *nul2domain.NUL2Domain*
- *nuhostinterface.NUHostInterface*
- *nume.NUMe*

## NUPOLICYGROUPTEMPLATE

`nupolicygrouptemplate.NUPolicyGroupTemplate(bambou.nurest_object.NUMetaRESTObject, ) :`

PolicyGroupTemplate represents the template of a policy group object. PolicyGroup is group of vports on which a user can policies like ACL, QoS, etc.

### 139.1 Attributes

- `evpn_community_tag`: An extended community or other similar BGP attribute to the specific EVPN / IP-VPN NLRI where the VM or network macro is being advertised.
- `name` (**Mandatory**): Name of the policy group
- `last_updated_by`: ID of the user who last updated the object.
- `description`: Describes this policy group
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external`: Indicates whether this PG is internal to VSP or not.
- `external_id`: External object ID. Used for integration with third party systems
- `type` (**Mandatory**): Type of policy group.

### 139.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nujob.NUJob</code>	<code>jobs</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>

### 139.3 Parents

- `nul2domaintemplate.NUL2DomainTemplate`
- `nudomaintemplate.NUDomainTemplate`



## NUPORT

`nuport.NUPort (bambou.nurest_object.NUMetaRESTObject, ) :`

Represents Port under a particular gateway object or redundant group object.

### 140.1 Attributes

- `vlan_range`: VLAN Range of the Port. Format must conform to a-b,c,d-f where a,b,c,d,f are integers between 0 and 4095.
- `name` (**Mandatory**): Name of the Port
- `last_updated_by`: ID of the user who last updated the object.
- `template_id`: The ID of the template that this Port was created from
- `permitted_action`: The permitted action to USE/EXTEND this Gateway.
- `description`: A description of the Port
- `physical_name` (**Mandatory**): Identifier of the Port
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `port_type` (**Mandatory**): Type of the Port.
- `is_resilient`: States if this port instance is resilient (redundant). An example would be a Multi-Chassis LAG port.
- `use_user_mnemonic`: determines whether to use user mnemonic of the Port
- `user_mnemonic`: user mnemonic of the Port
- `associated_egress_qos_policy_id`: ID of the Egress QOS Policy associated with this Vlan.
- `associated_redundant_port_id`: ID of the redundant port to which this Port instance may be associated to.
- `status`: Status of the port.
- `external_id`: External object ID. Used for integration with third party systems

## 140.2 Children

<b>class</b>	<b>fetcher</b>
<i>nupermision.NUPermission</i>	permissions
<i>numetadata.NUMetadata</i>	metadatas
<i>nuvlan.NUVLAN</i>	vlangs
<i>nualarm.NUAlarm</i>	alarms
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuenterpriasepermission.NUEnterprisePermission</i>	enterprise_permissions
<i>nueventlog.NUEventLog</i>	event_logs

## 140.3 Parents

- *nuredundancygroup.NURedundancyGroup*
- *nuautodiscoveredgateway.NUAutoDiscoveredGateway*
- *nugateway.NUGateway*

## **NUPORTMAPPING**

`nuportmapping.NUPortMapping(bambou.nurest_object.NUMetaRESTObject, ) :`  
missing documentation.

### **141.1 Attributes**

- `last_updated_by`: ID of the user who last updated the object.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `private_port`: The vport's port-number.
- `public_port`: The public port used for Static PAT.
- `external_id`: External object ID. Used for integration with third party systems

### **141.2 Parents**

- *nuvport.NUVPort*





## NUPORTTEMPLATE

`nuporttemplate.NUPortTemplate(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents a Port Template object under a given gateway template object.

### 142.1 Attributes

- `vlan_range`: VLAN Range of the Port. Format must conform to a-b,c,d-f where a,b,c,d,f are integers between 0 and 4095.
- `name` (**Mandatory**): Name of the Port
- `last_updated_by`: ID of the user who last updated the object.
- `description`: A description of the Port
- `physical_name` (**Mandatory**): Identifier of the Port
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `port_type` (**Mandatory**): Type of the Port - NETWORK, ACCESS Possible values are ACCESS, NETWORK, .
- `associated_egress_qos_policy_id`: ID of the Egress QOS Policy associated with this Vlan.
- `external_id`: External object ID. Used for integration with third party systems

### 142.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuvlantemplate.NUVLANTemplate</code>	<code>vlan_templates</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 142.3 Parents

- `nugatewaytemplate.NUGatewayTemplate`



## NUPUBLICNETWORKMACRO

`nupublicnetworkmacro.NUPublicNetworkMacro(bambou.nurest_object.NUMetaRESTObject, ) :`

Similar to the enterprise macros, the public network macro allows an administrator of an enterprise to define range of subnets that can be used by users in the ACL definition.

### 143.1 Attributes

- `ip_type`: IPv4 or IPv6(only IPv4 is supported in R1.0) Possible values are IPV4, IPV6, .
- `ipv6_address`: IPv6 address of the subnet defined. In case of zone, this is an optional field for and allows users to allocate an IP address range to a zone. The VSD will auto-assign IP addresses to subnets from this range if a specific IP address is not defined for the subnet
- `name` (**Mandatory**): Name of the current entity(Zone or zone template or subnet etc..) Valid characters are alphabets, numbers, space and hyphen( - ).
- `last_updated_by`: ID of the user who last updated the object.
- `address` (**Mandatory**): IP address of the subnet defined. In case of zone, this is an optional field for and allows users to allocate an IP address range to a zone. The VSD will auto-assign IP addresses to subnets from this range if a specific IP address is not defined for the subnet
- `netmask` (**Mandatory**): Netmask of the subnet defined
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 143.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>

### 143.3 Parents

- `numenterprise.NUEnterprise`



## NUQOS

`nuqos.NUQOS (bambou.nurest_object.NUMetaRESTObject, ) :`

The object manipulates the QoS parameters attached to a domain, zone, or subnet.

### 144.1 Attributes

- `fip_committed_burst_size`: Committed burst size setting in kilo-bytes (kilo-octets) for FIP Shaper.
- `fip_committed_information_rate`: Committed information rate setting in Mb/s for FIP Shaper.
- `fip_peak_burst_size`: Peak burst size setting in kilo-bytes (kilo-octets) for FIP rate limiting.
- `fip_peak_information_rate`: Peak rate setting for FIP rate limiting in Mb/s;
- `fip_rate_limiting_active`: Flag the indicates whether FIP rate limiting is enabled or disabled
- `bum_committed_burst_size`: Committed burst size setting in kilo-bytes (kilo-octets) for BUM Shaper.
- `bum_committed_information_rate`: Committed information rate setting in Mb/s for BUM Shaper.
- `bum_peak_burst_size`: Peak burst size setting in kilo-bytes (kilo-octets) for Broadcast/Multicast rate limiting (BUM).
- `bum_peak_information_rate`: Peak rate setting in Mb/s for Broadcast/Multicast rate limiting
- `bum_rate_limiting_active`: Flag the indicates whether Broadcast/Multicast rate limiting is enabled or disabled
- `name` (**Mandatory**): A unique name of the QoS object
- `last_updated_by`: ID of the user who last updated the object.
- `rate_limiting_active`: Identifies if rate limiting must be implemented
- `active`: If enabled, it means that this ACL or QOS entry is active
- `peak`: Peak Information Rate : Peak bandwidth that is allowed from each VM in Mb/s; only whole values allowed and 'INFINITY' if rate limiting is disabled.
- `service_class`: Class of service to be used. Service classes in order of priority are A(1), B(2), C(3), D(4), E(5), F(6), G(7) and H(8) Possible values are NONE, A, B, C, D, E, F, G, H, .
- `description`: A description of the QoS object
- `rewrite_forwarding_class`: Specifies if the rewrite flag is set for the QoS policy / template
- `egress_fip_committed_burst_size`: Committed burst size setting in kilo-bytes (kilo-octets) for FIP Shaper on the Egress.

- `egress_fip_committed_information_rate`: Committed information rate setting in Mb/s for FIP Shaper on the egress side.
- `egress_fip_peak_burst_size`: Peak burst size setting in kilo-bytes (kilo-octets) for Egress FIP rate limiting.
- `egress_fip_peak_information_rate`: Peak rate setting for FIP rate limiting on egress in Mb/s
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `committed_burst_size`: Committed Burst Size : Burst size associated with the rate limiter in kilo-bytes (kilo-octets); only whole values are supported.
- `committed_information_rate`: Committed Information Rate : Committed bandwidth that is allowed from each VM in Mb/s; only whole values supported.
- `trusted_forwarding_class`: Specifies if the trusted flag is set for the QoS policy / template
- `assoc_qos_id`: ID of object associated with this QoS object
- `associated_dscp_forwarding_class_table_id`: ID of the DSCP->Forwarding Class used by this Qos Policy
- `associated_dscp_forwarding_class_table_name`: Name of the DSCP->Forwarding Class used by this Qos Policy
- `burst`: Peak Burst Size : The maximum burst size associated with the rate limiter in kilo-bytes (kilo-octets); only whole values allowed and 'INFINITY' if rate limiting is disabled.
- `external_id`: External object ID. Used for integration with third party systems

## 144.2 Children

<b>class</b>	<b>fetcher</b>
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuvm.NUVM</i>	vms
<i>nucontainer.NUContainer</i>	containers
<i>nueventlog.NUEventLog</i>	event_logs

## 144.3 Parents

- *nuzone.NUZone*
- *nusubnettemplate.NUSubnetTemplate*
- *nudomain.NUDomain*
- *nuvport.NUVPort*
- *nusubnet.NUSubnet*
- *nuzonetemplate.NUZoneTemplate*
- *nubridgeinterface.NUBridgeInterface*
- *nul2domaintemplate.NUL2DomainTemplate*
- *nul2domain.NUL2Domain*

- *nuhostinterface.NUHostInterface*
- *nudomaintemplate.NUDomainTemplate*
- *nupolicydecision.NUPolicyDecision*





## NURATELIMITER

`nuratelimiter.NURateLimiter(bambou.nurest_object.NUMetaRESTObject, ) :`

Rate Limiter object that contains peak, burst and cir. It can be associated with Egress QOS policy objects.

### 145.1 Attributes

- **name (Mandatory):** A unique name of the Rate Limiter object
- **last\_updated\_by:** ID of the user who last updated the object.
- **peak\_burst\_size:** Peak Burst Size : The maximum burst size associated with the rate limiter in kilo-bits; only whole values are supported.
- **peak\_information\_rate:** Peak Information Rate : Peak bandwidth allowed in Mb/s; only whole values supported.
- **description:** A description of the Rate Limiter object
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **committed\_information\_rate:** Committed Information Rate : Committed bandwidth that is allowed in Mb/s; only whole values supported.
- **external\_id:** External object ID. Used for integration with third party systems

### 145.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 145.3 Parents

- `nume.NUMe`
- `nuenterprise.NUEnterprise`



## NUREDIRECTIONTARGET

`nuredirectiontarget.NURedirectionTarget (bambou.nurest_object.NUMetaRESTObject, ) :`

A group/collection of vports that belong to the same domain.

### 146.1 Attributes

- `esi`: ESI id, globally unique
- `name` (**Mandatory**): Name of this redirection target
- `last_updated_by`: ID of the user who last updated the object.
- `redundancy_enabled` (**Mandatory**): Allow/Disallow redundant appliances and VIP
- `template_id`: Template to which this redirection target belongs to
- `description`: Description of this redirection target
- `virtual_network_id`: Auto Generated by VSD. Each vPortTag with redundancy=enable and Endpoint-Type != none will have a globally unique ESI & VNID generated by VSD
- `end_point_type` (**Mandatory**): EndpointType defines the type of header rewrite and forwarding performed by VRS when the endpoint is used as a PBR destination. NONE type is deprecated. Possible values are NONE, L3, VIRTUAL\_WIRE, .
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `trigger_type`: Trigger type, THIS IS READ ONLY. Possible values are NONE, GARP, .
- `external_id`: External object ID. Used for integration with third party systems

### 146.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuvirtualip.NUVirtualIP</code>	<code>virtual_ips</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nujob.NUJob</code>	<code>jobs</code>
<code>nuvport.NUVPort</code>	<code>vports</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>

## 146.3 Parents

- *nucontainerinterface.NUContainerInterface*
- *nudomain.NUDomain*
- *nuvport.NUVPort*
- *nuvminterface.NUVMInterface*
- *nubridgeinterface.NUBridgeInterface*
- *nul2domain.NUL2Domain*
- *nuhostinterface.NUHostInterface*
- *nume.NUMe*

## NUREDIRECTIONTARGETTEMPLATE

`nuredirectiontargettemplate.NURedirectionTargetTemplate (bambou.nurest_object.NUMetaRESTObj`  
Template for a vporttag. It can be created only at the template level and available for all instances.

### 147.1 Attributes

- **name (Mandatory):** Name of this redirection target template
- **last\_updated\_by:** ID of the user who last updated the object.
- **redundancy\_enabled:** Allow/Disallow redundant appliances and VIP
- **description:** Description of this redirection target template
- **end\_point\_type (Mandatory):** VPortTagEndPointType is an enum. It defines the type of header rewrite and forwarding performed by VRS when the endpoint is used as a PBR destination. Possible values are NONE, L3, VIRTUAL\_WIRE.
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **trigger\_type:** Trigger type, could be NONE/GARP - THIS IS READONLY
- **external\_id:** External object ID. Used for integration with third party systems

### 147.2 Children

<b>class</b>	<b>fetcher</b>
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nujob.NUJob</i>	jobs
<i>nueventlog.NUEventLog</i>	event_logs

### 147.3 Parents

- *nul2domaintemplate.NUL2DomainTemplate*
- *nudomaintemplate.NUDomainTemplate*



## NUREDUNDANCYGROUP

`nuredundancygroup.NURedundancyGroup(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents Redundant Group formed by two Gateways.

### 148.1 Attributes

- **name (Mandatory):** Name of the Redundancy Group
- **last\_updated\_by:** ID of the user who last updated the object.
- **gateway\_peer1\_autodiscovered\_gateway\_id:** The Auto Discovered Gateway configuration owner in this Redundant Group.
- **gateway\_peer1\_id:** The gateway configuration owner in this Redundant Group. when Redundant Group is deleted this gateway will receive vport associations
- **gateway\_peer1\_name:** The gateway configuration owner name in this Redundant Group
- **gateway\_peer2\_autodiscovered\_gateway\_id:** The Auto Discovered Gateway peer in this Redundant Group
- **gateway\_peer2\_id:** The gateway peer in this Redundant Group. when Redundant Group is deleted this gateway will not receive vport associations
- **gateway\_peer2\_name:** The gateway peer name in this Redundant Group
- **redundant\_gateway\_status:** The status of Redundant Group, possible values are FAILED, SUCCESS  
Possible values are FAILED, SUCCESS, .
- **permitted\_action:** The permitted action to USE/EXTEND this Gateway Possible values are USE, READ, ALL, INSTITUTE, EXTEND, DEPLOY, .
- **personality:** derived personality of the Redundancy Group - VSG,VRSG,NSG,OTHER Possible values are VSG, VSA, VRSG, DC7X50, NSG, HARDWARE\_VTEP, OTHER, .
- **description:** Description of the Redundancy Group
- **enterprise\_id:** The enterprise associated with this Redundant Group. This is a read only attribute
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **vtep:** Represent the system ID or the Virtual IP of a service used by a Gateway (VSG for now) to establish a tunnel with a remote VSG or hypervisor. The format of this field is consistent with an IP address.
- **external\_id:** External object ID. Used for integration with third party systems

## 148.2 Children

<b>class</b>	<b>fetcher</b>
<i>nugateway.NUGateway</i>	gateways
<i>nupermision.NUPermission</i>	permissions
<i>nuwanservice.NUWANService</i>	wan_services
<i>numetadata.NUMetadata</i>	metadatas
<i>nualarm.NUAlarm</i>	alarms
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuenterpriasepermission.NUEnterprisePermission</i>	enterprise_permissions
<i>nuport.NUPort</i>	ports
<i>nuvsgredundantport.NUVsgRedundantPort</i>	vsg_redundant_ports
<i>nueventlog.NUEventLog</i>	event_logs

## 148.3 Parents

- *nume.NUMe*
- *nuenterpriase.NUEnterprise*



## NUREDUNDANTPORT

`nuredundantport.NURedundantPort(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents a Port under a particular gateway object or redundant group object.

### 149.1 Attributes

- `vlan_range`: VLAN Range of the Port. Format must conform to a-b,c,d-f where a,b,c,d,f are integers between 0 and 4095.
- `name` (**Mandatory**): Name of the Port
- `last_updated_by`: ID of the user who last updated the object.
- `permitted_action`: The permitted action to USE/EXTEND this Gateway.
- `description`: A description of the Port
- `physical_name` (**Mandatory**): Identifier of the Port
- `infrastructure_profile_id`: The ID of the infrastructure profile this instance is associated with.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `port_peer1_id`: The master gateway peer port id.
- `port_peer2_id`: The slave gateway peer port id.
- `port_type` (**Mandatory**): Type of the Port.
- `use_untagged_heartbeat_vlan`: A flag to indicate if for this redundant port an untagged heartbeat VLAN is to be used. If this is not set then will use the heartbeat VLAN set by the NS redundant group
- `use_user_mnemonic`: determines whether to use user mnemonic of the Port
- `user_mnemonic`: user mnemonic of the Port
- `associated_egress_qos_policy_id`: ID of the Egress QOS Policy associated with this Vlan.
- `status`: Status of the port.
- `external_id`: External object ID. Used for integration with third party systems

## 149.2 Children

<b>class</b>	<b>fetcher</b>
<i>numetadata.NUMetadata</i>	metadatas
<i>nuvlan.NUVLAN</i>	vlan
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nunsport.NUNSPort</i>	ns_ports

## 149.3 Parents

- *nunsredundantgatewaygroup.NUNSRedundantGatewayGroup*

## NUROUTINGPOLICY

```
nuroutingpolicy.NURoutingPolicy(bambou.nurest_object.NUMetaRESTObject,):
```

None

### 150.1 Attributes

- **name (Mandatory):** policy name, unique within an enterprise
- **default\_action (Mandatory):** accept/reject
- **description:** None
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **policy\_definition:** String blob
- **external\_id:** External object ID. Used for integration with third party systems

### 150.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas

### 150.3 Parents

- *nudomain.NUDomain*
- *nume.NUMe*
- *numenterprise.NUEnterprise*



## NUSHAREDNETWORKRESOURCE

`nusharednetworkresource.NUSharedNetworkResource(bambou.nurest_object.NUMetaRESTObject,)` :

This defines shared infrastructure resources that are created by user with CSPROOT role. These resources can be used by all the enterprises in the data center for various purposes. Examples of shared resources are public subnet, floating subnet, public L2 domain.

### 151.1 Attributes

- `ecmp_count`: Domain specific Equal-cost multi-path routing count, `ECMPCount = 1` means no ECMP
- `dhcp_managed`: true if DHCP is enabled else it is false. This value is always true for network resource of type PUBLIC or FLOATING.
- `back_haul_route_distinguisher`: `backHaulRouteDistinguisher` of the Shared Resource
- `back_haul_route_target`: `backHaulRouteTarget` of the Shared Resource
- `back_haul_vnid`: `backHaulVNID` of the Shared Resource
- `name` (**Mandatory**): Name of the shared resource. Valid characters are alphabets, numbers, space and hyphen(-).
- `last_updated_by`: ID of the user who last updated the object.
- `gateway`: Gatemask configured on the shared resource
- `gateway_mac_address`: MAC address for a public subnet or managed l2 domain
- `access_restriction_enabled`: Boolean indicates that this shared network resource is available to everyone by default or not
- `address` (**Mandatory**): Address configured on the shared resource
- `permitted_action_type`: Permitted action on this shared network resource
- `description`: Description of the shared resource
- `netmask` (**Mandatory**): Netmask configured on the shared resource
- `shared_resource_parent_id`: Parent ID of the floating IP subnet to which this FIP subnet must be attached. If empty it will be created in a new domain.
- `vn_id`: VNID of the Shared Resource
- `underlay`: Indicates whether this shared subnet is in underlay or not.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `domain_route_distinguisher`: Route distinguisher configured on the shared resource

- `domain_route_target`: Route target configured on the shared resource
- `uplink_gw_vlan_attachment_id`: VLAN ID to which this vport must be attached
- `uplink_interface_ip`: IP address of the host interface
- `uplink_interface_mac`: MAC address of the host interface
- `uplink_vport_name`: Name of the uplink vport
- `use_global_mac`: if this flag is enabled, the system configured globalMACAddress will be used as the gateway mac address
- `associated_pat_mapper_id`: The ID of the PatMapper entity to which this pool is associated to.
- `external_id`: External object ID. Used for integration with third party systems
- `dynamic_pat_allocation_enabled`: Indicates if PAT Mapping is enabled for the SharedNetworkResource or not
- `type` (**Mandatory**): Type of the shared resource.

## 151.2 Children

class	fetcher
<i>nupatipentry.NUPATIPEntry</i>	patip_entries
<i>nuaddressrange.NUAddressRange</i>	address_ranges
<i>numetadata.NUMetadata</i>	metadatas
<i>nudhcpoption.NUDHCPOption</i>	dhcp_options
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuenterprisepermission.NUEnterprisePermission</i>	enterprise_permissions
<i>nuvpnconnection.NUVPNConnection</i>	vpn_connections
<i>nustaticroute.NUStaticRoute</i>	static_routes

## 151.3 Parents

- *nume.NUMe*
- *nupatmapper.NUPATMapper*
- *nuenterprise.NUEnterprise*

## NUSITEINFO

```
nusiteinfo.NUSiteInfo(bambou.nurest_object.NUMetaRESTObject,):
```

Remote Site info.

## 152.1 Attributes

- **name (Mandatory)**: name of the Remote Site.
- **last\_updated\_by**: ID of the user who last updated the object.
- **address (Mandatory)**: unique fqdn/address of the remote site
- **description**: Description of the Remote Site.
- **site\_identifier**: unique identifier of the remote site
- **xmpp\_domain (Mandatory)**: unique xmpp domain name of the remote site
- **entity\_scope**: Specify if scope of entity is Data center or Enterprise level
- **external\_id**: External object ID. Used for integration with third party systems

## 152.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas

## 152.3 Parents

- *nume.NUMe*





## NUSSHKEY

`nusshkey.NUSSHKey(bambou.nurest_object.NUMetaRESTObject,):`

None

### 153.1 Attributes

- `name` (**Mandatory**): Name of the SSH Key.
- `description`: A description of the SSH Key.
- `key_type`: Type of SSH Key defined. Only RSA supported for now.
- `public_key`: Public Key of a SSH Key Pair.



## NUSTATICROUTE

`nustaticroute.NUStaticRoute (bambou.nurest_object.NUMetaRESTObject, ) :`

Static routes allow end users to define how traffic is routed through the dVRS in addition to the routes learned by VSC through VM activation. By using static routes, end users can define for example that all traffic with a destination address towards a specific subnet must be forwarded to a specific VM attached in the dVRS and this VM could be a firewall

### 154.1 Attributes

- `ip_type`: IPv4 or IPv6
- `ipv6_address`: IPv6 address of the route
- `last_updated_by`: ID of the user who last updated the object.
- `address` (**Mandatory**): IP address of the route
- `netmask` (**Mandatory**): Netmask associated with the route
- `next_hop_ip` (**Mandatory**): IP address of the next hop. This must be a VM attached to the dVRS
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `route_distinguisher`: Route distinguisher associated with the nexthop. System generates this identifier automatically
- `external_id`: External object ID. Used for integration with third party systems
- `type`: Type flag for static-route provisioning for exit-domain (break-to-underlay) prefixes.

### 154.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>

### 154.3 Parents

- `nucontainerinterface.NUContainerInterface`

- *nusharednetworkresource.NUSharedNetworkResource*
- *nudomain.NUDomain*
- *nuvminterface.NUVMInterface*
- *nul2domain.NUL2Domain*
- *nuhostinterface.NUHostInterface*
- *nume.NUMe*

## NUSTATISTICS

`nustatistics.NUStatistics(bambou.nurest_object.NUMetaRESTObject,):`

Retrieves the statistics for a particular domain, zone, subnet, or VM.

### 155.1 Attributes

- `version`: Version of this Sequence number.
- `end_time`: End time for the statistics to be retrieved
- `start_time`: Start time for the statistics to be retrieved
- `stats_data`: Map<TCAMetric, Long[]> TCAMetric is an Enum. Possible values are packets\_in, bytes\_in, packets\_in\_dropped, packets\_in\_errors, packets\_out, bytes\_out, packets\_out\_dropped, packets\_out\_errors, packets\_dropped\_rate\_limit
- `number_of_data_points`: Number of data points between start time and end time

### 155.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 155.3 Parents

- `nucontainerinterface.NUContainerInterface`
- `nuingressadvfwdentrytemplate.NUIngressAdvFwdEntryTemplate`
- `nuzone.NUZone`
- `nuegressaclentrytemplate.NUEgressACLEntryTemplate`
- `nuingressexternalservicetemplateentry.NUIngressExternalServiceTemplateEntry`
- `nudomain.NUDomain`
- `nuvport.NUVPort`
- `nusubnet.NUSubnet`

- *nuvminterface.NUVMInterface*
- *nubridgeinterface.NUBridgeInterface*
- *nutier.NUTier*
- *nuaddressmap.NUAddressMap*
- *nuingressaclentrytemplate.NUIngressACLEntryTemplate*
- *nul2domain.NUL2Domain*
- *nuhostinterface.NUHostInterface*
- *nunsport.NUNSPort*
- *nupatnatpool.NUPATNATPool*

## NUSTATISTICSPOLICY

`nustatisticpolicy.NUStatisticsPolicy(bambou.nurest_object.NUMetaRESTObject, ) :`

Defines the frequency of statistics collection associated with an object.

### 156.1 Attributes

- **name (Mandatory):** Name of statistics policy
- **last\_updated\_by:** ID of the user who last updated the object.
- **data\_collection\_frequency (Mandatory):** How frequent to collect statistics in seconds
- **description:** A description of the statistics policy
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **external\_id:** External object ID. Used for integration with third party systems

### 156.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas

### 156.3 Parents

- *nuzone.NUZone*
- *nudomain.NUDomain*
- *nuvport.NUVPort*
- *nusubnet.NUSubnet*
- *nutier.NUTier*
- *nuaddressmap.NUAddressMap*
- *nul2domain.NUL2Domain*
- *nunsport.NUNSport*
- *nupatnatpool.NUPATNATPool*





## NUSTATSCOLLECTORINFO

`nustatscollectorinfo.NUStatsCollectorInfo(bambou.nurest_object.NUMetaRESTObject, ) :`

Identifies the IP address of the stats collector entity that must be used.

### 157.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `address_type`: Type for stats collector address Possible values are ip, fqdn, .
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `port`: Port(s) of the stats collector process
- `ip_address`: IP address(es) of the stats collector process
- `proto_buf_port`: Protobuf Port(s) of the stats collector process
- `external_id`: External object ID. Used for integration with third party systems

### 157.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	metadatas
<code>nuglobalmetadata.NUGlobalMetadata</code>	global_metadatas

### 157.3 Parents

- `nume.NUme`



## NUSUBNET

`nusubnet.NUSubnet(bambou.nurest_object.NUMetaRESTObject,):`

This is the definition of a subnet associated with a Zone.

### 158.1 Attributes

- `pat_enabled`: None
- `dpi`: determines whether or not Deep packet inspection is enabled
- `ip_type`: IPv4 or IPv6
- `ipv6_address`: IP address of the subnet defined. In case of zone, this is an optional field for and allows users to allocate an IP address range to a zone. The VSD will auto-assign IP addresses to subnets from this range if a specific IP address is not defined for the subnet
- `ipv6_gateway`: The IPv6 address of the gateway of this subnet
- `maintenance_mode`: `maintenanceMode` is an enum that indicates if the SubNetwork is accepting VM activation requests.
- `name` (**Mandatory**): Name of the current entity (Zone or zone template or subnet etc..) Valid characters are alphabets, numbers, space and hyphen ( - ).
- `last_updated_by`: ID of the user who last updated the object.
- `gateway`: The IP address of the gateway of this subnet
- `gateway_mac_address`: None
- `address`: IP address of the subnet defined. In case of zone, this is an optional field for and allows users to allocate an IP address range to a zone. The VSD will auto-assign IP addresses to subnets from this range if a specific IP address is not defined for the subnet
- `default_action`: If PAT is disabled then this flag indicates what to do if routes don't exist in overlay, will default to drop | possible values `USE_UNDERLAY`, `DROP_TRAFFIC` Possible values are `USE_UNDERLAY`, `DROP_TRAFFIC`, .
- `template_id`: The ID of the subnet template that this subnet object was derived from
- `service_id`: The service ID used by the VSCs to identify this subnet
- `description`: A description field provided by the user that identifies the subnet
- `netmask`: Netmask of the subnet defined
- `vn_id`: Current Network's globally unique VXLAN network identifier generated by VSD

- `encryption`: Determines whether or not IPSEC is enabled.
- `underlay`: Boolean flag to indicate whether underlay is enabled directly or indirectly
- `underlay_enabled`: Indicates whether UNDERLAY is enabled for the subnets in this domain
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `policy_group_id`: PG ID for the subnet. This is unique per domain and will be in the range 1-4095
- `route_distinguisher`: The Route Distinguisher value assigned by VSD for this subnet that is used by the BGP-EVPN protocol in VSC
- `route_target`: The Route Target value assigned by VSD for this subnet that is used by the BGP-EVPN protocol in VSC
- `split_subnet`: Need to add correct description
- `proxy_arp`: when set VRS will act as ARP Proxy
- `use_global_mac`: if this flag is enabled, the system configured globalMACAddress will be used as the gateway mac address
- `associated_application_id`: The associated application ID.
- `associated_application_object_id`: The associated application object ID.
- `associated_application_object_type`: The associated application object type. Refer to API section for supported types.
- `associated_multicast_channel_map_id`: The ID of the Multi Cast Channel Map this Subnet/Subnet Template is associated with. This has to be set when enableMultiCast is set to ENABLED
- `associated_shared_network_resource_id`: The ID of public subnet that is associated with this subnet
- `public`: when set to true means public subnet under a public zone
- `multicast`: multicast is enum that indicates multicast policy on Subnet/Subnet Template.
- `external_id`: External object ID. Used for integration with third party systems

## 158.2 Children

<b>class</b>	<b>fetcher</b>
<i>nutca.NUTCA</i>	tcas
<i>nuaddressrange.NUAddressRange</i>	address_ranges
<i>nuvmresync.NUVMResync</i>	vm_resyncs
<i>numetadata.NUMetadata</i>	metadatas
<i>nubgpneighbor.NUBGPNeighbor</i>	bgp_neighbors
<i>nudhcpoption.NUDHCPOption</i>	dhcp_options
<i>nuvirtualip.NUVirtualIP</i>	virtual_ips
<i>nuikegatewayconnection.NUIKEGatewayConnection</i>	ike_gateway_connections
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuvm.NUVM</i>	vms
<i>nuvminterface.NUVMInterface</i>	vm_interfaces
<i>nucontainer.NUContainer</i>	containers
<i>nucontainerinterface.NUContainerInterface</i>	container_interfaces
<i>nucontainerresync.NUContainerResync</i>	container_resyncs
<i>nuqos.NUQOS</i>	qoss
<i>nuvport.NUVPort</i>	vports
<i>nuipreservation.NUIPReservation</i>	ip_reservations
<i>nustatistics.NUStatistics</i>	statistics
<i>nustatisticspolicy.NUStatisticsPolicy</i>	statistics_policies
<i>nueventlog.NUEventLog</i>	event_logs

## 158.3 Parents

- *nuzone.NUZone*
- *nusubnettemplate.NUSubnetTemplate*
- *nuikegatewayconnection.NUIKEGatewayConnection*
- *nudomain.NUDomain*
- *nume.NUMe*
- *nunsgateway.NUNSGateway*



## NUSUBNETTEMPLATE

`nusubnettemplate.NUSubnetTemplate(bambou.nurest_object.NUMetaRESTObject, ) :`

As domain and zone objects, subnet objects are created in VSD as derived by templates. This object describes the subnet template.

### 159.1 Attributes

- `dpi`: determines whether or not Deep packet inspection is enabled
- `ip_type`: IPv4 or IPv6
- `ipv6_gateway`: The IPv6 address of the gateway of this subnet
- `ipv6address`: IPv6 address of the subnet defined. In case of zone, this is an optional field for and allows users to allocate an IP address range to a zone. The VSD will auto-assign IP addresses to subnets from this range if a specific IP address is not defined for the subnet
- `name` (**Mandatory**): Name of the current entity (Zone or zone template or subnet etc..) Valid characters are alphabets, numbers, space and hyphen( - ).
- `last_updated_by`: ID of the user who last updated the object.
- `gateway`: The IP address of the gateway of this subnet
- `address` (**Mandatory**): IP address of the subnet defined. In case of zone, this is an optional field for and allows users to allocate an IP address range to a zone. The VSD will auto-assign IP addresses to subnets from this range if a specific IP address is not defined for the subnet
- `description`: A description field provided by the user that identifies the subnet
- `netmask` (**Mandatory**): Netmask of the subnet defined
- `encryption`: Determines whether or not IPSEC is enabled. Possible values are INHERITED, ENABLED, DISABLED, .
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `split_subnet`: Need to add correct description
- `proxy_arp`: when set VRS will act as ARP Proxy
- `use_global_mac`: if this flag is enabled, the system configured globalMACAddress will be used as the gateway mac address
- `associated_multicast_channel_map_id`: The ID of the Multi Cast Channel Map this Subnet/Subnet Template is associated with. This has to be set when enableMultiCast is set to ENABLED
- `multicast`: Indicates multicast policy on Subnet/Subnet Template.

- `external_id`: External object ID. Used for integration with third party systems

## 159.2 Children

class	fetcher
<i>nuaddressrange.NUAddressRange</i>	address_ranges
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuqos.NUQOS</i>	qoss
<i>nusubnet.NUSubnet</i>	subnets
<i>nueventlog.NUEventLog</i>	event_logs

## 159.3 Parents

- *nuzonetemplate.NUZoneTemplate*
- *nudomaintemplate.NUDomainTemplate*



## NUSYSTEMCONFIG

`nusystemconfig.NUSystemConfig(bambou.nurest_object.NUMetaRESTObject,)` :

The system configuration which can be dynamically managed using rest api.

### 160.1 Attributes

- `acl_allow_origin`: Defines the domains allowed for access control list.
- `ecmp_count`: System Default Equal-cost multi-path routing count, Every Domain derives ECMP count from this value unless specifically set for the domain
- `ldap_sync_interval`: LDAP Sync-Up task interval in seconds.
- `ldap_trust_store_certificate`: Location of the truststore which is need to store LDAP server certificates. Default is cacerts located in `java.home/lib/security/cacerts`. Uncomment below setting if you need to use a different file
- `ldap_trust_store_password`: Password to access the truststore. Uncomment below line to change its value.
- `ad_gateway_purge_time`: Timers in sec for undefined vms to be deleted(min =7200, max = 86400).
- `rd_lower_limit`: route distinguisher lower limit
- `rd_public_network_lower_limit`: route distinguisher public network lower limit
- `rd_public_network_upper_limit`: route distinguisher public network upper limit
- `rd_upper_limit`: route distinguisher upper limit
- `zfb_bootstrap_enabled`: Whether the NSG should auto bootstrap using ZFB
- `zfb_request_retry_timer`: Retry time for the ZFB daemon to recheck ZFBRequest Status in seconds
- `zfb_scheduler_stale_request_timeout`: Time for the ZFB scheduler to wait in seconds before deleting a stale request
- `dhcp_option_size`: Defines total DHCP options that can be set on a domain.
- `vm_cache_size`: LRU Map size for vm, this value has to set based on memory given to VSD jvm not finalized.
- `vm_purge_time`: Timers in sec for undefined vms to be deleted.
- `vm_resync_deletion_wait_time`: After resync on vm , if no controller returns with a VM request with in the below timeframe then it will get deleted deletion wait time in minutes.
- `vm_resync_outstanding_interval`: Outstanding VM resync interval (in secs). System wide value.
- `vm_unreachable_cleanup_time`: Timers in sec for unreachable VMs for cleanup.

- `vm_unreachable_time`: Timers in sec for unreachable VMs.
- `vnid_lower_limit`: Virtual network ID offset
- `vnid_public_network_lower_limit`: Virtual network ID public network lower limit
- `vnid_public_network_upper_limit`: Virtual network ID public network upper limit
- `vnid_upper_limit`: Virtual network ID upper limit
- `api_key_renewal_interval`: Defines the interval in seconds, before the expiry time, that can used to renew the `apiKey` by making me API call. Minimum value is 1 min and maximum is 5 min.
- `api_key_validity`: Defines the `apiKey` validity duration in seconds. Default is 24 hours and minimum value is 10 min.
- `vport_init_stateful_timer`: Defines the timeout in seconds for `vport` initialization to stateful. Default value is 300 secs and the timeout should be between 0 to 86400 seconds.
- `lru_cache_size_per_subnet`: LRU Map size per subnet (to hold the deleted vm's ip addresses).
- `vsc_on_same_version_as_vsd`: This flag is used to indicate that whether VSC is on the same version as VSD or not.
- `vsd_read_only_mode`: True means VSD readonly mode enabled. False means VSD readonly mode disabled
- `vsd_upgrade_is_complete`: This flag is used to indicate that whether VSD upgrade is complete, it is expected that `csproot` will set to true, after VSD upgrade is complete and also making sure that all VSC's audits and Gateway audits with VSD are done
- `as_number`: Autonomous System Number, Used for RT/RD auto-generation
- `rt_lower_limit`: route target lower limit
- `rt_public_network_lower_limit`: route target public network lower limit
- `rt_public_network_upper_limit`: route target public network upper limit
- `rt_upper_limit`: route target upper limit
- `evpnbgp_community_tag_as_number`: Autonomous System Number, Used for EVPNBGPCCommunityTag auto-generation
- `evpnbgp_community_tag_lower_limit`: EVPNBGPCCommunityTag lower limit
- `evpnbgp_community_tag_upper_limit`: EVPNBGPCCommunityTag upper limit
- `page_max_size`: Defines upper bound for the page size. Configured or input page size should be less than this max page size.
- `page_size`: Defines the page size for the results returned by the REST call.
- `last_updated_by`: ID of the user who last updated the object.
- `max_failed_logins`: Maximum failed login attempts before the account is locked (min = 5, max = 10). 0 = not enforced (unlimited attempts). This is not enforced if LDAP is used for authorization
- `max_response`: Defines maximum results returned by the REST call (allowed max=5000).
- `accumulate_licenses_enabled`: Whether the various VRS license flavours be merged in one pool
- `performance_path_selection_vnid`: performance Path Selection Virtual Network ID
- `service_id_upper_limit`: Service id upper limit system wide value
- `key_server_monitor_enabled`: Enable the keyserver debug monitor (ie. `ksmon` command)

- `key_server_vsd_data_synchronization_interval`: KeyServer time in seconds between full resyncs of VSD data (just in case of missed events)
- `offset_customer_id`: Customer id offset, this value has to be set before jboss starts , after that any change of value is ignored (minexclusive = 0, max = 20000) system wide value
- `offset_service_id`: Service id offset, this value has to be set before jboss starts during install time, after that any change of value is ignored (minexclusive = 0, max = 40000) system wide value
- `ejbca_nsg_certificate_profile`: EJBCA NSG Certificate Profile
- `ejbca_nsg_end_entity_profile`: EJBCA NSG End Entity Profile
- `ejbca_ocsp_responder_cn`: EJBCA OCSP Responder CommonName
- `ejbca_ocsp_responder_uri`: EJBCA OCSP Responder URI
- `ejbca_vsp_root_ca`: EJBCA VSP CA
- `alarms_max_per_object`: Maximum alarms per object for example max distinct alarms for specific VM (min = 5, max =20)
- `elastic_cluster_name`: Specifies the name of the Elastic Search Cluster.
- `elastic_search_ui_address`: Specifies the server address Elastic Search Cluster.
- `allow_enterprise_avatar_on_nsg`: Allow Enterprise Avatar to be populated on NSG Portal
- `global_mac_address`: the MAC Address to use for those subnets that have the useGlobalMAC flag enabled.
- `flow_collection_enabled`: Enables flow statistics collection. It is needed for the VSS feature, and requires a valid VSS license. This option requires “statisticsEnabled”.
- `inactive_timeout`: Defines the inactive timeout for the client. If the client is inactive for more than timeout, server clears off all the cache/information regarding the client. This value should be greater than event processor max timeout
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `domain_tunnel_type`: Default Domain Tunnel Type .Possible values are VXLAN,GRE Possible values are DC\_DEFAULT, GRE, VXLAN, .
- `post_processor_threads_count`: Post processor thread count.
- `group_key_default_sek_generation_interval`: Group Key Encryption Profile Default SEK Generation Interval
- `group_key_default_sek_lifetime`: Group Key Encryption Profile Default SEK Lifetime
- `group_key_default_sek_payload_encryption_algorithm`: Group Key Encryption Profile Default Sek Payload Encryption Algorithm.
- `group_key_default_sek_payload_signing_algorithm`: Group Key Encryption Profile Default Sek Payload Signing Algorithm.
- `group_key_default_seed_generation_interval`: Group Key Encryption Profile Default Seed Generation Interval
- `group_key_default_seed_lifetime`: Group Key Encryption Profile Default Seed Lifetime
- `group_key_default_seed_payload_authentication_algorithm`: Group Key Encryption Profile Default Seed Payload Authentication Algorithm.
- `group_key_default_seed_payload_encryption_algorithm`: Group Key Encryption Profile Default Seed Payload Encryption Algorithm.

- `group_key_default_seed_payload_signing_algorithm`: Group Key Encryption Profile Default Seed Payload Signature Algorithm.
- `group_key_default_traffic_authentication_algorithm`: Group Key Encryption Profile Default Traffic Authentication Algorithm.
- `group_key_default_traffic_encryption_algorithm`: Group Key Encryption Profile Default Traffic Encryption Algorithm.
- `group_key_default_traffic_encryption_key_lifetime`: Group Key Encryption Profile Default Traffic Encryption Key Lifetime
- `group_key_generation_interval_on_forced_re_key`: Time in seconds before new keys will be generated in the case of a forced re-key event
- `group_key_generation_interval_on_revoke`: Time in seconds before new keys will be generated in the case of a revoke event
- `group_key_minimum_sek_generation_interval`: Group Key Encryption Profile Minimum SEK Generation Interval
- `group_key_minimum_sek_lifetime`: Group Key Encryption Profile Minimum SEK Lifetime
- `group_key_minimum_seed_generation_interval`: Group Key Encryption Profile Default Seed Generation Interval
- `group_key_minimum_seed_lifetime`: Group Key Encryption Profile Default Seed Lifetime
- `group_key_minimum_traffic_encryption_key_lifetime`: Group Key Encryption Profile Minimum TEK Lifetime
- `nsg_bootstrap_endpoint`: NSG Bootstrap Endpoint
- `nsg_config_endpoint`: NSG Config Endpoint
- `nsg_local_ui_url`: NSG Local UI URL - will be redirected on NSG to localhost
- `esi_id`: ESI ID offset
- `csproot_authentication_method`: Authentication method for csproot when local authentication is not used for CSP organization
- `stack_trace_enabled`: True to enable stacktrace in the REST call.
- `stateful_acl_non_tcp_timeout`: Defines the timeout in seconds for stateful ACLs that are not of type TCP.
- `stateful_acltcp_timeout`: Defines the timeout in seconds for stateful ACLs that are of type TCP.
- `static_wan_service_purge_time`: Timers in sec for unreachable static WAN Services to be deleted.
- `statistics_enabled`: This flag is used to indicate if statistics is enabled in the system. CSProot is expected to activate this through the enable statistics script.
- `stats_collector_address`: Specify the ip address(es) of the stats collector.
- `stats_collector_port`: Specify the port number(s) of the stats collector.
- `stats_collector_proto_buf_port`: Specify the protobuf port number(s) of the stats collector.
- `stats_max_data_points`: Specifies the maximum number of data points to support.
- `stats_min_duration`: Default minimum duration for statistics to be displayed in UI is 30 days in seconds.
- `stats_number_of_data_points`: Specifies number of data points.
- `stats_tsdb_server_address`: Specifies the TSDB server location.

- `sticky_ecmp_idle_timeout`: sticky ECMP Idle Timeout in seconds
- `subnet_resync_interval`: After resync on a subnet , another resync on the same subnet is allowed based on the below value subnet resync complete wait time in min.
- `subnet_resync_outstanding_interval`: Outstanding subnet resync interval (in secs). System wide value.
- `customer_id_upper_limit`: Customer id upper limit, system wide value
- `customer_key`: Customer key associated with the license
- `avatar_base_path`: Defines location where image files needs to be copied. Above URL should be configured to read the file from this location.
- `avatar_base_url`: Defines the url to read the avatar image files
- `event_log_cleanup_interval`: Cleanup task run interval in seconds.
- `event_log_entry_max_age`: Maximum age in days for cleanup of the eventlog entries. On every periodic interval run, any eventlog entries older than this max age will be deleted.
- `event_processor_interval`: Defines time interval in milliseconds when events collected for a client should be processed.
- `event_processor_max_events_count`: Defines the maximum number of events to be collected in case of events burst.
- `event_processor_timeout`: Defines the maximum time period in milliseconds for the Rest server to wait before sending the events from the system.
- `two_factor_code_expiry`: Two Factor Code Expiry in Seconds
- `two_factor_code_length`: Two Factor Code Length
- `two_factor_code_seed_length`: Two Factor Seed length in bytes
- `external_id`: External object ID. Used for integration with third party systems
- `dynamic_wan_service_diff_time`: Timers in sec for dynamic WAN Services to be considered not seen by 7X50.
- `syslog_destination_host`: Specifies the remote syslog destination host
- `syslog_destination_port`: Specified the remote syslog destination port
- `sysmon_cleanup_task_interval`: Sysmon cleanup task run interval in seconds.
- `sysmon_node_presence_timeout`: Node presence timeout in seconds if no messages.
- `sysmon_probe_response_timeout`: Probe response timeout in seconds.
- `system_avatar_data`: CSP Avatar Data
- `system_avatar_type`: None

## 160.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

## 160.3 Parents

- *nume.NUMe*

## NUTCA

`nutca.NUTCA(bambou.nurest_object.NUMetaRESTObject, ) :`

Provides the definition of the Threshold Control Alarms.

### 161.1 Attributes

- `url_end_point`: URL endpoint to post Alarm data to when TCA is triggered
- `name` (**Mandatory**): The name of the TCA
- `target_policy_group_id`: Target policygroup when TCA is triggered
- `last_updated_by`: ID of the user who last updated the object.
- `action` (**Mandatory**): Action to be taken when TCA is fired - Alert or PolicyGroupChange
- `period` (**Mandatory**): The averaging period
- `description`: Description of the TCA
- `metric` (**Mandatory**): The metric associated with the TCA.
- `threshold` (**Mandatory**): The threshold that must be exceeded before an alarm is issued
- `throttle_time`: Throttle time in secs
- `disable`: This flag is used to indicate whether the watch(TCA) is enabled/disabled
- `display_status`: Explanation of the TCA status
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `count`: Count of the attempts by maintenance thread to create/update watcher
- `status`: This flag is used to indicate the status of TCA
- `external_id`: External object ID. Used for integration with third party systems
- `type` (**Mandatory**): Rolling average or sequence of samples over the averaging period.

## 161.2 Children

<b>class</b>	<b>fetcher</b>
<i>numetadata.NUMetadata</i>	metadatas
<i>nualarm.NUAlarm</i>	alarms
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nueventlog.NUEventLog</i>	event_logs

## 161.3 Parents

- *nucontainerinterface.NUContainerInterface*
- *nuzone.NUZone*
- *nudomain.NUDomain*
- *nuvport.NUVPort*
- *nusubnet.NUSubnet*
- *nuvminterface.NUVMInterface*
- *nubridgeinterface.NUBridgeInterface*
- *nutier.NUTier*
- *nul2domain.NUL2Domain*
- *nuhostinterface.NUHostInterface*
- *nume.NUMe*



## NUTIER

`nutier.NUTier(bambou.nurest_object.NUMetaRESTObject, ) :`

Tier represents a portion of an Application.

### 162.1 Attributes

- **name (Mandatory):** Name of the application tier.
- **last\_updated\_by:** ID of the user who last updated the object.
- **gateway:** The IP address of the gateway for this tier.
- **address:** IP address of the tier defined.
- **description:** Description of the application tier.
- **metadata:** Metadata field to store tier related data.
- **netmask:** Netmask for the tier.
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **associated\_application\_id:** The associated network macro ID.
- **associated\_floating\_ip\_pool\_id:** The associated floating IP Pool ID.
- **associated\_network\_macro\_id:** The associated network macro ID.
- **associated\_network\_object\_id:** The associated network object id.
- **associated\_network\_object\_type:** The associated network object type. Refer to API section for supported types.
- **external\_id:** External object ID. Used for integration with third party systems
- **type (Mandatory):** Type of the application tier.

## 162.2 Children

<b>class</b>	<b>fetcher</b>
<i>nutca.NUTCA</i>	tcas
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuvm.NUVM</i>	vms
<i>nucontainer.NUContainer</i>	containers
<i>nuvport.NUVPort</i>	vports
<i>nustatistics.NUStatistics</i>	statistics
<i>nustatisticspolicy.NUStatisticsPolicy</i>	statistics_policies
<i>nueventlog.NUEventLog</i>	event_logs

## NUUNDERLAY

`nuunderlay.NUUnderlay(bambou.nurest_object.NUMetaRESTObject,):`

None

### 163.1 Attributes

- `name`: Name of the underlay
- `description`: Description of the underlay
- `underlay_id`: None

### 163.2 Parents

- *nume.NUMe*
- *nuuplinkconnection.NUUplinkConnection*



## NUUPLINKCONNECTION

```
nuuplinkconnection.NUUplinkConnection(bambou.nurest_object.NUMetaRESTObject,):  
None
```

### 164.1 Attributes

- `dns_address`: DNS server address
- `password`: PPPoE password.
- `gateway`: IP address of the gateway bound to the port
- `address`: IP address for static configuration
- `advertisement_criteria`: Advertisement Criteria for Traffic Flow
- `netmask`: Subnet mask
- `mode`: Specify how to connect to the network. Possible values: Any, Dynamic (DHCP), Static (static configuration is required), PPPoE (pppoe configuration required). Default: Dynamic
- `role`: To allow prioritisation of traffic, the NSG network ports must be configured with an uplink type or tag value which will be used in the identification of packets being forwarded. That identification is at the base of the selection of which network port will serve in sending packets to the outside world. The default value is PRIMARY. Possible values are PRIMARY, SECONDARY, TERTIARY, UNKNOWN,
- `uplink_id`: ID that uniquely identifies the uplink.
- `username`: PPPoE username
- `assoc_underlay_id`: UUID of the underlay associated to the uplink.
- `associated_vsc_profile_id`: The ID of the infrastructure VSC profile this is associated with this instance of a vlan or vlan template.

### 164.2 Children

class	fetcher
<i>nuunderlay.NUUnderlay</i>	underlays

## 164.3 Parents

- *nuvlan.NUVLAN*
- *nuvlantemplate.NUVLANTemplate*

## NUUPLINKRD

`nuuplinkrd.NUUplinkRD(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents a network port uplink route distinguisher value.

### 165.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `route_distinguisher`: The uplink route distinguisher value is used to identify which route packets should be flowing through with regards to having multiple network ports on the VRS/NSG.
- `uplink_type`: Indicates the uplink type associated with the instance of Uplink Route Distinguisher.
- `external_id`: External object ID. Used for integration with third party systems

### 165.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas

### 165.3 Parents

- *nudomain.NUDomain*
- *nul2domain.NUL2Domain*
- *nume.NUMe*





## NUUSER

`nuuser.NUUser(bambou.nurest_object.NUMetaRESTObject, ) :`

Object that identifies the user functions.

## 166.1 Attributes

- `management_mode`: Management mode of the user object - allows for override of external authorization and syncup
- `password` (**Mandatory**): User password stored as a hash (SHA-1 encrypted)
- `last_name` (**Mandatory**): Last name of the user
- `last_updated_by`: ID of the user who last updated the object.
- `first_name` (**Mandatory**): First name of the user
- `disabled`: Status of the user account; true=disabled, false=not disabled; default value = false
- `email` (**Mandatory**): Email address of the user
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `mobile_number`: Mobile Number of the user
- `user_name` (**Mandatory**): Unique Username of the user. Valid characters are alphabets, numbers and hyphen(-).
- `avatar_data`: URL to the avatar data associated with the enterprise. If the avatarType is URL then value of avatarData should an URL of the image. If the avatarType BASE64 then avatarData should be BASE64 encoded value of the image
- `avatar_type`: Avatar type.
- `external_id`: External object ID. Used for integration with third party systems

## 166.2 Children

<b>class</b>	<b>fetcher</b>
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuvm.NUVM</i>	vms
<i>nucontainer.NUContainer</i>	containers
<i>nugroup.NUGroup</i>	groups
<i>nuavatar.NUAvatar</i>	avatars
<i>nueventlog.NUEventLog</i>	event_logs

## 166.3 Parents

- *nugroup.NUGroup*
- *nume.NUMe*
- *nuenterpise.NUEnterprise*

## NUVCENTER

`nuvcenter.NUVCenter(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents a VCenter.

### 167.1 Attributes

- `vrs_configuration_time_limit`: The maximum wait time limit in minutes to get VRS configured at cluster level
- `v_require_nuage_metadata`: Whether split-activation or not (Openstack/CloudStack)
- `name` (**Mandatory**): Name of the VCenter
- `password` (**Mandatory**): Password for the VCenter user
- `last_updated_by`: ID of the user who last updated the object.
- `data_dns1`: Data DNS 1
- `data_dns2`: Data DNS 2
- `data_gateway`: Data Gateway
- `data_network_portgroup`: Data Network Port Group
- `datapath_sync_timeout`: Datapath Sync Timeout
- `secondary_nuage_controller`: IP address of the secondary Controller (VSC)
- `generic_split_activation`: Whether split-activation is needed from VRO
- `separate_data_network`: Whether Data will use the management network or not
- `personality`: VRS/VRS-G
- `description`: Description of the VCenter
- `destination_mirror_port`: Extra Vnic to mirror access port
- `metadata_server_ip`: Metadata Server IP
- `metadata_server_listen_port`: Metadata Server Listen Port
- `metadata_server_port`: Metadata Server Port
- `metadata_service_enabled`: Metadata Service Enabled
- `network_uplink_interface`: Network Uplink Interface to support PAT/NAT with no tunnels on VRS-VM

- `network_uplink_interface_gateway`: Network Uplink Interface Gateway
- `network_uplink_interface_ip`: Ip Address to support PAT/NAT with no tunnels on VRS-VM
- `network_uplink_interface_netmask`: Network Uplink Interface Netmask
- `nfs_log_server`: IP address of NFS server to send the VRS log
- `nfs_mount_path`: Location to mount the NFS server
- `mgmt_dns1`: DNS server 1
- `mgmt_dns2`: DNS server 2
- `mgmt_gateway`: Gateway for the IP address
- `mgmt_network_portgroup`: Management Network Port group
- `dhcp_relay_server`: To provide IP address of the interface from which you will connect to the DHCP relay server
- `mirror_network_portgroup`: Mirror Port Group Name
- `site_id`: Site ID field for object profiles to support VSD Geo-redundancy
- `old_agency_name`: Old Agency Name
- `allow_data_dhcp`: Whether to get the Data IP for the VRS VM from DHCP or statically
- `allow_mgmt_dhcp`: Whether to get the management IP for the VRS VM from DHCP or statically
- `flow_eviction_threshold`: Flow Eviction Threshold
- `vm_network_portgroup`: VM Network Port Group Name
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `connection_status`: VCenter connection status.
- `portgroup_metadata`: Port Group Meta data
- `host_level_management`: Flag to say if host level management is enabled
- `nova_client_version`: Nova client Version
- `nova_metadata_service_auth_url`: Nova metadata service auth url
- `nova_metadata_service_endpoint`: Nova metadata service endpoint
- `nova_metadata_service_password`: Nova metadata service password
- `nova_metadata_service_tenant`: Nova metadata service tenant
- `nova_metadata_service_username`: Nova metadata service username
- `nova_metadata_shared_secret`: Nova metadata shared secret
- `nova_region_name`: Nova region name
- `ip_address` (**Mandatory**): IP Address of the VCenter
- `primary_nuage_controller`: IP address of the primary Controller (VSC)
- `vrs_config_id`: The ID of the template that this Port was created from
- `vrs_password`: VRS password to be used by toolbox to communicate with VRS
- `vrs_user_name`: VRS user name to be used by toolbox to communicate with VRS
- `user_name` (**Mandatory**): User name of the VCenter

- `static_route`: static route to be configured in the VRS
- `static_route_gateway`: Gateway for the static route given above
- `static_route_netmask`: Nova region name
- `ntp_server1`: IP of the NTP server 1
- `ntp_server2`: IP of the NTP server 1
- `http_port`: Http proxy port for VCenter
- `https_port`: Https proxy port of the VCenter
- `mtu`: Maximum Transmission Unit for eth2 interface
- `multi_vmssupport`: Whether Multi VM is to be used or not
- `multicast_receive_interface`: Multicast Receive Interface
- `multicast_receive_interface_ip`: IP address for eth3 interface
- `multicast_receive_interface_netmask`: Multicast Interface netmask
- `multicast_receive_range`: Allowed Range to receive the Multicast traffic from
- `multicast_send_interface`: Multicast Send Interface
- `multicast_send_interface_ip`: IP address for eth3 interface
- `multicast_send_interface_netmask`: Multicast Interface netmask
- `multicast_source_portgroup`: Multi Cast Source Port Group Name
- `customized_script_url`: To provide a URL to install a custom app on VRS
- `auto_resolve_frequency`: Frequency at which VCenter issues are to be resolved
- `ovf_url`: The url for the ovf
- `external_id`: External object ID. Used for integration with third party systems

## 167.2 Children

class	fetcher
<i>nuvcenterdatacenter.NUVCenterDataCenter</i>	<code>vcenter_data_centers</code>
<i>numetadata.NUMetadata</i>	<code>metadatas</code>
<i>nuglobalmetadata.NUGlobalMetadata</i>	<code>global_metadatas</code>
<i>nuijob.NUIJob</i>	<code>jobs</code>
<i>nuvrsaddressrange.NUVRSAddressRange</i>	<code>vrs_address_ranges</code>
<i>nuvrsredemptionpolicy.NUVRSRedemptionpolicy</i>	<code>vrs_redemptionpolicies</code>
<i>nuautodiscovereddatacenter.NUAutodiscovereddatacenter</i>	<code>autodiscovereddatacenters</code>

## 167.3 Parents

- *nume.NUMe*



## NUVCENTERCLUSTER

`nuvcentercluster.NUVCenterCluster(bambou.nurest_object.NUMetaRESTObject,):`

VCenter Clusters.

### 168.1 Attributes

- `vrs_configuration_time_limit`: The maximum wait time limit in minutes to get VRS configured at cluster level
- `v_require_nuage_metadata`: Whether split-activation or not (Openstack/CloudStack)
- `name` (**Mandatory**): Name of the Cluster
- `managed_object_id`: VCenter Managed Object ID of the Cluster.
- `last_updated_by`: ID of the user who last updated the object.
- `data_dns1`: Data DNS 1
- `data_dns2`: Data DNS 2
- `data_gateway`: Data Gateway
- `data_network_portgroup`: Data Network Port Group
- `datapath_sync_timeout`: Datapath Sync Timeout
- `scope`: Cluster in scope or not in scope.
- `secondary_nuage_controller`: IP address of the secondary Controller (VSC)
- `deleted_from_vcenter_data_center`: Set to true if the cluster is deleted from Vcenter
- `generic_split_activation`: Whether split-activation is needed from VRO
- `separate_data_network`: Whether Data will use the management network or not
- `personality`: VRS/VRS-G
- `description`: Description of the Cluster
- `destination_mirror_port`: Extra Vnic to mirror access port
- `metadata_server_ip`: Metadata Server IP
- `metadata_server_listen_port`: Metadata Server Listen Port
- `metadata_server_port`: Metadata Server Port
- `metadata_service_enabled`: Metadata Service Enabled

- `network_uplink_interface`: Network Uplink Interface to support PAT/NAT with no tunnels on VRS-VM
- `network_uplink_interface_gateway`: Network Uplink Interface Gateway
- `network_uplink_interface_ip`: Ip Address to support PAT/NAT with no tunnels on VRS-VM
- `network_uplink_interface_netmask`: Network Uplink Interface Netmask
- `nfs_log_server`: IP address of NFS server to send the VRS log
- `nfs_mount_path`: Location to mount the NFS server
- `mgmt_dns1`: DNS server 1
- `mgmt_dns2`: DNS server 2
- `mgmt_gateway`: Gateway for the IP address
- `mgmt_network_portgroup`: Management Network Port group
- `dhcp_relay_server`: To provide IP address of the interface from which you will connect to the DHCP relay server
- `mirror_network_portgroup`: Mirror Port Group Name
- `site_id`: Site ID field for object profiles to support VSD Geo-redundancy
- `allow_data_dhcp`: Whether to get the Data IP for the VRS VM from DHCP or statically
- `allow_mgmt_dhcp`: Whether to get the management IP for the VRS VM from DHCP or statically.
- `flow_eviction_threshold`: Flow Eviction Threshold
- `vm_network_portgroup`: VM Network Port Group Name
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `portgroup_metadata`: Port Group Meta data
- `nova_client_version`: Nova client Version
- `nova_metadata_service_auth_url`: Nova metadata service auth url
- `nova_metadata_service_endpoint`: Nova metadata service endpoint
- `nova_metadata_service_password`: Nova metadata service password
- `nova_metadata_service_tenant`: Nova metadata service tenant
- `nova_metadata_service_username`: Nova metadata service username
- `nova_metadata_shared_secret`: Nova metadata shared secret
- `nova_region_name`: Nova region name
- `primary_nuage_controller`: IP address of the primary Controller (VSC)
- `vrs_password`: VRS password to be used by toolbox to communicate with VRS
- `vrs_user_name`: VRS user name to be used by toolbox to communicate with VRS
- `assoc_vcenter_data_center_id`: The ID of the vcenter to which this host is attached
- `assoc_vcenter_id`: ID of the associated VCenter.
- `static_route`: static route to be configured in the VRS
- `static_route_gateway`: Gateway for the static route given above
- `static_route_netmask`: Nova region name



- `ntp_server1`: IP of the NTP server 1
- `ntp_server2`: IP of the NTP server 1
- `mtu`: Maximum Transmission Unit for eth2 interface
- `multi_vmssupport`: Whether Multi VM is to be used or not
- `multicast_receive_interface`: Multicast Receive Interface
- `multicast_receive_interface_ip`: IP address for eth3 interface
- `multicast_receive_interface_netmask`: Multicast Interface netmask
- `multicast_receive_range`: Allowed Range to receive the Multicast traffic from
- `multicast_send_interface`: Multicast Send Interface
- `multicast_send_interface_ip`: IP address for eth3 interface
- `multicast_send_interface_netmask`: Multicast Interface netmask
- `multicast_source_portgroup`: Multi Cast Source Port Group Name
- `customized_script_url`: To provide a URL to install a custom app on VRS
- `ovf_url`: ovf url
- `external_id`: External object ID. Used for integration with third party systems

## 168.2 Children

class	fetcher
<i>nuvcenterhypervisor.NUVCenterHypervisor</i>	<code>vcenter_hypervisors</code>
<i>numetadata.NUMetadata</i>	<code>metadatas</code>
<i>nuglobalmetadata.NUGlobalMetadata</i>	<code>global_metadatas</code>
<i>nujob.NUJob</i>	<code>jobs</code>
<i>nuvrsaddressrange.NUVRSAddressRange</i>	<code>vrs_address_ranges</code>
<i>nuvrsredemptionpolicy.NUVRSRedemptionpolicy</i>	<code>vrs_redemptionpolicies</code>
<i>nuautodiscoverhypervisorfromcluster.NUAutoDiscoverHypervisorFromCluster</i>	<code>auto_discover_hypervisor_from_clusters</code>

## 168.3 Parents

- *nuvcenterdatacenter.NUVCenterDataCenter*



## NUVCENTERDATACENTER

`nuvcenterdatacenter.NUVCenterDataCenter(bambou.nurest_object.NUMetaRESTObject,):`  
VCenter DataCenters.

### 169.1 Attributes

- `vrs_configuration_time_limit`: The maximum wait time limit in minutes to get VRS configured at cluster level
- `v_require_nuage_metadata`: Whether split-activation or not (Openstack/CloudStack)
- `name` (**Mandatory**): Name of the Datacenter
- `managed_object_id`: VCenter Managed Object ID of the Datacenter.
- `last_updated_by`: ID of the user who last updated the object.
- `data_dns1`: Data DNS 1
- `data_dns2`: Data DNS 2
- `data_gateway`: Data Gateway
- `data_network_portgroup`: Data Network Port Group
- `datapath_sync_timeout`: Datapath Sync Timeout
- `secondary_nuage_controller`: IP address of the secondary Controller (VSC)
- `deleted_from_vcenter`: Set to true if the datacenter is deleted from Vcenter
- `generic_split_activation`: Whether split-activation is needed from VRO
- `separate_data_network`: Whether Data will use the management network or not
- `personality`: VRS/VRS-G
- `description`: Description of the Datacenter
- `destination_mirror_port`: Extra Vnic to mirror access port
- `metadata_server_ip`: Metadata Server IP
- `metadata_server_listen_port`: Metadata Server Listen Port
- `metadata_server_port`: Metadata Server Port
- `metadata_service_enabled`: Metadata Service Enabled

- `network_uplink_interface`: Network Uplink Interface to support PAT/NAT with no tunnels on VRS-VM
- `network_uplink_interface_gateway`: Network Uplink Interface Gateway
- `network_uplink_interface_ip`: Ip Address to support PAT/NAT with no tunnels on VRS-VM
- `network_uplink_interface_netmask`: Network Uplink Interface Netmask
- `nfs_log_server`: IP address of NFS server to send the VRS log
- `nfs_mount_path`: Location to mount the NFS server
- `mgmt_dns1`: DNS server 1
- `mgmt_dns2`: DNS server 2
- `mgmt_gateway`: Gateway for the IP address
- `mgmt_network_portgroup`: Management Network Port group
- `dhcp_relay_server`: To provide IP address of the interface from which you will connect to the DHCP relay server
- `mirror_network_portgroup`: Mirror Port Group Name
- `site_id`: Site ID field for object profiles to support VSD Geo-redundancy
- `allow_data_dhcp`: Whether to get the Data IP for the VRS VM from DHCP or statically
- `allow_mgmt_dhcp`: Whether to get the management IP for the VRS VM from DHCP or statically
- `flow_eviction_threshold`: Flow Eviction Threshold
- `vm_network_portgroup`: VM Network Port Group Name
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `portgroup_metadata`: Port Group Meta data
- `nova_client_version`: Nova client Version
- `nova_metadata_service_auth_url`: Nova metadata service auth url
- `nova_metadata_service_endpoint`: Nova metadata service endpoint
- `nova_metadata_service_password`: Nova metadata service password
- `nova_metadata_service_tenant`: Nova metadata service tenant
- `nova_metadata_service_username`: Nova metadata service username
- `nova_metadata_shared_secret`: Nova metadata shared secret
- `nova_region_name`: Nova region name
- `primary_nuage_controller`: IP address of the primary Controller (VSC)
- `vrs_password`: VRS password to be used by toolbox to communicate with VRS
- `vrs_user_name`: VRS user name to be used by toolbox to communicate with VRS
- `associated_vcenter_id`: The ID of the vcenter to which this host is attached
- `static_route`: static route to be configured in the VRS
- `static_route_gateway`: Gateway for the static route given above
- `static_route_netmask`: Nova region name
- `ntp_server1`: IP of the NTP server 1

- `ntp_server2`: IP of the NTP server 1
- `mtu`: Maximum Transmission Unit for eth2 interface
- `multi_vmssupport`: Whether Multi VM is to be used or not
- `multicast_receive_interface`: Multicast Receive Interface
- `multicast_receive_interface_ip`: IP address for eth3 interface
- `multicast_receive_interface_netmask`: Multicast Interface netmask
- `multicast_receive_range`: Allowed Range to receive the Multicast traffic from
- `multicast_send_interface`: Multicast Send Interface
- `multicast_send_interface_ip`: IP address for eth3 interface
- `multicast_send_interface_netmask`: Multicast Interface netmask
- `multicast_source_portgroup`: Multi Cast Source Port Group Name
- `customized_script_url`: To provide a URL to install a custom app on VRS
- `ovf_url`: ovf url
- `external_id`: External object ID. Used for integration with third party systems

## 169.2 Children

class	fetcher
<i>nuvcentercluster.NUVCenterCluster</i>	<code>vcenter_clusters</code>
<i>nuvcenterhypervisor.NUVCenterHypervisor</i>	<code>vcenter_hypervisors</code>
<i>numetadata.NUMetadata</i>	<code>metadatas</code>
<i>nuglobalmetadata.NUGlobalMetadata</i>	<code>global_metadatas</code>
<i>nuvrsaddressrange.NUVRSAddressRange</i>	<code>vrs_address_ranges</code>
<i>nuvrsredemptionpolicy.NUVRSRedemptionpolicy</i>	<code>vrs_redemptionpolicies</code>
<i>nuautodiscovercluster.NUAutoDiscoverCluster</i>	<code>auto_discover_clusters</code>
<i>nuautodiscoverhypervisorfromcluster.NUAutoDiscoverHypervisorFromCluster</i>	<code>auto_discover_hypervisor_from_clusters</code>

## 169.3 Parents

- *nuvcenter.NUVCenter*



## NUVCENTEREAMCONFIG

`nuvcentereamconfig.NUVCenterEAMConfig(bambou.nurest_object.NUMetaRESTObject, ) :`

The EAM solution configuration.

### 170.1 Attributes

- `eam_server_ip` (**Mandatory**): The EAM server IP
- `eam_server_port_number` (**Mandatory**): The EAM server port number
- `eam_server_port_type` (**Mandatory**): The EAM server port Type
- `last_updated_by`: ID of the user who last updated the object.
- `vib_url`: The url for the optional vib
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `ovf_url` (**Mandatory**): The url for the ovf
- `extension_key`: Key of the extension that the solution registers
- `external_id`: External object ID. Used for integration with third party systems

### 170.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 170.3 Parents

- `nume.NUMe`





## NUVCENTERHYPERVISOR

`nuvcenterhypervisor.NUVCenterHypervisor(bambou.nurest_object.NUMetaRESTObject,):`  
Host or Hypervisors.

### 171.1 Attributes

- `vcenter_ip`: IP Address of the VCenter.
- `vcenter_password`: Password for VCenter.
- `vcenter_user`: Username for VCenter.
- `vrs_configuration_time_limit`: The maximum wait time limit in minutes to get VRS configured at cluster level
- `vrs_metrics_id`: ID of the VRS metrics object.
- `vrs_state`: Current state of the VRS VM on the hypervisor
- `v_require_nuage_metadata`: Whether split-activation or not (Openstack/CloudStack)
- `name` (**Mandatory**): Name of the Hypervisor
- `managed_object_id`: managed Object ID of hypervisor
- `last_updated_by`: ID of the user who last updated the object.
- `last_vrs_deployed_date`: Determines the time the vrs vm was last deployed.
- `data_dns1`: Data DNS 1
- `data_dns2`: Data DNS 2
- `data_gateway`: Data Gateway
- `data_ip_address`: Data IP Address
- `data_netmask`: Data NetMask
- `data_network_portgroup` (**Mandatory**): Data Network Port Group
- `datapath_sync_timeout`: Datapath Sync Timeout
- `scope`: Cluster in scope or not in scope.
- `secondary_nuage_controller`: IP address of the secondary Controller (VSC)
- `removed_from_vcenter_inventory`: Set to true if the hypervisor is removed from Vcenter inventory datacenter or cluster

- `generic_split_activation`: Whether split-activation is needed from VRO
- `separate_data_network`: Whether Data will use the management network or not
- `deployment_count`: The number of times the vrs was deployed on this hypervisor
- `personality`: VRS/VRS-G
- `description` (**Mandatory**): Description of the Hypervisor
- `destination_mirror_port`: Extra Vnic to mirror access port
- `metadata_server_ip`: Metadata Server IP
- `metadata_server_listen_port`: Metadata Server Listen Port
- `metadata_server_port`: Metadata Server Port
- `metadata_service_enabled`: Metadata Service Enabled
- `network_uplink_interface`: Network Uplink Interface to support PAT/NAT with no tunnels on VRS-VM
- `network_uplink_interface_gateway`: Network Uplink Interface Gateway
- `network_uplink_interface_ip`: Ip Address to support PAT/NAT with no tunnels on VRS-VM
- `network_uplink_interface_netmask`: Network Uplink Interface Netmask
- `nfs_log_server`: IP address of NFS server to send the VRS log
- `nfs_mount_path`: Location to mount the NFS server
- `mgmt_dns1`: DNS server 1
- `mgmt_dns2`: DNS server 2
- `mgmt_gateway`: Gateway for the IP address
- `mgmt_ip_address`: The Management IP address for VRS VM if needed to be given statically
- `mgmt_netmask`: Netmask of the IP address above
- `mgmt_network_portgroup` (**Mandatory**): Management Network Port group
- `dhcp_relay_server`: To provide IP address of the interface from which you will connect to the DHCP relay server
- `mirror_network_portgroup`: Mirror Port Group Name
- `site_id`: Site ID field for object profiles to support VSD Geo-redundancy
- `allow_data_dhcp`: Whether to get the Data IP for the VRS VM from DHCP or statically
- `allow_mgmt_dhcp`: Whether to get the management IP for the VRS VM from DHCP or statically
- `flow_eviction_threshold`: Flow Eviction Threshold
- `vm_network_portgroup` (**Mandatory**): VM Network Port Group Name
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `toolbox_deployment_mode`: Flag to specify if VRS is deployed using tool box.
- `toolbox_group`: Deployment Toolbox Group.
- `toolbox_ip`: Deployment Toolbox IP.
- `toolbox_password`: Deployment Toolbox password.
- `toolbox_user_name`: Deployment Toolbox username.

- portgroup\_metadata: Port Group Meta data
- nova\_client\_version: Nova client Version
- nova\_metadata\_service\_auth\_url: Nova metadata service auth url
- nova\_metadata\_service\_endpoint: Nova metadata service endpoint
- nova\_metadata\_service\_password: Nova metadata service password
- nova\_metadata\_service\_tenant: Nova metadata service tenant
- nova\_metadata\_service\_username: Nova metadata service username
- nova\_metadata\_shared\_secret: Nova metadata shared secret
- nova\_region\_name: Nova region name
- primary\_nuage\_controller: IP address of the primary Controller (VSC)
- vrs\_id: VCenter Name or Id used by toolbox to identify the VRS virtual machine
- vrs\_password: VRS password to be used by toolbox to communicate with VRS
- vrs\_user\_name: VRS user name to be used by toolbox to communicate with VRS
- static\_route: static route to be configured in the VRS
- static\_route\_gateway: Gateway for the static route given above
- static\_route\_netmask: Nova region name
- ntp\_server1: IP of the NTP server 1
- ntp\_server2: IP of the NTP server 1
- mtu: Maximum Transmission Unit for eth2 interface
- multi\_vmssupport: Whether Multi VM is to be used or not
- multicast\_receive\_interface: Multicast Receive Interface
- multicast\_receive\_interface\_ip: IP address for eth3 interface
- multicast\_receive\_interface\_netmask: Multicast Interface netmask
- multicast\_receive\_range: Allowed Range to receive the Multicast traffic from
- multicast\_send\_interface: Multicast Send Interface
- multicast\_send\_interface\_ip: IP address for eth3 interface
- multicast\_send\_interface\_netmask: Multicast Interface netmask
- multicast\_source\_portgroup: Multi Cast Source Port Group Name
- customized\_script\_url: To provide a URL to install a custom app on VRS
- available\_networks: List of the available network list for the hypervisor.
- ovf\_url: ovf url
- external\_id: External object ID. Used for integration with third party systems
- hypervisor\_ip (**Mandatory**): IP Address of the Hypervisor
- hypervisor\_password (**Mandatory**): Hypervisor username
- hypervisor\_user (**Mandatory**): Hypervisor username

## 171.2 Children

<b>class</b>	<b>fetcher</b>
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nujob.NUJob</i>	jobs
<i>nuvrsaddressrange.NUVRSAddressRange</i>	vrs_address_ranges
<i>nuvrsmetrics.NUVRSMetrics</i>	vrs_metrics
<i>nuvrsredemptionpolicy.NUVRSRedemptionpolicy</i>	vrs_redemptionpolicies

## 171.3 Parents

- *nuvcentercluster.NUVCenterCluster*
- *nuvcenterdatacenter.NUVCenterDataCenter*
- *nume.NUMe*

## NUVCENTERVRSCONFIG

`nuvcentervrsconfig.NUVCenterVRSConfig(bambou.nurest_object.NUMetaRESTObject,):`

Default VRS Configuration parameters

### 172.1 Attributes

- `v_require_nuage_metadata`: Whether split-activation or not (Openstack/CloudStack)
- `last_updated_by`: ID of the user who last updated the object.
- `data_dns1`: Data DNS 1
- `data_dns2`: Data DNS 2
- `data_gateway`: Data Gateway
- `data_network_portgroup`: Data Network Port Group
- `datapath_sync_timeout`: Datapath Sync Timeout
- `secondary_nuage_controller`: IP address of the secondary Controller (VSC)
- `generic_split_activation`: Whether split-activation is needed from VRO
- `separate_data_network`: Whether Data will use the management network or not
- `personality`: VRS/VRS-G
- `metadata_server_ip`: Metadata Server IP
- `metadata_server_listen_port`: Metadata Server Listen Port
- `metadata_server_port`: Metadata Server Port
- `metadata_service_enabled`: Metadata Service Enabled
- `network_uplink_interface`: Network Uplink Interface to support PAT/NAT with no tunnels on VRS-VM
- `network_uplink_interface_gateway`: Network Uplink Interface Gateway
- `network_uplink_interface_ip`: Ip Address to support PAT/NAT with no tunnels on VRS-VM
- `network_uplink_interface_netmask`: Network Uplink Interface Netmask
- `nfs_log_server`: IP address of NFS server to send the VRS log
- `nfs_mount_path`: Location to mount the NFS server
- `mgmt_dns1`: DNS server 1

- `mgmt_dns2`: DNS server 2
- `mgmt_gateway`: Gateway for the IP address
- `mgmt_network_portgroup`: Management Network Port group
- `dhcp_relay_server`: To provide IP address of the interface from which you will connect to the DHCP relay server
- `site_id`: Site ID field for object profiles to support VSD Geo-redundancy
- `allow_data_dhcp`: Whether to get the Data IP for the VRS VM from DHCP or statically
- `allow_mgmt_dhcp`: Whether to get the management IP for the VRS VM from DHCP or statically
- `flow_eviction_threshold`: Flow Eviction Threshold
- `vm_network_portgroup`: VM Network Port Group Name
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `portgroup_metadata`: Port Group Meta data
- `nova_client_version`: Nova client Version
- `nova_metadata_service_auth_url`: Nova metadata service auth url
- `nova_metadata_service_endpoint`: Nova metadata service endpoint
- `nova_metadata_service_password`: Nova metadata service password
- `nova_metadata_service_tenant`: Nova metadata service tenant
- `nova_metadata_service_username`: Nova metadata service username
- `nova_metadata_shared_secret`: Nova metadata shared secret
- `nova_region_name`: Nova region name
- `primary_nuage_controller`: IP address of the primary Controller (VSC)
- `vrs_password`: VRS password to be used by toolbox to communicate with VRS
- `vrs_user_name`: VRS user name to be used by toolbox to communicate with VRS
- `static_route`: static route to be configured in the VRS
- `static_route_gateway`: Gateway for the static route given above
- `static_route_netmask`: Nova region name
- `ntp_server1`: IP of the NTP server 1
- `ntp_server2`: IP of the NTP server 1
- `mtu`: Maximum Transmission Unit for eth2 interface
- `multi_vmssupport`: Whether Multi VM is to be used or not
- `multicast_receive_interface`: Multicast Receive Interface
- `multicast_receive_interface_ip`: IP address for eth3 interface
- `multicast_receive_interface_netmask`: Multicast Interface netmask
- `multicast_receive_range`: Allowed Range to receive the Multicast traffic from
- `multicast_send_interface`: Multicast Send Interface
- `multicast_send_interface_ip`: IP address for eth3 interface

- `multicast_send_interface_netmask`: Multicast Interface netmask
- `multicast_source_portgroup`: Multi Cast Source Port Group Name
- `customized_script_url`: To provide a URL to install a custom app on VRS
- `external_id`: External object ID. Used for integration with third party systems

## 172.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nuvrsaddressrange.NUVRSAddressRange</code>	<code>vrs_address_ranges</code>
<code>nuvrsredemptionpolicy.NUVRSRedemptionpolicy</code>	<code>vrs_redemptionpolicies</code>

## 172.3 Parents

- `nume.NUMe`





## NUVIA

`nuvia.NUVia(bambou.nurest_object.NUMetaRESTObject,):`

None

### 173.1 Attributes

- `next_hops`: A set of NextHop objects. A NextHop can be either an InetAddress (IPv4 or IPV6) address or a VLAN ID (for NSGBR)



## NUVIRTUALIP

`nuvirtualip.NUVirtualIP(bambou.nurest_object.NUMetaRESTObject,):`

Virtual IP address.

### 174.1 Attributes

- `mac`: The MAC address of the virtual port
- `last_updated_by`: ID of the user who last updated the object.
- `virtual_ip` (**Mandatory**): Virtual IP address
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `associated_floating_ip_id`: Id of Floating IP address associated to this virtual ip
- `subnet_id`: Id of subnet to which this ip address belongs
- `external_id`: External object ID. Used for integration with third party systems

### 174.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>

### 174.3 Parents

- `nuredirectiontarget.NURedirectionTarget`
- `nuvport.NUVPort`
- `nusubnet.NUSubnet`



## NUVLAN

`nuvlan.NUVLAN(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents VLAN object under a given Port object.

### 175.1 Attributes

- **value (Mandatory):** value of VLAN
- **last\_updated\_by:** ID of the user who last updated the object.
- **gateway\_id:** The Gateway associated with this VLAN . This is a read only attribute
- **readonly:** Determines whether this entity is read only. Read only objects cannot be modified or deleted.
- **template\_id:** The ID of the template that this Port was created from
- **permitted\_action:** The permitted action to USE/EXTEND this Gateway.
- **description:** A description of the Port
- **restricted:** Determines whether this entity can be used in associations with other properties.
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **vport\_id:** The Vport associated with this VLAN . This is a read only attribute
- **use\_user\_mnemonic:** determines whether to use user mnemonic of the Port
- **user\_mnemonic:** user mnemonic of the Port
- **associated\_bgp\_profile\_id:** The ID of the associated BGP profile
- **associated\_egress\_qos\_policy\_id:** ID of the Egress QOS Policy associated with this VLAN.
- **associated\_uplink\_connection\_id:** Associated uplink connection ID
- **associated\_vsc\_profile\_id:** The associated VSC profile for the uplink VLANS. This should be only be valid for the uplinks
- **status:** Status of the VLAN.
- **duc\_vlan:** When set to true, this specifies that this VLAN instance serves as an underlay connection endpoint on an NSG-UBR gateway.
- **external\_id:** External object ID. Used for integration with third party systems

## 175.2 Children

<b>class</b>	<b>fetcher</b>
<i>nupatnatpool.NUPATNATPool</i>	patnat_pools
<i>nupermision.NUPermission</i>	permissions
<i>numetadata.NUMetadata</i>	metadatas
<i>nubgpneighbor.NUBGPNeighbor</i>	bgp_neighbors
<i>nuikegatewayconnection.NUIKEGatewayConnection</i>	ike_gateway_connections
<i>nualarm.NUAlarm</i>	alarms
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuenterprisepermission.NUEnterprisePermission</i>	enterprise_permissions
<i>nuuplinkconnection.NUUplinkConnection</i>	uplink_connections
<i>nubrconnection.NUBRConnection</i>	br_connections
<i>nueventlog.NUEventLog</i>	event_logs

## 175.3 Parents

- *nuvsgredundantport.NUVsgRedundantPort*
- *nuredundantport.NURedundantPort*
- *nuport.NUPort*
- *nunsport.NUNSPort*

## NUVLANTEMPLATE

`nuvlantemplate.NUVLANTemplate(bambou.nurest_object.NUMetaRESTObject,)` :

Represents VLAN Template under a Port Template object.

### 176.1 Attributes

- `value`: value of VLAN
- `last_updated_by`: ID of the user who last updated the object.
- `description`: A description of the Port
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `associated_egress_qos_policy_id`: ID of the Egress QOS Policy associated with this Vlan.
- `associated_vsc_profile_id`: The ID of the infrastructure VSC profile this is associated with this instance of a vlan or vlan template.
- `duc_vlan`: When set to true, this specifies that this VLAN template instance serves as an underlay connection endpoint on an NSG-UBR gateway.
- `external_id`: External object ID. Used for integration with third party systems

### 176.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nuuplinkconnection.NUUplinkConnection</code>	<code>uplink_connections</code>
<code>nubrconnection.NUBRConnection</code>	<code>br_connections</code>

### 176.3 Parents

- `nunsporttemplate.NUNSPortTemplate`
- `nuporttemplate.NUPortTemplate`





## NUVM

**nuvm.NUVM(bambou.nurest\_object.NUMetaRESTObject,):**

Read only API that can retrieve the VMs associated with a domain, zone or subnet for mediation created VM's for REST created VM's you need to set the additional proxy user header in http request : X-Nuage-ProxyUser value of the header has to be either : 1) `enterpriseName@UserName` (example : Alcatel `Lucent@bob`), or 2) external ID of user in VSD, typically is UUID generally decided by the CMS tool in question User needs to have CMS privileges to use proxy user header.

### 177.1 Attributes

- `l2_domain_ids`: Array of IDs of the l2 domain that the VM is connected to
- `vrsid`: Id of the VRS that this VM is attached to.
- `uuid` (**Mandatory**): UUID of the VM
- `name` (**Mandatory**): Name of the VM
- `last_updated_by`: ID of the user who last updated the object.
- `reason_type`: Reason of the event associated with the VM.
- `delete_expiry`: reflects the VM Deletion expiry timer in secs , deleteMode needs to be non-null value for deleteExpiry to be taken in to effect. CMS created VM's will always have deleteMode set to TIMER
- `delete_mode`: reflects the mode of VM Deletion - TIMER Possible values are TIMER, .
- `resync_info`: Information of the status of the resync operation of a VM
- `site_identifier`: This property specifies the site the VM belongs to, for Geo-redundancy.
- `interfaces`: List of VM interfaces associated with the VM
- `enterprise_id`: ID of the enterprise that this VM belongs to
- `enterprise_name`: Name of the enterprise that this VM belongs to
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `domain_ids`: Array of IDs of the domain that the VM is connected to
- `zone_ids`: Array of IDs of the zone that this VM is attached to
- `orchestration_id`: Orchestration ID
- `user_id`: ID of the user that created this VM
- `user_name`: Username of the user that created this VM

- `status`: Status of the VM.
- `subnet_ids`: Array of IDs of the subnets that the VM is connected to
- `external_id`: External object ID. Used for integration with third party systems
- `hypervisor_ip`: IP address of the hypervisor that this VM is currently running in

## 177.2 Children

<b>class</b>	<b>fetcher</b>
<i>nuvmresync.NUVMResync</i>	<code>vm_resyncs</code>
<i>numetadata.NUMetadata</i>	<code>metadatas</code>
<i>nualarm.NUAlarm</i>	<code>alarms</code>
<i>nuglobalmetadata.NUGlobalMetadata</i>	<code>global_metadatas</code>
<i>nuvminterface.NUVMInterface</i>	<code>vm_interfaces</code>
<i>nuvrs.NUVRS</i>	<code>vrss</code>
<i>nueventlog.NUEventLog</i>	<code>event_logs</code>

## 177.3 Parents

- *nuqos.NUQOS*
- *nuzone.NUZone*
- *nudomain.NUDomain*
- *nuvport.NUVPort*
- *nusubnet.NUSubnet*
- *nuvrs.NUVRS*
- *nutier.NUTier*
- *nul2domain.NUL2Domain*
- *nume.NUMe*
- *nuegressacltemplate.NUEgressACLTemplate*
- *nuuser.NUUser*
- *nuenterprise.NUEnterprise*
- *nuingressacltemplate.NUIngressACLTemplate*

## NUVMINTERFACE

**`nuvminterface.NUVMInterface(bambou.nurest_object.NUMetaRESTObject, ) :`**

Read only API that can retrieve the VM interface associated with a domain, zone or subnet for mediation created VM's for REST created VM interfaces you need to set the additional proxy header in http request : X-Nuage-ProxyUser value of the header has to be either :1) `enterpriseName@UserName` (example :`bob@Alcatel` Lucent), or 2) external ID of user in VSD, typically is UUID generally decided by the CMS tool in question User needs to have CMS privileges to use proxy user header.

### 178.1 Attributes

- `mac`: MAC address of the interface
- `vmuuid`: UUID of the associated virtual machine
- `ip_address`: IP address of the interface
- `vport_id`: ID of the vport that the interface is attached to
- `vport_name`: Name of the vport that the VM is attached to
- `name`: Device name associated with this interface
- `last_updated_by`: ID of the user who last updated the object.
- `gateway`: Gateway of the subnet that the VM is connected to
- `netmask`: Netmask of the subnet that the VM is attached to
- `network_name`: Name of the network that the VM is attached to
- `tier_id`: ID of the tier that the interface is attached to.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `policy_decision_id`: The policy decision ID for this particular interface
- `domain_id`: ID of the domain that the VM is attached to
- `domain_name`: Name of the domain that the VM is attached to
- `zone_id`: ID of the zone that the interface is attached to
- `zone_name`: Name of the zone that the VM is attached to
- `associated_floating_ip_address`: Floating Ip Address of this network interface eg: 10.1.2.1
- `attached_network_id`: ID of the l2 domain or Subnet that the VM is attached to
- `attached_network_type`: l2 domain or Subnet that the interface is attached to

- `multi_nic_vport_name`: Name of the Multi NIC VPort associated with this VM Interface
- `external_id`: External object ID. Used for integration with third party systems

## 178.2 Children

<b>class</b>	<b>fetcher</b>
<i>nutca.NUTCA</i>	<code>tcas</code>
<i>nuredirectiontarget.NURedirectionTarget</i>	<code>redirection_targets</code>
<i>numetadata.NUMetadata</i>	<code>metadatas</code>
<i>nudhcpoption.NUDHCPOption</i>	<code>dhcp_options</code>
<i>nuglobalmetadata.NUGlobalMetadata</i>	<code>global_metadatas</code>
<i>nupolicydecision.NUPolicyDecision</i>	<code>policy_decisions</code>
<i>nupolicygroup.NUPolicyGroup</i>	<code>policy_groups</code>
<i>nustaticroute.NUStaticRoute</i>	<code>static_routes</code>
<i>nustatistics.NUStatistics</i>	<code>statistics</code>
<i>numulticastchannelmap.NUMultiCastChannelMap</i>	<code>multi_cast_channel_maps</code>
<i>nueventlog.NUEventLog</i>	<code>event_logs</code>

## 178.3 Parents

- *nuzone.NUZone*
- *nudomain.NUDomain*
- *nuvport.NUVPort*
- *nusubnet.NUSubnet*
- *nul2domain.NUL2Domain*
- *nume.NUMe*
- *nuvm.NUVM*

## NUVMRESYNC

`nuvmresync.NUVMResync(bambou.nurest_object.NUMetaRESTObject, ) :`

Provide information about the state of a VM resync request.

## 179.1 Attributes

- `last_request_timestamp`: Time of the last timestamp received
- `last_time_resync_initiated`: Time that the resync was initiated
- `last_updated_by`: ID of the user who last updated the object.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `status`: Status of the resync
- `external_id`: External object ID. Used for integration with third party systems

## 179.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

## 179.3 Parents

- `nusubnet.NUSubnet`
- `nuvm.NUVM`



## NUVPNCONNECTION

`nuvpnconnection.NUVPNConnection(bambou.nurest_object.NUMetaRESTObject,)` :

This is the definition of a VPN Connect which holds the PE service association with a DOMAIN.

### 180.1 Attributes

- **name (Mandatory)**: Name of the VPNConnect
- **last\_updated\_by**: ID of the user who last updated the object.
- **description**: A description of the VPNConnect
- **entity\_scope**: Specify if scope of entity is Data center or Enterprise level
- **associated\_wan\_service\_id**: Associated WAN Service
- **external\_id**: External object ID. Used for integration with third party systems

### 180.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas

### 180.3 Parents

- *nusharednetworkresource.NUSharedNetworkResource*
- *nudomain.NUDomain*
- *nul2domain.NUL2Domain*





## NUVPORT

`nuvport.NUVPort(bambou.nurest_object.NUMetaRESTObject, ) :`

VPorts are a new level in the domain hierarchy, intended to provide more granular configuration than at subnet, and also support a split workflow, where the vPort is configured and associated with a VM port (or gateway port) before the port exists.

## 181.1 Attributes

- `vlanid`: associated Vlan of this vport - applicable for type host/bridge
- `dpi`: determines whether or not Deep packet inspection is enabled
- `name` (**Mandatory**): Name of the vport. Valid characters are alphabets, numbers, space and hyphen( - ).
- `has_attached_interfaces`: Indicates that this vport has attached interfaces
- `last_updated_by`: ID of the user who last updated the object.
- `active`: Indicates if this vport is up or down
- `address_spoofing` (**Mandatory**): Indicates if address spoofing is ENABLED/DISABLED/INHERITED for this vport Possible values are INHERITED, ENABLED, DISABLED, .
- `description`: Description for this vport
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `domain_id`: ID the Domain associated with the VPort
- `zone_id`: ID the Zone associated with the VPort
- `operational_state`: Operational State of the VPort - RUNNING/SHUTDOWN Possible values are INIT, UP, DOWN, .
- `associated_floating_ip_id`: Id of Floating IP address associated to this vport
- `associated_multicast_channel_map_id`: The ID of the receive Multicast Channel Map this Vport is associated with. This has to be set when enableMultiCast is set to ENABLED
- `associated_send_multicast_channel_map_id`: The ID of the send Multicast Channel Map this Vport is associated with. This has to be set when enableMultiCast is set to ENABLED
- `multi_nic_vport_id`: ID of the Multi NIC VPort associated with the VPort
- `multicast`: Indicates multicast policy on Vport.
- `external_id`: External object ID. Used for integration with third party systems

- **type (Mandatory):** Type of vport - possible values VM/HOST/BRIDGE Possible values are VM, HOST, BRIDGE, .
- **system\_type:** Indicates what system it is.

## 181.2 Children

class	fetcher
<i>nutca.NUTCA</i>	tcas
<i>nuredirectiontarget.NURedirectionTarget</i>	redirection_targets
<i>numetadata.NUMetadata</i>	metadatas
<i>nuaggregatemetadata.NUAggregateMetadata</i>	aggregate_metadatas
<i>nudhcpoption.NUDHCPOption</i>	dhcp_options
<i>nuvirtualip.NUVirtualIP</i>	virtual_ips
<i>nualarm.NUAlarm</i>	alarms
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuvm.NUVM</i>	vms
<i>nuvminterface.NUVMInterface</i>	vm_interfaces
<i>nupolicygroup.NUPolicyGroup</i>	policy_groups
<i>nucontainer.NUContainer</i>	containers
<i>nucontainerinterface.NUContainerInterface</i>	container_interfaces
<i>nuportmapping.NUPortMapping</i>	port_mappings
<i>nuqos.NUQOS</i>	qoss
<i>nuhostinterface.NUHostInterface</i>	host_interfaces
<i>nuvportmirror.NUVPortMirror</i>	vport_mirrors
<i>nuapplicationperformancemanagement.NUApplicationperformancemanagement</i>	applicationperformancemanagements
<i>nubridgeinterface.NUBridgeInterface</i>	bridge_interfaces
<i>nuvrs.NUVRS</i>	vrss
<i>nustatistics.NUStatistics</i>	statistics
<i>nustatisticspolicy.NUStatisticsPolicy</i>	statistics_policies
<i>nueventlog.NUEventLog</i>	event_logs

## 181.3 Parents

- *nuzone.NUZone*
- *nuredirectiontarget.NURedirectionTarget*
- *nudomain.NUDomain*
- *nusubnet.NUSubnet*
- *nuvrs.NUVRS*
- *nutier.NUTier*
- *numultinicvport.NUMultiNICVPort*
- *nul2domain.NUL2Domain*
- *nufloatingip.NUFloatingIp*
- *nupolicygroup.NUPolicyGroup*

## NUVPORTMIRROR

`nuvportmirror.NUVPortMirror(bambou.nurest_object.NUMetaRESTObject,):`

VPort Mirror represents the relationship between a vport and a mirror destination.

### 182.1 Attributes

- `vport_name`: Name of the vport to which the mirror destination is associated with.
- `last_updated_by`: ID of the user who last updated the object.
- `network_name`: Name of the network to which the vport belongs to
- `mirror_destination_id`: Destination ID of the mirror destination object.
- `mirror_destination_name`: Name of the mirror destination
- `mirror_direction`: Describes what type of traffic needs to be mirrored.
- `enterprise_name`: Enterprise to which the vport associated with the mirror destination belongs to.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `domain_name`: Domain name of the vport associated with the mirror destination
- `vport_id`: Id of the vport to which the mirror destination is associated with.
- `attached_network_type`: Type of the network attached - L2/L3
- `external_id`: External object ID. Used for integration with third party systems

### 182.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 182.3 Parents

- `nuvport.NUVPort`
- `numirrordestination.NUMirrorDestination`



## **NUVRS**

`nuvrs.NUVRS(bambou.nurest_object.NUMetaRESTObject, ) :`

System Monitoring details for VRS connected to VSC or HSC

### **183.1 Attributes**

- `jsonrpc_connection_state`: The current JSON RPC connection status.
- `name`: Identifies the entity with a name.
- `management_ip`: The management IP of the VRS entity
- `parent_ids`: Holds VRS controllers ids
- `last_event_name`: The last event name from the hypervisor.
- `last_event_object`: The last event object (including metadata) from the hypervisor.
- `last_event_timestamp`: The last event timestamp from the hypervisor.
- `last_state_change`: Last state change timestamp (in millis).
- `last_updated_by`: ID of the user who last updated the object.
- `db_synced`: Flag to indicate if the ovs database is synced between the NSG pair part of a redundant group
- `address`: The IP of the VRS entity
- `peak_cpuusage`: Peek CPU usage percentage.
- `peak_memory_usage`: Peek memory usage percentage.
- `peer`: The redundant peer id for the current VRS.
- `personality`: VRS personality.
- `description`: Description of the entity.
- `messages`: An array of degraded messages.
- `revert_behavior_enabled`: Flag to indicate if the revert behavior took place or not.
- `revert_completed`: Flag indicates whether revert was completed successfully.
- `revert_count`: Indicates the number of retries for the revert to take place.
- `revert_failed_count`: This value indicates the number of failed attempts for the revert to happen successfully.
- `licensed_state`: Licensed state.

- `disks`: Set of disk usage details.
- `cluster_node_role`: Indicate that the controller associated is primary, secondary or unknown.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `location`: Identifies the entity to be associated with a location.
- `role`: Flag to indicate that VRS-G redundancy state (active/standby/standalone). Only applicable for gateways.
- `uptime`: How long the VRS was up.
- `primary_vsc_connection_lost`: Flag indicates whether the connection with the primary is lost, which will help trigger alarms.
- `product_version`: Product version supported by this entity.
- `is_resilient`: Flag to indicate that the VRS is part of a redundant group.
- `vsc_config_state`: Indicates the configured state of the VSC.
- `vsc_current_state`: Indicates the current state of the VSC, which may or maybe not be same as the configured state.
- `status`: Computed status of the entity.
- `multi_nic_vport_enabled`: VRS is in Multi-NIC VPORT Mode
- `number_of_bridge_interfaces`: Number of bridge interfaces defined in this VRS.
- `number_of_containers`: Number of containers defined in this VRS.
- `number_of_host_interfaces`: Number of host interfaces defined in this VRS.
- `number_of_virtual_machines`: Number of VMs defined in this VRS.
- `current_cpuusage`: Current CPU usage percentage.
- `current_memory_usage`: Current memory usage percentage.
- `average_cpuusage`: Average CPU usage percentage.
- `average_memory_usage`: Average memory usage percentage.
- `external_id`: External object ID. Used for integration with third party systems
- `dynamic`: Flag to indicate it is dynamically configured or not.
- `hypervisor_connection_state`: The VRS connection state with the hypervisor.
- `hypervisor_identifier`: The hypervisor IP (or name) associated with the VRS.
- `hypervisor_name`: The hypervisor name associated with the VRS.
- `hypervisor_type`: The hypervisor type associated with the VRS.

## 183.2 Children

<b>class</b>	<b>fetcher</b>
<i>numetadata.NUMetadata</i>	metadatas
<i>nualarm.NUAlarm</i>	alarms
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuvm.NUVM</i>	vms
<i>nujob.NUJob</i>	jobs
<i>numonitoringport.NUMonitoringPort</i>	monitoring_ports
<i>nucontainer.NUContainer</i>	containers
<i>nuvport.NUVPort</i>	vports
<i>nuhsc.NUHSC</i>	hscs
<i>nuvsc.NUVSC</i>	vscs
<i>numultinicvport.NUMultiNICVPort</i>	multi_nic_vports
<i>nueventlog.NUEventLog</i>	event_logs

## 183.3 Parents

- *nuvport.NUVPort*
- *nucontainer.NUContainer*
- *nuhsc.NUHSC*
- *nuvsc.NUVSC*
- *nuvm.NUVM*





## NUVRSADDRESSRANGE

`nuvrsaddressrange.NUVRSAddressRange(bambou.nurest_object.NUMetaRESTObject,):`

This is the definition of a Address Range associated with a VRS.

### 184.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `max_address` (**Mandatory**): Highest address in the address range
- `min_address` (**Mandatory**): Lowest address in the address range
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 184.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 184.3 Parents

- `nuvcentervrsconfig.NUVCenterVRSCfg`
- `nuvcenterhypervisor.NUVCenterHypervisor`
- `nuvcentercluster.NUVCenterCluster`
- `nuvcenterdatacenter.NUVCenterDataCenter`
- `nuvcenter.NUVCenter`



## NUVRSMETRICS

```
nuvrsmetrics.NUVRSMetrics(bambou.nurest_object.NUMetaRESTObject,):
```

None

### 185.1 Attributes

- `al_ubr0_status`: alubr0 status
- `cpu_utilization`: cpu utilization
- `vrs_process`: vrs vsc process status
- `vrsvsc_status`: vrs vrs connection status
- `last_updated_by`: ID of the user who last updated the object.
- `re_deploy`: re-Deploy
- `receiving_metrics`: Is the VRS VM Sending Metrics to the hypervisor on VCIN
- `memory_utilization`: Memory Utilization
- `jesxmon_process`: jesxmon process status
- `agent_name`: VRS Agent Name
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `associated_vcenter_hypervisor_id`: None
- `external_id`: External object ID. Used for integration with third party systems

### 185.2 Parents

- `nuvcenterhypervisor`: *NUVCenterHypervisor*



## NUVRSREDEPLOYMENTPOLICY

```
nuvrsredeploymentpolicy.NUVRSRedeploymentpolicy(bambou.nurest_object.NUMetaRESTObject,):  
None
```

### 186.1 Attributes

- `al_ubr0_status_redeployment_enabled`: ALU br0 Status Redeployment Enabled
- `cpu_utilization_redeployment_enabled`: CPU Utilization Redeployment Enabled
- `cpu_utilization_threshold`: CPU Utilization Threshold
- `vrs_corrective_action_delay`: VRS Corrective Action Delay in seconds
- `vrs_process_redeployment_enabled`: VRS Process Redeployment Enabled
- `vrsvsc_status_redeployment_enabled`: VRSVSC Status Redeployment Enabled
- `last_updated_by`: ID of the user who last updated the object.
- `redeployment_delay`: redeployment Delay
- `memory_utilization_redeployment_enabled`: memory Utilization Redeployment Enabled
- `memory_utilization_threshold`: memory Utilization Threshold
- `deployment_count_threshold`: deployment count threshold
- `jesxmon_process_redeployment_enabled`: jesxmon Process Redeployment Enabled
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 186.2 Parents

- *nuvcentervrsconfig.NUVCenterVRSConfig*
- *nuvcenterhypervisor.NUVCenterHypervisor*
- *nuvcentercluster.NUVCenterCluster*
- *nuvcenterdatacenter.NUVCenterDataCenter*
- *nuvcenter.NUVCenter*



## **NUVSC**

**nuvsc.NUVSC (bambou.nurest\_object.NUMetaRESTObject, ) :**

System Monitoring details for VSC.

### **187.1 Attributes**

- **name:** Identifies the entity with a name.
- **management\_ip:** The management IP of the VSC/HSC entity
- **last\_state\_change:** Last state change timestamp (in millis).
- **last\_updated\_by:** ID of the user who last updated the object.
- **address:** The IP of the VRS entity
- **peak\_cpuusage:** Peek CPU usage percentage.
- **peak\_memory\_usage:** Peek memory usage percentage.
- **description:** Description of the entity.
- **messages:** An array of degraded messages.
- **disks:** Set of disk usage details.
- **already\_marked\_for\_unavailable:** Flag to indicate that it is already marked a unavailable.
- **unavailable\_timestamp:** The duration the controller is unavailable (in millis).
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **location:** Identifies the entity to be associated with a location.
- **product\_version:** Product version supported by this entity.
- **vsds:** A collection of VSD id(s) which are identified by this controller.
- **status:** Computed status of the entity.
- **current\_cpuusage:** Current CPU usage percentage.
- **current\_memory\_usage:** Current memory usage percentage.
- **average\_cpuusage:** Average CPU usage percentage.
- **average\_memory\_usage:** Average memory usage percentage.
- **external\_id:** External object ID. Used for integration with third party systems

## 187.2 Children

<b>class</b>	<b>fetcher</b>
<i>numetadata.NUMetadata</i>	metadatas
<i>nubgppeer.NUBGPPeer</i>	bgp_peers
<i>nualarm.NUAlarm</i>	alarms
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nujob.NUJob</i>	jobs
<i>numonitoringport.NUMonitoringPort</i>	monitoring_ports
<i>nuvrs.NUVRS</i>	vrss
<i>nueventlog.NUEventLog</i>	event_logs

## 187.3 Parents

- *nuvsp.NUVSP*
- *nuvrs.NUVRS*



**nuvsd.NUVSD (bambou.nurest\_object.NUMetaRESTObject, ) :**

System Monitoring details for VSD.

## 188.1 Attributes

- **url**: An optional web url for management.
- **name**: Identifies the entity with a name.
- **management\_ip**: An optional management IP to log into this component.
- **last\_state\_change**: Last state change timestamp (in millis).
- **last\_updated\_by**: ID of the user who last updated the object.
- **address**: An optional IP to access this component.
- **peak\_cpuusage**: Peek CPU usage percentage.
- **peak\_memory\_usage**: Peek memory usage percentage.
- **peer\_addresses**: A comma separated list of peer addresses, if it is in cluster mode.
- **description**: Description of the entity.
- **messages**: An array of degraded messages.
- **disks**: Set of disk usage details.
- **already\_marked\_for\_unavailable**: Flag to indicate that it is already marked a unavailable.
- **unavailable\_timestamp**: The duration the controller is unavailable (in millis).
- **entity\_scope**: Specify if scope of entity is Data center or Enterprise level
- **location**: Identifies the entity to be associated with a location.
- **mode**: Standalone or cluster mode.
- **product\_version**: Product version supported by this entity.
- **status**: Computed status of the entity.
- **current\_cpuusage**: Current CPU usage percentage.
- **current\_memory\_usage**: Current memory usage percentage.
- **average\_cpuusage**: Average CPU usage percentage.
- **average\_memory\_usage**: Average memory usage percentage.

- `external_id`: External object ID. Used for integration with third party systems

## 188.2 Children

<b>class</b>	<b>fetcher</b>
<i>numetadata.NUMetadata</i>	metadatas
<i>nualarm.NUAlarm</i>	alarms
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>njob.NUJob</i>	jobs
<i>nuvsdcomponent.NUVSDComponent</i>	vsd_components
<i>nueventlog.NUEventLog</i>	event_logs

## 188.3 Parents

- *nuvsp.NUVSP*

## NUVSDCOMPONENT

`nuvsdcomponent.NUVSDComponent (bambou.nurest_object.NUMetaRESTObject, ) :`

System Monitoring details for components of VSD system.

### 189.1 Attributes

- `name`: Identifies the entity with a name.
- `management_ip`: An optional management IP to log into this component.
- `address`: An optional IP to access this component.
- `description`: Description of the entity.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `location`: Identifies the entity to be associated with a location.
- `product_version`: Product version supported by this entity.
- `status`: Current status of the entity. Possible values are UP, DOWN, ADMIN\_DOWN, .
- `external_id`: External object ID. Used for integration with third party systems
- `type`: Type of the component.

### 189.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 189.3 Parents

- `nuvsd.NUVSD`



## NUVSGREDUNDANTPORT

`nuvsgredundantport.NUVsgRedundantPort (bambou.nurest_object.NUMetaRESTObject, ) :`

Represents a redundant Port under a particular gateway object or redundant group object.

### 190.1 Attributes

- `vlan_range`: VLAN Range of the Port. Format must conform to a-b,c,d-f where a,b,c,d,f are integers between 0 and 4095.
- `name` (**Mandatory**): Name of the Port
- `last_updated_by`: ID of the user who last updated the object.
- `permitted_action`: The permitted action to USE/EXTEND this Gateway.
- `description`: A description of the Port
- `physical_name` (**Mandatory**): Identifier of the Port
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `port_peer1_id`: The master gateway peer port id.
- `port_peer2_id`: The slave gateway peer port id.
- `port_type` (**Mandatory**): Type of the Port.
- `use_user_mnemonic`: determines whether to use user mnemonic of the Port
- `user_mnemonic`: user mnemonic of the Port
- `associated_egress_qos_policy_id`: ID of the Egress QOS Policy associated with this Vlan.
- `status`: Status of the port.
- `external_id`: External object ID. Used for integration with third party systems

### 190.2 Children

class	fetcher
<code>nupermision.NUPermission</code>	permissions
<code>numetadata.NUMetadata</code>	metadatas
<code>nuvlan.NUVLAN</code>	vlangs
<code>nualarm.NUAlarm</code>	alarms
<code>nuglobalmetadata.NUGlobalMetadata</code>	global_metadatas
<code>nuenterpisepermission.NUEnterprisePermission</code>	enterprise_permissions

## 190.3 Parents

- *nuredundancygroup.NURedundancyGroup*

## NUVSP

**nuvsp.NUVSP (bambou.nurest\_object.NUMetaRESTObject, ) :**

System Monitoring details for VSP.

### 191.1 Attributes

- **name:** Name of the VSP
- **last\_updated\_by:** ID of the user who last updated the object.
- **description:** Description of the VSP
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **location:** Installed location of the VSP product
- **product\_version:** Product version number for VSP
- **external\_id:** External object ID. Used for integration with third party systems

### 191.2 Children

<b>class</b>	<b>fetcher</b>
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuhsc.NUHSC</i>	hscs
<i>nuvsc.NUVSC</i>	vscs
<i>nuvsd.NUVSD</i>	vsds
<i>nueventlog.NUEventLog</i>	event_logs

### 191.3 Parents

- *nume.NUMe*





## NUWANSERVICE

`nuwanservice.NUWANService(bambou.nurest_object.NUMetaRESTObject,):`

Represents a WAN Service Object.

### 192.1 Attributes

- `wan_service_identifier`: Identifier of the WAN Service
- `irb_enabled`: Determines whether Integrated Routing and Bridging is enabled on the WAN Service
- `name` (**Mandatory**): Name of the WAN Service
- `last_updated_by`: ID of the user who last updated the object.
- `permitted_action`: The permitted action to USE/EXTEND this Gateway.
- `service_policy`: Name of 7X50 Policy associated with the service
- `service_type` (**Mandatory**): Type of the service.
- `description`: A description of the WAN Service
- `vn_id`: VNID of the BackHaul Subnet of L3Domain /L2Domain to which this WANService is associated
- `enterprise_name`: The associated enterprise name.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `domain_name`: The associated domain name.
- `config_type`: Type of the CONFIG.
- `orphan`: Indicates if this WAN Service is orphan or not.
- `use_user_mnemonic`: Determines whether to use user mnemonic of the WAN Service
- `user_mnemonic`: user mnemonic of the WAN Service
- `associated_domain_id`: ID of the entity to which the WAN Service is attached to. This could be ID DOMAIN/L2DOMAIN
- `associated_vpn_connect_id`: The associated vpn connect ID.
- `tunnel_type`: Type of the tunnel.
- `external_id`: External object ID. Used for integration with third party systems
- `external_route_target`: Route target associated with the WAN. It is an optional parameter that can be provided by the user

## 192.2 Children

<b>class</b>	<b>fetcher</b>
<i>nupermision.NUPermission</i>	permissions
<i>numetadata.NUMetadata</i>	metadatas
<i>nualarm.NUAlarm</i>	alarms
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuenterprisepermission.NUEnterprisePermission</i>	enterprise_permissions
<i>nueventlog.NUEventLog</i>	event_logs

## 192.3 Parents

- *nuredundancygroup.NURedundancyGroup*
- *nuautodiscoveredgateway.NUAutoDiscoveredGateway*
- *nugateway.NUGateway*

## NUZFBAUTOASSIGNMENT

`nuzfbautoassignment.NUZFBAutoAssignment(bambou.nurest_object.NUMetaRESTObject,):`  
ZFB Enterprise Auto Assignment

### 193.1 Attributes

- `zfb_match_attribute`: Attribute to auto match on
- `zfb_match_attribute_values`: Array of values to match on
- `name` (**Mandatory**): Name of the ZFB auto assignment criteria.
- `last_updated_by`: ID of the user who last updated the object.
- `description`: Description of the ZFB auto assignment criteria.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `priority` (**Mandatory**): Priority of the Auto Assignment
- `associated_enterprise_id`: Associated Enterprise ID
- `associated_enterprise_name`: The name of the associated Enterprise
- `external_id`: External object ID. Used for integration with third party systems

### 193.2 Parents

- *nume.NUMe*



## NUZFBREQUEST

`nuzfbrequest.NUZFBRequest(bambou.nurest_object.NUMetaRESTObject,):`

A ZFB Request from an NSG

### 194.1 Attributes

- `mac_address`: MAC Address fo the NSG Port1 interface
- `zfb_approval_status`: the status of the request
- `zfb_bootstrap_enabled`: whether the NSG should bootstrap, or just simulate bootstrap. Set from System Config
- `zfb_info`: The Base64 encoded JSON string of ZFB Attributes
- `zfb_request_retry_timer`: ZFB Request retry timer on NSG. Set from System Config
- `sku`: The part number of the NSG
- `ip_address`: IP Address of the NSG
- `cpu_type`: Processor Type
- `nsg_version`: The Nuage NSG Version
- `uuid`: Redhat UUID
- `family`: NSG Type
- `last_connected_time`: the time in which the last GET was made from the NSG
- `last_updated_by`: ID of the user who last updated the object.
- `serial_number`: The NSG's Serial Number
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `hostname`: hostname of the NSG
- `associated_enterprise_id`: the ID of the associated enterprise
- `associated_enterprise_name`: name of the associated Enterprise
- `associated_ns_gateway_id`: ID of the assigned NSG
- `associated_ns_gateway_name`: name of the associated NSG
- `status_string`: Extra status info
- `external_id`: External object ID. Used for integration with third party systems

## 194.2 Children

<b>class</b>	<b>fetcher</b>
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nujob.NUJob</i>	jobs

## 194.3 Parents

- *nume.NUMe*
- *numenterprise.NUEnterprise*

## NUZONE

`nuzone.NUZone(bambou.nurest_object.NUMetaRESTObject, ) :`

The zone is a collection of subnets attached to a domain. The zone concept enables the definition of policies for collections of subnets.

## 195.1 Attributes

- `dpi`: determines whether or not Deep packet inspection is enabled
- `ip_type`: IPv4 or IPv6
- `maintenance_mode`: Indicates if the Zone is accepting VM activation requests.
- `name` (**Mandatory**): Name of the current entity (Zone or zone template or subnet etc..) Valid characters are alphabets, numbers, space and hyphen ( - ).
- `last_updated_by`: ID of the user who last updated the object.
- `address`: IP address of the subnet defined. In case of zone, this is an optional field for and allows users to allocate an IP address range to a zone. The VSD will auto-assign IP addresses to subnets from this range if a specific IP address is not defined for the subnet
- `template_id`: The ID of the template that this zone was derived from
- `description`: A description of the zone
- `netmask`: Netmask of the subnet defined
- `encryption`: Determines whether or not IPSEC is enabled.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `policy_group_id`: PG ID for the subnet. This is unique per domain and will be in the range 1-4095
- `associated_application_id`: The associated application ID.
- `associated_application_object_id`: The associated application object ID.
- `associated_application_object_type`: The associated application object type. Refer to API section for supported types.
- `associated_multicast_channel_map_id`: The ID of the Multi Cast Channel Map this zone/zone template is associated with. This has to be set when `enableMultiCast` is set to `ENABLED`
- `public_zone`: If a zone is marked as public, then it is lined to the public network associated with this data center
- `multicast`: Indicates multicast policy on zone.

- `number_of_hosts_in_subnets`: Number of hosts in each of the subnets that can be created under a zone and are auto-assigned IP addresses
- `external_id`: External object ID. Used for integration with third party systems

## 195.2 Children

class	fetcher
<i>nutca.NUTCA</i>	<code>tcas</code>
<i>nupermision.NUPermission</i>	<code>permissions</code>
<i>numetadata.NUMetadata</i>	<code>metadatas</code>
<i>nudhcpoption.NUDHCPOption</i>	<code>dhcp_options</code>
<i>nuglobalmetadata.NUGlobalMetadata</i>	<code>global_metadatas</code>
<i>nuvm.NUVM</i>	<code>vms</code>
<i>nuvminterface.NUVMInterface</i>	<code>vm_interfaces</code>
<i>nucontainer.NUContainer</i>	<code>containers</code>
<i>nucontainerinterface.NUContainerInterface</i>	<code>container_interfaces</code>
<i>nuqos.NUQOS</i>	<code>qoss</code>
<i>nuvport.NUVPort</i>	<code>vports</code>
<i>nugroup.NUGroup</i>	<code>groups</code>
<i>nustatistics.NUStatistics</i>	<code>statistics</code>
<i>nustatisticspolicy.NUStatisticsPolicy</i>	<code>statistics_policies</code>
<i>nusubnet.NUSubnet</i>	<code>subnets</code>
<i>nueventlog.NUEventLog</i>	<code>event_logs</code>

## 195.3 Parents

- *nudomain.NUDomain*
- *nume.NUMe*



## NUZONETEMPLATE

`nuzonetemplate.NUZoneTemplate(bambou.nurest_object.NUMetaRESTObject, ) :`

As in domains and subnets, zones are derived from templates. This object provides the definition of the template.

### 196.1 Attributes

- `dpi`: determines whether or not Deep packet inspection is enabled
- `ip_type`: IPv4 or IPv6(only IPv4 is supported in R1.0) Possible values are IPV4, IPV6, .
- `name` (**Mandatory**): Name of the current entity(Zone or zone template or subnet etc..) Valid characters are alphabets, numbers, space and hyphen( - ).
- `last_updated_by`: ID of the user who last updated the object.
- `address`: IP address of the subnet defined. In case of zone, this is an optional field for and allows users to allocate an IP address range to a zone. The VSD will auto-assign IP addresses to subnets from this range if a specific IP address is not defined for the subnet
- `description`: A description of the Zone template
- `netmask`: Netmask of the subnet defined
- `encryption`: Determines whether or not IPSEC is enabled.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `associated_multicast_channel_map_id`: The ID of the Multi Cast Channel Map this zone/zone template is associated with. This has to be set when enableMultiCast is set to ENABLED
- `public_zone`: Identifies if the zone is a public zone, in which case any subnets associated with this zone are actually connected to the public subnet of the data center
- `multicast`: Indicates multicast policy on zone template.
- `number_of_hosts_in_subnets`: Number of hosts in the subnets where IP addresses are automatically assigned from the zone IP pool
- `external_id`: External object ID. Used for integration with third party systems

## 196.2 Children

<b>class</b>	<b>fetcher</b>
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuqos.NUQOS</i>	qoss
<i>nusubnettemplate.NUSubnetTemplate</i>	subnet_templates
<i>nueventlog.NUEventLog</i>	event_logs

## 196.3 Parents

- *nudomaintemplate.NUDomainTemplate*

## NUADDRESSRANGE

`nuaddressrange.NUAddressRange(bambou.nurest_object.NUMetaRESTObject,):`

This is the definition of a Address Range associated with a Network.

### 197.1 Attributes

- `dhcp_pool_type`: DHCPPoolType is an enum that indicates if the DHCP Pool is for HOST/BRIDGE.
- `last_updated_by`: ID of the user who last updated the object.
- `max_address` (**Mandatory**): Highest address in the address range
- `min_address` (**Mandatory**): Lowest address in the address range
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 197.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>

### 197.3 Parents

- `nusharednetworkresource.NUSharedNetworkResource`
- `nusubnettemplate.NUSubnetTemplate`
- `nusubnet.NUSubnet`
- `nul2domaintemplate.NUL2DomainTemplate`
- `nul2domain.NUL2Domain`



## NUAGGREGATEMETADATA

`nuaggregatemetadata.NUAggregateMetadata(bambou.nurest_object.NUMetaRESTObject, ) :`

Metadata associated to a entity

### 198.1 Attributes

- `name`: Name of the Metadata.
- `description`: Description of the Metadata.
- `metadata_tag_ids`: Metadata tag IDs associated with this metadata. You can filter metadata based on this attribute for example X-Nuage-Filter: '2d6fb627-603b-421c-b63a-eb0a6d712761' IN metadataTagIDs
- `network_notification_disabled`: Specifies metadata changes need to be notified to controller, by default it is notified
- `blob` (**Mandatory**): Metadata that describes about the entity attached to it.
- `global_metadata`: Specifies whether the metadata is global or local
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 198.2 Parents

- `nuvport.NUVPort`



## NUALARM

```
nualarm.NUAlarm(bambou.nurest_object.NUMetaRESTObject,):
```

The alarm API allows the management of system alarms.

## 199.1 Attributes

- **name (Mandatory):** The alarm name. Each type of alarm will generate its own name
- **target\_object:** Identifies affected Entity. Example: Alarm generated by TCA for Domain domain1(Packets towards a VM, Breach)
- **last\_updated\_by:** ID of the user who last updated the object.
- **acknowledged:** Flag to indicate that alarm is already acknowledged or not
- **reason:** Provides a description of the alarm
- **description:** Description of the alarm
- **severity:** Severity of the alarm.
- **timestamp:** Indicates the time that the alarm was triggered
- **enterprise\_id:** Enterprise that this alarm belongs to
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **error\_condition:** Identifies the error condition
- **number\_of\_occurrences:** Number of times that the alarm was triggered
- **external\_id:** External object ID. Used for integration with third party systems

## 199.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas

## 199.3 Parents

- *nuredundancygroup.NURedundancyGroup*

- *nutca.NUTCA*
- *nuvsgredundantport.NUVsgRedundantPort*
- *nuwanservice.NUWANService*
- *nuvsd.NUVSD*
- *nuvport.NUVPort*
- *nuport.NUPort*
- *nuvrs.NUVRS*
- *nuhsc.NUHSC*
- *nugateway.NUGateway*
- *nunsgateway.NUNSGateway*
- *nuvsc.NUVSC*
- *nuvm.NUVM*
- *nunsport.NUNSPort*
- *nunsredundantgatewaygroup.NUNSRedundantGatewayGroup*
- *nuvlan.NUVLAN*
- *nuenterprise.NUEnterprise*



## NUAPP

`nuapp.NUApp(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents a real life application like a vendor website, or a social network.

### 200.1 Attributes

- **name (Mandatory):** Name of the application.
- **last\_updated\_by:** ID of the user who last updated the object.
- **description:** Description of the application.
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **assoc\_egress\_acl\_template\_id:** The ID of the ACL template that this application is pointing to.
- **assoc\_ingress\_acl\_template\_id:** The ID of the ACL template that this application is pointing to
- **associated\_domain\_id (Mandatory):** Domain id where the application is running.
- **associated\_domain\_type (Mandatory):** Type of domain (DOMAIN, L2DOMAIN). Refer to API section for supported types.
- **associated\_network\_object\_id:** ID of the network object that this App is associated with.
- **associated\_network\_object\_type:** Type of network object this App is associated with (ENTERPRISE, DOMAIN) Refer to API section for supported types.
- **external\_id:** External object ID. Used for integration with third party systems

### 200.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nutier.NUTier</i>	tiers
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuflow.NUFlow</i>	flows
<i>nujob.NUJob</i>	jobs
<i>nueventlog.NUEventLog</i>	event_logs

### 200.3 Parents

- *nuenterprise.NUEnterprise*



## NUAPPLICATIONSERVICE

`nuapplicationservice.NUApplicationService(bambou.nurest_object.NUMetaRESTObject,)` :

Represents a networking communication service.

### 201.1 Attributes

- `dscp`: DSCP match condition to be set in the rule. It is either \* or from 0-63. Required for etherType 0x0800
- `name` (**Mandatory**): Name of the application service.
- `last_updated_by`: ID of the user who last updated the object.
- `description`: Description of the application service.
- `destination_port` (**Mandatory**): The destination port to be matched if protocol is UDP or TCP. Value should be either \* or single port number or a port range.
- `direction` (**Mandatory**): Direction of the service. Default is UNIDIRECTIONAL.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `source_port` (**Mandatory**): Source port to be matched if protocol is UDP or TCP. Value can be either \* or single port number or a port range.
- `protocol` (**Mandatory**): Protocol that must be matched. Needs to be 6 (TCP) or 17 (UDP)
- `ether_type` (**Mandatory**): Ether type of the packet to be matched. Ether type can be \* or a valid hexadecimal value
- `external_id`: External object ID. Used for integration with third party systems

### 201.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>

### 201.3 Parents

- `nume.NUMe`

- *nuenterprise.NUEnterprise*

## NUAUTODISCOVEREDGATEWAY

`nuautodiscoveredgateway.NUAutoDiscoveredGateway(bambou.nurest_object.NUMetaRESTObject, ) :`  
Represents Auto discovered Gateway.

### 202.1 Attributes

- **name (Mandatory):** Name of the Gateway
- **last\_updated\_by:** ID of the user who last updated the object.
- **gateway\_id:** The Gateway associated with this Auto Discovered Gateway . This is a read only attribute
- **peer:** The System ID of the peer gateway associated with this Gateway instance when it is discovered by the network manager (VSD) as being redundant.
- **personality (Mandatory):** Personality of the Gateway - VSG,VRSG,NONE,OTHER, cannot be changed after creation.
- **description:** A description of the Gateway
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **controllers:** Controllers to which this gateway instance is associated with.
- **vtep:** Represent the system ID or the Virtual IP of a service used by a Gateway (VSG for now) to establish a tunnel with a remote VSG or hypervisor. The format of this field is consistent with an IP address.
- **external\_id:** External object ID. Used for integration with third party systems
- **system\_id:** Identifier of the Gateway

### 202.2 Children

class	fetcher
<i>nuwanservice.NUWANService</i>	wan_services
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuport.NUPort</i>	ports
<i>nunsport.NUNSPort</i>	ns_ports
<i>nueventlog.NUEventLog</i>	event_logs

## 202.3 Parents

- *nume.NUMe*

## NUBGPPEER

`nubgppeer.NUBGPPeer(bambou.nurest_object.NUMetaRESTObject, ) :`

Encapsulates the BGP peer information for system monitor entity.

### 203.1 Attributes

- `last_state_change`: Last state change timestamp.
- `address`: IP of the BGP peer.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `status`: Current connection status of the BGP peer.
- `external_id`: External object ID. Used for integration with third party systems

### 203.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 203.3 Parents

- `nuhsc.NUHSC`
- `nuvsc.NUVSC`





## NUBOOTSTRAP

`nubootstrap.NUBootstrap(bambou.nurest_object.NUMetaRESTObject, ) :`

Gateway bootstrap details.

### 204.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `installer_id`: The Installer ID
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `status`: Bootstrap status.
- `external_id`: External object ID. Used for integration with third party systems

### 204.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 204.3 Parents

- `nunsgateway.NUNSGateway`



## NUBOOTSTRAPACTIVATION

`nubootstrapactivation.NUBootstrapActivation(bambou.nurest_object.NUMetaRESTObject, ) :`  
NSG Gateway initiated Bootstrap Activation

### 205.1 Attributes

- `ca_cert`: The CA Certificate Chain
- `hash`: The authentication hash of this request
- `last_updated_by`: ID of the user who last updated the object.
- `action`: The bootstrap action to perform.
- `seed`: The random seed for this request
- `cert`: The signed Certificate
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `config_url`: The configuration URL
- `tpm_owner_password`: TPM owner passphrase
- `srk_password`: TPM SRK passphrase
- `vsd_time`: VSD Server time when an NSG is initiating a Bootstrapping request
- `csr`: The CSR of the request
- `status`: The agent status for the request
- `external_id`: External object ID. Used for integration with third party systems

### 205.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 205.3 Parents

- `nunsgateway.NUNSGateway`



## NUBRIDGEINTERFACE

`nubridgeinterface.NUBridgeInterface(bambou.nurest_object.NUMetaRESTObject, ) :`

Provides information for each bridge interface.

### 206.1 Attributes

- `vport_id`: ID of the vport that the interface is attached to
- `vport_name`: Name of the vport that the VM is attached to
- `name`: Device name associated with this interface
- `last_updated_by`: ID of the user who last updated the object.
- `gateway`: Gateway of the subnet that the VM is connected to
- `netmask`: Netmask of the subnet that the VM is attached to
- `network_name`: Name of the network that the VM is attached to
- `tier_id`: ID of the tier that the interface is attached to.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `policy_decision_id`: The policy decision ID for this particular interface
- `domain_id`: ID of the domain that the VM is attached to
- `domain_name`: Name of the domain that the VM is attached to
- `zone_id`: ID of the zone that the interface is attached to
- `zone_name`: Name of the zone that the VM is attached to.
- `associated_floating_ip_address`: Floating Ip Address of this network interface eg: 10.1.2.1
- `attached_network_id`: ID of the l2 domain or Subnet that the VM is attached to
- `attached_network_type`: l2 domain or Subnet that the interface is attached to
- `external_id`: External object ID. Used for integration with third party systems

## 206.2 Children

<b>class</b>	<b>fetcher</b>
<i>nutca.NUTCA</i>	tcas
<i>nuredirectiontarget.NURedirectionTarget</i>	redirection_targets
<i>numetadata.NUMetadata</i>	metadatas
<i>nudhcpoption.NUDHCPOption</i>	dhcp_options
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nupolicydecision.NUPolicyDecision</i>	policy_decisions
<i>nupolicygroup.NUPolicyGroup</i>	policy_groups
<i>nuqos.NUQOS</i>	qoss
<i>nustatistics.NUStatistics</i>	statistics
<i>nueventlog.NUEventLog</i>	event_logs

## 206.3 Parents

- *nudomain.NUDomain*
- *nuvport.NUVPort*
- *nul2domain.NUL2Domain*

## NUCERTIFICATE

`nucertificate.NUCertificate(bambou.nurest_object.NUMetaRESTObject,)` :

This object represents a X509 Certificate Request

### 207.1 Attributes

- `pem_encoded`: The PEM encoded certificate.
- `serial_number`: The serial number of this certificate.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `issuer_dn`: The distinguished name of the authority that issued this certificate.
- `subject_dn`: The distinguished name of this certificate's end entity.
- `public_key`: The public key contained in this certificate.
- `external_id`: External object ID. Used for integration with third party systems

### 207.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 207.3 Parents

- `nume.NUMe`





## NUCLOUDMGMTSYSTEM

`nucloudmgmtsystem.NUCloudMgmtSystem(bambou.nurest_object.NUMetaRESTObject, ) :`

Object that identifies a cloud management system.

### 208.1 Attributes

- `name`: Name of the cloud management system
- `last_updated_by`: ID of the user who last updated the object.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 208.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 208.3 Parents

- `nume.NUMe`



## NUDHCOPTION

`nudhcption.NUDHCOption(bambou.nurest_object.NUMetaRESTObject, ) :`

Allows the definition of one or more DHCP options that will be provided to all VMs that are associated with a given domain. DHCP options are provided as Type- Length-Value (TLV). There is no validation by VSD on whether these options are valid or not. It is up to the user to guarantee that the options make sense for their application.

### 209.1 Attributes

- **value (Mandatory):** DHCP option value. Value should be a hexadecimal value(ie. Hex value 0xac40 would be passed as 'ac40')
- **last\_updated\_by:** ID of the user who last updated the object.
- **actual\_type:** This will be used to send actual type instead of the hexadecimal. Note: If actualType is set, it will override the entry set in the type attribute
- **actual\_values:** This will be used to send actual values instead of the hexadecimal. Note: If actualValues are set, it will override entry set in the value attribute
- **length (Mandatory):** DHCP option length. Length should be a hexadecimal value(ie. Hex value 0x04 would be passed as '04')
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **external\_id:** External object ID. Used for integration with third party systems
- **type (Mandatory):** DHCP option type. Type should be a hexadecimal value(ie. Hex value 0x06 would be passed as '06')

### 209.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>

### 209.3 Parents

- `nusharednetworkresource.NUSharedNetworkResource`

- *nuzone.NUZone*
- *nudomain.NUDomain*
- *nuvport.NUVPort*
- *nusubnet.NUSubnet*
- *nuvminterface.NUVMInterface*
- *nubridgeinterface.NUBridgeInterface*
- *nul2domain.NUL2Domain*
- *nuhostinterface.NUHostInterface*

## **NUDISKSTAT**

**nudiskstat.NUDiskStat (bambou.nurest\_object.NUMetaRESTObject, ) :**

Encapsulates the disk usage metrics for system monitor entity.

### **210.1 Attributes**

- **name:** Name of the disk.
- **size:** Total disk space.
- **unit:** Storage unit type (example: bytes, KB, MB, etc.,).
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **used:** Disk space used.
- **available:** Available disk space.
- **external\_id:** External object ID. Used for integration with third party systems



## NUDOMAIN

`nudomain.NUDomain(bambou.nurest_object.NUMetaRESTObject, ) :`

This object is used to manipulate domain state. A domain corresponds to a distributed Virtual Router and Switch (dVRS).

### 211.1 Attributes

- `pat_enabled`: Indicates whether PAT is enabled for the subnets in this domain - ENABLED/DISABLED Possible values are INHERITED, ENABLED, DISABLED, .
- `ecmp_count`: Domain specific Equal-cost multi-path routing count, ECMPCount = 1 means no ECMP
- `dhcp_behavior`: DHCPBehaviorType is an enum that indicates DHCP Behavior of VRS having VM's under this domain. Possible values are FLOOD, CONSUME ,RELAY Possible values are CONSUME, FLOOD, RELAY, .
- `dhcp_server_address`: when DHCPBehaviorType is RELAY, then DHCP Server IP Address needs to be set
- `label_id`: The label associated with the dVRS. This is a read only attribute
- `back_haul_route_distinguisher`: Route distinguisher associated with the BackHaul Service in dVRS. If not provided during creation, System generates this identifier automatically
- `back_haul_route_target`: Route target associated with the BackHaul Service in dVRS. If not provided during creation, System generates this identifier automatically
- `back_haul_vnid`: Current BackHaul Network's globally unique VXLAN network identifier generated by VSD
- `maintenance_mode`: maintenanceMode is an enum that indicates if the Domain is accepting VM activation requests. Possible values are DISABLED, ENABLED and ENABLED\_INHERITED Possible values are .
- `name` (**Mandatory**): The name of the domain. Valid characters are alphabets, numbers, space and hyphen( - ).
- `last_updated_by`: ID of the user who last updated the object.
- `leaking_enabled`: Indicates if this domain is a leakable domain or not - boolean true/false
- `secondary_dhcp_server_address`: when DHCPBehaviorType is RELAY, then DHCP Server IP Address needs to be set
- `template_id` (**Mandatory**): The ID of the template that this domain was created from. This should be set when instantiating a domain
- `permitted_action`: The permitted action to USE/DEPLOY for the Domain Possible values are USE, READ, ALL, INSTANTIATE, EXTEND, DEPLOY, .

- `service_id`: The serviceID of the Virtual Router created in VSC and is associated with this object. This is auto-generated by VSD
- `description`: A description string of the domain that is provided by the user
- `dhcp_server_addresses`: when DHCPBehaviorType is RELAY, then DHCP Server IP Address needs to be set
- `global_routing_enabled`: Indicates if this domain is a globally routable domain or not - boolean true/false
- `import_route_target`: Route distinguisher associated with the dVRS. It is an optional parameter that can be provided by the user or auto-managed by VSD. System generates this identifier automatically, if not provided
- `encryption`: Determines whether IPSEC is enabled Possible values are ENABLED, DISABLED, .
- `underlay_enabled`: Indicates whether UNDERLAY is enabled for the subnets in this domain
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `policy_change_status`: None
- `route_distinguisher`: Route distinguisher associated with the dVRS. It is an optional parameter that can be provided by the user or auto-managed by VSD. System generates this identifier automatically, if not provided
- `route_target`: Route target associated with the dVRS. It is an optional parameter that can be provided by the user or auto-managed by VSD System generates this identifier automatically, if not provided
- `uplink_preference`: Indicates the preferential path selection for network traffic in this domain - Default is Primary 1 and Secondary 2. Possible values are PRIMARY\_SECONDARY, SECONDARY\_PRIMARY, PRIMARY, SECONDARY, SYMMETRIC, .
- `application_deployment_policy`: Application deployment policy.
- `associated_multicast_channel_map_id`: The ID of the Multi Cast Channel Map this domain is associated with. This has to be set when enableMultiCast is set to ENABLED
- `stretched`: Indicates whether this domain is stretched, if so remote VM resolutions will be allowed
- `multicast`: multicast is enum that indicates multicast policy on domain. Possible values are ENABLED, DISABLED and INHERITED Possible values are INHERITED, ENABLED, DISABLED, .
- `tunnel_type`: Default Domain Tunnel Type
- `customer_id`: The customerID that is created in the VSC and identifies this dVRS. This is auto-generated by VSD
- `export_route_target`: Route target associated with the dVRS. It is an optional parameter that can be provided by the user or auto-managed by VSD System generates this identifier automatically, if not provided
- `external_id`: External object ID. Used for integration with third party systems

## 211.2 Children

class	fetcher
<i>nutca.NUTCA</i>	tcas
<i>nuredirectiontarget.NURedirectionTarget</i>	redirection_targets
<i>nupermission.NUPermission</i>	permissions
<i>numetadata.NUMetadata</i>	metadatas
<i>nuegressaclentrytemplate.NUEgressACLEntryTemplate</i>	egress_acl_entry_templates

Continued on next page



Table 211.1 – continued from previous page

<i>nuegressacltemplate.NUEgressACLTemplate</i>	<code>egress_acl_templates</code>
<i>nudomainfipacltemplate.NUDomainFIPACLTemplate</i>	<code>domain_fip_acl_templates</code>
<i>nufloatingipacltemplate.NUFloatingIPACLTemplate</i>	<code>floating_ipacl_templates</code>
<i>nudhcpoption.NUDHCPOption</i>	<code>dhcp_options</code>
<i>nufloatingip.NUFloatingIp</i>	<code>floating_ips</code>
<i>nuglobalmetadata.NUGlobalMetadata</i>	<code>global_metadata</code>
<i>nuvm.NUVM</i>	<code>vms</code>
<i>nuvminterface.NUVMInterface</i>	<code>vm_interfaces</code>
<i>nuingressaclentrytemplate.NUIngressACLEntryTemplate</i>	<code>ingress_acl_entry_templates</code>
<i>nuingressacltemplate.NUIngressACLTemplate</i>	<code>ingress_acl_templates</code>
<i>nuingressadvfwdtemplate.NUIngressAdvFwdTemplate</i>	<code>ingress_adv_fwd_templates</code>
<i>nuingressexternalservicetemplate.NUIngressExternalServiceTemplate</i>	<code>ingress_external_service_templates</code>
<i>nujob.NUJob</i>	<code>jobs</code>
<i>nupolicygroup.NUPolicyGroup</i>	<code>policy_groups</code>
<i>nudomain.NUDomain</i>	<code>domains</code>
<i>nudomaintemplate.NUDomainTemplate</i>	<code>domain_templates</code>
<i>nuzone.NUZone</i>	<code>zones</code>
<i>nuqos.NUQOS</i>	<code>qoss</code>
<i>nuhostinterface.NUHostInterface</i>	<code>host_interfaces</code>
<i>nuuplinkrd.NUUplinkRD</i>	<code>uplink_rds</code>
<i>nuvpnconnection.NUVPNConnection</i>	<code>vpn_connections</code>
<i>nuvport.NUVPort</i>	<code>vports</code>
<i>nubridgeinterface.NUBridgeInterface</i>	<code>bridge_interfaces</code>
<i>nugroup.NUGroup</i>	<code>groups</code>
<i>nustaticroute.NUStaticRoute</i>	<code>static_routes</code>
<i>nustatistics.NUStatistics</i>	<code>statistics</code>
<i>nustatisticspolicy.NUStatisticsPolicy</i>	<code>statistics_policies</code>
<i>nusubnet.NUSubnet</i>	<code>subnets</code>
<i>nueventlog.NUEventLog</i>	<code>event_logs</code>
<i>nuexternalappservice.NUExternalAppService</i>	<code>external_app_services</code>

## 211.3 Parents

- *nudomain.NUDomain*
- *nume.NUMe*
- *nudomaintemplate.NUDomainTemplate*
- *nuenterprise.NUEnterprise*



## NUDOMAINFIPACLTEMPLATE

`nudomainfipacltemplate.NUDomainFIPACLTemplate(bambou.nurest_object.NUMetaRESTObject,)` :

Defines the template for an Domain Floating IP ACL

### 212.1 Attributes

- `name`: The name of the entity
- `last_updated_by`: ID of the user who last updated the object.
- `active`: If enabled, it means that this ACL or QOS entry is active
- `default_allow_ip`: If enabled a default ACL of Allow All is added as the last entry in the list of ACL entries
- `default_allow_non_ip`: If enabled, non ip traffic will be dropped
- `description`: A description of the entity
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `entries`: List of Egress Domain ACL entries associated with this ACL
- `policy_state`: State of the policy
- `priority`: The priority of the ACL entry that determines the order of entries
- `priority_type`:
- `associated_live_entity_id`: ID of the associated live entity
- `external_id`: External object ID. Used for integration with third party systems

### 212.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nudomainfipacltemplateentry.NUDomainFIPACLTemplateEntry</code>	<code>domain_fip_acl_template_entries</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

## 212.3 Parents

- *nudomain.NUDomain*
- *nume.NUMe*
- *nudomaintemplate.NUDomainTemplate*

## NUDOMAINFIPACLTEMPLATEENTRY

`nudomainfipacltemplateentry.NUDomainFIPACLTemplateEntry(bambou.nurest_object.NUMetaRESTObj`

Defines the template of Egress Domain ACL Template entries

### 213.1 Attributes

- `dscp`: DSCP match condition to be set in the rule. It is either \* or from 0-63
- `last_updated_by`: ID of the user who last updated the object.
- `action`: The action of the ACL entry DROP or FORWARD or REDIRECT. Action REDIRECT is allowed only for IngressAdvancedForwardingEntry
- `action_details`: Type of action to be performed when a ACL match criteria succeeds
- `address_override`: Overrides the source IP for Ingress and destination IP for Egress, macentries will use this adress as the match criteria.
- `reflexive`: True means that this ACL entry is reflexive, so there will be a corresponding rule that will be created by OVS in the network. False means that there is no corresponding rule created by OVS in the network.
- `description`: Description of the ACL entry
- `dest_pg_id`: In case of PG this will be its EVPNBGPCommunity String, incase of network it will be network cidr
- `dest_pg_type`: In case of PG this will be its EVPNBGPCommunity String, incase of network it will be network cidr
- `destination_port`: The destination port to be matched if protocol is UDP or TCP. Value should be either \* or single port number or a port range
- `destination_type`: Network Type - either PolicyGroup or Network
- `destination_value`: In case of PG this will be its EVPNBGPCommunity String, incase of network it will be network cidr
- `network_id`: The destination network entity that is referenced(subnet/zone/macro)
- `network_type`: Type of the source network - VM\_SUBNET or VM\_ZONE or VM\_DOMAIN or SUBNET or ZONE or ENTERPRISE\_NETWORK or PUBLIC\_NETWORK or ANY
- `flow_logging_enabled`: Is flow logging enabled for this particular template
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `location_id`: The ID of the location entity (Subnet/Zone/VportTag)

- `location_type`: Type of the location entity - ANY or SUBNET or ZONE or VPORTTAG
- `policy_state`: State of the policy.
- `source_pg_id`: In case of PG this will be its EVPNBGPCommunity String, incase of network it will be network cidr
- `source_pg_type`: In case of PG this will be its EVPNBGPCommunity String, incase of network it will be network cidr
- `source_port`: Source port to be matched if protocol is UDP or TCP. Value can be either \* or single port number or a port range
- `source_type`: Location Type - either PolicyGroup or Network
- `source_value`: In case of PG this will be its EVPNBGPCommunity String, incase of network it will be network cidr
- `priority`: The priority of the ACL entry that determines the order of entries
- `protocol`: Protocol number that must be matched
- `associated_application_id`: The associated application ID
- `associated_application_object_id`: The associated application object ID
- `associated_application_object_type`: The associated application object type
- `associated_live_entity_id`: ID of the associated live entity
- `stats_id`: The statsID that is created in the VSD and identifies this ACL Template Entry. This is auto-generated by VSD
- `stats_logging_enabled`: Is stats logging enabled for this particular template
- `ether_type`: Ether type of the packet to be matched. etherType can be \* or a valid hexadecimal value
- `external_id`: External object ID. Used for integration with third party systems

## 213.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas

## 213.3 Parents

- *nudomainfipacltemplate.NUDomainFIPAcTemplate*

## NUDOMAINTEMPLATE

`nudomaintemplate.NUDomainTemplate(bambou.nurest_object.NUMetaRESTObject, ) :`

Domains in VSD are created from domain templates. This object provides the definition of the DomainTemplate.

### 214.1 Attributes

- **name (Mandatory):** The name of the domain template, that is unique within an enterprise. Valid characters are alphabets, numbers, space and hyphen( - ).
- **last\_updated\_by:** ID of the user who last updated the object.
- **description:** Domain template description provided by the user
- **encryption:** Determines whether IPSEC is enabled. Possible values are ENABLED, DISABLED, .
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **policy\_change\_status:** None
- **associated\_multicast\_channel\_map\_id:** The ID of the Multi Cast Channel Map this domain template is associated with. This has to be set when enableMultiCast is set to ENABLED
- **multicast:** Indicates multicast policy on domain.
- **external\_id:** External object ID. Used for integration with third party systems

## 214.2 Children

class	fetcher
<i>nuredirectiontargettemplate.NURedirectionTargetTemplate</i>	redirection_target_templates
<i>nupermision.NUPermission</i>	permissions
<i>numetadata.NUMetadata</i>	metadatas
<i>nuegressacltemplate.NUEgressACLTemplate</i>	egress_acl_templates
<i>nudomainfipacltemplate.NUDomainFIPACLTemplate</i>	domain_fip_acl_templates
<i>nufloatingipacltemplate.NUFloatingIPACLTemplate</i>	floating_ipacl_templates
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>ningressacltemplate.NUIngressACLTemplate</i>	ingress_acl_templates
<i>ningressadvfwdtemplate.NUIngressAdvFwdTemplate</i>	ingress_adv_fwd_templates
<i>ningressexternalservicetemplate.NUIngressExternalServiceTemplate</i>	ingress_external_service_templates
<i>nujob.NUJob</i>	jobs
<i>nupolicygrouptemplate.NUPolicyGroupTemplate</i>	policy_group_templates
<i>nudomain.NUDomain</i>	domains
<i>nuzonetemplate.NUZoneTemplate</i>	zone_templates
<i>nuqos.NUQOS</i>	qoss
<i>nugroup.NUGroup</i>	groups
<i>nusubnettemplate.NUSubnetTemplate</i>	subnet_templates
<i>nueventlog.NUEventLog</i>	event_logs

## 214.3 Parents

- *nudomain.NUDomain*
- *nuenterprise.NUEnterprise*



## NUDSCPFORWARDINGCLASSMAPPING

`nudscpforwardingclassmapping.NUDSCPForwardingClassMapping(bambou.nurest_object.NUMetaRESTO`

Provides the definition of a single DSCP to a Forwarding class mapping that is part of a Table used in QoS policies.

### 215.1 Attributes

- `dscp`: DSCP value range from enumeration of 65 values : \*, 0, 1, ..., 63
- `last_updated_by`: ID of the user who last updated the object.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `forwarding_class`: Class of service to be used. Service classes in order of priority are A, B, C, D, E, F, G, and H.
- `external_id`: External object ID. Used for integration with third party systems

### 215.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 215.3 Parents

- `nudscpforwardingclasstable.NUDSCPForwardingClassTable`



## NUDSCPFORWARDINGCLASSTABLE

`nudscpforwardingclasstable.NUDSCPForwardingClassTable (bambou.nurest_object.NUMetaRESTObject`

Provides the definition of a table that holds multiple DSCP to Forwarding class mappings. Used in QoS policies.

### 216.1 Attributes

- **name (Mandatory):** A unique name of the dscp-fc mapping table.
- **last\_updated\_by:** ID of the user who last updated the object.
- **description:** A description of the dscp-fc mapping table.
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **external\_id:** External object ID. Used for integration with third party systems

### 216.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nudscpforwardingclassmapping.NUDSCPForwardingClassMapping</i>	dscp_forwarding_class_mappings

### 216.3 Parents

- *nuenterprise.NUEnterprise*



## NUEGRESSACLENTRYTEMPLATE

`nuegressaclentrytemplate.NUEgressACLEntryTemplate(bambou.nurest_object.NUMetaRESTObject,)` :

Defines the template of Egress ACL Template entries

### 217.1 Attributes

- `dscp` (**Mandatory**): DSCP match condition to be set in the rule. It is either \* or from 0-63
- `last_updated_by`: ID of the user who last updated the object.
- `action` (**Mandatory**): The action of the ACL entry.
- `address_override`: Overrides the source IP for Ingress and destination IP for Egress, macentries will use this address as the match criteria.
- `reflexive`: True means that this ACL entry is reflexive, so there will be a corresponding rule that will be created by OVS in the network. False means that there is no corresponding rule created by OVS in the network.
- `description`: Description of the ACL entry
- `destination_port` (**Mandatory**): The destination port to be matched if protocol is UDP or TCP. Value should be either \* or single port number or a port range
- `network_id`: The destination network entity that is referenced(subnet/zone/macro)
- `network_type` (**Mandatory**): Type of the source network.
- `flow_logging_enabled`: Is flow logging enabled for this particular template
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `location_id`: The ID of the location entity (Subnet/Zone/VportTag)
- `location_type` (**Mandatory**): Type of the location entity.
- `policy_state`: State of the policy. Possible values are DRAFT, LIVE, .
- `source_port` (**Mandatory**): Source port to be matched if protocol is UDP or TCP. Value can be either \* or single port number or a port range
- `priority`: The priority of the ACL entry that determines the order of entries
- `protocol` (**Mandatory**): Protocol number that must be matched
- `associated_application_id`: The associated application ID
- `associated_application_object_id`: The associated application object ID

- `associated_application_object_type`: The associated application object type Refer to API section for supported types.
- `associated_live_entity_id`: In the draft mode, the ACL entry refers to this LiveEntity. In non-drafted mode, this is null.
- `stats_id`: The statsID that is created in the VSD and identifies this ACL Template Entry. This is auto-generated by VSD
- `stats_logging_enabled`: Is stats logging enabled for this particular template
- `ether_type` (**Mandatory**): Ether type of the packet to be matched. etherType can be \* or a valid hexadecimal value
- `external_id`: External object ID. Used for integration with third party systems

## 217.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nujob.NUJob</i>	jobs
<i>nustatistics.NUStatistics</i>	statistics

## 217.3 Parents

- *nudomain.NUDomain*
- *nul2domain.NUL2Domain*
- *nume.NUMe*
- *nuegressacltemplate.NUEgressACLTemplate*

## NUEGRESSACLTEMPLATE

`nuegressacltemplate.NUEgressACLTemplate(bambou.nurest_object.NUMetaRESTObject, ) :`

Defines the template for an Egress ACL.

### 218.1 Attributes

- **name (Mandatory):** The name of the entity
- **last\_updated\_by:** ID of the user who last updated the object.
- **active:** If enabled, it means that this ACL or QOS entry is active
- **default\_allow\_ip:** If enabled a default ACL of Allow All is added as the last entry in the list of ACL entries
- **default\_allow\_non\_ip:** If enabled, non ip traffic will be dropped
- **default\_install\_acl\_implicit\_rules:** If enabled, implicit rule will allow intra domain traffic by default
- **description:** A description of the entity
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **policy\_state:**
- **priority:** The priority of the ACL entry that determines the order of entries
- **priority\_type:**
- **associated\_live\_entity\_id:** In the draft mode, the ACL entry refers to this LiveEntity. In non-drafted mode, this is null.
- **external\_id:** External object ID. Used for integration with third party systems

### 218.2 Children

<b>class</b>	<b>fetcher</b>
<i>numetadata.NUMetadata</i>	metadatas
<i>nuegressaclentrytemplate.NUEgressACLEntryTemplate</i>	egress_acl_entry_templates
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuvm.NUVM</i>	vms
<i>nujob.NUJob</i>	jobs
<i>nueventlog.NUEventLog</i>	event_logs

## 218.3 Parents

- *nudomain.NUDomain*
- *nul2domaintemplate.NUL2DomainTemplate*
- *nul2domain.NUL2Domain*
- *nume.NUMe*
- *nudomaintemplate.NUDomainTemplate*



## NUEGRESSQOSPOLICY

`nuegressqospolicy.NUEgressQOSPolicy(bambou.nurest_object.NUMetaRESTObject, ) :`

The object manipulates Egress QoS parameters attached to a Access Port / VLAN or Network port.

### 219.1 Attributes

- **name (Mandatory):** A unique name of the QoS object
- **parent\_queue\_associated\_rate\_limiter\_id:** ID of the parent rate limiter associated with this Egress QOS policy.
- **last\_updated\_by:** ID of the user who last updated the object.
- **description:** A description of the QoS object
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **assoc\_egress\_qos\_id:** ID of object associated with this QoS object
- **queue1\_associated\_rate\_limiter\_id:** ID of the queue1 rate limiter associated with this Egress QOS policy.
- **queue1\_forwarding\_classes:** Queue1 Forwarding Classes for this Egress QOS Policy Possible values are NONE, A, B, C, D, E, F, G, H, .
- **queue2\_associated\_rate\_limiter\_id:** ID of the queue2 rate limiter associated with this Egress QOS policy.
- **queue2\_forwarding\_classes:** Queue2 Forwarding Classes for this Egress QOS Policy Possible values are NONE, A, B, C, D, E, F, G, H, .
- **queue3\_associated\_rate\_limiter\_id:** ID of the queue3 rate limiter associated with this Egress QOS policy.
- **queue3\_forwarding\_classes:** Queue3 Forwarding Classes for this Egress QOS Policy Possible values are NONE, A, B, C, D, E, F, G, H, .
- **queue4\_associated\_rate\_limiter\_id:** ID of the queue4 rate limiter associated with this Egress QOS policy.
- **queue4\_forwarding\_classes:** Queue4 Forwarding Classes for this Egress QOS Policy Possible values are NONE, A, B, C, D, E, F, G, H, .
- **external\_id:** External object ID. Used for integration with third party systems

## 219.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas

## 219.3 Parents

- *nume.NUMe*
- *nuenterprise.NUEnterprise*

## NUENDPOINT

`nuendpoint.NUEndPoint(bambou.nurest_object.NUMetaRESTObject, ) :`

Representation of End Point

### 220.1 Attributes

- `name`: unique name of the External Service.
- `last_updated_by`: ID of the user who last updated the object.
- `description`: Description of the External Service.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 220.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nueventlog.NUEventLog</i>	event_logs

### 220.3 Parents

- *nuexternalservice.NUExternalService*



## NUENTERPRISE

`nenterprise.NUEnterprise(bambou.nurest_object.NUMetaRESTObject, ) :`

Definition of the enterprise object. This is the top level object that represents an organization.

### 221.1 Attributes

- `ldap_authorization_enabled`: Read-only flag - indicates if LDAP is used for authorization for the enterprise. For detailed explanation, see definition in `LDAPConfiguration` class
- `ldap_enabled`: Read-only flag - indicates if LDAP is used for authentication for the enterprise. For detailed explanation, see definition in `LDAPConfiguration` class
- `dhcp_lease_interval`: DHCP Lease Interval (in hrs) to be used by an enterprise.
- `name` (**Mandatory**): The unique name of the enterprise. Valid characters are alphabets, numbers, space and hyphen( - ).
- `last_updated_by`: ID of the user who last updated the object.
- `receive_multi_cast_list_id`: Readonly Id of the auto generated receive multicast list associated with this enterprise profile
- `send_multi_cast_list_id`: Readonly Id of the auto generated send multicast list associated with this enterprise profile
- `description`: A description of the enterprise
- `allow_advanced_qos_configuration`: Controls whether this enterprise has access to advanced QoS settings
- `allow_gateway_management`: This flag indicates if the enterprise/organization can manage gateways. If yes then it can create gateway templates, instantiate them etc.
- `allow_trusted_forwarding_class`: Controls whether QoS policies and templates created under this enterprise set the trusted flag to true
- `allowed_forwarding_classes`: Allowed Forwarding Classes for this enterprise. Possible values are NONE, A, B, C, D, E, F, G, H, .
- `floating_ips_quota`: Quota set for the number of floating IPs to be used by an enterprise.
- `floating_ips_used`: Number of floating IPs used by the enterprise from the assigned `floatingIPsQuota`
- `encryption_management_mode`: Readonly encryption management mode of the associated profile
- `enterprise_profile_id`: Enterprise profile id for this enterprise
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level

- `associated_enterprise_security_id`: Readonly Id of the associated group key encryption profile
- `associated_group_key_encryption_profile_id`: Readonly Id of the associated group key encryption profile
- `associated_key_server_monitor_id`: Readonly Id of the associated keyserver monitor
- `customer_id`: CustomerID that is used by VSC to identify this enterprise. This is a read only attribute.
- `avatar_data`: URL to the avatar data associated with the enterprise. If the `avatarType` is URL then value of `avatarData` should an URL of the image. If the `avatarType` BASE64 then `avatarData` should be BASE64 encoded value of the image
- `avatar_type`: Avatar type - URL or BASE64 Possible values are URL, BASE64, COMPUTEDURL, .
- `external_id`: External object ID. Used for integration with third party systems

## 221.2 Children

class	fetcher
<i>nul2domain.NUL2Domain</i>	<code>l2_domains</code>
<i>nul2domaintemplate.NUL2DomainTemplate</i>	<code>l2_domain_templates</code>
<i>nuratelimiter.NURateLimiter</i>	<code>rate_limiters</code>
<i>nugateway.NUGateway</i>	<code>gateways</code>
<i>nugatewaytemplate.NUGatewayTemplate</i>	<code>gateway_templates</code>
<i>nupatnatpool.NUPATNATPool</i>	<code>patnat_pools</code>
<i>nuldapconfiguration.NULDAPConfiguration</i>	<code>ldap_configurations</code>
<i>nuredundancygroup.NURedundancyGroup</i>	<code>redundancy_groups</code>
<i>numetadata.NUMetadata</i>	<code>metadatas</code>
<i>numetadatatag.NUMetadataTag</i>	<code>metadata_tags</code>
<i>nunetworkmacrogroup.NUNetworkMacroGroup</i>	<code>network_macro_groups</code>
<i>nukeyservermonitor.NUKeyServerMonitor</i>	<code>key_server_monitors</code>
<i>nuegressqospolicy.NUEgressQOSPolicy</i>	<code>egress_qos_policies</code>
<i>nusharednetworkresource.NUSharedNetworkResource</i>	<code>shared_network_resources</code>
<i>nualarm.NUAlarm</i>	<code>alarms</code>
<i>nualarm.NUAlarm</i>	<code>alarms</code>
<i>nuglobalmetadata.NUGlobalMetadata</i>	<code>global_metadatas</code>
<i>nuvm.NUVM</i>	<code>vms</code>
<i>nuinfrastructureportprofile.NUInfrastructurePortProfile</i>	<code>infrastructure_port_profiles</code>
<i>nuenterprisenetwork.NUEnterpriseNetwork</i>	<code>enterprise_networks</code>
<i>nuenterprisecurity.NUEnterpriseSecurity</i>	<code>enterprise_securities</code>
<i>nujob.NUJob</i>	<code>jobs</code>
<i>nudomain.NUDomain</i>	<code>domains</code>
<i>nudomaintemplate.NUDomainTemplate</i>	<code>domain_templates</code>
<i>nuapp.NUApp</i>	<code>apps</code>
<i>nuapplicationsevice.NUApplicationService</i>	<code>application_services</code>
<i>nugroup.NUGroup</i>	<code>groups</code>
<i>nugroupkeyencryptionprofile.NUGroupKeyEncryptionProfile</i>	<code>group_key_encryption_profiles</code>
<i>nudscpforwardingclasstable.NUDSCPForwardingClassTable</i>	<code>dscp_forwarding_class_tables</code>
<i>nuuser.NUUser</i>	<code>users</code>
<i>nunsgateway.NUNSGateway</i>	<code>ns_gateways</code>
<i>nunsgatewaytemplate.NUNSGatewayTemplate</i>	<code>ns_gateway_templates</code>

Continued on next page

Table 221.1 – continued from previous page

<i>nunsredundantgatewaygroup.NUNSRedundantGatewayGroup</i>	ns_redundant_gateway_groups
<i>nupublicnetworkmacro.NUPublicNetworkMacro</i>	public_network_macros
<i>numulticastlist.NUMultiCastList</i>	multi_cast_lists
<i>nueventlog.NUEventLog</i>	event_logs
<i>nuexternalappservice.NUExternalAppService</i>	external_app_services
<i>nuexternalservice.NUExternalService</i>	external_services

## 221.3 Parents

- *nuenterpriseprofile.NUEnterpriseProfile*
- *nume.NUMe*





## NUENTERPRISENETWORK

`numenterprisenetwork.NUEnterpriseNetwork(bambou.nurest_object.NUMetaRESTObject, ) :`

Administrators of an enterprise can define macros that are set of IP addresses that identify enterprise networks. These macros can be used in the ACL definitions by network designers and other users to identify access restrictions towards specific enterprise networks.

### 222.1 Attributes

- `ip_type`: IPv4 or IPv6(only IPv4 is supported in R1.0) Possible values are IPV4, IPV6, .
- `name` (**Mandatory**): Name of the current entity(Zone or zone template or subnet etc..) Valid characters are alphabets, numbers, space and hyphen( - ).
- `last_updated_by`: ID of the user who last updated the object.
- `address` (**Mandatory**): IP address of the subnet defined. In case of zone, this is an optional field for and allows users to allocate an IP address range to a zone. The VSD will auto-assign IP addresses to subnets from this range if a specific IP address is not defined for the subnet
- `netmask` (**Mandatory**): Netmask of the subnet defined
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 222.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nunetworkmacrogroup.NUNetworkMacroGroup</i>	network_macro_groups
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nueventlog.NUEventLog</i>	event_logs

### 222.3 Parents

- *nunetworkmacrogroup.NUNetworkMacroGroup*
- *numenterprise.NUEnterprise*



## NUENTERPRISEPERMISSION

`numenterprisepermission.NUEnterprisePermission(bambou.nurest_object.NUMetaRESTObject,)` :

Represents Enterprise Permission for a CSP entity.

### 223.1 Attributes

- `name`: Name of the Permission
- `last_updated_by`: ID of the user who last updated the object.
- `permitted_action` (**Mandatory**): The permitted action.
- `permitted_entity_description`: Description for the permittedEntity
- `permitted_entity_id`: The enterprise permitted to use/extend this Gateway
- `permitted_entity_name`: Name of the entity for which we have given permission.
- `permitted_entity_type`: Type of the entity for which we have given permission.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 223.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 223.3 Parents

- `nusharednetworkresource.NUSharedNetworkResource`
- `nuredundancygroup.NURedundancyGroup`
- `nuvsgredundantport.NUVsgRedundantPort`
- `nupatnatpool.NUPATNATPool`
- `nuwanservice.NUWANService`
- `nuport.NUPort`

- *nugateway.NUGateway*
- *nunsgateway.NUNSGateway*
- *nunsport.NUNSPort*
- *nuvlan.NUVLAN*

## NUENTERPRISEPROFILE

`numenterpriseprofile.NUEnterpriseProfile(bambou.nurest_object.NUMetaRESTObject,):`

Enterprise profile, used to store an enterprise's policies, quota etc.

### 224.1 Attributes

- `dhcp_lease_interval`: DHCP Lease Interval (in hours) to be used by an enterprise.
- `name` (**Mandatory**): The unique name of the enterprise. Valid characters are alphabets, numbers, space and hyphen ( - ).
- `last_updated_by`: ID of the user who last updated the object.
- `receive_multi_cast_list_id`: Readonly ID of the auto generated receive multicast list associated with this enterprise profile
- `send_multi_cast_list_id`: Readonly ID of the auto generated send multicast list associated with this enterprise profile
- `description`: A description of the enterprise/organisation profile.
- `allow_advanced_qos_configuration`: Controls whether this enterprise has access to advanced QoS settings.
- `allow_gateway_management`: If set to true lets the enterprise admin create gateway templates and instances.
- `allow_trusted_forwarding_class`: Controls whether QoS policies and templates created under this enterprise set the trusted flag to true
- `allowed_forwarding_classes`: Allowed Forwarding Classes for this enterprise. Possible values are NONE, A, B, C, D, E, F, G, H, .
- `floating_ips_quota`: Quota set for the number of floating IPs to be used by an enterprise.
- `encryption_management_mode`: encryption management mode for this enterprise Possible values are DISABLED, MANAGED, .
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

## 224.2 Children

<b>class</b>	<b>fetcher</b>
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuenterprise.NUEnterprise</i>	enterprises
<i>numulticastlist.NUMultiCastList</i>	multi_cast_lists
<i>nueventlog.NUEventLog</i>	event_logs
<i>nuexternalservice.NUExternalService</i>	external_services

## 224.3 Parents

- *nume.NUMe*

## NUENTERPRISESECUREDATA

`numenterprisesecuredata.NUEnterpriseSecuredData(bambou.nurest_object.NUMetaRESTObject, ) :`

This object represents the secured data object under the enterprise

### 225.1 Attributes

- `hash`: authentication hash
- `last_updated_by`: ID of the user who last updated the object.
- `data`: encrypted data
- `sek_id`: Seed Encryption Key id that encrypted this data
- `keyserver_cert_serial_number`: Serial Number of the certificate needed to verify the encrypted data
- `signed_hash`: private key signed data
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 225.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 225.3 Parents

- `numenterprisesecurity.NUEnterpriseSecurity`





## NUENTERPRISESECURITY

`numenterprisesecurity.NUEnterpriseSecurity(bambou.nurest_object.NUMetaRESTObject, ) :`

This object represents the enterprise security

### 226.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `gateway_security_revision`: change revision number for the gateway security data
- `revision`: revision number for the enterprise security data
- `enterprise_id`: The enterprise associated with this object. This is a read only attribute
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 226.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>numenterprisesecureddata.NUEnterpriseSecuredData</code>	<code>enterprise_secured_datas</code>

### 226.3 Parents

- `numenterprise.NUEnterprise`



## NUEVENTLOG

`nueventlog.NUEventLog(bambou.nurest_object.NUMetaRESTObject, ) :`

The API retrieves the events related to a particular entity

### 227.1 Attributes

- `diff`: Holds the results of diff between two objects of same type.
- `enterprise`: The enterprise name of the user who triggered this event.
- `entities`: List of entities associated with the event.
- `entity_id`: The entity id associated with this event.
- `entity_parent_id`: The entity parent id associated with this event. It can be null.
- `entity_parent_type`: Event parent entity type. Generally reported against enterprise.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `entity_type`: The entity type of this event. It may be Domain, VirtualMachine, etc.,
- `user`: The authenticated user who triggered this event.
- `event_received_time`: The time that event was received.
- `external_id`: External object ID. Used for integration with third party systems
- `type`: The event type (CREATE, UPDATE or DELETE).

### 227.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 227.3 Parents

- `nuqos.NUQOS`
- `nuvirtualip.NUVirtualIP`
- `numulticastchannelmap.NUMultiCastChannelMap`

- *nuredundancygroup.NURedundancyGroup*
- *nutca.NUTCA*
- *nugroup.NUGroup*
- *nuzone.NUZone*
- *nuapp.NUApp*
- *nuflowsecuritypolicy.NUFlowSecurityPolicy*
- *nuenterprisenetwork.NUEnterpriseNetwork*
- *nuppermission.NUPermission*
- *nuireservation.NUIPReservation*
- *nuredirectiontargettemplate.NURedirectionTargetTemplate*
- *nusubnettemplate.NUSubnetTemplate*
- *numetadatatag.NUMetadataTag*
- *nuredirectiontarget.NURedirectionTarget*
- *numetadata.NUMetadata*
- *nuvsp.NUVSP*
- *nudomain.NUDomain*
- *nuwanservice.NUWANService*
- *nuvsd.NUVSD*
- *nuvport.NUVPort*
- *nuflowforwardingpolicy.NUFlowForwardingPolicy*
- *nuport.NUPort*
- *nusubnet.NUSubnet*
- *nupolicygroup.NUPolicyGroup*
- *nuzonetemplate.NUZoneTemplate*
- *nuvrs.NUVRS*
- *nuhsc.NUHSC*
- *nustaticroute.NUStaticRoute*
- *nuvminterface.NUVMInterface*
- *nulicense.NULicense*
- *nuenterpriseprofile.NUEnterpriseProfile*
- *nupolicygrouptemplate.NUPolicyGroupTemplate*
- *nubridgeinterface.NUBridgeInterface*
- *numulticastrange.NUMultiCastRange*
- *nuautodiscoveredgateway.NUAutoDiscoveredGateway*
- *nutier.NUTier*
- *nugateway.NUGateway*

- *nul2domaintemplate.NUL2DomainTemplate*
- *nuapplicationservice.NUApplicationService*
- *nul2domain.NUL2Domain*
- *nuhostinterface.NUHostInterface*
- *nuendpoint.NUEndPoint*
- *nudhcption.NUDHCPOption*
- *nunsgateway.NUNSGateway*
- *nuvsc.NUVSC*
- *nuvm.NUVM*
- *nunsport.NUNSPort*
- *nupublicnetworkmacro.NUPublicNetworkMacro*
- *nuaddressrange.NUAddressRange*
- *nudomaintemplate.NUDomainTemplate*
- *nufloatingip.NUFloatingIp*
- *nuegressacltemplate.NUEgressACLTemplate*
- *nunsredundantgatewaygroup.NUNSRedundantGatewayGroup*
- *nuexternalservice.NUExternalService*
- *nuvlan.NUVLAN*
- *nuuser.NUUser*
- *nuflow.NUFlow*
- *nuenterprise.NUEnterprise*
- *nuingressacltemplate.NUIngressACLTemplate*



## NUEXTERNALAPPSERVICE

`nuexternalappservice.NUExternalAppService(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents an External Service in the Application Designer.

### 228.1 Attributes

- `name` (**Mandatory**): Name of the flow.
- `last_updated_by`: ID of the user who last updated the object.
- `description`: Description of the flow.
- `destination_nat_address`: Destination NAT Address
- `destination_nat_enabled`: Boolean flag to indicate whether source NAT is enabled
- `destination_nat_mask`: netmask of the Destination NAT
- `metadata`: metadata
- `egress_type`: Egress type.
- `virtual_ip`: Virtual IP Address
- `virtual_ip_required`: Boolean flag to indicate whether we require a VIP
- `ingress_type`: Ingress type.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `source_nat_address`: Source NAT Address
- `source_nat_enabled`: Boolean flag to indicate whether source NAT is enabled
- `associated_service_egress_group_id`: ID of service port group identifying the output ports
- `associated_service_egress_redirect_id`: the redirect target ID that identifies the output ports
- `associated_service_ingress_group_id`: ID of service port group identifying the input ports
- `associated_service_ingress_redirect_id`: the redirect target ID that identifies the input ports
- `external_id`: External object ID. Used for integration with third party systems

## 228.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas

## 228.3 Parents

- *nudomain.NUDomain*
- *nume.NUMe*
- *numenterprise.NUEnterprise*



## NUEXTERNALSERVICE

`nuexternalservice.NUExternalService(bambou.nurest_object.NUMetaRESTObject, ) :`

Representation of External Service.

### 229.1 Attributes

- `name` (**Mandatory**): unique name of the External Service.
- `last_updated_by`: ID of the user who last updated the object.
- `service_type` (**Mandatory**): Type of the service.
- `description`: Description of the External Service.
- `direction`: Direction
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `stage`: Stage - START,END Possible values are START, .
- `external_id`: External object ID. Used for integration with third party systems

### 229.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>numetadatatag.NUMetadataTag</code>	<code>metadata_tags</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nuendpoint.NUEndPoint</code>	<code>end_points</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>

### 229.3 Parents

- `nuenterpriseprofile.NUEnterpriseProfile`
- `nume.NUMe`
- `nuenterprise.NUEnterprise`



## NUFLOATINGIP

`nufloatingip.NUFloatingIp(bambou.nurest_object.NUMetaRESTObject, ) :`

Floating IP that is associated to a Domain. This floating IP could be used in the VM interface for NAT functionality.

### 230.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `access_control`: If access control is enabled this FIP is part of the Internet PG.
- `address`: Floating IP address assigned to the Domain
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `assigned`: True if this floating IP is assigned to a network interface else the value is false
- `assigned_to_object_type`: The object type to which this floating ip is assigned. Eg. vport or virtualip
- `associated_shared_network_resource_id` (**Mandatory**): Id of the shared network resource subnet which was used to get this floating IP address
- `external_id`: External object ID. Used for integration with third party systems

### 230.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nuvport.NUVPort</code>	<code>vports</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>

### 230.3 Parents

- `nudomain.NUDomain`
- `nume.NUMe`



## NUFLOATINGIPACLTEMPLATE

`nufloatingipacltemplate.NUFloatingIPACLTemplate(bambou.nurest_object.NUMetaRESTObject, ) :`

Defines the template for an Floating IP ACL

### 231.1 Attributes

- `name`: The name of the entity
- `last_updated_by`: ID of the user who last updated the object.
- `active`: If enabled, it means that this ACL or QOS entry is active
- `default_allow_ip`: If enabled a default ACL of Allow All is added as the last entry in the list of ACL entries
- `default_allow_non_ip`: If enabled, non ip traffic will be dropped
- `description`: A description of the entity
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `policy_state`: State of the policy
- `priority`: The priority of the ACL entry that determines the order of entries
- `priority_type`: Priority type
- `associated_live_entity_id`: ID of the associated live entity
- `external_id`: External object ID. Used for integration with third party systems

### 231.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nufloatingipacltemplateentry.NUFloatingIPACLTemplateEntry</code>	<code>floating_ipacl_template_entries</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 231.3 Parents

- `nudomain.NUDomain`

- *nume.NUMe*
- *nudomaintemplate.NUDomainTemplate*

## NUFLOATINGIPACLTEMPLATEENTRY

`nufloatingipacltemplateentry.NUFloatingIPACLTemplateEntry(bambou.nurest_object.NUMetaRESTO`

Defines the template of Egress ACL Template entries

### 232.1 Attributes

- `dscp`: DSCP match condition to be set in the rule. It is either \* or from 0-63
- `last_updated_by`: ID of the user who last updated the object.
- `action`: The action of the ACL entry DROP or FORWARD or REDIRECT. Action REDIRECT is allowed only for IngressAdvancedForwardingEntry
- `address_override`: Overrides the source IP for Ingress and destination IP for Egress, macentries will use this address as the match criteria.
- `reflexive`: True means that this ACL entry is reflexive, so there will be a corresponding rule that will be created by OVS in the network. False means that there is no corresponding rule created by OVS in the network.
- `description`: Description of the ACL entry
- `destination_port`: The destination port to be matched if protocol is UDP or TCP. Value should be either \* or single port number or a port range
- `network_id`: The destination network entity that is referenced(subnet/zone/macro)
- `network_type`: Type of the source network - VM\_SUBNET or VM\_ZONE or VM\_DOMAIN or SUBNET or ZONE or ENTERPRISE\_NETWORK or PUBLIC\_NETWORK or ANY
- `flow_logging_enabled`: Is flow logging enabled for this particular template
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `location_id`: The ID of the location entity (Subnet/Zone/VportTag)
- `location_type`: Type of the location entity - ANY or SUBNET or ZONE or VPORTTAG
- `policy_state`: State of the policy.
- `source_port`: Source port to be matched if protocol is UDP or TCP. Value can be either \* or single port number or a port range
- `priority`: The priority of the ACL entry that determines the order of entries
- `protocol`: Protocol number that must be matched
- `associated_application_id`: The associated application ID
- `associated_application_object_id`: The associated application object ID

- `associated_application_object_type`: The associated application object type
- `associated_live_entity_id`: ID of the associated live entity
- `stats_id`: The statsID that is created in the VSD and identifies this ACL Template Entry. This is auto-generated by VSD
- `stats_logging_enabled`: Is stats logging enabled for this particular template
- `ether_type`: Ether type of the packet to be matched. etherType can be \* or a valid hexadecimal value
- `external_id`: External object ID. Used for integration with third party systems

## 232.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas

## 232.3 Parents

- *nufloatingipacltemplate.NUFloatingIPACLTemplate*



## NUFLOW

`nuflow.NUFlow(bambou.nurest_object.NUMetaRESTObject,):`

Flow represents the traffic between two different application tiers.

### 233.1 Attributes

- **name (Mandatory):** Name of the flow.
- **last\_updated\_by:** ID of the user who last updated the object.
- **description:** Description of the flow.
- **destination\_tier\_id:** Flow destination tier id.
- **metadata:** Metadata field to store flow related data.
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **origin\_tier\_id:** Flow origin tier id.
- **external\_id:** External object ID. Used for integration with third party systems

### 233.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nuflowforwardingpolicy.NUFlowForwardingPolicy</code>	<code>flow_forwarding_policies</code>
<code>nuflowsecuritypolicy.NUFlowSecurityPolicy</code>	<code>flow_security_policies</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>

### 233.3 Parents

- `nuapp.NUApp`



## NUFLOWFORWARDINGPOLICY

`nuflowforwardingpolicy.NUFlowForwardingPolicy(bambou.nurest_object.NUMetaRESTObject,)` :

The redirect policy on the flow.

### 234.1 Attributes

- `redirect_target_id`: The associated service id.
- `destination_address_rewrite`: The destination address rewrite. Needs to be in CIDR format `x.x.x.x/n`
- `flow_id`: The associated service id.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `source_address_rewrite`: The source address rewrite. Needs to be in CIDR format `x.x.x.x/n`
- `associated_application_service_id`: The associated service id.
- `associated_network_object_id`: The associated network object id.
- `associated_network_object_type`: The associated network object type. Refer to API section for supported types.
- `external_id`: External object ID. Used for integration with third party systems
- `type`: The redirect type.

### 234.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>

### 234.3 Parents

- `nuflow.NUFlow`



## NUFLOWSECURITYPOLICY

`nuflowsecuritypolicy.NUFlowSecurityPolicy(bambou.nurest_object.NUMetaRESTObject, ) :`

The security policy on the flow.

### 235.1 Attributes

- `action`: The flow action. The action can be either FORWARD or DROP.
- `destination_address_overwrite`: The destination address overwrite. Needs to be in CIDR format x.x.x.x/n
- `flow_id`: The associated service id.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `source_address_overwrite`: The source address overwrite. Needs to be in CIDR format x.x.x.x/n
- `priority`: The priority of the flow security policy that determines the order of entries.
- `associated_application_service_id`: The associated service id.
- `associated_network_object_id`: The associated network object id.
- `associated_network_object_type`: The associated network object type. Refer to API section for supported types.
- `external_id`: External object ID. Used for integration with third party systems

### 235.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>

### 235.3 Parents

- `nuflow.NUFlow`



## NUGATEWAY

`nugateway.NUGateway(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents Gateway object.

### 236.1 Attributes

- **name (Mandatory):** Name of the Gateway
- **last\_updated\_by:** ID of the user who last updated the object.
- **redundancy\_group\_id:** The Redundancy Gateway Group associated with this Gateway Instance. This is a read only attribute
- **peer:** The System ID of the peer gateway associated with this Gateway instance when it is discovered by the network manager (VSD) as being redundant.
- **template\_id:** The ID of the template that this Gateway was created from. This should be set when instantiating a Gateway
- **pending:** Indicates that this gateway is pending state or state. When in pending state it cannot be modified from REST.
- **permitted\_action:** The permitted action to USE/EXTEND this Gateway.
- **personality (Mandatory):** Personality of the Gateway, cannot be changed after creation.
- **description:** A description of the Gateway
- **enterprise\_id:** The enterprise associated with this Gateway. This is a read only attribute
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **vtep:** Represent the system ID or the Virtual IP of a service used by a Gateway (VSG for now) to establish a tunnel with a remote VSG or hypervisor. The format of this field is consistent with an IP address.
- **auto\_disc\_gateway\_id:** The Auto Discovered Gateway associated with this Gateway Instance
- **external\_id:** External object ID. Used for integration with third party systems
- **system\_id:** Identifier of the Gateway, cannot be modified after creation

## 236.2 Children

<b>class</b>	<b>fetcher</b>
<i>nupatnatpool.NUPATNATPool</i>	patnat_pools
<i>nupermision.NUPermission</i>	permissions
<i>nuwanservice.NUWANService</i>	wan_services
<i>numetadata.NUMetadata</i>	metadatas
<i>nualarm.NUAlarm</i>	alarms
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>numenterprisepermission.NUEnterprisePermission</i>	enterprise_permissions
<i>nujob.NUJob</i>	jobs
<i>nuport.NUPort</i>	ports
<i>nueventlog.NUEventLog</i>	event_logs

## 236.3 Parents

- *nuredundancygroup.NURedundancyGroup*
- *nume.NUMe*
- *numenterprise.NUEnterprise*



## NUGATEWAYSECUREDData

`nugatewaysecureddata.NUGatewaySecuredData (bambou.nurest_object.NUMetaRESTObject, ) :`

This object represents the secured data object under the gateway

### 237.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `data`: encrypted data
- `gateway_cert_serial_number`: Serial Number of the certificate of the public key that encrypted this data
- `keyserver_cert_serial_number`: Serial Number of the certificate needed to verify the encrypted data
- `signed_data`: private key signed data
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 237.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 237.3 Parents

- `nugatewaysecurityprofile.NUGatewaySecurityProfile`
- `nugatewaysecurity.NUGatewaySecurity`



## NUGATEWAYSECURITY

`nugatewaysecurity.NUGatewaySecurity(bambou.nurest_object.NUMetaRESTObject,)` :

This object represents the gateway security object

### 238.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `gateway_id`: The gateway associated with this object. This is a read only attribute
- `revision`: change revision number for the gateway security data
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 238.2 Children

class	fetcher
<i>nugatewaysecureddata.NUGatewaySecuredData</i>	<code>gateway_secured_datas</code>
<i>numetadata.NUMetadata</i>	<code>metadatas</code>
<i>nuglobalmetadata.NUGlobalMetadata</i>	<code>global_metadatas</code>

### 238.3 Parents

- *nunsgateway.NUNSGateway*



## NUGATEWAYSECURITYPROFILE

`nugatewaysecurityprofile.NUGatewaySecurityProfile(bambou.nurest_object.NUMetaRESTObject,)` :

This object represents the gateway security object

### 239.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `gateway_id`: The gateway associated with this object. This is a read only attribute
- `revision`: revision number for the gateway security profile data
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 239.2 Children

class	fetcher
<i>nugatewaysecureddata.NUGatewaySecuredData</i>	<code>gateway_secured_datas</code>
<i>numetadata.NUMetadata</i>	<code>metadatas</code>
<i>nuglobalmetadata.NUGlobalMetadata</i>	<code>global_metadatas</code>

### 239.3 Parents

- *nunsgateway.NUNSGateway*



## NUGATEWAYTEMPLATE

`nugatewaytemplate.NUGatewayTemplate(bambou.nurest_object.NUMetaRESTObject,)` :

Represents Gateway Template object.

### 240.1 Attributes

- **name (Mandatory)**: Name of the Gateway
- **last\_updated\_by**: ID of the user who last updated the object.
- **personality (Mandatory)**: Personality of the Gateway, cannot be changed after creation.
- **description**: A description of the Gateway
- **enterprise\_id**: The enterprise associated with this Gateway. This is a read only attribute
- **entity\_scope**: Specify if scope of entity is Data center or Enterprise level
- **external\_id**: External object ID. Used for integration with third party systems

### 240.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuporttemplate.NUPortTemplate</i>	port_templates

### 240.3 Parents

- *nume.NUMe*
- *numenterprise.NUEnterprise*





## NUGLOBALMETADATA

`nuglobalmetadata.NUGlobalMetadata (bambou.nurest_object.NUMetaRESTObject, ) :`

Metadata associated to a entity.

### 241.1 Attributes

- `name`: name of the Metadata.
- `last_updated_by`: ID of the user who last updated the object.
- `description`: Description of the Metadata.
- `metadata_tag_ids`: metadata tag IDs associated with this metadata you can filter metadata based on this attribute for example X-Nuage-Filter: '2d6fb627-603b-421c-b63a-eb0a6d712761' IN metadataTagIDs
- `network_notification_disabled`: specifies metadata changes need to be notified to controller,by default it is notified
- `blob` (**Mandatory**): Metadata that describes about the entity attached to it.
- `global_metadata`: specifies metadata is global or local
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 241.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>numetadataatag.NUMetadataTag</code>	<code>metadata_tags</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 241.3 Parents

- `nugroupkeyencryptionprofile.NUGroupKeyEncryptionProfile`
- `nuqos.NUQOS`
- `nubgppeer.NUBGPPeer`
- `nusharednetworkresource.NUSharedNetworkResource`

- *nuvirtualip.NUVirtualIP*
- *nudscpforwardingclasstable.NUDSCPForwardingClassTable*
- *numulticastchannelmap.NUMultiCastChannelMap*
- *nuredundancygroup.NURedundancyGroup*
- *nutca.NUTCA*
- *nugroup.NUGroup*
- *nuvsgredundantport.NUVsgRedundantPort*
- *nuzone.NUZone*
- *nuapp.NUApp*
- *nupatnatpool.NUPATNATPool*
- *nuinfrastructuregatewayprofile.NUInfrastructureGatewayProfile*
- *nuflowsecuritypolicy.NUFlowSecurityPolicy*
- *nuvcentereamconfig.NUVCenterEAMConfig*
- *nulocation.NULocation*
- *nuenterprisesecurity.NUEnterpriseSecurity*
- *nuvcentervrsconfig.NUVCenterVRSCConfig*
- *nuenterprisenetwork.NUEnterpriseNetwork*
- *nuinfrastructurevscprofile.NUInfrastructureVscProfile*
- *nupermision.NUPermission*
- *nuipreservation.NUIPReservation*
- *nuredirectiontargettemplate.NURedirectionTargetTemplate*
- *nusubnettemplate.NUSubnetTemplate*
- *numetadatatag.NUMetadataTag*
- *nufloatingipacltemplate.NUFloatingIPACLTemplate*
- *nuredirectiontarget.NURedirectionTarget*
- *nunetworklayout.NUNetworkLayout*
- *nuingressexternalservicetemplateentry.NUIngressExternalServiceTemplateEntry*
- *nubootstrapactivation.NUBootstrapActivation*
- *nucloudmgmtsystem.NUCloudMgmtSystem*
- *nuvsp.NUVSP*
- *nudomain.NUDomain*
- *nuredundantport.NURedundantPort*
- *nudscpforwardingclassmapping.NUDSCPForwardingClassMapping*
- *nuegressaclentrytemplate.NUEgressACLEntryTemplate*
- *nuwanservice.NUWANService*
- *nuvsd.NUVSD*

- *nuvport.NUVPort*
- *nuflowforwardingpolicy.NUFlowForwardingPolicy*
- *nuport.NUPort*
- *nustatisticspolicy.NUStatisticsPolicy*
- *nusubnet.NUSubnet*
- *nupolicygroup.NUPolicyGroup*
- *nuratelimiter.NURateLimiter*
- *nukeyservermonitorencryptedseed.NUKeyServerMonitorEncryptedSeed*
- *nuzonetemplate.NUZoneTemplate*
- *nukeyservermonitorseed.NUKeyServerMonitorSeed*
- *nugatewaytemplate.NUGatewayTemplate*
- *nuvrs.NUVRS*
- *nuhsc.NUHSC*
- *nuldapconfiguration.NULDAPConfiguration*
- *nuvsdcomponent.NUVSDComponent*
- *nuenterprisepermission.NUEnterprisePermission*
- *nustaticroute.NUStaticRoute*
- *nujob.NUJob*
- *nuvminterface.NUVMInterface*
- *nugatewaysecureddata.NUGatewaySecuredData*
- *nuvcenterhypervisor.NUVCenterHypervisor*
- *nukeyservermonitor.NUKeyServerMonitor*
- *nueventlog.NUEventLog*
- *nulicense.NULicense*
- *nuenterpriseprofile.NUEnterpriseProfile*
- *nuvcenterdatacenter.NUVCenterDataCenter*
- *nupolicygrouptemplate.NUPolicyGroupTemplate*
- *nubridgeinterface.NUBridgeInterface*
- *nuvcentercluster.NUVCenterCluster*
- *numulticastrange.NUMultiCastRange*
- *nunetworkmacrogroup.NUNetworkMacroGroup*
- *nuingressadvfwdentrytemplate.NUIngressAdvFwdEntryTemplate*
- *numulticastlist.NUMultiCastList*
- *nuautodiscoveredgateway.NUAutoDiscoveredGateway*
- *numirrordestination.NUMirrorDestination*
- *nutier.NUTier*

- *nudomainfipacltemplate.NUDomainFIPAcITemplate*
- *nugateway.NUGateway*
- *numultinicvport.NUMultiNICVPort*
- *nustatistics.NUStatistics*
- *nunsporttemplate.NUNSPortTemplate*
- *nucertificate.NUCertificate*
- *nugatewaysecurityprofile.NUGatewaySecurityProfile*
- *nul2domaintemplate.NUL2DomainTemplate*
- *nuapplicationservice.NUApplicationService*
- *nustatscollectorinfo.NUStatsCollectorInfo*
- *nuvcenter.NUVCenter*
- *nuingressaclentrytemplate.NUIngressACLEntryTemplate*
- *nul2domain.NUL2Domain*
- *nuhostinterface.NUHostInterface*
- *nuenterprisesecureddata.NUEnterpriseSecuredData*
- *nuexternalappservice.NUExternalAppService*
- *nuingressexternalservicetemplate.NUIngressExternalServiceTemplate*
- *nuporttemplate.NUPortTemplate*
- *nume.NUMe*
- *nuendpoint.NUEndPoint*
- *nudhcption.NUDHCPOption*
- *nukeyservermember.NUKeyServerMember*
- *nunsgateway.NUNSGateway*
- *nunsgatewaytemplate.NUNSGatewayTemplate*
- *nuvsc.NUVSC*
- *nuuplinkrd.NUUplinkRD*
- *nuvm.NUVM*
- *nunsport.NUNSPort*
- *nuvrsaddressrange.NUVRSAddressRange*
- *nuegressqospolicy.NUEgressQOSPolicy*
- *nupublicnetworkmacro.NUPublicNetworkMacro*
- *nudomainfipacltemplateentry.NUDomainFIPAcITemplateEntry*
- *nuaddressrange.NUAddressRange*
- *nudomaintemplate.NUDomainTemplate*
- *nusiteinfo.NUSiteInfo*
- *nuvmresync.NUVMResync*

- *nupolicydecision.NUPolicyDecision*
- *nufloatingip.NUFloatingIp*
- *nuegressacltemplate.NUEgressACLTemplate*
- *numonitoringport.NUMonitoringPort*
- *nuinfrastructureportprofile.NUInfrastructurePortProfile*
- *nuvpnconnection.NUVPNConnection*
- *nunsredundantgatewaygroup.NUNSRedundantGatewayGroup*
- *nuexternalservice.NUExternalService*
- *nukeyservermonitorek.NUKeyServerMonitorSEK*
- *nuingressadvfwdtemplate.NUIngressAdvFwdTemplate*
- *nuvportmirror.NUVPortMirror*
- *nunspportstaticconfiguration.NUNSPortStaticConfiguration*
- *nualarm.NUAlarm*
- *nuvlan.NUVLAN*
- *nusystemconfig.NUSystemConfig*
- *nufloatingipacltemplateentry.NUFloatingIPACLTemplateEntry*
- *nuinfrastructureconfig.NUInfrastructureConfig*
- *nuuser.NUUser*
- *nunatmapentry.NUNATMapEntry*
- *nualarm.NUAlarm*
- *nubootstrap.NUBootstrap*
- *nuflow.NUFlow*
- *nuvlantemplate.NUVLANTemplate*
- *nugatewaysecurity.NUGatewaySecurity*
- *nuglobalmetadata.NUGlobalMetadata*
- *nuenterprise.NUEnterprise*
- *nuingressacltemplate.NUIngressACLTemplate*



## NUGROUP

`nugroup.NUGroup(bambou.nurest_object.NUMetaRESTObject, ) :`

Identifies a group within an enterprise

### 242.1 Attributes

- **name (Mandatory):** A unique name of the group
- **management\_mode:** Management mode of the user object - allows for override of external authorization and syncup
- **last\_updated\_by:** ID of the user who last updated the object.
- **account\_restrictions:** Determines whether group is disabled or not.
- **description:** Description of the group
- **restriction\_date:** When the group was disabled.
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **role:** The role associated with this group.
- **private:** A private group is visible only by the owner of the group. Public groups are visible by all users in the enterprise
- **external\_id:** External object ID. Used for integration with third party systems

### 242.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuuser.NUUser</i>	users
<i>nueventlog.NUEventLog</i>	event_logs

### 242.3 Parents

- *nuzone.NUZone*
- *nudomain.NUDomain*

- *nul2domaintemplate.NUL2DomainTemplate*
- *nul2domain.NUL2Domain*
- *nudomaintemplate.NUDomainTemplate*
- *nuuser.NUUser*
- *numenterprise.NUEnterprise*



## NUGROUPKEYENCRYPTIONPROFILE

`nugroupkeyencryptionprofile.NUGroupKeyEncryptionProfile(bambou.nurest_object.NUMetaRESTObj`

Represents a Group Key Profile

### 243.1 Attributes

- `sek_generation_interval`: Group Key SEK Generation Interval in Seconds. Min=1, Max=86400
- `sek_lifetime`: Group Key SEK Lifetime in Seconds. Min=1, Max=86400
- `sek_payload_encryption_algorithm`: Group Key SEK Payload Encryption Algorithm.
- `sek_payload_encryption_bc_algorithm`: Group Key Sek Payload Encryption BC Algorithm (read only)
- `sek_payload_encryption_key_length`: Group Key Sek Payload Encryption Key Length (read only)
- `sek_payload_signing_algorithm`: Group Key SEK Payload Signature Algorithm.
- `name`: Name of the Encryption Profile
- `last_updated_by`: ID of the user who last updated the object.
- `seed_generation_interval`: Group Key SEED Generation Interval in Seconds.
- `seed_lifetime`: Group Key SEED Lifetime in Seconds. Min=1, Max=86400
- `seed_payload_authentication_algorithm`: Group Key SEK Payload Signature Algorithm.
- `seed_payload_authentication_bc_algorithm`: Group Key Seed Payload Authentication Algorithm (read only)
- `seed_payload_authentication_key_length`: Group Key Seed Payload Authentication Key Length (read only)
- `seed_payload_encryption_algorithm`: Group Key SEED Payload Encryption Algorithm.
- `seed_payload_encryption_bc_algorithm`: Group Key Seed Payload Encryption Algorithm (read only)
- `seed_payload_encryption_key_length`: Group Key Seed Payload Encryption Key Length (read only)
- `seed_payload_signing_algorithm`: Group Key Seed Payload Signature Algorithm.
- `description`: A description of the Profile instance created.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level

- `traffic_authentication_algorithm`: Group Key traffic Authentication Algorithm. Possible values are HMAC\_SHA1, HMAC\_SHA256, HMAC\_SHA384, HMAC\_SHA512, HMAC\_MD5, .
- `traffic_encryption_algorithm`: Group Key traffic Encryption Algorithm. Possible values are AES\_128\_CBC, AES\_192\_CBC, AES\_256\_CBC, TRIPLE\_DES\_CBC, .
- `traffic_encryption_key_lifetime`: Group Key Traffic Encryption Key Lifetime in Seconds. Min=1, Max=86400
- `external_id`: External object ID. Used for integration with third party systems

## 243.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

## 243.3 Parents

- `numenterprise.NUEnterprise`

## NUHOSTINTERFACE

`nuhostinterface.NUHostInterface(bambou.nurest_object.NUMetaRESTObject, ) :`

Provides information for each host interface.

### 244.1 Attributes

- **mac (Mandatory):** MAC address of the interface, cannot be modified after creation.
- **ip\_address:** IP address of the interface
- **vport\_id:** ID of the vport that the interface is attached to
- **vport\_name:** Name of the vport that the VM is attached to
- **name:** Device name associated with this interface
- **last\_updated\_by:** ID of the user who last updated the object.
- **gateway:** Gateway of the subnet that the VM is connected to
- **netmask:** Netmask of the subnet that the VM is attached to
- **network\_name:** Name of the network that the VM is attached to
- **tier\_id:** ID of the tier that the interface is attached to.
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **policy\_decision\_id:** The policy decision ID for this particular interface
- **domain\_id:** ID of the domain that the VM is attached to
- **domain\_name:** Name of the domain that the VM is attached to
- **zone\_id:** ID of the zone that the interface is attached to
- **zone\_name:** Name of the zone that the VM is attached to
- **associated\_floating\_ip\_address:** Floating Ip Address of this network interface eg: 10.1.2.1
- **attached\_network\_id:** ID of the l2 domain or Subnet that the VM is attached to
- **attached\_network\_type:** l2 domain or Subnet that the interface is attached to
- **external\_id:** External object ID. Used for integration with third party systems

## 244.2 Children

<b>class</b>	<b>fetcher</b>
<i>nutca.NUTCA</i>	tcas
<i>nuredirectiontarget.NURedirectionTarget</i>	redirection_targets
<i>numetadata.NUMetadata</i>	metadatas
<i>nudhcpoption.NUDHCPOption</i>	dhcp_options
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nupolicydecision.NUPolicyDecision</i>	policy_decisions
<i>nupolicygroup.NUPolicyGroup</i>	policy_groups
<i>nuqos.NUQOS</i>	qoss
<i>nustaticroute.NUStaticRoute</i>	static_routes
<i>nustatistics.NUStatistics</i>	statistics
<i>numulticastchannelmap.NUMultiCastChannelMap</i>	multi_cast_channel_maps
<i>nueventlog.NUEventLog</i>	event_logs

## 244.3 Parents

- *nudomain.NUDomain*
- *nuvport.NUVPort*
- *nul2domain.NUL2Domain*
- *nume.NUMe*

`nuhsc.NUHSC(bambou.nurest_object.NUMetaRESTObject, ) :`

System Monitoring details for hardware service controllers.

## 245.1 Attributes

- `name`: Identifies the entity with a name.
- `management_ip`: The management IP of the VSC/HSC entity
- `last_state_change`: Last state change timestamp (in millis).
- `last_updated_by`: ID of the user who last updated the object.
- `address`: The IP of the VRS entity
- `peak_cpuusage`: Peek CPU usage percentage.
- `peak_memory_usage`: Peek memory usage percentage.
- `description`: Description of the entity.
- `messages`: An array of degraded messages.
- `disks`: Set of disk usage details.
- `already_marked_for_unavailable`: Flag to indicate that it is already marked a unavailable.
- `unavailable_timestamp`: The duration the controller is unavailable (in millis).
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `location`: Identifies the entity to be associated with a location.
- `model`: The model of the hardware service controller
- `product_version`: Product version supported by this entity.
- `vsds`: A collection of VSD id(s) which are identified by this controller.
- `status`: Computed status of the entity. Possible values are UP, DOWN, ADMIN\_DOWN, .
- `current_cpuusage`: Current CPU usage percentage.
- `current_memory_usage`: Current memory usage percentage.
- `average_cpuusage`: Average CPU usage percentage.
- `average_memory_usage`: Average memory usage percentage.
- `external_id`: External object ID. Used for integration with third party systems

- `type`: The type of the hardware service controller.

## 245.2 Children

<b>class</b>	<b>fetcher</b>
<i>numetadata.NUMetadata</i>	metadatas
<i>nubgppeer.NUBGPPeer</i>	bgp_peers
<i>nualarm.NUAlarm</i>	alarms
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nujob.NUJob</i>	jobs
<i>numonitoringport.NUMonitoringPort</i>	monitoring_ports
<i>nuvrs.NUVRS</i>	vrss
<i>nueventlog.NUEventLog</i>	event_logs

## 245.3 Parents

- *nuvsp.NUVSP*
- *nuvrs.NUVRS*

## NUINFRASTRUCTURECONFIG

`nuinfrastructureconfig.NUInfrastructureConfig(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents Infrastructure Config

### 246.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `config`: Infrastructure Config
- `config_status`: Status of the configuration application
- `external_id`: External object ID. Used for integration with third party systems

### 246.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 246.3 Parents

- `nunsgateway.NUNSGateway`





## NUINFRASTRUCTUREGATEWAYPROFILE

`nuinfrastructuregatewayprofile.NUInfrastructureGatewayProfile (bambou.nurest_object.NUMetaR`

Represents Infrastructure Gateway Profile

### 247.1 Attributes

- `ntp_server_key`: If set, this represents the security key for the Gateway to communicate with the NTP server (a VSC).
- `ntp_server_key_id`: Correspond to the key ID on the NTP server that matches the `ntpServerKey` value. Valid values are from 1 to 255 as specified by SR-OS and 0 to specify unused (VSD/NSG only).
- `name` (**Mandatory**): Name of the Infrastructure Profile
- `last_updated_by`: ID of the user who last updated the object.
- `datapath_sync_timeout`: Datapath flows sync-time-interval specified in milliseconds (default: 1000)
- `dead_timer`: Time, in seconds, allowed for a Gateway to be inactive before the VSD revokes its certificates and marks it as untrusted.
- `remote_log_dir_path`: Path on the remote log server where the logs generated by the NSG are to be stored. This field is only useful for SCP and SFTP.
- `remote_log_mode`: Type of Log Server for system logs generated by Gateways associated with this Infrastructure Profile.
- `remote_log_password`: Password to be used when accessing the remote log server via SCP or SFTP. This field is only useful for SCP and SFTP.
- `remote_log_server_address`: Primary Log Server for system logs generated by Gateways associated with this Infrastructure Profile. Can be an IP address or a URL. This field is optional.
- `remote_log_server_port`: Port to be used to access the Remote Syslog server. By default, this is port 514.
- `remote_log_username`: Username to be used when accessing the remote log server via SCP or SFTP. This field is only useful for SCP and SFTP.
- `description`: A description of the Profile instance created.
- `metadata_upgrade_path`: Path/URL to retrieve the NSG Upgrade information meta data files. From that meta data, the NSG will be able to retrieve the upgrade package files and perform some validations. It is expected that the meta data file is in JSON format. RFC 2616 states that there are no 'official' maximum length for a URL but different browsers and servers have limits. Our friendly Internet Explorer has a maximum of 'around' 2048 characters, we shall use this as a limit here.

- `flow_eviction_threshold`: Number of flows at which eviction from kernel flow table will be triggered (default: 2500)
- `enterprise_id`: Enterprise/Organisation associated with this Profile instance.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `upgrade_action`: Upgrade action for NSG associated with this Infrastructure Gateway Profile instance.
- `proxy_dns_name` (**Mandatory**): Proxy DNS Name : DNS Name of the system acting as a proxy between the NSG instances and the VSD.
- `use_two_factor`: Use Two Factor : When set to true, the use of two independent authentication factors will be used to secure the installed NSG. When set to false, there is an assumption that the NSG is being installed in a secure environment and the installer is also trusted. The default value is true, using 2-factor.
- `stats_collector_port`: The port to open by the proxy for stats collector to use
- `external_id`: External object ID. Used for integration with third party systems
- `system_sync_scheduler`: Time in a Cron format when configuration update are being applied on the Gateway (NSG). This property is linked to `systemSyncWindow`. Default value is every midnight (0 0 \* \* \*). Format: Minutes Hours DayOfMonth Month DayOfWeek
- `system_sync_window`: Length of time, in seconds, given to a Gateway to apply a configuration change. This property is closely linked to `systemSyncScheduler`.

## 247.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

## 247.3 Parents

- `nume.NUMe`

## NUINFRASTRUCTUREPORTPROFILE

`nuinfrastructureportprofile.NUInfrastructurePortProfile (bambou.nurest_object.NUMetaRESTObj`

Represents an Infrastructure Port Profile.

### 248.1 Attributes

- **name (Mandatory):** Name of the Infrastructure Profile
- **last\_updated\_by:** ID of the user who last updated the object.
- **description:** A description of the Profile instance created.
- **enterprise\_id:** Enterprise/Organisation associated with this Profile instance.
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **speed:** Port Speed in Mb/s : Supported Ethernet speeds are 10 (10Base-T), 100 (Fast-ethernet 100Base-TX), 1000 (Gigabit Ethernet 1000Base-T), 10 000 (10 Gigabit Ethernet 10GBase-X), and Auto-Negotiate.
- **uplink\_tag:** To allow prioritisation of traffic, the NSG network ports must be configured with an uplink type or tag value which will be used in the identification of packets being forwarded. That identification is at the base of the selection of which network port will serve in sending packets to the outside world. The default value is PRIMARY. Possible values are PRIMARY, SECONDARY, TERTIARY, UNKNOWN, .
- **mtu:** Port MTU (Maximum Transmission Unit) : The size in octets of the largest protocol data unit (PDU) that the layer can pass on. The default value is normally 1500 octets for Ethernet v2 and can go up to 9198 for Jumbo Frames.
- **duplex:** Port Duplex : Supported values are FULL where both parties can communicate to the other simultaneously and HALF where each party can only communicate to each other in one direction at a time.
- **external\_id:** External object ID. Used for integration with third party systems

### 248.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

## 248.3 Parents

- *nume.NUMe*
- *nuenterprise.NUEnterprise*

## NUINFRASTRUCTUREVSCPROFILE

`nuinfrastructurevscprofile.NUInfrastructureVscProfile (bambou.nurest_object.NUMetaRESTObject`

Represents an Infrastructure VSC Profile.

### 249.1 Attributes

- **name (Mandatory):** Name of the Infrastructure Profile
- **last\_updated\_by:** ID of the user who last updated the object.
- **second\_controller:** Second VSC Controller : IP Address of the secondary VSC system NSG instances associated to this profile will be reaching for.
- **description:** A description of the Profile instance created.
- **first\_controller:** First VSC Controller : IP Address of the first VSC system NSG instances associated to this profile will be reaching for.
- **enterprise\_id:** Enterprise/Organisation associated with this Profile instance.
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **probe\_interval:** Openflow echo timer in millisecond
- **external\_id:** External object ID. Used for integration with third party systems

### 249.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas

### 249.3 Parents

- *nume.NUMe*



## NUINGRESSACLENTRYTEMPLATE

`nuingressaclentrytemplate.NUIngressACLEntryTemplate(bambou.nurest_object.NUMetaRESTObject, )`

Defines the template of Ingress ACL entries

### 250.1 Attributes

- `dscp` (**Mandatory**): DSCP match condition to be set in the rule. It is either \* or from 0-63
- `last_updated_by`: ID of the user who last updated the object.
- `action` (**Mandatory**): The action of the ACL entry DROP or FORWARD or REDIRECT. Action REDIRECT is allowed only for IngressAdvancedForwardingEntry Possible values are DROP, FORWARD, REDIRECT, .
- `address_override`: Overrides the source IP for Ingress and destination IP for Egress, macentries will use this adress as the match criteria.
- `reflexive`: True means that this ACL entry is reflexive, so there will be a corresponding rule that will be created by OVS in the network. False means that there is no corresponding rule created by OVS in the network.
- `description`: Description of the ACL entry
- `destination_port` (**Mandatory**): The destination port to be matched if protocol is UDP or TCP. Value should be either \* or single port number or a port range
- `network_id`: The destination network entity that is referenced(subnet/zone/macro)
- `network_type` (**Mandatory**): Type of the source network.
- `flow_logging_enabled`: Is flow logging enabled for this particular template
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `location_id`: The ID of the location entity (Subnet/Zone/VportTag)
- `location_type` (**Mandatory**): Type of the location entity.
- `policy_state`: State of the policy.
- `source_port` (**Mandatory**): Source port to be matched if protocol is UDP or TCP. Value can be either \* or single port number or a port range
- `priority`: The priority of the ACL entry that determines the order of entries
- `protocol` (**Mandatory**): Protocol number that must be matched
- `associated_application_id`: The associated application ID
- `associated_application_object_id`: The associated application object ID
- `associated_application_object_type`: The associated application object type Refer to API section for supported types.

- `associated_live_entity_id`: In the draft mode, the ACL entry refers to this LiveEntity. In non-drafted mode, this is null.
- `stats_id`: The statsID that is created in the VSD and identifies this ACL Template Entry. This is auto-generated by VSD
- `stats_logging_enabled`: Is stats logging enabled for this particular template
- `ether_type` (**Mandatory**): Ether type of the packet to be matched. etherType can be \* or a valid hexadecimal value
- `external_id`: External object ID. Used for integration with third party systems

## 250.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nujob.NUJob</i>	jobs
<i>nustatistics.NUStatistics</i>	statistics

## 250.3 Parents

- *nudomain.NUDomain*
- *nul2domain.NUL2Domain*
- *nume.NUMe*
- *nuingressacltemplate.NUIngressACLTemplate*



## NUINGRESSACLTEMPLATE

`nuingressacltemplate.NUIngressACLTemplate (bambou.nurest_object.NUMetaRESTObject, ) :`

Defines the template for an Ingress ACL.

### 251.1 Attributes

- **name (Mandatory):** The name of the entity
- **last\_updated\_by:** ID of the user who last updated the object.
- **active:** If enabled, it means that this ACL or QOS entry is active
- **default\_allow\_ip:** If enabled a default ACL of Allow All is added as the last entry in the list of ACL entries
- **default\_allow\_non\_ip:** If enabled, non ip traffic will be dropped
- **description:** A description of the entity
- **allow\_l2\_address\_spoof:** If enabled, it will disable the default anti-spoof ACL for this domain that essentially prevents any VM to send packets that do not originate from that particular VM
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **policy\_state:**
- **priority:** The priority of the ACL entry that determines the order of entries
- **priority\_type:**
- **assoc\_acl\_template\_id:** ID of the ACL template associated with this ACL template
- **associated\_live\_entity\_id:** In the draft mode, the ACL entry refers to this LiveEntity. In non-drafted mode, this is null.
- **external\_id:** External object ID. Used for integration with third party systems

## 251.2 Children

<b>class</b>	<b>fetcher</b>
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuvm.NUVM</i>	vms
<i>nuingressaclentrytemplate.NUIngressACLEntryTemplate</i>	ingress_acl_entry_templates
<i>nuljob.NULJob</i>	jobs
<i>nueventlog.NUEventLog</i>	event_logs

## 251.3 Parents

- *nudomain.NUDomain*
- *nul2domaintemplate.NUL2DomainTemplate*
- *nul2domain.NUL2Domain*
- *nume.NUMe*
- *nudomaintemplate.NUDomainTemplate*

## NUINGRESSADVFWENTRYTEMPLATE

`nuingressadvfwentrytemplate.NUIngressAdvFwdEntryTemplate (bambou.nurest_object.NUMetaRESTO`

Defines the template of Ingress Advanced Forwarding entries

### 252.1 Attributes

- `fc_override`: Value of the Service Class to be overridden in the packet when the match conditions are satisfied Possible values are NONE, A, B, C, D, E, F, G, H, .
- `dscp` (**Mandatory**): DSCP match condition to be set in the rule. It is either \* or from 0-63
- `last_updated_by`: ID of the user who last updated the object.
- `action` (**Mandatory**): The action of the ACL entry DROP or FORWARD or REDIRECT. Action REDIRECT is allowed only for IngressAdvancedForwardingEntry Possible values are DROP, FORWARD, REDIRECT, .
- `address_override`: Overrides the source IP for Ingress and destination IP for Egress, macentries will use this adress as the match criteria.
- `redirect_vport_tag_id`: VPort tag to which traffic will be redirected to, when ACL entry match criteria succeeds
- `description`: Description of the ACL entry
- `destination_port` (**Mandatory**): The destination port to be matched if protocol is UDP or TCP. Value should be either \* or single port number or a port range
- `network_id`: The destination network entity that is referenced(subnet/zone/macro)
- `network_type` (**Mandatory**): Type of the source network.
- `flow_logging_enabled`: Is flow logging enabled for this particular template
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `location_id`: The ID of the location entity (Subnet/Zone/VportTag)
- `location_type` (**Mandatory**): Type of the location entity.
- `policy_state`: State of the policy. Possible values are DRAFT, LIVE, .
- `source_port` (**Mandatory**): Source port to be matched if protocol is UDP or TCP. Value can be either \* or single port number or a port range
- `uplink_preference`: Indicates the preferencial path selection for network traffic for this ACL - Default is Primary 1 and Secondary 2 when the attribute is applicable.
- `priority`: The priority of the ACL entry that determines the order of entries

- `protocol` (**Mandatory**): Protocol number that must be matched
- `associated_application_id`: The associated application ID
- `associated_application_object_id`: The associated application object ID
- `associated_application_object_type`: The associated application object type Refer to API section for supported types.
- `associated_live_entity_id`: In the draft mode, the ACL entry refers to this LiveEntity. In non-drafted mode, this is null.
- `stats_id`: The statsID that is created in the VSD and identifies this ACL Template Entry. This is auto-generated by VSD
- `stats_logging_enabled`: Is stats logging enabled for this particular template
- `ether_type` (**Mandatory**): Ether type of the packet to be matched. etherType can be \* or a valid hexadecimal value
- `external_id`: External object ID. Used for integration with third party systems

## 252.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nujob.NUJob</i>	jobs
<i>nustatistics.NUStatistics</i>	statistics

## 252.3 Parents

- *nume.NUMe*
- *nuingressadvfwdtemplate.NUIngressAdvFwdTemplate*

## NUINGRESSADVFWDTEMPLATE

`nuingressadvfwdtemplate.NUIngressAdvFwdTemplate(bambou.nurest_object.NUMetaRESTObject, ) :`

Defines the template for an Ingress Advanced Forwarding.

### 253.1 Attributes

- `name`: The name of the entity
- `last_updated_by`: ID of the user who last updated the object.
- `active`: If enabled, it means that this ACL or QOS entry is active
- `description`: A description of the entity
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `policy_state`:
- `priority`: The priority of the ACL entry that determines the order of entries
- `priority_type`:
- `associated_live_entity_id`: In the draft mode, the ACL entry refers to this LiveEntity. In non-drafted mode, this is null.
- `external_id`: External object ID. Used for integration with third party systems

### 253.2 Children

<b>class</b>	<b>fetcher</b>
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nuingressadvfwdentrytemplate.NUIngressAdvFwdEntryTemplate</code>	<code>ingress_adv_fwd_entry_templates</code>
<code>nujob.NUJob</code>	<code>jobs</code>

### 253.3 Parents

- `nudomain.NUDomain`
- `nul2domaintemplate.NUL2DomainTemplate`

- *nul2domain.NUL2Domain*
- *nudomaintemplate.NUDomainTemplate*

## NUINGRESSEXTERNALSERVICETEMPLATE

`nuingressexternalservicetemplate.NUIngressExternalServiceTemplate (bambou.nurest_object.NUMetadata)`

Defines the template for an Ingress External Service Acls.

### 254.1 Attributes

- `name`: The name of the entity
- `active`: If enabled, it means that this ACL or QOS entry is active
- `description`: A description of the entity
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `policy_state`:
- `priority`: The priority of the ACL entry that determines the order of entries
- `priority_type`:
- `associated_live_entity_id`: In the draft mode, the ACL entry refers to this LiveEntity. In non-drafted mode, this is null.
- `external_id`: External object ID. Used for integration with third party systems

### 254.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nuingressexternalservicetemplateentry.NUIngressExternalServiceTemplateEntry</code>	<code>ingress_external_service_template_entries</code>
<code>nujob.NUJob</code>	<code>jobs</code>

### 254.3 Parents

- `nudomain.NUDomain`
- `nul2domaintemplate.NUL2DomainTemplate`
- `nul2domain.NUL2Domain`

- *nudomaintemplate.NUDomainTemplate*



## NUINGRESSEXTERNALSERVICETEMPLATEENTRY

`nuingressexternalservicetemplateentry.NUIngressExternalServiceTemplateEntry(bambou.nurest_`

Defines the template of Ingress External Service ACL entries

### 255.1 Attributes

- `dscp` (**Mandatory**): DSCP match condition to be set in the rule. It is either \* or from 0-63
- `last_updated_by`: ID of the user who last updated the object.
- `action` (**Mandatory**): The action of the ACL entry DROP or FORWARD or REDIRECT. Action REDIRECT is allowed only for IngressAdvancedForwardingEntry
- `address_override`: Overrides the source IP for Ingress and destination IP for Egress, macentries will use this address as the match criteria.
- `redirect_external_service_end_point_id`: VPort tag to which traffic will be redirected to, when ACL entry match criteria succeeds
- `description`: Description of the ACL entry
- `destination_port` (**Mandatory**): The destination port to be matched if protocol is UDP or TCP. Value should be either \* or single port number or a port range
- `network_id`: The destination network entity that is referenced(subnet/zone/macro)
- `network_type` (**Mandatory**): Type of the source network.
- `flow_logging_enabled`: Is flow logging enabled for this particular template
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `location_id`: The ID of the location entity (Subnet/Zone/VportTag)
- `location_type` (**Mandatory**): Type of the location entity.
- `policy_state`: State of the policy.
- `source_port` (**Mandatory**): Source port to be matched if protocol is UDP or TCP. Value can be either \* or single port number or a port range
- `priority`: The priority of the ACL entry that determines the order of entries
- `protocol` (**Mandatory**): Protocol number that must be matched
- `associated_application_id`: The associated application ID
- `associated_application_object_id`: The associated application object ID

- `associated_application_object_type`: The associated application object type Refer to API section for supported types.
- `associated_live_entity_id`: In the draft mode, the ACL entry refers to this LiveEntity. In non-drafted mode, this is null.
- `stats_id`: The statsID that is created in the VSD and identifies this ACL Template Entry. This is auto-generated by VSD
- `stats_logging_enabled`: Is stats logging enabled for this particular template
- `ether_type` (**Mandatory**): Ether type of the packet to be matched. etherType can be \* or a valid hexadecimal value
- `external_id`: External object ID. Used for integration with third party systems

## 255.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nujob.NUJob</i>	jobs
<i>nustatistics.NUStatistics</i>	statistics

## 255.3 Parents

- *nuingressexternalservicetemplate.NUIngressExternalServiceTemplate*

## NUIPRESERVATION

`nuipreservation.NUIPReservation(bambou.nurest_object.NUMetaRESTObject,):`

This is the definition of a IP Bindings associated with in a Network.

### 256.1 Attributes

- `mac` (**Mandatory**): MAC Address
- `ip_address` (**Mandatory**): Static IP Address
- `last_updated_by`: ID of the user who last updated the object.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems
- `dynamic_allocation_enabled`: Binding is static or dynamic

### 256.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>

### 256.3 Parents

- `nusubnet.NUSubnet`



## NUJOB

`nujob.NUJob(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents JOB entity. The job API accepts a command and parameters and executes the job and returns the results. Jobs API are typically used for long running tasks.

### 257.1 Attributes

- `parameters`: Additional arguments required for the specific command. Differs based on types of command.
- `last_updated_by`: ID of the user who last updated the object.
- `result`: Results from the execution of the job
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `command` (**Mandatory**): Name of the command.
- `progress`: Indicates the progress of the job as a faction. eg : 0.5 means 50% done.
- `assoc_entity_type`: Entity with which this job is associated Refer to API section for supported types.
- `status`: Current status of the job. Possible values are RUNNING, FAILED, SUCCESS, .
- `external_id`: External object ID. Used for integration with third party systems

### 257.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 257.3 Parents

- `nuapp.NUApp`
- `nuredirectiontargettemplate.NURedirectionTargetTemplate`
- `nuredirectiontarget.NURedirectionTarget`
- `nuingressexternalservicetemplateentry.NUIngressExternalServiceTemplateEntry`
- `nudomain.NUDomain`

- *nuegressaclentrytemplate.NUEgressACLEntryTemplate*
- *nuvsd.NUVSD*
- *nupolicygroup.NUPolicyGroup*
- *nuvrs.NUVRS*
- *nuhsc.NUHSC*
- *nuvcenterhypervisor.NUVCenterHypervisor*
- *nupolicygrouptemplate.NUPolicyGroupTemplate*
- *nuingressadvfwdentrytemplate.NUIngressAdvFwdEntryTemplate*
- *nugateway.NUGateway*
- *nul2domaintemplate.NUL2DomainTemplate*
- *nuvcenter.NUVCenter*
- *nuingressaclentrytemplate.NUIngressACLEntryTemplate*
- *nul2domain.NUL2Domain*
- *nuingressexternalservicetemplate.NUIngressExternalServiceTemplate*
- *nume.NUMe*
- *nunsgateway.NUNSGateway*
- *nuvsc.NUVSC*
- *nudomaintemplate.NUDomainTemplate*
- *nuegressacltemplate.NUEgressACLTemplate*
- *nuingressadvfwdtemplate.NUIngressAdvFwdTemplate*
- *nuenterprise.NUEnterprise*
- *nuingressacltemplate.NUIngressACLTemplate*

## NUKEYSERVERMEMBER

`nukeyservermember.NUKeyServerMember(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents a KeyServer

### 258.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `pem_encoded`: PEM Encoded Certificate
- `certificate_serial_number`: Certificate serial number associated to the keyserver private key which it is currently signing with
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `fqdn`: FQDN of the keyserver member
- `issuer_dn`: Issuer DN
- `subject_dn`: Subject DN
- `public_key`: Public Key
- `external_id`: External object ID. Used for integration with third party systems

### 258.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 258.3 Parents

- `nume.NUMe`





## NUKEYSERVERMONITOR

`nukeyservermonitor.NUKeyServerMonitor(bambou.nurest_object.NUMetaRESTObject,)` :

Represents a Keyserver Monitor Snapshot.

### 259.1 Attributes

- `last_update_time`: The time the latest SEK or Seed was created/removed (milliseconds since epoch)
- `last_updated_by`: ID of the user who last updated the object.
- `gateway_secured_data_record_count`: Total number of Gateway Secured Data records
- `keyserver_monitor_encrypted_sek_count`: Total number of Keyserver Monitor Encrypted SEK records
- `keyserver_monitor_encrypted_seed_count`: Total number of Keyserver Monitor Encrypted Seed records
- `keyserver_monitor_sek_count`: Total number of Keyserver Monitor SEK records
- `keyserver_monitor_seed_count`: Total number of Keyserver Monitor Seed records
- `enterprise_secured_data_record_count`: Total number of Enterprise Secured Data records
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 259.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nukeyservermonitorencrypted-seed.NUKeyServerMonitorEncryptedSeed</code>	<code>key_server_monitor_encrypted_seeds</code>
<code>nukeyservermonitorseed.NUKeyServerMonitorSeed</code>	<code>key_server_monitor_seeds</code>
<code>nukeyservermonitorek.NUKeyServerMonitorSEK</code>	<code>key_server_monitor_seks</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 259.3 Parents

- `nuenterprise.NUEnterprise`



## NUKEYSERVERMONITORENCRYPTEDSEED

`nukeyservermonitorencryptedseed.NUKeyServerMonitorEncryptedSeed(bambou.nurest_object.NUMeta`

Represents a Keyserver Monitor Encrypted Seed Snapshot.

### 260.1 Attributes

- `sek_creation_time`: SEK Creation Time
- `last_updated_by`: ID of the user who last updated the object.
- `key_server_certificate_serial_number`: KeyServer Certificate Serial Number
- `enterprise_secured_data_id`: Enterprise Secured ID record this monitor represents
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `associated_key_server_monitor_sek_creation_time`: The creation time of the associated KeyServer Monitor Seed ID
- `associated_key_server_monitor_sekid`: The ID of the associated KeyServer Monitor SEK ID
- `associated_key_server_monitor_seed_creation_time`: The creation time of the associated KeyServer Monitor Seed ID
- `associated_key_server_monitor_seed_id`: The ID of the associated KeyServer Monitor Seed ID
- `external_id`: External object ID. Used for integration with third party systems

### 260.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 260.3 Parents

- `nukeyservermonitorseed.NUKeyServerMonitorSeed`
- `nukeyservermonitor.NUKeyServerMonitor`
- `nukeyservermonitorek.NUKeyServerMonitorSEK`



## NUKEYSERVERMONITORSEED

`nukeyservermonitorseed.NUKeyServerMonitorSeed(bambou.nurest_object.NUMetaRESTObject,)` :

Represents a Keyserver Monitor Seed Snapshot.

### 261.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `seed_traffic_authentication_algorithm`: Seed traffic Authentication Algorithm.
- `seed_traffic_encryption_algorithm`: Seed traffic Encryption Algorithm.
- `seed_traffic_encryption_key_lifetime`: Seed Traffic Encryption Key Lifetime in Seconds
- `lifetime`: The lifetime of this entry (seconds)
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `creation_time`: The time this entry was created (milliseconds since epoch)
- `start_time`: The time this entry was activated (milliseconds since epoch)
- `external_id`: External object ID. Used for integration with third party systems

### 261.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nukeyservermonitorencrypted-seed.NUKeyServerMonitorEncryptedSeed</code>	<code>key_server_monitor_encrypted_seeds</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 261.3 Parents

- `nukeyservermonitor.NUKeyServerMonitor`



## NUKEYSERVERMONITORSEK

`nukeyservermonitorsek.NUKeyServerMonitorSEK(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents a Keyserver Monitor SEK Snapshot

### 262.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `seed_payload_authentication_algorithm`: SEK Payload Signature Algorithm Possible values are HMAC\_SHA1, HMAC\_SHA256, HMAC\_SHA512, .
- `seed_payload_encryption_algorithm`: SEK Payload Encryption Algorithm Possible values are AES\_128\_CBC, AES\_256\_CBC, TRIPLE\_DES\_CBC, .
- `lifetime`: The lifetime of this entry (seconds)
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `creation_time`: The time this entry was created (milliseconds since epoch)
- `start_time`: The time this entry was activated (milliseconds since epoch)
- `external_id`: External object ID. Used for integration with third party systems

### 262.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nukeyservermonitorencrypted-seed.NUKeyServerMonitorEncryptedSeed</code>	<code>key_server_monitor_encrypted_seeds</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 262.3 Parents

- `nukeyservermonitor.NUKeyServerMonitor`





## NUL2DOMAIN

`nul2domain.NUL2Domain(bambou.nurest_object.NUMetaRESTObject,):`

This is the definition of a l2 domain associated with a Enterprise.

### 263.1 Attributes

- `dhcp_managed`: decides whether L2Domain / L2Domain template DHCP is managed by VSD
- `ip_type`: IPv4 or IPv6
- `maintenance_mode`: `maintenanceMode` is an enum that indicates if the L2Domain is accepting VM activation requests. Possible values are `DISABLED`, `ENABLED` and `ENABLED_INHERITED` Possible values are .
- `name` (**Mandatory**): Name of the L2Domain / L2Domain template,has to be unique within a Enterprise. Valid characters are alphabets, numbers, space and hyphen( - ).
- `last_updated_by`: ID of the user who last updated the object.
- `gateway`: The IP address of the gateway of this l2 domain
- `gateway_mac_address`: The MAC address of the Gateway.
- `address`: Network address of the L2Domain / L2Domain template defined.
- `template_id`: The ID of the L2 Domain template that this L2 Domain object was derived from
- `service_id`: The service ID used by the VSCs to identify this subnet
- `description`: A description field provided by the user that identifies the L2Domain / L2Domain template.
- `netmask`: Netmask of the L2Domain / L2Domain template defined
- `vn_id`: Current Network's globally unique VXLAN network identifier generated by VSD
- `encryption`: Determines whether IPSEC is enabled Possible values are `ENABLED`, `DISABLED`, .
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `policy_change_status`: None
- `route_distinguisher`: The Route Distinguisher value assigned by VSD for this subnet that is used by the BGP-EVPN protocol in VSC
- `route_target`: The Route Target value assigned by VSD for this subnet that is used by the BGP-EVPN protocol in VSC

- `uplink_preference`: Indicates the preferential path selection for network traffic in this domain - Default is Primary 1 and Secondary 2. Possible values are PRIMARY\_SECONDARY, SECONDARY\_PRIMARY, PRIMARY, SECONDARY, SYMMETRIC, .
- `associated_multicast_channel_map_id`: The ID of the Multi Cast Channel Map this L2Domain / L2Domain template template is associated with. This has to be set when `enableMultiCast` is set to ENABLED
- `associated_shared_network_resource_id`: The ID of the L2 Domain that this L2 Domain object is pointing to
- `stretched`: Indicates whether this domain is stretched, if so remote VM resolutions will be allowed
- `multicast`: Indicates multicast policy on L2Domain.
- `external_id`: External object ID. Used for integration with third party systems

## 263.2 Children

class	fetcher
<code>nutca.NUTCA</code>	<code>tcas</code>
<code>nuaddressrange.NUAddressRange</code>	<code>address_ranges</code>
<code>nuredirectiontarget.NURedirectionTarget</code>	<code>redirection_targets</code>
<code>nupermission.NUPermission</code>	<code>permissions</code>
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuegressaclentrytemplate.NUEgressACLEntryTemplate</code>	<code>egress_acl_entry_templates</code>
<code>nuegressacltemplate.NUEgressACLTemplate</code>	<code>egress_acl_templates</code>
<code>nudhcpoption.NUDHCPOption</code>	<code>dhcp_options</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nuvm.NUVM</code>	<code>vms</code>
<code>nuvminterface.NUVMInterface</code>	<code>vm_interfaces</code>
<code>nuingressaclentrytemplate.NUIngressACLEntryTemplate</code>	<code>ingress_acl_entry_templates</code>
<code>nuingressacltemplate.NUIngressACLTemplate</code>	<code>ingress_acl_templates</code>
<code>nuingressadvfwdtemplate.NUIngressAdvFwdTemplate</code>	<code>ingress_adv_fwd_templates</code>
<code>nuingressexternalservicetemplate.NUIngressExternalServiceTemplate</code>	<code>ingress_external_service_templates</code>
<code>nujob.NUJob</code>	<code>jobs</code>
<code>nupolicygroup.NUPolicyGroup</code>	<code>policy_groups</code>
<code>nuqos.NUQOS</code>	<code>qoss</code>
<code>nuhostinterface.NUHostInterface</code>	<code>host_interfaces</code>
<code>nuuplinkrd.NUUplinkRD</code>	<code>uplink_rds</code>
<code>nuvpnconnection.NUVPNConnection</code>	<code>vpn_connections</code>
<code>nuvport.NUVPort</code>	<code>vports</code>
<code>nubridgeinterface.NUBridgeInterface</code>	<code>bridge_interfaces</code>
<code>nugroup.NUGroup</code>	<code>groups</code>
<code>nustaticroute.NUStaticRoute</code>	<code>static_routes</code>
<code>nustatistics.NUStatistics</code>	<code>statistics</code>
<code>nustatisticspolicy.NUStatisticsPolicy</code>	<code>statistics_policies</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>

## 263.3 Parents

- `nul2domaintemplate.NUL2DomainTemplate`

- *nume.NUMe*
- *nuenterprise.NUEnterprise*



## NUL2DOMAINTEMPLATE

`nul2domaintemplate.NUL2DomainTemplate(bambou.nurest_object.NUMetaRESTObject, ) :`

L2 Domain in VSD as derived by templates. This object describes the L2 Domain template.

### 264.1 Attributes

- `dhcp_managed`: decides whether L2Domain / L2Domain template DHCP is managed by VSD
- `ip_type`: IPv4 or IPv6
- `name` (**Mandatory**): Name of the L2Domain / L2Domain template, has to be unique within a Enterprise. Valid characters are alphabets, numbers, space and hyphen( - ).
- `last_updated_by`: ID of the user who last updated the object.
- `gateway`: The IP address of the gateway of this L2 domain
- `address`: Network address of the L2Domain / L2Domain template defined.
- `description`: A description field provided by the user that identifies the L2Domain / L2Domain template.
- `netmask`: Netmask of the L2Domain / L2Domain template defined
- `encryption`: Determines whether IPSEC is enabled Possible values are ENABLED, DISABLED, .
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `policy_change_status`:
- `associated_multicast_channel_map_id`: The ID of the Multi Cast Channel Map this L2Domain / L2Domain template template is associated with. This has to be set when enableMultiCast is set to ENABLED
- `multicast`: Indicates multicast policy on L2Domain template.
- `external_id`: External object ID. Used for integration with third party systems

## 264.2 Children

class	fetcher
<i>nul2domain.NUL2Domain</i>	l2_domains
<i>nuaddressrange.NUAddressRange</i>	address_ranges
<i>nuredirectiontargettemplate.NURedirectionTargetTemplate</i>	redirection_target_templates
<i>nupermission.NUPermission</i>	permissions
<i>numetadata.NUMetadata</i>	metadatas
<i>nuegressacltemplate.NUEgressACLTemplate</i>	egress_acl_templates
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>ningressacltemplate.NUIngressACLTemplate</i>	ingress_acl_templates
<i>ningressadvfwdtemplate.NUIngressAdvFwdTemplate</i>	ingress_adv_fwd_templates
<i>ningressexternalservice-template.NUIngressExternalServiceTemplate</i>	ingress_external_service_templates
<i>nujob.NUJob</i>	jobs
<i>nupolicygrouptemplate.NUPolicyGroupTemplate</i>	policy_group_templates
<i>nuqos.NUQOS</i>	qoss
<i>nugroup.NUGroup</i>	groups
<i>nueventlog.NUEventLog</i>	event_logs

## 264.3 Parents

- *nuenterprise.NUEnterprise*

## NULDAPCONFIGURATION

```
nuldapconfiguration.NULDAPConfiguration(bambou.nurest_object.NUMetaRESTObject,):
```

Configuration of LDAP parameters associated with an enterprise. This will enable authentication through an external LDAP server for this enterprise.

### 265.1 Attributes

- `ssl_enabled`: Enable SSL for communication with the LDAP server
- `password`: This attribute is a mandatory field for LDAP authorization. Password that will be used to verify the integrity of groups and users in LDAP server for the enterprise.
- `last_updated_by`: ID of the user who last updated the object.
- `accept_all_certificates`: Accept all certificates from the LDAP server
- `certificate`: The certificate to authenticate with the LDAP server
- `server` (**Mandatory**): The LDAP server IP or FQDN
- `enabled`: To enable LDAP authentication for an enterprise, set this attribute to true. If enabled is set to false, `authorizationEnabled` attribute is ignored and LDAP is not used for authentication as well as authorization. The relationship between `enabled` and `authorizationEnabled` attributes is as follows, `enabled = true, authorizationEnabled = false`, LDAP is used only for Authentication `enabled = true, authorizationEnabled = true`, LDAP is used for both authentication and authorization. `enabled = false, authorizationEnabled = true`, LDAP is not used. `enabled = false, authorizationEnabled = false`, LDAP is not used.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `port` (**Mandatory**): Port to be used for the LDAP server
- `group_dn` (**Mandatory**): This attribute is a mandatory field for LDAP authorization. When LDAP is used for authorization for an enterprise, the group DN will be used to get the list of VSD specific groups in LDAP server for the enterprise. For example, `OU=VSDGroups,DC=company,DC=com`
- `user_dn_template` (**Mandatory**): The DN template to be used for authentication. The template needs to have a string `_USERID_` in it. This will be replaced by the `userId` of the user who makes the REST API call. For example, template `UID=_USERID_,OU=company,DC=com` will be converted to `UID=admin,OU=company,DC=com` and this will be used as DN for LDAP authentication.
- `authorization_enabled`: To enable LDAP authorization for an enterprise, both `authorizationEnabled` and `enabled` attributes must be set to true. If `enabled` attribute is not set, this attribute is ignored. The relationship between `enabled` and `authorizationEnabled` attributes is as follows, `enabled = true, authorizationEnabled = false`, LDAP is used only for Authentication. `enabled = true, authorizationEnabled = true`, LDAP is used for both

authentication and authorization. enabled = false, authorizationEnabled = true, LDAP is not used. enabled = false, authorizationEnabled = false, LDAP is not used.

- `authorizing_user_dn` (**Mandatory**): This attribute is a mandatory field for LDAP authorization. When LDAP is used for authorization for an enterprise, the user DN that will be used to verify the integrity of groups and users in LDAP server for the enterprise. For example, CN=groupAdmin,OU=VSD\_USERS,OU=Personal,OU=Domain Users,DC=company,DC=com
- `external_id`: External object ID. Used for integration with third party systems

## 265.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas

## 265.3 Parents

- *numenterprise.NUEnterprise*



## NULICENSE

`nulicense.NULicense(bambou.nurest_object.NUMetaRESTObject, ) :`

Enables retrieval/modification and creation of license files. Most of the attributes are retrieved from the encrypted license. The create API simply provides the encrypted license that is in base64 format.

### 266.1 Attributes

- `major_release`: Major software release associated with this license
- `last_updated_by`: ID of the user who last updated the object.
- `phone`: Phone number of the owner associated with the license file
- `license` (**Mandatory**): Base 64 value of the license
- `license_encryption`: License encryption
- `license_id`: Unique identifier of the license file
- `license_type`:
- `minor_release`: Minor software release for which this license has been issued
- `zip`: Zipcode of the owner associated with the license file
- `city`: City of the owner associated with the license file
- `allowed_cpes_count`: Maximum number of CPEs enabled with this license. A value of -1 indicates an unlimited number of CPEs
- `allowed_nics_count`: Maximum number of NICs allowed. A value of -1 indicates unlimited number of NICs
- `allowed_vms_count`: Maximum number of VMs enabled with this license. A value of -1 indicates an unlimited number of VMs
- `allowed_vrsgs_count`: Maximum number of VRSGs enabled with this license. A value of -1 indicates an unlimited number of VRSGs
- `allowed_vrss_count`: Maximum number of VRSSs enabled with this license. A value of -1 indicates an unlimited number of VRSSs
- `email`: Email of the owner associated with the license file
- `encryption_mode`: Indicates if the system is associated with a license that allows encryption or not
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `company`: Company of the owner associated with the license file

- `country`: Country of the owner associated with the license file
- `product_version`: Version of the product that this license applies to
- `provider`: Provider of the license file
- `is_cluster_license`: Indicates if the license is associated with standalone or cluster setup of VSD
- `user_name`: The name of the user associated with the license
- `state`: State of the owner associated with the license file
- `street`: Address of the owner associated with the license file
- `customer_key`: Customer key associated with the license
- `expiration_date`: Expiration date of this license
- `external_id`: External object ID. Used for integration with third party systems

## 266.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nueventlog.NUEventLog</i>	event_logs

## 266.3 Parents

- *nume.NUme*

## **NULOCATION**

**nulocation.NULocation(bambou.nurest\_object.NUMetaRESTObject,):**

Gateway location details.

### **267.1 Attributes**

- **last\_updated\_by**: ID of the user who last updated the object.
- **latitude**: Latitude in decimal format.
- **address**: Formatted address including property number, street name, suite or office number, ...
- **ignore\_geocode**: Request BSS to perform a geocode on the address - If no value passed, requestGeocode will be set to true
- **time\_zone\_id**: Time zone in which the Gateway is located. This can be in the form of a UTC/GMT offset, continent/city location, or country/region. The available time zones can be found in /usr/share/zoneinfo on a Linux machine or retrieved with `TimeZone.getAvailableIDs()` in Java. Refer to the IANA (Internet Assigned Numbers Authority) for a list of time zones. URL : <http://www.iana.org/time-zones> Default value is UTC (translating to Etc/Zulu)
- **entity\_scope**: Specify if scope of entity is Data center or Enterprise level
- **locality**: Locality/City/County
- **longitude**: Longitude in decimal format.
- **country**: Country
- **state**: State/Province/Region
- **external\_id**: External object ID. Used for integration with third party systems

### **267.2 Children**

<b>class</b>	<b>fetcher</b>
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas

## 267.3 Parents

- *nunsgateway.NUNSGateway*

## NUME

`nume.NUME(bambou.nurest_object.NUMetaRESTObject, ) :`

Object that identifies the user functions

### 268.1 Attributes

- `password` (**Mandatory**): User password stored as a hash (SHA-1 encrypted)
- `last_name` (**Mandatory**): Last name of the user
- `last_updated_by`: ID of the user who last updated the object.
- `first_name` (**Mandatory**): First name of the user
- `disabled`: Status of the user account; true=disabled, false=not disabled; default value = false
- `email` (**Mandatory**): Email address of the user
- `enterprise_id`: Identifier of the enterprise.
- `enterprise_name`: Name of the enterprise.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `mobile_number`: Mobile Number of the user
- `role`: Role of the user.
- `user_name` (**Mandatory**): Unique Username of the user. Valid characters are alphabets, numbers and hyphen(-).
- `avatar_data`: URL to the avatar data associated with the enterprise. If the avatarType is URL then value of avatarData should an URL of the image. If the avatarType BASE64 then avatarData should be BASE64 encoded value of the image
- `avatar_type`: Avatar type.
- `external_id`: External object ID. Used for integration with third party systems

### 268.2 Children

class	fetcher
<code>nul2domain.NUL2Domain</code>	<code>l2_domains</code>

Continued on next page

Table 268.1 – continued from previous page

<i>nuvcentreamconfig.NUVCenterEAMConfig</i>	<i>vcenter_eam_configs</i>
<i>nuratelimiter.NURateLimiter</i>	<i>rate_limiters</i>
<i>nugateway.NUGateway</i>	<i>gateways</i>
<i>nugatewaytemplate.NUGatewayTemplate</i>	<i>gateway_templates</i>
<i>nupatnatpool.NUPATNATPool</i>	<i>patnat_pools</i>
<i>nutca.NUTCA</i>	<i>tcas</i>
<i>nuvcenter.NUVCenter</i>	<i>vcenters</i>
<i>nuvcenterhypervisor.NUVCenterHypervisor</i>	<i>vcenter_hypervisors</i>
<i>nuredirectiontarget.NURedirectionTarget</i>	<i>redirection_targets</i>
<i>nuredundancygroup.NURedundancyGroup</i>	<i>redundancy_groups</i>
<i>nucertificate.NUCertificate</i>	<i>certificates</i>
<i>numetadata.NUMetadata</i>	<i>metadatas</i>
<i>numetadatatag.NUMetadataTag</i>	<i>metadata_tags</i>
<i>nunetworklayout.NUNetworkLayout</i>	<i>network_layouts</i>
<i>nukeyservermember.NUKeyServerMember</i>	<i>key_server_members</i>
<i>nuegressaclentrytemplate.NUEgressACLEntryTemplate</i>	<i>egress_acl_entry_templates</i>
<i>nuegressacltemplate.NUEgressACLTemplate</i>	<i>egress_acl_templates</i>
<i>nudomainfipacltemplate.NUDomainFIPACLTemplate</i>	<i>domain_fip_acl_templates</i>
<i>nufloatingipacltemplate.NUFloatingIPACLTemplate</i>	<i>floating_ipacl_templates</i>
<i>nuegressqospolicy.NUEgressQOSPolicy</i>	<i>egress_qos_policies</i>
<i>nusharednetworkresource.NUSharedNetworkResource</i>	<i>shared_network_resources</i>
<i>nulicense.NULicense</i>	<i>licenses</i>
<i>numirrordestination.NUMirrorDestination</i>	<i>mirror_destinations</i>
<i>nusiteinfo.NUSiteInfo</i>	<i>site_infos</i>
<i>nufloatingip.NUFloatingIp</i>	<i>floating_ips</i>
<i>nuglobalmetadata.NUGlobalMetadata</i>	<i>global_metadatas</i>
<i>nuvm.NUVM</i>	<i>vms</i>
<i>nuvminterface.NUVMInterface</i>	<i>vm_interfaces</i>
<i>nucloudmgmtsystem.NUCloudMgmtSystem</i>	<i>cloud_mgmt_systems</i>
<i>nuinfrastructuregatewayprofile.NUInfrastructureGatewayProfile</i>	<i>infrastructure_gateway_profiles</i>
<i>nuinfrastructureportprofile.NUInfrastructurePortProfile</i>	<i>infrastructure_port_profiles</i>
<i>nuinfrastructurevscprofile.NUInfrastructureVscProfile</i>	<i>infrastructure_vsc_profiles</i>
<i>nuingressaclentrytemplate.NUIngressACLEntryTemplate</i>	<i>ingress_acl_entry_templates</i>
<i>nuingressacltemplate.NUIngressACLTemplate</i>	<i>ingress_acl_templates</i>
<i>nuingressadvfwdentrytemplate.NUIngressAdvFwdEntryTemplate</i>	<i>ingress_adv_fwd_entry_templates</i>
<i>nuenterprise.NUEnterprise</i>	<i>enterprises</i>
<i>nuenterpriseprofile.NUEnterpriseProfile</i>	<i>enterprise_profiles</i>
<i>nujob.NUJob</i>	<i>jobs</i>
<i>nupolicygroup.NUPolicyGroup</i>	<i>policy_groups</i>
<i>nudomain.NUDomain</i>	<i>domains</i>
<i>nuzone.NUZone</i>	<i>zones</i>
<i>nuhostinterface.NUHostInterface</i>	<i>host_interfaces</i>
<i>nuuplinkrd.NUUplinkRD</i>	<i>uplink_rds</i>
<i>nuapplicationsevice.NUApplicationService</i>	<i>application_services</i>
<i>nuvcentervrsconfig.NUVCenterVRSCfg</i>	<i>vcenter_vrs_configs</i>
<i>nuuser.NUUser</i>	<i>users</i>
<i>nunsgateway.NUNSGateway</i>	<i>ns_gateways</i>
<i>nunsgatewaytemplate.NUNSGatewayTemplate</i>	<i>ns_gateway_templates</i>
<i>nunsredundantgatewaygroup.NUNSRedundantGatewayGroup</i>	<i>ns_redundant_gateway_groups</i>
<i>nuvsp.NUVSP</i>	<i>vsps</i>

Continued on next page

Table 268.1 – continued from previous page

<i>nunsportstaticconfiguration.NUNSPortStaticConfiguration</i>	ns_port_static_configurations
<i>nustaticroute.NUStaticRoute</i>	static_routes
<i>nustatscollectorinfo.NUStatsCollectorInfo</i>	stats_collector_infos
<i>nusubnet.NUSubnet</i>	subnets
<i>numulticastchannelmap.NUMultiCastChannelMap</i>	multi_cast_channel_maps
<i>nuautodiscoveredgateway.NUAutoDiscoveredGateway</i>	auto_discovered_gateways
<i>nuexternalappservice.NUExternalAppService</i>	external_app_services
<i>nuexternalservice.NUExternalService</i>	external_services
<i>nusystemconfig.NUSystemConfig</i>	system_configs





## NUMETADATA

`numetadata.NUMetadata(bambou.nurest_object.NUMetaRESTObject, ) :`

Metadata associated to a entity.

### 269.1 Attributes

- `name`: name of the Metadata.
- `description`: Description of the Metadata.
- `metadata_tag_ids`: metadata tag IDs associated with this metadata you can filter metadata based on this attribute for example X-Nuage-Filter: '2d6fb627-603b-421c-b63a-eb0a6d712761' IN metadataTagIDs
- `network_notification_disabled`: specifies metadata changes need to be notified to controller, by default it is notified
- `blob` (**Mandatory**): Metadata that describes about the entity attached to it.
- `global_metadata`: specifies metadata is global or local
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 269.2 Children

class	fetcher
<code>numetadatatag.NUMetadataTag</code>	<code>metadata_tags</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>

### 269.3 Parents

- `nugroupkeyencryptionprofile.NUGroupKeyEncryptionProfile`
- `nuqos.NUQOS`
- `nubgppeer.NUBGPPeer`
- `nusharednetworkresource.NUSharedNetworkResource`
- `nuvirtualip.NUVirtualIP`

- *nudscpforwardingclasstable.NUDSCPForwardingClassTable*
- *numulticastchannelmap.NUMultiCastChannelMap*
- *nuredundancygroup.NURedundancyGroup*
- *nutca.NUTCA*
- *nugroup.NUGroup*
- *nuvsgredundantport.NUVsgRedundantPort*
- *nuzone.NUZone*
- *nuapp.NUApp*
- *nupatnatpool.NUPATNATPool*
- *nuinfrastructuregatewayprofile.NUInfrastructureGatewayProfile*
- *nuflowsecuritypolicy.NUFlowSecurityPolicy*
- *nuvcentereamconfig.NUVCenterEAMConfig*
- *nulocation.NULocation*
- *nuenterprisesecurity.NUEnterpriseSecurity*
- *nuvcentervrsconfig.NUVCenterVRSCfg*
- *nuenterprisenetwork.NUEnterpriseNetwork*
- *nuinfrastructurevscprofile.NUInfrastructureVscProfile*
- *nupermision.NUPermission*
- *nuipreservation.NUIPReservation*
- *nuredirectiontargettemplate.NURedirectionTargetTemplate*
- *nusubnettemplate.NUSubnetTemplate*
- *numetadatag.NUMetadataTag*
- *nufloatingipacltemplate.NUFloatingIPACLTemplate*
- *nuredirectiontarget.NURedirectionTarget*
- *nunetworklayout.NUNetworkLayout*
- *nuingressexternalservicetemplateentry.NUIngressExternalServiceTemplateEntry*
- *nubootstrapactivation.NUBootstrapActivation*
- *nucloudmgmtsystem.NUCloudMgmtSystem*
- *nuvsp.NUVSP*
- *nudomain.NUDomain*
- *nuredundantport.NURedundantPort*
- *nudscpforwardingclassmapping.NUDSCPForwardingClassMapping*
- *nuegressaclentrytemplate.NUEgressACLEntryTemplate*
- *nuwanservice.NUWANService*
- *nuvsd.NUVSD*
- *nuvport.NUVPort*

- *nuflowforwardingpolicy.NUFlowForwardingPolicy*
- *nuport.NUPort*
- *nustatisticspolicy.NUStatisticsPolicy*
- *nusubnet.NUSubnet*
- *nupolicygroup.NUPolicyGroup*
- *nuratelimiter.NURateLimiter*
- *nukeyservermonitorencryptedseed.NUKeyServerMonitorEncryptedSeed*
- *nuzonetemplate.NUZoneTemplate*
- *nukeyservermonitorseed.NUKeyServerMonitorSeed*
- *nugatewaytemplate.NUGatewayTemplate*
- *nuvrs.NUVRS*
- *nuhsc.NUHSC*
- *nuldapconfiguration.NULDAPConfiguration*
- *nuvsdcomponent.NUVSDComponent*
- *nuenterprisepermission.NUEnterprisePermission*
- *nustaticroute.NUStaticRoute*
- *nujob.NUJob*
- *nuvminterface.NUVMInterface*
- *nugatewaysecureddata.NUGatewaySecuredData*
- *nuvcenterhypervisor.NUVCenterHypervisor*
- *nukeyservermonitor.NUKeyServerMonitor*
- *nueventlog.NUEventLog*
- *nulicense.NULicense*
- *nuenterpriseprofile.NUEnterpriseProfile*
- *nuvcenterdatacenter.NUVCenterDataCenter*
- *nupolicygrouptemplate.NUPolicyGroupTemplate*
- *nubridgeinterface.NUBridgeInterface*
- *nuvcentercluster.NUVCenterCluster*
- *numulticastrange.NUMultiCastRange*
- *nunetworkmacrogroup.NUNetworkMacroGroup*
- *nuingressadvfwdentrytemplate.NUIngressAdvFwdEntryTemplate*
- *numulticastlist.NUMultiCastList*
- *nuautodiscoveredgateway.NUAutoDiscoveredGateway*
- *numirrordestination.NUMirrorDestination*
- *nutier.NUTier*
- *nudomainfipacltemplate.NUDomainFIPACLTemplate*

- *nugateway.NUGateway*
- *numultinicvport.NUMultiNICVPort*
- *nustatistics.NUStatistics*
- *nunsporttemplate.NUNSPortTemplate*
- *nucertificate.NUCertificate*
- *nugatewaysecurityprofile.NUGatewaySecurityProfile*
- *nul2domaintemplate.NUL2DomainTemplate*
- *nuapplicationservice.NUApplicationService*
- *nustatscollectorinfo.NUStatsCollectorInfo*
- *nuvcenter.NUVCenter*
- *nuingressaclentrytemplate.NUIngressACLEntryTemplate*
- *nul2domain.NUL2Domain*
- *nuhostinterface.NUHostInterface*
- *nuenterprisesecureddata.NUEnterpriseSecuredData*
- *nuexternalappservice.NUExternalAppService*
- *nuingressexternalservicetemplate.NUIngressExternalServiceTemplate*
- *nuporttemplate.NUPortTemplate*
- *nume.NUMe*
- *nuendpoint.NUEndPoint*
- *nudhcption.NUDHCPOption*
- *nukeyservermember.NUKeyServerMember*
- *nunsgateway.NUNSGateway*
- *nunsgatewaytemplate.NUNSGatewayTemplate*
- *nuvsc.NUVSC*
- *nuuplinkrd.NUUplinkRD*
- *nuvm.NUVM*
- *nunsport.NUNSPort*
- *nuvrsaddressrange.NUVRSAddressRange*
- *nuegressqospolicy.NUEgressQOSPolicy*
- *nupublicnetworkmacro.NUPublicNetworkMacro*
- *nudomainfipacltemplateentry.NUDomainFIPACLTemplateEntry*
- *nuaddressrange.NUAddressRange*
- *nudomaintemplate.NUDomainTemplate*
- *nusiteinfo.NUSiteInfo*
- *nuvmresync.NUVMResync*
- *nupolicydecision.NUPolicyDecision*

- *nufloatingip.NUFloatingIp*
- *nuegressacltemplate.NUEgressACLTemplate*
- *numonitoringport.NUMonitoringPort*
- *nuinfrastructureportprofile.NUInfrastructurePortProfile*
- *nuvpnconnection.NUVPNConnection*
- *nunsredundantgatewaygroup.NUNSRedundantGatewayGroup*
- *nuexternalservice.NUExternalService*
- *nukeyservermonitorsek.NUKeyServerMonitorSEK*
- *nuingressadvfwdtemplate.NUIngressAdvFwdTemplate*
- *nuvportmirror.NUVPortMirror*
- *nunspportstaticconfiguration.NUNSPortStaticConfiguration*
- *nualarm.NUAlarm*
- *nuvlan.NUVLAN*
- *nusystemconfig.NUSystemConfig*
- *nufloatingipacltemplateentry.NUFloatingIPACLTemplateEntry*
- *nuinfrastructureconfig.NUInfrastructureConfig*
- *nuuser.NUUser*
- *nunatmapentry.NUNATMapEntry*
- *nualarm.NUAlarm*
- *nubootstrap.NUBootstrap*
- *nuflow.NUFlow*
- *nuvlantemplate.NUVLANTemplate*
- *nugatewaysecurity.NUGatewaySecurity*
- *nuglobalmetadata.NUGlobalMetadata*
- *nuenterprise.NUEnterprise*
- *nuingressacltemplate.NUIngressACLTemplate*



## NUMETADATATAG

`numetadatatag.NUMMetadataTag(bambou.nurest_object.NUMetaRESTObject, ) :`

Metadata tag associated to a metadata.

### 270.1 Attributes

- **name (Mandatory):** name of the Metadata tag.
- **last\_updated\_by:** ID of the user who last updated the object.
- **description:** Description of the Metadata tag.
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **associated\_external\_service\_id:** ID of the entity to which the Metadata tag is associated to
- **auto\_created:** set to true if it is the default metadata tag created as part of external service creation
- **external\_id:** External object ID. Used for integration with third party systems

### 270.2 Children

class	fetcher
<i>numetadata.NUMMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nueventlog.NUEventLog</i>	event_logs

### 270.3 Parents

- *numetadata.NUMMetadata*
- *nume.NUMe*
- *nuexternalservice.NUExternalService*
- *nuglobalmetadata.NUGlobalMetadata*
- *nuenterprise.NUEnterprise*





## NUMIRRORDESTINATION

`numirrordestination.NUMirrorDestination(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents a mirror destination.

### 271.1 Attributes

- `name`: Name of the mirror destination. Valid characters are alphabets, numbers, space and hyphen( - ).
- `last_updated_by`: ID of the user who last updated the object.
- `service_id`: Service ID of the mirror destination.
- `destination_ip`: IP address of the destination server where you want your traffic to be mirrored.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 271.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuvportmirror.NUVPortMirror</i>	vport_mirrors

### 271.3 Parents

- *nume.NUMe*



## NUMONITORINGPORT

`numonitoringport.NUMonitoringPort(bambou.nurest_object.NUMetaRESTObject,):`

Encapsulates the port information for system monitoring entity.

### 272.1 Attributes

- `name`: Name for the port.
- `last_state_change`: Last port state change timestamp.
- `access`: Flag to indicate that it is a access port or network port.
- `description`: Optional port description.
- `resiliency_state`:
- `resilient`: Flag to indicate if an ACCESS port is resilient or not.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `uplink`: Flag to indicate that is an uplink or downlink port.
- `state`: The current state of the port.
- `external_id`: External object ID. Used for integration with third party systems

### 272.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 272.3 Parents

- `nuvrs.NUVRS`
- `nuhsc.NUHSC`
- `nuvsc.NUVSC`



## NUMULTICASTCHANNELMAP

`nummulticastchannelmap.NUMMultiCastChannelMap(bambou.nurest_object.NUMetaRESTObject,):`

This is the definition of a MultiCast Channel Map.

### 273.1 Attributes

- **name (Mandatory):** Name of the current entity
- **last\_updated\_by:** ID of the user who last updated the object.
- **description:** A description field provided by the user that identifies the MultiCast Channel Map
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **external\_id:** External object ID. Used for integration with third party systems

### 273.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>numulticastrange.NUMultiCastRange</i>	multi_cast_ranges
<i>nueventlog.NUEventLog</i>	event_logs

### 273.3 Parents

- *nuvminterface.NUVMInterface*
- *numulticastlist.NUMultiCastList*
- *nuhostinterface.NUHostInterface*
- *nume.NUMe*



## NUMULTICASTLIST

```
nummulticastlist.NUMMultiCastList(bambou.nurest_object.NUMetaRESTObject,):
```

This is the definition of a MultiCast Channel List.

### 274.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `mcast_type`: Type of multicast list.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 274.2 Children

class	fetcher
<i>numetadata.NUMMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>numulticastchannelmap.NUMMultiCastChannelMap</i>	multi_cast_channel_maps

### 274.3 Parents

- *numenterpriseprofile.NUEnterpriseProfile*
- *numenterprise.NUEnterprise*





## NUMULTICASTRANGE

`nummulticasterange.NUMMultiCastRange(bambou.nurest_object.NUMetaRESTObject,):`

This is the definition of a MultiCast Range associated with a MultiCast Channel Map.

### 275.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `max_address` (**Mandatory**): Highest address in the MultiCast range
- `min_address` (**Mandatory**): Lowest address in the MultiCast range
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 275.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nueventlog.NUEventLog</i>	event_logs

### 275.3 Parents

- *numulticastchannelmap.NUMultiCastChannelMap*



## NUMULTINICVPORT

`numultinicvport.NUMMultiNICVPort(bambou.nurest_object.NUMetaRESTObject,)` :

Encapsulates the Multi NIC VPort information for system monitoring entity.

### 276.1 Attributes

- `name`: Name for the Multi NIC VPort.
- `last_updated_by`: ID of the user who last updated the object.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 276.2 Children

class	fetcher
<code>numetadata.NUMMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nuvport.NUVPort</code>	<code>vports</code>

### 276.3 Parents

- `nuvrs.NUVRS`



## NUNATMAPENTRY

`nunatmapentry.NUNATMapEntry(bambou.nurest_object.NUMetaRESTObject, ) :`

Defines a MAP between the private ip and public ip.

### 277.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `private_ip` (**Mandatory**): Private IP address of the interface
- `associated_patnat_pool_id`: Indicates which PATNATPool this entry belongs to
- `public_ip` (**Mandatory**): Public IP address of the interface
- `external_id`: External object ID. Used for integration with third party systems

### 277.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 277.3 Parents

- `nupatnatpool.NUPATNATPool`



## NUNETWORKLAYOUT

`nunetworklayout.NUNetworkLayout(bambou.nurest_object.NUMetaRESTObject,)` :

This API defines the AS number that should be used in the data center as well as the IP address of the route reflector.

### 278.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `service_type`: Identifies whether L3 or L2 services are supported.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `route_reflector_ip`: The IP address of the route reflector that can be used by the VSCs
- `autonomous_system_num`: The AS number associated with this data center
- `external_id`: External object ID. Used for integration with third party systems

### 278.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 278.3 Parents

- `nume.NUMe`





## NUNETWORKMACROGROUP

`nunetworkmacrogroup.NUNetworkMacroGroup(bambou.nurest_object.NUMetaRESTObject, ) :`

Administrators of an enterprise can define macros that are set of IP addresses that identify enterprise networks. These macros can be used in the ACL definitions by network designers and other users to identify access restrictions towards specific enterprise networks.

### 279.1 Attributes

- **name (Mandatory):** Name of the macro group
- **last\_updated\_by:** ID of the user who last updated the object.
- **description:** Description of the macro group
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **external\_id:** External object ID. Used for integration with third party systems

### 279.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>numenterprisenetwork.NUEnterpriseNetwork</i>	enterprise_networks

### 279.3 Parents

- *numenterprisenetwork.NUEnterpriseNetwork*
- *numenterprise.NUEnterprise*



## NUNSGATEWAY

`nunsgateway.NUNSGateway(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents Network Service Gateway object.

### 280.1 Attributes

- `nat_traversal_enabled`: Boolean value that states if the NSG instance is in a network that is behind a NAT device and will use NAT Traversal procedures to talk to other NSGs and the Internet.
- `name` (**Mandatory**): Name of the Gateway
- `last_updated_by`: ID of the user who last updated the object.
- `datapath_id`: Identifier of the Gateway, based on the `systemId`
- `redundancy_group_id`: The Redundancy Gateway Group associated with this Gateway Instance. This is a read only attribute
- `template_id` (**Mandatory**): The ID of the template that this Gateway was created from. This should be set when instantiating a Gateway
- `pending`: Indicates that this gateway is pending state or state. When in pending state it cannot be modified from REST.
- `permitted_action`: The permitted action to USE/EXTEND this Gateway.
- `personality`: Personality of the Gateway - NSG, cannot be changed after creation.
- `description`: A description of the Gateway
- `enterprise_id`: The enterprise associated with this Gateway. This is a read only attribute
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `location_id`: The NSGateway's Location. NOTE: this is a read only property, it can only be set through the location object
- `configuration_reload_state`:
- `configuration_status`:
- `bootstrap_id`: The bootstrap details associated with this NSGateway. NOTE: this is a read only property, it can only be set during creation of an NSG
- `bootstrap_status`: The bootstrap status of this NSGateway. NOTE: this is a read only property
- `associated_gateway_security_id`: Readonly Id of the associated gateway security object
- `associated_gateway_security_profile_id`: Readonly Id of the associated gateway security profile object

- `auto_disc_gateway_id`: The Auto Discovered Gateway associated with this Gateway Instance
- `external_id`: External object ID. Used for integration with third party systems
- `system_id`: Identifier of the Gateway, cannot be modified after creation

## 280.2 Children

class	fetcher
<i>nugatewaysecurity.NUGatewaySecurity</i>	gateway_securities
<i>nugatewaysecurityprofile.NUGatewaySecurityProfile</i>	gateway_security_profiles
<i>nupatnatpool.NUPATNATPool</i>	patnat_pools
<i>nupermision.NUPermission</i>	permissions
<i>numetadata.NUMetadata</i>	metadatas
<i>nualarm.NUAlarm</i>	alarms
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuinfrastructureconfig.NUInfrastructureConfig</i>	infrastructure_configs
<i>nueenterprisepermission.NUEenterprisePermission</i>	enterprise_permissions
<i>nujob.NUJob</i>	jobs
<i>nulocation.NULocation</i>	locations
<i>nubootstrap.NUBootstrap</i>	bootstraps
<i>nubootstrapactivation.NUBootstrapActivation</i>	bootstrap_activations
<i>nunsport.NUNSPORT</i>	ns_ports
<i>nueventlog.NUEventLog</i>	event_logs

## 280.3 Parents

- *nume.NUme*
- *nunsredundantgatewaygroup.NUNSRedundantGatewayGroup*
- *nueenterprise.NUEenterprise*

## NUNSGATEWAYTEMPLATE

`nunsgatewaytemplate.NUNSGatewayTemplate(bambou.nurest_object.NUMetaRESTObject,)` :

Represents a Network Service Gateway Template.

### 281.1 Attributes

- **name (Mandatory)**: Name of the Gateway
- **last\_updated\_by**: ID of the user who last updated the object.
- **description**: A description of the Gateway
- **infrastructure\_profile\_id (Mandatory)**: The ID of the infrastructure gateway profile this instance of a Gateway is associated with.
- **enterprise\_id**: The enterprise associated with this Gateway. This is a read only attribute
- **entity\_scope**: Specify if scope of entity is Data center or Enterprise level
- **external\_id**: External object ID. Used for integration with third party systems

### 281.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nunsporttemplate.NUNSPortTemplate</i>	ns_port_templates

### 281.3 Parents

- *nume.NUMe*
- *nuenterprise.NUEnterprise*



## NUNSPORT

`nunsport.NUNSPort(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents a Port of a particular NS gateway object.

### 282.1 Attributes

- `vlan_range`: VLAN Range of the Port. Format must conform to a-b,c,d-f where a,b,c,d,f are integers between 0 and 4095.
- `name` (**Mandatory**): Name of the Port
- `last_updated_by`: ID of the user who last updated the object.
- `template_id`: The ID of the template that this Port was created from
- `permitted_action`: The permitted action to USE/EXTEND this Gateway.
- `description`: A description of the Port
- `physical_name` (**Mandatory**): Identifier of the Port
- `infrastructure_profile_id`: The ID of the infrastructure profile this instance is associated with.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `port_type` (**Mandatory**): Type of the Port.
- `use_user_mnemonic`: determines whether to use user mnemonic of the Port
- `user_mnemonic` (**Mandatory**): user mnemonic of the Port
- `associated_egress_qos_policy_id`: ID of the Egress QOS Policy associated with this Vlan.
- `associated_redundant_port_id`: ID of the redundant port to which the Port is associated to.
- `associated_vsc_profile_id`: The ID of the infrastructure VSC profile this is associated with this instance of a port or port template.
- `status`: Status of the port.
- `external_id`: External object ID. Used for integration with third party systems

## 282.2 Children

<b>class</b>	<b>fetcher</b>
<i>nupermision.NUPermission</i>	permissions
<i>numetadata.NUMetadata</i>	metadatas
<i>nuvlan.NUVLAN</i>	vlans
<i>nualarm.NUAlarm</i>	alarms
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuenterpriasepermission.NUEnterprisePermission</i>	enterprise_permissions
<i>nunsportstaticconfiguration.NUNSPortStaticConfiguration</i>	ns_port_static_configurations
<i>nustatistics.NUStatistics</i>	statistics
<i>nustatisticspolicy.NUStatisticsPolicy</i>	statistics_policies
<i>nueventlog.NUEventLog</i>	event_logs

## 282.3 Parents

- *nuredundantport.NURedundantPort*
- *nuautodiscoveredgateway.NUAutoDiscoveredGateway*
- *nunsgateway.NUNSGateway*



## NUNSPORTSTATICCONFIGURATION

`nunsportstaticconfiguration.NUNSPORTStaticConfiguration(bambou.nurest_object.NUMetaRESTObj`

Represents a network port static configuration in the context of an Network Services Gateway.

### 283.1 Attributes

- `dns_address`: DNS Address for Network NSPort.
- `last_updated_by`: ID of the user who last updated the object.
- `gateway`: IP address of the gateway bound to the Network NSPort.
- `address`: IP address of the Network NSPort.
- `netmask`: IP address netmask of the Network NSPort.
- `enabled`: Boolean value that states if the NSG Port static configuration needs to be applied.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 283.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 283.3 Parents

- `nume.NUMe`
- `nunsport.NUNSPORT`



## NUNSPORTTEMPLATE

`nunsporttemplate.NUNSPORTTemplate (bambou.nurest_object.NUMetaRESTObject, ) :`

Represents Port Template object under a given gateway template object.

### 284.1 Attributes

- `vlan_range`: VLAN Range of the Port. Format must conform to a-b,c,d-f where a,b,c,d,f are integers between 0 and 4095.
- `name` (**Mandatory**): Name of the Port
- `last_updated_by`: ID of the user who last updated the object.
- `description`: A description of the Port
- `physical_name` (**Mandatory**): Identifier of the Port
- `infrastructure_profile_id`: The ID of the infrastructure profile this instance is associated with.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `port_type` (**Mandatory**): Type of the Port.
- `associated_egress_qos_policy_id`: ID of the Egress QOS Policy associated with this Vlan.
- `associated_vsc_profile_id`: The ID of the infrastructure VSC profile this is associated with this instance of a port or port template.
- `external_id`: External object ID. Used for integration with third party systems

### 284.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuvlantemplate.NUVLANTemplate</code>	<code>vlan_templates</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 284.3 Parents

- `nunsgatewaytemplate.NUNSGatewayTemplate`



## NUNSREDUNDANTGATEWAYGROUP

`nunsredundantgatewaygroup.NUNSRedundantGatewayGroup(bambou.nurest_object.NUMetaRESTObject, )`

Represents Redundant Group formed by two VNS Gateways.

### 285.1 Attributes

- **name (Mandatory):** Name of the Redundancy Group
- **last\_updated\_by:** ID of the user who last updated the object.
- **gateway\_peer1\_autodiscovered\_gateway\_id:** The Auto Discovered Gateway configuration owner in this Redundant Group.
- **gateway\_peer1\_id:** The gateway configuration owner in this Redundant Group. when Redundant Group is deleted this gateway will receive vport associations
- **gateway\_peer1\_name:** The gateway configuration owner name in this Redundant Group
- **gateway\_peer2\_autodiscovered\_gateway\_id:** The Auto Discovered Gateway peer in this Redundant Group
- **gateway\_peer2\_name:** The gateway peer name in this Redundant Group
- **heartbeat\_interval:** Heartbeat interval in milliseconds to declare the neighbor dead.
- **heartbeat\_vlanid:** Heartbeat VLAN used for BFD.
- **redundancy\_port\_ids:** Collections resilient port ids associated with this redundant group.
- **redundant\_gateway\_status:** The status of Redundant Group.
- **permitted\_action:** The permitted action to USE/EXTEND this Gateway.
- **personality:** Derived personality of the Redundancy Group.
- **description:** Description of the Redundancy Group
- **enterprise\_id:** The enterprise associated with this Redundant Group. This is a read only attribute
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **consecutive\_failures\_count:** Consecutive failure count.
- **external\_id:** External object ID. Used for integration with third party systems

## 285.2 Children

<b>class</b>	<b>fetcher</b>
<i>numetadata.NUMetadata</i>	metadatas
<i>nualarm.NUAlarm</i>	alarms
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nunsgateway.NUNSGateway</i>	ns_gateways
<i>nuredundantport.NURedundantPort</i>	redundant_ports
<i>nueventlog.NUEventLog</i>	event_logs

## 285.3 Parents

- *nume.NUMe*
- *numenterprise.NUEnterprise*

## NUPATNATPOOL

`nupatnatpool.NUPATNATPool(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents a PAT NAT Pool object.

### 286.1 Attributes

- **name (Mandatory):** Name of the PATNATPool
- **last\_updated\_by:** ID of the user who last updated the object.
- **address\_range (Mandatory):** Pool of IP Address that is available for use ex : 130.12.0.0/16
- **default\_patip:** Default PAT IP Address, must belong to the pool above
- **permitted\_action:** The permitted action to USE/EXTEND this Gateway.
- **description:** A description of the PATNATPool
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **associated\_gateway\_id:** Default PAT IP Address, must belong to the pool above
- **associated\_gateway\_type:** None
- **external\_id:** External object ID. Used for integration with third party systems

### 286.2 Children

class	fetcher
<i>nunatmapentry.NUNATMapEntry</i>	<i>nat_map_entries</i>
<i>numetadata.NUMetadata</i>	<i>metadatas</i>
<i>nuglobalmetadata.NUGlobalMetadata</i>	<i>global_metadatas</i>
<i>nuenterprisepermission.NUEnterprisePermission</i>	<i>enterprise_permissions</i>

### 286.3 Parents

- *nugateway.NUGateway*
- *nume.NUMe*
- *nunsgateway.NUNSGateway*

- *numenterprise.NUEnterprise*



## NUPERMISSION

`nupermmission.NUPermission(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents Permitted action on an entity for a group.

### 287.1 Attributes

- `name`: Name of the Permission
- `last_updated_by`: ID of the user who last updated the object.
- `permitted_action` (**Mandatory**): The permitted action to USE/EXTEND/READ/INSTANTIATE an entity.
- `permitted_entity_description`: Description for the permittedEntity
- `permitted_entity_id` (**Mandatory**): The entity ID for which this permission action is associated against.
- `permitted_entity_name`: Name of the entity for which we have given permission.
- `permitted_entity_type`: Type of the entity for which we have given permission.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 287.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nueventlog.NUEventLog</i>	event_logs

### 287.3 Parents

- *nuredundancygroup.NURedundancyGroup*
- *nuvsgredundantport.NUVsgRedundantPort*
- *nuzone.NUZone*
- *nudomain.NUDomain*

- *nuwanservice.NUWANService*
- *nuport.NUPort*
- *nugateway.NUGateway*
- *nul2domaintemplate.NUL2DomainTemplate*
- *nul2domain.NUL2Domain*
- *nunsgateway.NUNSGateway*
- *nunsport.NUNSPort*
- *nudomaintemplate.NUDomainTemplate*
- *nuvlan.NUVLAN*

## NUPOLICYDECISION

`nupolicydecision.NUPolicyDecision(bambou.nurest_object.NUMetaRESTObject,)` :

This object is a read only object that provides the policy decisions for a particular VM interface.

### 288.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `egress_acls`: List of actual Egress ACLs that will be applied on the interface of this VM
- `egress_qos`: Egress QoS primitive that was selected
- `fip_acls`: List of actual Egress ACLs that will be applied on the interface of this VM
- `ingress_acls`: List of actual Ingress ACLs that will be applied on the interface of this VM
- `ingress_adv_fwd`: List of actual Ingress Redirect ACLs that will be applied on the interface of this VM
- `ingress_external_service_acls`: List of actual Ingress External Service ACLs that will be applied on the interface of this VM
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `qos`: QoS primitive that was selected based on inheritance policies
- `stats`: Stats primitive that was selected based on inheritance policies
- `external_id`: External object ID. Used for integration with third party systems

### 288.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nuqos.NUQOS</code>	<code>qoss</code>

### 288.3 Parents

- `nuvminterface.NUVMInterface`
- `nubridgeinterface.NUBridgeInterface`

- *nuhostinterface.NUHostInterface*

## NUPOLICYGROUP

`nupolicygroup.NUPolicyGroup(bambou.nurest_object.NUMetaRESTObject, ) :`

PolicyGroup is group of polycys on which a user can policies like ACL, QoS, etc.

### 289.1 Attributes

- `evpn_community_tag`: Assigned by VSD. An extended community or other similar BGP attribute to the specific EVPN / IP-VPN NLRI where the VM or network macro is being advertised.
- `name` (**Mandatory**): Name of the policy group
- `last_updated_by`: ID of the user who last updated the object.
- `template_id`: Determines which template ID this policy group belongs to.
- `description`: Describes this policy group
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `policy_group_id`: PG ID for the subnet. This is unique per domain and will be in the range 1-4095
- `external`: Indicates whether this PG is internal to VSP or not.
- `external_id`: External object ID. Used for integration with third party systems
- `type` (**Mandatory**): Type of policy group.

### 289.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nujob.NUJob</code>	<code>jobs</code>
<code>nuvport.NUVPort</code>	<code>vports</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>

### 289.3 Parents

- `nudomain.NUDomain`
- `nuvport.NUVPort`

- *nuvminterface.NUVMInterface*
- *nubridgeinterface.NUBridgeInterface*
- *nul2domain.NUL2Domain*
- *nuhostinterface.NUHostInterface*
- *nume.NUMe*

## NUPOLICYGROUPTEMPLATE

`nupolicygrouptemplate.NUPolicyGroupTemplate(bambou.nurest_object.NUMetaRESTObject,)` :

PolicyGroupTemplate represents the template of a policy group object. PolicyGroup is group of vports on which a user can policies like ACL, QoS, etc.

### 290.1 Attributes

- `evpn_community_tag`: An extended community or other similar BGP attribute to the specific EVPN / IP-VPN NLRI where the VM or network macro is being advertised.
- `name` (**Mandatory**): Name of the policy group
- `last_updated_by`: ID of the user who last updated the object.
- `description`: Describes this policy group
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external`: Indicates whether this PG is internal to VSP or not.
- `external_id`: External object ID. Used for integration with third party systems
- `type` (**Mandatory**): Type of policy group.

### 290.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nujob.NUJob</code>	<code>jobs</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>

### 290.3 Parents

- `nul2domaintemplate.NUL2DomainTemplate`
- `nudomaintemplate.NUDomainTemplate`





## NUPORT

`nuport.NUPort (bambou.nurest_object.NUMetaRESTObject, ) :`

Represents Port under a particular gateway object or redundant group object.

### 291.1 Attributes

- `vlan_range`: VLAN Range of the Port. Format must conform to a-b,c,d-f where a,b,c,d,f are integers between 0 and 4095.
- `name` (**Mandatory**): Name of the Port
- `last_updated_by`: ID of the user who last updated the object.
- `template_id`: The ID of the template that this Port was created from
- `permitted_action`: The permitted action to USE/EXTEND this Gateway.
- `description`: A description of the Port
- `physical_name` (**Mandatory**): Identifier of the Port
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `port_type` (**Mandatory**): Type of the Port.
- `is_resilient`: States if this port instance is resilient (redundant). An example would be a Multi-Chassis LAG port.
- `use_user_mnemonic`: determines whether to use user mnemonic of the Port
- `user_mnemonic`: user mnemonic of the Port
- `associated_egress_qos_policy_id`: ID of the Egress QOS Policy associated with this Vlan.
- `associated_redundant_port_id`: ID of the redundant port to which this Port instance may be associated to.
- `status`: Status of the port.
- `external_id`: External object ID. Used for integration with third party systems

## 291.2 Children

<b>class</b>	<b>fetcher</b>
<i>nupermision.NUPermission</i>	permissions
<i>numetadata.NUMetadata</i>	metadatas
<i>nuvlan.NUVLAN</i>	vlangs
<i>nualarm.NUAlarm</i>	alarms
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuenterpriasepermission.NUEnterprisePermission</i>	enterprise_permissions
<i>nueventlog.NUEventLog</i>	event_logs

## 291.3 Parents

- *nuredundancygroup.NURedundancyGroup*
- *nuautodiscoveredgateway.NUAutoDiscoveredGateway*
- *nugateway.NUGateway*

## NUPORTTEMPLATE

`nuporttemplate.NUPortTemplate(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents a Port Template object under a given gateway template object.

### 292.1 Attributes

- `vlan_range`: VLAN Range of the Port. Format must conform to a-b,c,d-f where a,b,c,d,f are integers between 0 and 4095.
- `name` (**Mandatory**): Name of the Port
- `last_updated_by`: ID of the user who last updated the object.
- `description`: A description of the Port
- `physical_name` (**Mandatory**): Identifier of the Port
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `port_type` (**Mandatory**): Type of the Port - NETWORK, ACCESS Possible values are ACCESS, NETWORK, .
- `associated_egress_qos_policy_id`: ID of the Egress QOS Policy associated with this Vlan.
- `external_id`: External object ID. Used for integration with third party systems

### 292.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuvlantemplate.NUVLANTemplate</code>	<code>vlan_templates</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 292.3 Parents

- `nugatewaytemplate.NUGatewayTemplate`



## NUPUBLICNETWORKMACRO

`nupublicnetworkmacro.NUPublicNetworkMacro(bambou.nurest_object.NUMetaRESTObject, ) :`

Similar to the enterprise macros, the public network macro allows an administrator of an enterprise to define range of subnets that can be used by users in the ACL definition

### 293.1 Attributes

- `ip_type`: IPv4 or IPv6(only IPv4 is supported in R1.0) Possible values are IPV4, IPV6, .
- `name` (**Mandatory**): Name of the current entity(Zone or zone template or subnet etc..) Valid characters are alphabets, numbers, space and hyphen( - ).
- `last_updated_by`: ID of the user who last updated the object.
- `address` (**Mandatory**): IP address of the subnet defined. In case of zone, this is an optional field for and allows users to allocate an IP address range to a zone. The VSD will auto-assign IP addresses to subnets from this range if a specific IP address is not defined for the subnet
- `netmask` (**Mandatory**): Netmask of the subnet defined
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 293.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nueventlog.NUEventLog</i>	event_logs

### 293.3 Parents

- *nuenterprise.NUEnterprise*



## NUQOS

`nuqos.NUQOS (bambou.nurest_object.NUMetaRESTObject, ) :`

The object manipulates the QoS parameters attached to a domain, zone, or subnet.

### 294.1 Attributes

- `fip_committed_burst_size`: Committed burst size setting in kilo-bytes (kilo-octets) for FIP Shaper.
- `fip_committed_information_rate`: Committed information rate setting in Mb/s for FIP Shaper.
- `fip_peak_burst_size`: Peak burst size setting in kilo-bytes (kilo-octets) for FIP rate limiting.
- `fip_peak_information_rate`: Peak rate setting for FIP rate limiting in Mb/s;
- `fip_rate_limiting_active`: Flag the indicates whether FIP rate limiting is enabled or disabled
- `bum_committed_burst_size`: Committed burst size setting in kilo-bytes (kilo-octets) for BUM Shaper.
- `bum_committed_information_rate`: Committed information rate setting in Mb/s for BUM Shaper.
- `bum_peak_burst_size`: Peak burst size setting in kilo-bytes (kilo-octets) for Broadcast/Multicast rate limiting (BUM).
- `bum_peak_information_rate`: Peak rate setting in Mb/s for Broadcast/Multicast rate limiting
- `bum_rate_limiting_active`: Flag the indicates whether Broadcast/Multicast rate limiting is enabled or disabled
- `name` (**Mandatory**): A unique name of the QoS object
- `last_updated_by`: ID of the user who last updated the object.
- `rate_limiting_active`: Identifies if rate limiting must be implemented
- `active`: If enabled, it means that this ACL or QOS entry is active
- `peak`: Peak Information Rate : Peak bandwidth that is allowed from each VM in Mb/s; only whole values allowed and 'INFINITY' if rate limiting is disabled.
- `service_class` (**Mandatory**): Class of service to be used. Service classes in order of priority are A(1), B(2), C(3), D(4), E(5), F(6), G(7) and H(8) Possible values are NONE, A, B, C, D, E, F, G, H, .
- `description`: A description of the QoS object
- `rewrite_forwarding_class`: Specifies if the rewrite flag is set for the QoS policy / template
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level

- `committed_burst_size`: Committed Burst Size : Burst size associated with the rate limiter in kilo-bytes (kilo-octets); only whole values are supported.
- `committed_information_rate`: Committed Information Rate : Committed bandwidth that is allowed from each VM in Mb/s; only whole values supported.
- `trusted_forwarding_class`: Specifies if the trusted flag is set for the QoS policy / template
- `assoc_qos_id`: ID of object associated with this QoS object
- `associated_dscp_forwarding_class_table_id`: ID of the DSCP->Forwarding Class used by this Qos Policy
- `associated_dscp_forwarding_class_table_name`: Name of the DSCP->Forwarding Class used by this Qos Policy
- `burst`: Peak Burst Size : The maximum burst size associated with the rate limiter in kilo-bytes (kilo-octets); only whole values allowed and 'INFINITY' if rate limiting is disabled.
- `external_id`: External object ID. Used for integration with third party systems

## 294.2 Children

<b>class</b>	<b>fetcher</b>
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuvm.NUVM</i>	vms
<i>nueventlog.NUEventLog</i>	event_logs

## 294.3 Parents

- *nuzone.NUZone*
- *nusubnettemplate.NUSubnetTemplate*
- *nudomain.NUDomain*
- *nuvport.NUVPort*
- *nusubnet.NUSubnet*
- *nuzonemplate.NUZoneTemplate*
- *nubridgeinterface.NUBridgeInterface*
- *nul2domaintemplate.NUL2DomainTemplate*
- *nul2domain.NUL2Domain*
- *nuhostinterface.NUHostInterface*
- *nudomaintemplate.NUDomainTemplate*
- *nupolicydecision.NUPolicyDecision*



## NURATELIMITER

`nuratelimiter.NURateLimiter(bambou.nurest_object.NUMetaRESTObject, ) :`

Rate Limiter object that contains peak, burst and cir. It can be associated with Egress QOS policy objects.

### 295.1 Attributes

- **name (Mandatory):** A unique name of the Rate Limiter object
- **last\_updated\_by:** ID of the user who last updated the object.
- **peak\_burst\_size:** Peak Burst Size : The maximum burst size associated with the rate limiter in kilo-bits; only whole values are supported.
- **peak\_information\_rate:** Peak Information Rate : Peak bandwidth allowed in Mb/s; only whole values supported.
- **description:** A description of the Rate Limiter object
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **committed\_information\_rate:** Committed Information Rate : Committed bandwidth that is allowed in Mb/s; only whole values supported.
- **external\_id:** External object ID. Used for integration with third party systems

### 295.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 295.3 Parents

- `nume.NUMe`
- `nuenterprise.NUEnterprise`



## NUREDIRECTIONTARGET

`nuredirectiontarget.NURedirectionTarget (bambou.nurest_object.NUMetaRESTObject, ) :`

A group/collection of vports that belong to the same domain.

### 296.1 Attributes

- `esi`: ESI id, globally unique
- `name` (**Mandatory**): Name of this redirection target
- `last_updated_by`: ID of the user who last updated the object.
- `redundancy_enabled` (**Mandatory**): Allow/Disallow redundant appliances and VIP
- `template_id`: Template to which this redirection target belongs to
- `description`: Description of this redirection target
- `virtual_network_id`: Auto Generated by VSD. Each vPortTag with redundancy=enable and Endpoint-Type != none will have a globally unique ESI & VNID generated by VSD
- `end_point_type` (**Mandatory**): EndpointType defines the type of header rewrite and forwarding performed by VRS when the endpoint is used as a PBR destination. NONE type is deprecated. Possible values are NONE, L3, VIRTUAL\_WIRE, .
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `trigger_type`: Trigger type, THIS IS READ ONLY. Possible values are NONE, GARP, .
- `external_id`: External object ID. Used for integration with third party systems

### 296.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuvirtualip.NUVirtualIP</code>	<code>virtual_ips</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nujob.NUJob</code>	<code>jobs</code>
<code>nuvport.NUVPort</code>	<code>vports</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>

## 296.3 Parents

- *nudomain.NUDomain*
- *nuvport.NUVPort*
- *nuvminterface.NUVMInterface*
- *nubridgeinterface.NUBridgeInterface*
- *nul2domain.NUL2Domain*
- *nuhostinterface.NUHostInterface*
- *nume.NUMe*

## NUREDIRECTIONTARGETTEMPLATE

`nuredirectiontargettemplate.NURedirectionTargetTemplate (bambou.nurest_object.NUMetaRESTObj`

Template for a vporttag. It can be created only at the template level and available for all instances.

### 297.1 Attributes

- **name (Mandatory):** Name of this redirection target template
- **last\_updated\_by:** ID of the user who last updated the object.
- **redundancy\_enabled:** Allow/Disallow redundant appliances and VIP
- **description:** Description of this redirection target template
- **end\_point\_type (Mandatory):** VPortTagEndPointType is an enum. It defines the type of header rewrite and forwarding performed by VRS when the endpoint is used as a PBR destination. Possible values are NONE, L3, VIRTUAL\_WIRE.
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **trigger\_type:** Trigger type, could be NONE/GARP - THIS IS READONLY
- **external\_id:** External object ID. Used for integration with third party systems

### 297.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nujob.NUJob</i>	jobs
<i>nueventlog.NUEventLog</i>	event_logs

### 297.3 Parents

- *nul2domaintemplate.NUL2DomainTemplate*
- *nudomaintemplate.NUDomainTemplate*



## NUREDUNDANCYGROUP

`nuredundancygroup.NURedundancyGroup(bambou.nurest_object.NUMetaRESTObject,)` :

Represents Redundant Group formed by two Gateways.

### 298.1 Attributes

- **name (Mandatory)**: Name of the Redundancy Group
- **last\_updated\_by**: ID of the user who last updated the object.
- **gateway\_peer1\_autodiscovered\_gateway\_id**: The Auto Discovered Gateway configuration owner in this Redundant Group.
- **gateway\_peer1\_id**: The gateway configuration owner in this Redundant Group. when Redundant Group is deleted this gateway will receive vport associations
- **gateway\_peer1\_name**: The gateway configuration owner name in this Redundant Group
- **gateway\_peer2\_autodiscovered\_gateway\_id**: The Auto Discovered Gateway peer in this Redundant Group
- **gateway\_peer2\_name**: The gateway peer name in this Redundant Group
- **redundant\_gateway\_status**: The status of Redundant Group, possible values are FAILED, SUCCESS  
Possible values are FAILED, SUCCESS, .
- **permitted\_action**: The permitted action to USE/EXTEND this Gateway Possible values are USE, READ, ALL, INSTITUTE, EXTEND, DEPLOY, .
- **personality**: derived personality of the Redundancy Group - VSG,VRSG,NSG,OTHER Possible values are VSG, VSA, VRSG, DC7X50, NSG, HARDWARE\_VTEP, OTHER, .
- **description**: Description of the Redundancy Group
- **enterprise\_id**: The enterprise associated with this Redundant Group. This is a read only attribute
- **entity\_scope**: Specify if scope of entity is Data center or Enterprise level
- **vtep**: Represent the system ID or the Virtual IP of a service used by a Gateway (VSG for now) to establish a tunnel with a remote VSG or hypervisor. The format of this field is consistent with an IP address.
- **external\_id**: External object ID. Used for integration with third party systems

## 298.2 Children

<b>class</b>	<b>fetcher</b>
<i>nugateway.NUGateway</i>	gateways
<i>nupermision.NUPermission</i>	permissions
<i>nuwanservice.NUWANService</i>	wan_services
<i>numetadata.NUMetadata</i>	metadatas
<i>nualarm.NUAlarm</i>	alarms
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuenterprisepermission.NUEnterprisePermission</i>	enterprise_permissions
<i>nuport.NUPort</i>	ports
<i>nuvsgredundantport.NUVsgRedundantPort</i>	vsg_redundant_ports
<i>nuvsgredundantport.NUVsgRedundantPort</i>	vsg_redundant_ports
<i>nueventlog.NUEventLog</i>	event_logs

## 298.3 Parents

- *nume.NUMe*
- *nuenterprise.NUEnterprise*



## NUREDUNDANTPORT

`nuredundantport.NURedundantPort(bambou.nurest_object.NUMetaRESTObject,)` :

Represents a Port under a particular gateway object or redundant group object.

### 299.1 Attributes

- `vlan_range`: VLAN Range of the Port. Format must conform to a-b,c,d-f where a,b,c,d,f are integers between 0 and 4095.
- `name` (**Mandatory**): Name of the Port
- `last_updated_by`: ID of the user who last updated the object.
- `permitted_action`: The permitted action to USE/EXTEND this Gateway.
- `description`: A description of the Port
- `physical_name` (**Mandatory**): Identifier of the Port
- `infrastructure_profile_id`: The ID of the infrastructure profile this instance is associated with.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `port_peer1_id`: The master gateway peer port id.
- `port_peer2_id`: The slave gateway peer port id.
- `port_type` (**Mandatory**): Type of the Port.
- `use_untagged_heartbeat_vlan`: A flag to indicate if for this redundant port an untagged heartbeat VLAN is to be used. If this is not set then will use the heartbeat VLAN set by the NS redundant group
- `use_user_mnemonic`: determines whether to use user mnemonic of the Port
- `user_mnemonic`: user mnemonic of the Port
- `associated_egress_qos_policy_id`: ID of the Egress QOS Policy associated with this Vlan.
- `status`: Status of the port.
- `external_id`: External object ID. Used for integration with third party systems

## 299.2 Children

<b>class</b>	<b>fetcher</b>
<i>numetadata.NUMetadata</i>	metadatas
<i>nuvlan.NUVLAN</i>	vlan
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nunsport.NUNSPort</i>	ns_ports

## 299.3 Parents

- *nunsredundantgatewaygroup.NUNSRedundantGatewayGroup*

## NUSHAREDNETWORKRESOURCE

`nusharednetworkresource.NUSharedNetworkResource(bambou.nurest_object.NUMetaRESTObject,)` :

This defines shared infrastructure resources that are created by user with CSPROOT role. These resources can be used by all the enterprises in the data center for various purposes. Examples of shared resources are public subnet, floating subnet, public L2 domain, etc.

### 300.1 Attributes

- `ecmp_count`: Domain specific Equal-cost multi-path routing count, `ECMPCount = 1` means no ECMP
- `dhcp_managed`: `true` if DHCP is enabled else it is `false`. This value is always `true` for network resource of type `PUBLIC` or `FLOATING`.
- `back_haul_route_distinguisher`: `backHaulRouteDistinguisher` of the Shared Resource
- `back_haul_route_target`: `backHaulRouteTarget` of the Shared Resource
- `back_haul_vnid`: `backHaulVNID` of the Shared Resource
- `name` (**Mandatory**): Name of the shared resource. Valid characters are alphabets, numbers, space and hyphen(-).
- `last_updated_by`: ID of the user who last updated the object.
- `gateway`: Gatemask configured on the shared resource
- `access_restriction_enabled`: Boolean indicates that this shared network resource is available to everyone by default or not
- `address` (**Mandatory**): Address configured on the shared resource
- `permitted_action_type`: Permitted action on this shared network resource
- `description`: Description of the shared resource
- `netmask` (**Mandatory**): Netmask configured on the shared resource
- `shared_resource_parent_id`: Parent ID of the floating IP subnet to which this FIP subnet must be attached. If empty it will be created in a new domain.
- `vn_id`: VNID of the Shared Resource
- `underlay`: Indicates whether this shared subnet is in underlay or not.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `domain_route_distinguisher`: Route distinguisher configured on the shared resource
- `domain_route_target`: Route target configured on the shared resource
- `uplink_gw_vlan_attachment_id`: VLAN ID to which this vport must be attached

- `uplink_interface_ip`: IP address of the host interface
- `uplink_interface_mac`: MAC address of the host interface
- `uplink_vport_name`: Name of the uplink vport
- `external_id`: External object ID. Used for integration with third party systems
- `type` (**Mandatory**): Type of the shared resource.

## 300.2 Children

class	fetcher
<i>nuaddressrange.NUAddressRange</i>	<code>address_ranges</code>
<i>numetadata.NUMetadata</i>	<code>metadatas</code>
<i>nudhcpoption.NUDHCPOption</i>	<code>dhcp_options</code>
<i>nuglobalmetadata.NUGlobalMetadata</i>	<code>global_metadatas</code>
<i>nuenterprisepermission.NUEnterprisePermission</i>	<code>enterprise_permissions</code>
<i>nuvpnconnection.NUVPNConnection</i>	<code>vpn_connections</code>
<i>nustaticroute.NUStaticRoute</i>	<code>static_routes</code>

## 300.3 Parents

- *nume.NUMe*
- *nuenterprise.NUEnterprise*

## NUSITEINFO

```
nusiteinfo.NUSiteInfo(bambou.nurest_object.NUMetaRESTObject,):
```

Remote Site info.

### 301.1 Attributes

- **name (Mandatory)**: name of the Remote Site.
- **last\_updated\_by**: ID of the user who last updated the object.
- **address (Mandatory)**: unique fqdn/address of the remote site
- **description**: Description of the Remote Site.
- **site\_identifier**: unique identifier of the remote site
- **xmpp\_domain (Mandatory)**: unique xmpp domain name of the remote site
- **entity\_scope**: Specify if scope of entity is Data center or Enterprise level
- **external\_id**: External object ID. Used for integration with third party systems

### 301.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas

### 301.3 Parents

- *nume.NUMe*



## NUSTATICROUTE

`nustaticroute.NUStaticRoute (bambou.nurest_object.NUMetaRESTObject, ) :`

Static routes allow end users to define how traffic is routed through the dVRS in addition to the routes learned by VSC through VM activation. By using static routes, end users can define for example that all traffic with a destination address towards a specific subnet must be forwarded to a specific VM attached in the dVRS and this VM could be a firewall.

### 302.1 Attributes

- `ip_type`: IPv4 or IPv6
- `last_updated_by`: ID of the user who last updated the object.
- `address` (**Mandatory**): IP address of the route
- `netmask` (**Mandatory**): Netmask associated with the route
- `next_hop_ip` (**Mandatory**): IP address of the next hop. This must be a VM attached to the dVRS
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `route_distinguisher`: Route distinguisher associated with the nexthop. System generates this identifier automatically
- `external_id`: External object ID. Used for integration with third party systems

### 302.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>

### 302.3 Parents

- `nusharednetworkresource.NUSharedNetworkResource`
- `nudomain.NUDomain`
- `nuvminterface.NUVMInterface`

- *nul2domain.NUL2Domain*
- *nuhostinterface.NUHostInterface*
- *nume.NUMe*



## NUSTATISTICS

`nustatistics.NUStatistics(bambou.nurest_object.NUMetaRESTObject, ) :`

Retrieves the statistics for a particular domain, zone, subnet, or VM.

### 303.1 Attributes

- `version`: Version of this Sequence number.
- `end_time`: End time for the statistics to be retrieved
- `start_time`: Start time for the statistics to be retrieved
- `stats_data`: Map<TCAMetric, Long[]> TCAMetric is an Enum. Possible values are packets\_in, bytes\_in, packets\_in\_dropped, packets\_in\_errors, packets\_out, bytes\_out, packets\_out\_dropped, packets\_out\_errors, packets\_dropped\_rate\_limit
- `number_of_data_points`: Number of data points between start time and end time

### 303.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 303.3 Parents

- `nuzone.NUZone`
- `nuingressexternalservicetemplateentry.NUIngressExternalServiceTemplateEntry`
- `nudomain.NUDomain`
- `nuegressaclentrytemplate.NUEgressACLEntryTemplate`
- `nuvport.NUVPort`
- `nusubnet.NUSubnet`
- `nuvminterface.NUVMInterface`
- `nubridgeinterface.NUBridgeInterface`

- *nuingressadvfwdentrytemplate.NUIngressAdvFwdEntryTemplate*
- *nutier.NUTier*
- *nuingressaclentrytemplate.NUIngressACLEntryTemplate*
- *nul2domain.NUL2Domain*
- *nuhostinterface.NUHostInterface*
- *nunsport.NUNSPort*

## NUSTATISTICSPOLICY

`nustatisticspolicy.NUStatisticsPolicy(bambou.nurest_object.NUMetaRESTObject,)` :

Defines the frequency of statistics collection associated with an object.

### 304.1 Attributes

- **name (Mandatory)**: Name of statistics policy
- **last\_updated\_by**: ID of the user who last updated the object.
- **data\_collection\_frequency (Mandatory)**: How frequent to collect statistics in seconds
- **description**: A description of the statistics policy
- **entity\_scope**: Specify if scope of entity is Data center or Enterprise level
- **external\_id**: External object ID. Used for integration with third party systems

### 304.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas

### 304.3 Parents

- *nuzone.NUZone*
- *nudomain.NUDomain*
- *nuvport.NUVPort*
- *nusubnet.NUSubnet*
- *nutier.NUTier*
- *nul2domain.NUL2Domain*
- *nunsport.NUNSPort*



## NUSTATSCOLLECTORINFO

`nustatscollectorinfo.NUStatsCollectorInfo(bambou.nurest_object.NUMetaRESTObject, ) :`

Identifies the IP address of the stats collector entity that must be used.

### 305.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `address_type`: Type for stats collector address Possible values are ip, fqdn, .
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `port`: Port(s) of the stats collector process
- `ip_address`: IP address(es) of the stats collector process
- `proto_buf_port`: Protobuf Port(s) of the stats collector process
- `external_id`: External object ID. Used for integration with third party systems

### 305.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	metadatas
<code>nuglobalmetadata.NUGlobalMetadata</code>	global_metadatas

### 305.3 Parents

- `nume.NUme`



## NUSUBNET

```
nusubnet.NUSubnet (bambou.nurest_object.NUMetaRESTObject, ) :
```

This is the definition of a subnet associated with a zone.

### 306.1 Attributes

- `pat_enabled`: None
- `ip_type`: IPv4 or IPv6
- `maintenance_mode`: `maintenanceMode` is an enum that indicates if the SubNetwork is accepting VM activation requests.
- `name` (**Mandatory**): Name of the current entity (Zone or zone template or subnet etc..) Valid characters are alphabets, numbers, space and hyphen ( - ).
- `last_updated_by`: ID of the user who last updated the object.
- `gateway`: The IP address of the gateway of this subnet
- `gateway_mac_address`: None
- `address`: IP address of the subnet defined. In case of zone, this is an optional field for and allows users to allocate an IP address range to a zone. The VSD will auto-assign IP addresses to subnets from this range if a specific IP address is not defined for the subnet
- `template_id`: The ID of the subnet template that this subnet object was derived from
- `service_id`: The service ID used by the VSCs to identify this subnet
- `description`: A description field provided by the user that identifies the subnet
- `netmask`: Netmask of the subnet defined
- `vn_id`: Current Network's globally unique VXLAN network identifier generated by VSD
- `encryption`: Determines whether or not IPSEC is enabled.
- `underlay_enabled`: Indicates whether UNDERLAY is enabled for the subnets in this domain
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `policy_group_id`: PG ID for the subnet. This is unique per domain and will be in the range 1-4095
- `route_distinguisher`: The Route Distinguisher value assigned by VSD for this subnet that is used by the BGP-EVPN protocol in VSC
- `route_target`: The Route Target value assigned by VSD for this subnet that is used by the BGP-EVPN protocol in VSC

- `split_subnet`: Need to add correct description
- `proxy_arp`: when set VRS will act as ARP Proxy
- `associated_application_id`: The associated application ID.
- `associated_application_object_id`: The associated application object ID.
- `associated_application_object_type`: The associated application object type. Refer to API section for supported types.
- `associated_multicast_channel_map_id`: The ID of the Multi Cast Channel Map this Subnet/Subnet Template is associated with. This has to be set when `enableMultiCast` is set to `ENABLED`
- `associated_shared_network_resource_id`: The ID of public subnet that is associated with this subnet
- `public`: when set to true means public subnet under a public zone
- `multicast`: multicast is enum that indicates multicast policy on Subnet/Subnet Template.
- `external_id`: External object ID. Used for integration with third party systems

## 306.2 Children

class	fetcher
<i>nutca.NUTCA</i>	<code>tcas</code>
<i>nuaddressrange.NUAddressRange</i>	<code>address_ranges</code>
<i>nuvmresync.NUVMResync</i>	<code>vm_resyncs</code>
<i>numetadata.NUMetadata</i>	<code>metadatas</code>
<i>nudhcpoption.NUDHCPOption</i>	<code>dhcp_options</code>
<i>nuvirtualip.NUVirtualIP</i>	<code>virtual_ips</code>
<i>nuglobalmetadata.NUGlobalMetadata</i>	<code>global_metadatas</code>
<i>nuvm.NUVM</i>	<code>vms</code>
<i>nuvminterface.NUVMInterface</i>	<code>vm_interfaces</code>
<i>nuqos.NUQOS</i>	<code>qoss</code>
<i>nuvport.NUVPort</i>	<code>vports</code>
<i>nuipreservation.NUIPReservation</i>	<code>ip_reservations</code>
<i>nustatistics.NUStatistics</i>	<code>statistics</code>
<i>nustatisticspolicy.NUStatisticsPolicy</i>	<code>statistics_policies</code>
<i>nueventlog.NUEventLog</i>	<code>event_logs</code>

## 306.3 Parents

- *nuzone.NUZone*
- *nusubnettemplate.NUSubnetTemplate*
- *nudomain.NUDomain*
- *nume.NUMe*



## NUSUBNETTEMPLATE

`nusubnettemplate.NUSubnetTemplate(bambou.nurest_object.NUMetaRESTObject, ) :`

As domain and zone objects, subnet objects are created in VSD as derived by templates. This object describes the subnet template.

### 307.1 Attributes

- `ip_type`: IPv4 or IPv6
- `name` (**Mandatory**): Name of the current entity (Zone or zone template or subnet etc..) Valid characters are alphabets, numbers, space and hyphen( - ).
- `last_updated_by`: ID of the user who last updated the object.
- `gateway`: The IP address of the gateway of this subnet
- `address` (**Mandatory**): IP address of the subnet defined. In case of zone, this is an optional field for and allows users to allocate an IP address range to a zone. The VSD will auto-assign IP addresses to subnets from this range if a specific IP address is not defined for the subnet
- `description`: A description field provided by the user that identifies the subnet
- `netmask` (**Mandatory**): Netmask of the subnet defined
- `encryption`: Determines whether or not IPSEC is enabled. Possible values are INHERITED, ENABLED, DISABLED, .
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `split_subnet`: Need to add correct description
- `proxy_arp`: when set VRS will act as ARP Proxy
- `associated_multicast_channel_map_id`: The ID of the Multi Cast Channel Map this Subnet/Subnet Template is associated with. This has to be set when enableMultiCast is set to ENABLED
- `multicast`: Indicates multicast policy on Subnet/Subnet Template.
- `external_id`: External object ID. Used for integration with third party systems

## 307.2 Children

<b>class</b>	<b>fetcher</b>
<i>nuaddressrange.NUAddressRange</i>	address_ranges
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuqos.NUQOS</i>	qoss
<i>nusubnet.NUSubnet</i>	subnets
<i>nueventlog.NUEventLog</i>	event_logs

## 307.3 Parents

- *nuzonetemplate.NUZoneTemplate*
- *nudomaintemplate.NUDomainTemplate*

## NUSYSTEMCONFIG

`nusystemconfig.NUSystemConfig(bambou.nurest_object.NUMetaRESTObject,)` :

The system configuration which can be dynamically managed using rest api.

### 308.1 Attributes

- `acl_allow_origin`: Defines the domains allowed for access control list.
- `ecmp_count`: System Default Equal-cost multi-path routing count, Every Domain derives ECMP count from this value unless specifically set for the domain
- `ldap_sync_interval`: LDAP Sync-Up task interval in seconds.
- `ldap_trust_store_certificate`: Location of the truststore which is need to store LDAP server certificates. Default is cacerts located in `java.home/lib/security/cacerts`. Uncomment below setting if you need to use a different file
- `ldap_trust_store_password`: Password to access the truststore. Uncomment below line to change its value.
- `ad_gateway_purge_time`: Timers in sec for undefined vms to be deleted(min =7200, max = 86400).
- `rd_lower_limit`: route distinguisher lower limit
- `rd_public_network_lower_limit`: route distinguisher public network lower limit
- `rd_public_network_upper_limit`: route distinguisher public network upper limit
- `rd_upper_limit`: route distinguisher upper limit
- `dhcp_option_size`: Defines total DHCP options that can be set on a domain.
- `vm_cache_size`: LRU Map size for vm, this value has to set based on memory given to VSD jvm not finalized.
- `vm_purge_time`: Timers in sec for undefined vms to be deleted.
- `vm_resync_deletion_wait_time`: After resync on vm , if no controller returns with a VM request with in the below timeframe then it will get deleted deletion wait time in minutes.
- `vm_resync_outstanding_interval`: Outstanding VM resync interval (in secs). System wide value.
- `vm_unreachable_cleanup_time`: Timers in sec for unreachable VMs for cleanup.
- `vm_unreachable_time`: Timers in sec for unreachable VMs.
- `vnid_lower_limit`: Virtual network ID offset
- `vnid_public_network_lower_limit`: Virtual network ID public network lower limit

- `vnid_public_network_upper_limit`: Virtual network ID public network upper limit
- `vnid_upper_limit`: Virtual network ID upper limit
- `api_key_renewal_interval`: Defines the interval in seconds, before the expiry time, that can used to renew the apiKey by making me API call. Minimum value is 1 min and maximum is 5 min.
- `api_key_validity`: Defines the apiKey validity duration in seconds. Default is 24 hours and minimum value is 10 min.
- `lru_cache_size_per_subnet`: LRU Map size per subnet (to hold the deleted vm's ip addresses).
- `vsc_on_same_version_as_vsd`: This flag is used to indicate that whether VSC is on the same version as VSD or not.
- `vsd_read_only_mode`: True means VSD readonly mode enabled. False means VSD readonly mode disabled
- `vsd_upgrade_is_complete`: This flag is used to indicate that whether VSD upgrade is complete, it is expected that csproot will set to true, after VSD upgrade is complete and also making sure that all VSC's audits and Gateway audits with VSD are done
- `as_number`: Autonomous System Number, Used for RT/RD auto-generation
- `rt_lower_limit`: route target lower limit
- `rt_public_network_lower_limit`: route target public network lower limit
- `rt_public_network_upper_limit`: route target public network upper limit
- `rt_upper_limit`: route target upper limit
- `evpnbgp_community_tag_as_number`: Autonomous System Number, Used for EVPNBGPCCommunityTag auto-generation
- `evpnbgp_community_tag_lower_limit`: EVPNBGPCCommunityTag lower limit
- `evpnbgp_community_tag_upper_limit`: EVPNBGPCCommunityTag upper limit
- `page_max_size`: Defines upper bound for the page size. Configured or input page size should be less than this max page size.
- `page_size`: Defines the page size for the results returned by the REST call.
- `last_updated_by`: ID of the user who last updated the object.
- `max_failed_logins`: Maximum failed login attempts before the account is locked (min = 5, max = 10). 0 = not enforced (unlimited attempts). This is not enforced if LDAP is used for authorization
- `max_response`: Defines maximum results returned by the REST call (allowed max=5000).
- `reflexive_acl_timeout`: Defines the timeout in seconds for reflexive ACLs. This value applies for both TCP and UDP connections. Default value is 180 seconds and the timeout should be between 10 to 86400 seconds.
- `service_id_upper_limit`: Service id upper limit system wide value
- `key_server_monitor_enabled`: Enable the keyserver debug monitor (ie. ksmon command)
- `key_server_vsd_data_synchronization_interval`: KeyServer time in seconds between full resyncs of VSD data (just in case of missed events)
- `offset_customer_id`: Customer id offset, this value has to be set before jboss starts , after that any change of value is ignored (minexclusive = 0, max = 20000) system wide value
- `offset_service_id`: Service id offset, this value has to be set before jboss starts during install time, after that any change of value is ignored (minexclusive = 0, max = 40000) system wide value

- `ejbca_nsg_certificate_profile`: EJBCA NSG Certificate Profile
- `ejbca_nsg_end_entity_profile`: EJBCA NSG End Entity Profile
- `ejbca_ocsp_responder_cn`: EJBCA OCSP Responder CommonName
- `ejbca_ocsp_responder_uri`: EJBCA OCSP Responder URI
- `ejbca_vsp_root_ca`: EJBCA VSP CA
- `alarms_max_per_object`: Maximum alarms per object for example max distinct alarms for specific VM (min = 5, max =20)
- `elastic_cluster_name`: Specifies the name of the Elastic Search Cluster.
- `inactive_timeout`: Defines the inactive timeout for the client. If the client is inactive for more than timeout, server clears off all the cache/information regarding the client. This value should be greater than event processor max timeout
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `domain_tunnel_type`: Default Domain Tunnel Type .Possible values are VXLAN,GRE Possible values are DC\_DEFAULT, GRE, VXLAN, .
- `post_processor_threads_count`: Post processor thread count.
- `group_key_default_sek_generation_interval`: Group Key Encryption Profile Default SEK Generation Interval
- `group_key_default_sek_lifetime`: Group Key Encryption Profile Default SEK Lifetime
- `group_key_default_sek_payload_encryption_algorithm`: Group Key Encryption Profile Default Sek Payload Encryption Algorithm.
- `group_key_default_sek_payload_signing_algorithm`: Group Key Encryption Profile Default Sek Payload Signing Algorithm.
- `group_key_default_seed_generation_interval`: Group Key Encryption Profile Default Seed Generation Interval
- `group_key_default_seed_lifetime`: Group Key Encryption Profile Default Seed Lifetime
- `group_key_default_seed_payload_authentication_algorithm`: Group Key Encryption Profile Default Seed Payload Authentication Algorithm.
- `group_key_default_seed_payload_encryption_algorithm`: Group Key Encryption Profile Default Seed Payload Encryption Algorithm.
- `group_key_default_seed_payload_signing_algorithm`: Group Key Encryption Profile Default Seed Payload Signature Algorithm.
- `group_key_default_traffic_authentication_algorithm`: Group Key Encryption Profile Default Traffic Authentication Algorithm.
- `group_key_default_traffic_encryption_algorithm`: Group Key Encryption Profile Default Traffic Encryption Algorithm.
- `group_key_default_traffic_encryption_key_lifetime`: Group Key Encryption Profile Default Traffic Encryption Key Lifetime
- `group_key_generation_interval_on_forced_re_key`: Time in seconds before new keys will be generated in the case of a forced re-key event
- `group_key_generation_interval_on_revoke`: Time in seconds before new keys will be generated in the case of a revoke event

- `group_key_minimum_sek_generation_interval`: Group Key Encryption Profile Minimum SEK Generation Interval
- `group_key_minimum_sek_lifetime`: Group Key Encryption Profile Minimum SEK Lifetime
- `group_key_minimum_seed_generation_interval`: Group Key Encryption Profile Default Seed Generation Interval
- `group_key_minimum_seed_lifetime`: Group Key Encryption Profile Default Seed Lifetime
- `group_key_minimum_traffic_encryption_key_lifetime`: Group Key Encryption Profile Minimum TEK Lifetime
- `nsg_bootstrap_endpoint`: NSG Bootstrap Endpoint
- `nsg_config_endpoint`: NSG Config Endpoint
- `nsg_local_ui_url`: NSG Local UI URL - will be redirected on NSG to localhost
- `esi_id`: ESI ID offset
- `stack_trace_enabled`: True to enable stacktrace in the REST call.
- `static_wan_service_purge_time`: Timers in sec for unreachable static WAN Services to be deleted.
- `statistics_enabled`: This flag is used to indicate if statistics is enabled in the system. CSProot is expected to activate this through the enable statistics script.
- `stats_collector_address`: Specify the ip address(es) of the stats collector.
- `stats_collector_port`: Specify the port number(s) of the stats collector.
- `stats_collector_proto_buf_port`: Specify the protobuf port number(s) of the stats collector.
- `stats_max_data_points`: Specifies the maximum number of data points to support.
- `stats_min_duration`: Default minimum duration for statistics to be displayed in UI is 30 days in seconds.
- `stats_number_of_data_points`: Specifies number of data points.
- `stats_tsdb_server_address`: Specifies the TSDB server location.
- `subnet_resync_interval`: After resync on a subnet , another resync on the same subnet is allowed based on the below value subnet resync complete wait time in min.
- `subnet_resync_outstanding_interval`: Outstanding subnet resync interval (in secs). System wide value.
- `customer_id_upper_limit`: Customer id upper limit, system wide value
- `customer_key`: Customer key associated with the license
- `avatar_base_path`: Defines location where image files needs to be copied. Above URL should be configured to read the file from this location.
- `avatar_base_url`: Defines the url to read the avatar image files
- `event_log_cleanup_interval`: Cleanup task run interval in seconds.
- `event_log_entry_max_age`: Maximum age in days for cleanup of the eventlog entries. On every periodic interval run, any eventlog entries older than this max age will be deleted.
- `event_processor_interval`: Defines time interval in milliseconds when events collected for a client should be processed.
- `event_processor_max_events_count`: Defines the maximum number of events to be collected in case of events burst.

- `event_processor_timeout`: Defines the maximum time period in milliseconds for the Rest server to wait before sending the events from the system.
- `two_factor_code_expiry`: Two Factor Code Expiry in Seconds
- `two_factor_code_length`: Two Factor Code Length
- `two_factor_code_seed_length`: Two Factor Seed length in bytes
- `external_id`: External object ID. Used for integration with third party systems
- `dynamic_wan_service_diff_time`: Timers in sec for dynamic WAN Services to be considered not seen by 7X50.
- `syslog_destination_host`: Specifies the remote syslog destination host
- `syslog_destination_port`: Specified the remote syslog destination port
- `sysmon_cleanup_task_interval`: Sysmon cleanup task run interval in seconds.
- `sysmon_node_presence_timeout`: Node presence timeout in seconds if no messages.
- `sysmon_probe_response_timeout`: Probe response timeout in seconds.

## 308.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

## 308.3 Parents

- `nume.NUMe`





## NUTCA

`nutca.NUTCA(bambou.nurest_object.NUMetaRESTObject, ) :`

Provides the definition of the Threshold Control Alarms.

### 309.1 Attributes

- `url_end_point`: URL endpoint to post Alarm data to when TCA is triggered
- `name` (**Mandatory**): The name of the TCA
- `last_updated_by`: ID of the user who last updated the object.
- `scope` (**Mandatory**): GLOBAL or LOCAL scope. Global refers to aggregate values across subnets, zones or domains. Local refers to traffic from/to individual VMs.
- `period` (**Mandatory**): The averaging period
- `description`: Description of the TCA
- `metric` (**Mandatory**): The metric associated with the TCA.
- `threshold` (**Mandatory**): The threshold that must be exceeded before an alarm is issued
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems
- `type` (**Mandatory**): Rolling average or sequence of samples over the averaging period.

### 309.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nualarm.NUAlarm</code>	<code>alarms</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>

### 309.3 Parents

- `nuzone.NUZone`
- `nudomain.NUDomain`

- *nuvport.NUVPort*
- *nusubnet.NUSubnet*
- *nuvminterface.NUVMInterface*
- *nubridgeinterface.NUBridgeInterface*
- *nutier.NUTier*
- *nul2domain.NUL2Domain*
- *nuhostinterface.NUHostInterface*
- *nume.NUMe*

## NUTIER

`nutier.NUTier(bambou.nurest_object.NUMetaRESTObject, ) :`

Tier represents a portion of an Application.

### 310.1 Attributes

- **name (Mandatory):** Name of the application tier.
- **last\_updated\_by:** ID of the user who last updated the object.
- **gateway:** The IP address of the gateway for this tier.
- **address:** IP address of the tier defined.
- **description:** Description of the application tier.
- **metadata:** Metadata field to store tier related data.
- **netmask:** Netmask for the tier.
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **associated\_application\_id:** The associated network macro ID.
- **associated\_floating\_ip\_pool\_id:** The associated floating IP Pool ID.
- **associated\_network\_macro\_id:** The associated network macro ID.
- **associated\_network\_object\_id:** The associated network object id.
- **associated\_network\_object\_type:** The associated network object type. Refer to API section for supported types.
- **external\_id:** External object ID. Used for integration with third party systems
- **type (Mandatory):** Type of the application tier.

## 310.2 Children

<b>class</b>	<b>fetcher</b>
<i>nutca.NUTCA</i>	tcas
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuvm.NUVM</i>	vms
<i>nuvport.NUVPort</i>	vports
<i>nustatistics.NUStatistics</i>	statistics
<i>nustatisticspolicy.NUStatisticsPolicy</i>	statistics_policies
<i>nueventlog.NUEventLog</i>	event_logs

## 310.3 Parents

- *nuapp.NUApp*

## NUUPLINKRD

`nuuplinkrd.NUUplinkRD(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents a network port uplink route distinguisher value.

### 311.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `route_distinguisher`: The uplink route distinguisher value is used to identify which route packets should be flowing through with regards to having multiple network ports on the VRS/NSG.
- `uplink_type`: Indicates the uplink type associated with the instance of Uplink Route Distinguisher.
- `external_id`: External object ID. Used for integration with third party systems

### 311.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas

### 311.3 Parents

- *nudomain.NUDomain*
- *nul2domain.NUL2Domain*
- *nume.NUMe*



## NUUSER

`nuuser.NUUser(bambou.nurest_object.NUMetaRESTObject, ) :`

Object that identifies the user functions.

### 312.1 Attributes

- `management_mode`: Management mode of the user object - allows for override of external authorization and syncup
- `password` (**Mandatory**): User password stored as a hash (SHA-1 encrypted)
- `last_name` (**Mandatory**): Last name of the user
- `last_updated_by`: ID of the user who last updated the object.
- `first_name` (**Mandatory**): First name of the user
- `disabled`: Status of the user account; true=disabled, false=not disabled; default value = false
- `email` (**Mandatory**): Email address of the user
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `mobile_number`: Mobile Number of the user
- `user_name` (**Mandatory**): Unique Username of the user. Valid characters are alphabets, numbers and hyphen(-).
- `avatar_data`: URL to the avatar data associated with the enterprise. If the avatarType is URL then value of avatarData should an URL of the image. If the avatarType BASE64 then avatarData should be BASE64 encoded value of the image
- `avatar_type`: Avatar type.
- `external_id`: External object ID. Used for integration with third party systems

### 312.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nuvm.NUVM</code>	<code>vms</code>
<code>nugroup.NUGroup</code>	<code>groups</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>

## 312.3 Parents

- *nugroup.NUGroup*
- *nume.NUMe*
- *numenterprise.NUEnterprise*



## NUVCENTER

`nuvcenter.NUVCenter(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents a VCenter.

### 313.1 Attributes

- `arp_reply`: Whether ARP Reply is enabled/disabled
- `v_require_nuage_metadata`: Whether split-activation or not (Openstack/CloudStack)
- `name` (**Mandatory**): Name of the VCenter
- `password` (**Mandatory**): Password for the VCenter user
- `last_updated_by`: ID of the user who last updated the object.
- `data_dns1`: Data DNS 1
- `data_dns2`: Data DNS 2
- `data_gateway`: Data Gateway
- `data_network_portgroup`: Data Network Port Group
- `datapath_sync_timeout`: Datapath Sync Timeout
- `secondary_nuage_controller`: IP address of the secondary Controller (VSC)
- `generic_split_activation`: Split-activation is needed from VRO
- `separate_data_network`: Whether Data will use the management network or not
- `personality`: VRS/VRS-G
- `description`: Description of the VCenter
- `metadata_server_ip`: Metadata Server IP
- `metadata_server_listen_port`: Metadata Server Listen Port
- `metadata_server_port`: Metadata Server Port
- `metadata_service_enabled`: Metadata Service Enabled
- `network_uplink_interface`: Network Uplink Interface to support PAT/NAT with no tunnels on VRS-VM
- `network_uplink_interface_gateway`: Network Uplink Interface Gateway
- `network_uplink_interface_ip`: Ip Address to support PAT/NAT with no tunnels on VRS-VM

- `network_uplink_interface_netmask`: Network Uplink Interface Netmask
- `nfs_log_server`: IP address of NFS server to send the VRS log
- `nfs_mount_path`: Location to mount the NFS server
- `mgmt_dns1`: DNS server 1
- `mgmt_dns2`: DNS server 2
- `mgmt_gateway`: Gateway for the IP address
- `mgmt_network_portgroup`: Management Network Port group
- `dhcp_relay_server`: To provide IP address of the interface from which you will connect to the DHCP relay server
- `site_id`: Site ID field for object profiles to support VSD Geo-redundancy
- `allow_data_dhcp`: Whether to get the Data IP for the VRS VM from DHCP or statically
- `allow_mgmt_dhcp`: Whether to get the management IP for the VRS VM from DHCP or statically
- `flow_eviction_threshold`: Flow Eviction Threshold
- `vm_network_portgroup`: VM Network Port Group Name
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `portgroup_metadata`: Port Group Meta data
- `host_level_management`: Flag to say if host level management is enabled
- `nova_client_version`: Nova client Version
- `nova_metadata_service_auth_url`: Nova metadata service auth url
- `nova_metadata_service_endpoint`: Nova metadata service endpoint
- `nova_metadata_service_password`: Nova metadata service password
- `nova_metadata_service_tenant`: Nova metadata service tenant
- `nova_metadata_service_username`: Nova metadata service username
- `nova_metadata_shared_secret`: Nova metadata shared secret
- `nova_region_name`: Nova region name
- `ip_address` (**Mandatory**): IP Address of the VCenter
- `primary_nuage_controller`: IP address of the primary Controller (VSC)
- `vrs_config_id`: The ID of the template that this Port was created from
- `vrs_password`: VRS password to be used by toolbox to communicate with VRS
- `vrs_user_name`: VRS user name to be used by toolbox to communicate with VRS
- `user_name` (**Mandatory**): User name of the VCenter
- `static_route`: static route to be configured in the VRS
- `static_route_gateway`: Gateway for the static route given above
- `static_route_netmask`: Nova region name
- `ntp_server1`: IP of the NTP server 1
- `ntp_server2`: IP of the NTP server 1

- `http_port`: Http proxy port for VCenter
- `https_port`: Https proxy port of the VCenter
- `mtu`: Maximum Transmission Unit for eth2 interface
- `multi_vmssupport`: Whether Multi VM is to be used or not
- `multicast_receive_interface`: Multicast Receive Interface
- `multicast_receive_interface_ip`: IP address for eth3 interface
- `multicast_receive_interface_netmask`: Multicast Interface netmask
- `multicast_receive_range`: Allowed Range to receive the Multicast traffic from
- `multicast_send_interface`: Multicast Send Interface
- `multicast_send_interface_ip`: IP address for eth3 interface
- `multicast_send_interface_netmask`: Multicast Interface netmask
- `multicast_source_portgroup`: Multi Cast Source Port Group Name
- `customized_script_url`: To provide a URL to install a custom app on VRS
- `auto_resolve_frequency`: Frequency at which VCenter issues are to be resolved
- `ovf_url`: The url for the ovf
- `external_id`: External object ID. Used for integration with third party systems

## 313.2 Children

class	fetcher
<i>nuvcenterdatacenter.NUVCenterDataCenter</i>	<code>vcenter_data_centers</code>
<i>numetadata.NUMetadata</i>	<code>metadatas</code>
<i>nuglobalmetadata.NUGlobalMetadata</i>	<code>global_metadatas</code>
<i>nujob.NUJob</i>	<code>jobs</code>
<i>nuvrsaddressrange.NUVRSAddressRange</i>	<code>vrs_address_ranges</code>

## 313.3 Parents

- *nume.NUMe*



## NUVCENTERCLUSTER

`nuvcentercluster.NUVCenterCluster(bambou.nurest_object.NUMetaRESTObject,):`  
VCenter Clusters.

### 314.1 Attributes

- `arp_reply`: Whether ARP Reply is enabled/disabled
- `v_require_nuage_metadata`: Whether split-activation or not (Openstack/CloudStack)
- `name` (**Mandatory**): Name of the Cluster
- `last_updated_by`: ID of the user who last updated the object.
- `data_dns1`: Data DNS 1
- `data_dns2`: Data DNS 2
- `data_gateway`: Data Gateway
- `data_network_portgroup`: Data Network Port Group
- `datapath_sync_timeout`: Datapath Sync Timeout
- `secondary_nuage_controller`: IP address of the secondary Controller (VSC)
- `generic_split_activation`: Split-activation is needed from VRO
- `separate_data_network`: Whether Data will use the management network or not
- `personality`: VRS/VRS-G
- `description`: Description of the Cluster
- `metadata_server_ip`: Metadata Server IP
- `metadata_server_listen_port`: Metadata Server Listen Port
- `metadata_server_port`: Metadata Server Port
- `metadata_service_enabled`: Metadata Service Enabled
- `network_uplink_interface`: Network Uplink Interface to support PAT/NAT with no tunnels on VRS-VM
- `network_uplink_interface_gateway`: Network Uplink Interface Gateway
- `network_uplink_interface_ip`: Ip Address to support PAT/NAT with no tunnels on VRS-VM
- `network_uplink_interface_netmask`: Network Uplink Interface Netmask

- `nfs_log_server`: IP address of NFS server to send the VRS log
- `nfs_mount_path`: Location to mount the NFS server
- `mgmt_dns1`: DNS server 1
- `mgmt_dns2`: DNS server 2
- `mgmt_gateway`: Gateway for the IP address
- `mgmt_network_portgroup`: Management Network Port group
- `dhcp_relay_server`: To provide IP address of the interface from which you will connect to the DHCP relay server
- `site_id`: Site ID field for object profiles to support VSD Geo-redundancy
- `allow_data_dhcp`: Whether to get the Data IP for the VRS VM from DHCP or statically
- `allow_mgmt_dhcp`: Whether to get the management IP for the VRS VM from DHCP or statically.
- `flow_eviction_threshold`: Flow Eviction Threshold
- `vm_network_portgroup`: VM Network Port Group Name
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `portgroup_metadata`: Port Group Meta data
- `nova_client_version`: Nova client Version
- `nova_metadata_service_auth_url`: Nova metadata service auth url
- `nova_metadata_service_endpoint`: Nova metadata service endpoint
- `nova_metadata_service_password`: Nova metadata service password
- `nova_metadata_service_tenant`: Nova metadata service tenant
- `nova_metadata_service_username`: Nova metadata service username
- `nova_metadata_shared_secret`: Nova metadata shared secret
- `nova_region_name`: Nova region name
- `primary_nuage_controller`: IP address of the primary Controller (VSC)
- `vrs_password`: VRS password to be used by toolbox to communicate with VRS
- `vrs_user_name`: VRS user name to be used by toolbox to communicate with VRS
- `assoc_vcenter_data_center_id`: The ID of the vcenter to which this host is attached
- `static_route`: static route to be configured in the VRS
- `static_route_gateway`: Gateway for the static route given above
- `static_route_netmask`: Nova region name
- `ntp_server1`: IP of the NTP server 1
- `ntp_server2`: IP of the NTP server 1
- `mtu`: Maximum Transmission Unit for eth2 interface
- `multi_vmssupport`: Whether Multi VM is to be used or not
- `multicast_receive_interface`: Multicast Receive Interface
- `multicast_receive_interface_ip`: IP address for eth3 interface

- `multicast_receive_interface_netmask`: Multicast Interface netmask
- `multicast_receive_range`: Allowed Range to receive the Multicast traffic from
- `multicast_send_interface`: Multicast Send Interface
- `multicast_send_interface_ip`: IP address for eth3 interface
- `multicast_send_interface_netmask`: Multicast Interface netmask
- `multicast_source_portgroup`: Multi Cast Source Port Group Name
- `customized_script_url`: To provide a URL to install a custom app on VRS
- `external_id`: External object ID. Used for integration with third party systems

## 314.2 Children

class	fetcher
<i>nuvcenterhypervisor.NUVCenterHypervisor</i>	<code>vcenter_hypervisors</code>
<i>numetadata.NUMetadata</i>	<code>metadatas</code>
<i>nuglobalmetadata.NUGlobalMetadata</i>	<code>global_metadatas</code>
<i>nuvrsaddressrange.NUVRSAddressRange</i>	<code>vrs_address_ranges</code>

## 314.3 Parents

- *nuvcenterdatacenter.NUVCenterDataCenter*





## NUVCENTERDATACENTER

`nuvcenterdatacenter.NUVCenterDataCenter(bambou.nurest_object.NUMetaRESTObject,):`  
VCenter DataCenters.

### 315.1 Attributes

- `arp_reply`: Whether ARP Reply is enabled/disabled
- `v_require_nuage_metadata`: Whether split-activation or not (Openstack/CloudStack)
- `name` (**Mandatory**): Name of the Datacenter
- `last_updated_by`: ID of the user who last updated the object.
- `data_dns1`: Data DNS 1
- `data_dns2`: Data DNS 2
- `data_gateway`: Data Gateway
- `data_network_portgroup`: Data Network Port Group
- `datapath_sync_timeout`: Datapath Sync Timeout
- `secondary_nuage_controller`: IP address of the secondary Controller (VSC)
- `generic_split_activation`: Split-activation is needed from VRO
- `separate_data_network`: Whether Data will use the management network or not
- `personality`: VRS/VRS-G
- `description`: Description of the Datacenter
- `metadata_server_ip`: Metadata Server IP
- `metadata_server_listen_port`: Metadata Server Listen Port
- `metadata_server_port`: Metadata Server Port
- `metadata_service_enabled`: Metadata Service Enabled
- `network_uplink_interface`: Network Uplink Interface to support PAT/NAT with no tunnels on VRS-VM
- `network_uplink_interface_gateway`: Network Uplink Interface Gateway
- `network_uplink_interface_ip`: Ip Address to support PAT/NAT with no tunnels on VRS-VM
- `network_uplink_interface_netmask`: Network Uplink Interface Netmask

- `nfs_log_server`: IP address of NFS server to send the VRS log
- `nfs_mount_path`: Location to mount the NFS server
- `mgmt_dns1`: DNS server 1
- `mgmt_dns2`: DNS server 2
- `mgmt_gateway`: Gateway for the IP address
- `mgmt_network_portgroup`: Management Network Port group
- `dhcp_relay_server`: To provide IP address of the interface from which you will connect to the DHCP relay server
- `site_id`: Site ID field for object profiles to support VSD Geo-redundancy
- `allow_data_dhcp`: Whether to get the Data IP for the VRS VM from DHCP or statically
- `allow_mgmt_dhcp`: Whether to get the management IP for the VRS VM from DHCP or statically
- `flow_eviction_threshold`: Flow Eviction Threshold
- `vm_network_portgroup`: VM Network Port Group Name
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `portgroup_metadata`: Port Group Meta data
- `nova_client_version`: Nova client Version
- `nova_metadata_service_auth_url`: Nova metadata service auth url
- `nova_metadata_service_endpoint`: Nova metadata service endpoint
- `nova_metadata_service_password`: Nova metadata service password
- `nova_metadata_service_tenant`: Nova metadata service tenant
- `nova_metadata_service_username`: Nova metadata service username
- `nova_metadata_shared_secret`: Nova metadata shared secret
- `nova_region_name`: Nova region name
- `primary_nuage_controller`: IP address of the primary Controller (VSC)
- `vrs_password`: VRS password to be used by toolbox to communicate with VRS
- `vrs_user_name`: VRS user name to be used by toolbox to communicate with VRS
- `associated_vcenter_id`: The ID of the vcenter to which this host is attached
- `static_route`: static route to be configured in the VRS
- `static_route_gateway`: Gateway for the static route given above
- `static_route_netmask`: Nova region name
- `ntp_server1`: IP of the NTP server 1
- `ntp_server2`: IP of the NTP server 1
- `mtu`: Maximum Transmission Unit for eth2 interface
- `multi_vmssupport`: Whether Multi VM is to be used or not
- `multicast_receive_interface`: Multicast Receive Interface
- `multicast_receive_interface_ip`: IP address for eth3 interface

- `multicast_receive_interface_netmask`: Multicast Interface netmask
- `multicast_receive_range`: Allowed Range to receive the Multicast traffic from
- `multicast_send_interface`: Multicast Send Interface
- `multicast_send_interface_ip`: IP address for eth3 interface
- `multicast_send_interface_netmask`: Multicast Interface netmask
- `multicast_source_portgroup`: Multi Cast Source Port Group Name
- `customized_script_url`: To provide a URL to install a custom app on VRS
- `external_id`: External object ID. Used for integration with third party systems

## 315.2 Children

class	fetcher
<i>nuvcentercluster.NUVCenterCluster</i>	<code>vcenter_clusters</code>
<i>numetadata.NUMetadata</i>	<code>metadatas</code>
<i>nuglobalmetadata.NUGlobalMetadata</i>	<code>global_metadatas</code>
<i>nuvrsaddressrange.NUVRSAddressRange</i>	<code>vrs_address_ranges</code>

## 315.3 Parents

- *nuvcenter.NUVCenter*



## NUVCENTEREAMCONFIG

`nuvcentereamconfig.NUVCenterEAMConfig(bambou.nurest_object.NUMetaRESTObject, ) :`

The EAM solution configuration.

### 316.1 Attributes

- `eam_server_ip` (**Mandatory**): The EAM server IP
- `eam_server_port_number` (**Mandatory**): The EAM server port number
- `eam_server_port_type` (**Mandatory**): The EAM server port Type
- `last_updated_by`: ID of the user who last updated the object.
- `vib_url`: The url for the optional vib
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `ovf_url` (**Mandatory**): The url for the ovf
- `extension_key`: Key of the extension that the solution registers
- `external_id`: External object ID. Used for integration with third party systems

### 316.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 316.3 Parents

- `nume.NUMe`



## NUVCENTERHYPERVISOR

`nuvcenterhypervisor.NUVCenterHypervisor(bambou.nurest_object.NUMetaRESTObject,):`  
Host or Hypervisors.

### 317.1 Attributes

- `vcenter_ip`: IP Address of the VCenter.
- `vcenter_password`: Password for VCenter.
- `vcenter_user`: Username for VCenter.
- `arp_reply`: Whether ARP Reply is enabled/disabled
- `v_require_nuage_metadata`: Whether split-activation or not (Openstack/CloudStack)
- `name` (**Mandatory**): Name of the Hypervisor
- `last_updated_by`: ID of the user who last updated the object.
- `data_dns1`: Data DNS 1
- `data_dns2`: Data DNS 2
- `data_gateway`: Data Gateway
- `data_ip_address`: Data IP Address
- `data_netmask`: Data NetMask
- `data_network_portgroup` (**Mandatory**): Data Network Port Group
- `datapath_sync_timeout`: Datapath Sync Timeout
- `secondary_nuage_controller`: IP address of the secondary Controller (VSC)
- `generic_split_activation`: Split-activation is needed from VRO
- `separate_data_network`: Whether Data will use the management network or not
- `personality`: VRS/VRS-G
- `description` (**Mandatory**): Description of the Hypervisor
- `metadata_server_ip`: Metadata Server IP
- `metadata_server_listen_port`: Metadata Server Listen Port
- `metadata_server_port`: Metadata Server Port
- `metadata_service_enabled`: Metadata Service Enabled

- `network_uplink_interface`: Network Uplink Interface to support PAT/NAT with no tunnels on VRS-VM
- `network_uplink_interface_gateway`: Network Uplink Interface Gateway
- `network_uplink_interface_ip`: Ip Address to support PAT/NAT with no tunnels on VRS-VM
- `network_uplink_interface_netmask`: Network Uplink Interface Netmask
- `nfs_log_server`: IP address of NFS server to send the VRS log
- `nfs_mount_path`: Location to mount the NFS server
- `mgmt_dns1`: DNS server 1
- `mgmt_dns2`: DNS server 2
- `mgmt_gateway`: Gateway for the IP address
- `mgmt_ip_address`: The Management IP address for VRS VM if needed to be given statically
- `mgmt_netmask`: Netmask of the IP address above
- `mgmt_network_portgroup` (**Mandatory**): Management Network Port group
- `dhcp_relay_server`: To provide IP address of the interface from which you will connect to the DHCP relay server
- `site_id`: Site ID field for object profiles to support VSD Geo-redundancy
- `allow_data_dhcp`: Whether to get the Data IP for the VRS VM from DHCP or statically
- `allow_mgmt_dhcp`: Whether to get the management IP for the VRS VM from DHCP or statically
- `flow_eviction_threshold`: Flow Eviction Threshold
- `vm_network_portgroup` (**Mandatory**): VM Network Port Group Name
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `toolbox_deployment_mode`: Flag to specify if VRS is deployed using tool box.
- `toolbox_group`: Deployment Toolbox Group.
- `toolbox_ip`: Deployment Toolbox IP.
- `toolbox_password`: Deployment Toolbox password.
- `toolbox_user_name`: Deployment Toolbox username.
- `portgroup_metadata`: Port Group Meta data
- `nova_client_version`: Nova client Version
- `nova_metadata_service_auth_url`: Nova metadata service auth url
- `nova_metadata_service_endpoint`: Nova metadata service endpoint
- `nova_metadata_service_password`: Nova metadata service password
- `nova_metadata_service_tenant`: Nova metadata service tenant
- `nova_metadata_service_username`: Nova metadata service username
- `nova_metadata_shared_secret`: Nova metadata shared secret
- `nova_region_name`: Nova region name
- `primary_nuage_controller`: IP address of the primary Controller (VSC)
- `vrs_id`: VCenter Name or Id used by toolbox to identify the VRS virtual machine



- `vrs_password`: VRS password to be used by toolbox to communicate with VRS
- `vrs_user_name`: VRS user name to be used by toolbox to communicate with VRS
- `associated_cluster_id`: The ID of the cluster to which this host is attached
- `static_route`: static route to be configured in the VRS
- `static_route_gateway`: Gateway for the static route given above
- `static_route_netmask`: Nova region name
- `ntp_server1`: IP of the NTP server 1
- `ntp_server2`: IP of the NTP server 1
- `mtu`: Maximum Transmission Unit for eth2 interface
- `multi_vmssupport`: Whether Multi VM is to be used or not
- `multicast_receive_interface`: Multicast Receive Interface
- `multicast_receive_interface_ip`: IP address for eth3 interface
- `multicast_receive_interface_netmask`: Multicast Interface netmask
- `multicast_receive_range`: Allowed Range to receive the Multicast traffic from
- `multicast_send_interface`: Multicast Send Interface
- `multicast_send_interface_ip`: IP address for eth3 interface
- `multicast_send_interface_netmask`: Multicast Interface netmask
- `multicast_source_portgroup`: Multi Cast Source Port Group Name
- `customized_script_url`: To provide a URL to install a custom app on VRS
- `external_id`: External object ID. Used for integration with third party systems
- `hypervisor_ip` (**Mandatory**): IP Address of the Hypervisor
- `hypervisor_password` (**Mandatory**): Hypervisor username
- `hypervisor_user` (**Mandatory**): Hypervisor username

## 317.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nujob.NUJob</code>	<code>jobs</code>
<code>nuvrsaddressrange.NUVRSAddressRange</code>	<code>vrs_address_ranges</code>

## 317.3 Parents

- `nuvcentercluster.NUVCenterCluster`
- `nume.NUMe`



## NUVCENTERVRSCONFIG

`nuvcentervrsconfig.NUVCenterVRSConfig(bambou.nurest_object.NUMetaRESTObject,):`

Default VRS Configuration parameters

### 318.1 Attributes

- `arp_reply`: Whether ARP Reply is enabled/disabled
- `v_require_nuage_metadata`: Whether split-activation or not (Openstack/CloudStack)
- `last_updated_by`: ID of the user who last updated the object.
- `data_dns1`: Data DNS 1
- `data_dns2`: Data DNS 2
- `data_gateway`: Data Gateway
- `data_network_portgroup`: Data Network Port Group
- `datapath_sync_timeout`: Datapath Sync Timeout
- `secondary_nuage_controller`: IP address of the secondary Controller (VSC)
- `generic_split_activation`: Whether split-activation is needed from VRO
- `separate_data_network`: Whether Data will use the management network or not
- `personality`: VRS/VRS-G
- `metadata_server_ip`: Metadata Server IP
- `metadata_server_listen_port`: Metadata Server Listen Port
- `metadata_server_port`: Metadata Server Port
- `metadata_service_enabled`: Metadata Service Enabled
- `network_uplink_interface`: Network Uplink Interface to support PAT/NAT with no tunnels on VRS-VM
- `network_uplink_interface_gateway`: Network Uplink Interface Gateway
- `network_uplink_interface_ip`: Ip Address to support PAT/NAT with no tunnels on VRS-VM
- `network_uplink_interface_netmask`: Network Uplink Interface Netmask
- `nfs_log_server`: IP address of NFS server to send the VRS log
- `nfs_mount_path`: Location to mount the NFS server

- `mgmt_dns1`: DNS server 1
- `mgmt_dns2`: DNS server 2
- `mgmt_gateway`: Gateway for the IP address
- `mgmt_network_portgroup`: Management Network Port group
- `dhcp_relay_server`: To provide IP address of the interface from which you will connect to the DHCP relay server
- `site_id`: Site ID field for object profiles to support VSD Geo-redundancy
- `allow_data_dhcp`: Whether to get the Data IP for the VRS VM from DHCP or statically
- `allow_mgmt_dhcp`: Whether to get the management IP for the VRS VM from DHCP or statically
- `flow_eviction_threshold`: Flow Eviction Threshold
- `vm_network_portgroup`: VM Network Port Group Name
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `portgroup_metadata`: Port Group Meta data
- `nova_client_version`: Nova client Version
- `nova_metadata_service_auth_url`: Nova metadata service auth url
- `nova_metadata_service_endpoint`: Nova metadata service endpoint
- `nova_metadata_service_password`: Nova metadata service password
- `nova_metadata_service_tenant`: Nova metadata service tenant
- `nova_metadata_service_username`: Nova metadata service username
- `nova_metadata_shared_secret`: Nova metadata shared secret
- `nova_region_name`: Nova region name
- `primary_nuage_controller`: IP address of the primary Controller (VSC)
- `vrs_password`: VRS password to be used by toolbox to communicate with VRS
- `vrs_user_name`: VRS user name to be used by toolbox to communicate with VRS
- `static_route`: static route to be configured in the VRS
- `static_route_gateway`: Gateway for the static route given above
- `static_route_netmask`: Nova region name
- `ntp_server1`: IP of the NTP server 1
- `ntp_server2`: IP of the NTP server 1
- `mtu`: Maximum Transmission Unit for eth2 interface
- `multi_vmssupport`: Whether Multi VM is to be used or not
- `multicast_receive_interface`: Multicast Receive Interface
- `multicast_receive_interface_ip`: IP address for eth3 interface
- `multicast_receive_interface_netmask`: Multicast Interface netmask
- `multicast_receive_range`: Allowed Range to receive the Multicast traffic from
- `multicast_send_interface`: Multicast Send Interface

- `multicast_send_interface_ip`: IP address for eth3 interface
- `multicast_send_interface_netmask`: Multicast Interface netmask
- `multicast_source_portgroup`: Multi Cast Source Port Group Name
- `customized_script_url`: To provide a URL to install a custom app on VRS
- `external_id`: External object ID. Used for integration with third party systems

## 318.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuvrsaddressrange.NUVRSAddressRange</i>	vrs_address_ranges

## 318.3 Parents

- *nume.NUMe*



## NUVIRTUALIP

`nuvirtualip.NUVirtualIP(bambou.nurest_object.NUMetaRESTObject,):`

Virtual IP address.

### 319.1 Attributes

- `mac`: The MAC address of the virtual port
- `last_updated_by`: ID of the user who last updated the object.
- `virtual_ip` (**Mandatory**): Virtual IP address
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `associated_floating_ip_id`: Id of Floating IP address associated to this virtual ip
- `subnet_id`: Id of subnet to which this ip address belongs
- `external_id`: External object ID. Used for integration with third party systems

### 319.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>
<code>nueventlog.NUEventLog</code>	<code>event_logs</code>

### 319.3 Parents

- `nuredirectiontarget.NURedirectionTarget`
- `nuvport.NUVPort`
- `nusubnet.NUSubnet`





## NUVLAN

`nuvlan.NUVLAN(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents VLAN object under a given PORT object.

### 320.1 Attributes

- **value (Mandatory):** value of VLAN
- **last\_updated\_by:** ID of the user who last updated the object.
- **gateway\_id:** The Gateway associated with this VLAN . This is a read only attribute
- **readonly:** Determines whether this entity is read only. Read only objects cannot be modified or deleted.
- **template\_id:** The ID of the template that this Port was created from
- **permitted\_action:** The permitted action to USE/EXTEND this Gateway.
- **description:** A description of the Port
- **restricted:** Determines whether this entity can be used in associations with other properties.
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **vport\_id:** The Vport associated with this VLAN . This is a read only attribute
- **use\_user\_mnemonic:** determines whether to use user mnemonic of the Port
- **user\_mnemonic (Mandatory):** user mnemonic of the Port
- **associated\_egress\_qos\_policy\_id:** ID of the Egress QOS Policy associated with this Vlan.
- **status:** Status of the VLAN.
- **external\_id:** External object ID. Used for integration with third party systems

### 320.2 Children

class	fetcher
<i>nupermision.NUPermission</i>	permissions
<i>numetadata.NUMetadadata</i>	metadatas
<i>nualarm.NUAlarm</i>	alarms
<i>nuglobalmetadadata.NUGlobalMetadadata</i>	global_metadatas
<i>nuenterpriasepermission.NUEnterpriasePermission</i>	enterprise_permissions
<i>nueventlog.NUEventLog</i>	event_logs

## 320.3 Parents

- *nuvsgredundantport.NUVsgRedundantPort*
- *nuredundantport.NURedundantPort*
- *nuport.NUPort*
- *nunsport.NUNSPort*

## NUVLANTEMPLATE

`nuvlantemplate.NUVLANTemplate(bambou.nurest_object.NUMetaRESTObject,)` :

Represents VLAN Template under a Port Template object.

### 321.1 Attributes

- `value`: value of VLAN
- `last_updated_by`: ID of the user who last updated the object.
- `description`: A description of the Port
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `associated_egress_qos_policy_id`: ID of the Egress QOS Policy associated with this Vlan.
- `external_id`: External object ID. Used for integration with third party systems

### 321.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 321.3 Parents

- `nunsporttemplate.NUNSPortTemplate`
- `nuporttemplate.NUPortTemplate`



## NUVM

`nuvm.NUVM(bambou.nurest_object.NUMetaRESTObject, ) :`

Read only API that can retrieve the VMs associated with a domain, zone or subnet for mediation created VM's for REST created VM's you need to set the additional proxy user header in http request : X-Nuage-ProxyUser value of the header has to be either : 1) `enterpriseName@UserName` (example : Alcatel `Lucent@bob`), or 2) external ID of user in VSD, typically is UUID generally decided by the CMS tool in question User needs to have CMS privileges to use proxy user header.

### 322.1 Attributes

- `l2_domain_ids`: Array of IDs of the l2 domain that the VM is connected to
- `vrsid`: Id of the VRS that this VM is attached to.
- `uuid` (**Mandatory**): UUID of the VM
- `name` (**Mandatory**): Name of the VM
- `last_updated_by`: ID of the user who last updated the object.
- `reason_type`: Reason of the event associated with the VM.
- `delete_expiry`: reflects the VM Deletion expiry timer in secs , deleteMode needs to be non-null value for deleteExpiry to be taken in to effect. CMS created VM's will always have deleteMode set to TIMER
- `delete_mode`: reflects the mode of VM Deletion - TIMER Possible values are TIMER, .
- `resync_info`: Information of the status of the resync operation of a VM
- `site_identifier`: This property specifies the site the VM belongs to, for Geo-redundancy.
- `interfaces`: List of VM interfaces associated with the VM
- `enterprise_id`: ID of the enterprise that this VM belongs to
- `enterprise_name`: Name of the enterprise that this VM belongs to
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `domain_ids`: Array of IDs of the domain that the VM is connected to
- `zone_ids`: Array of IDs of the zone that this VM is attached to
- `app_name`: Application name that this VM belongs to
- `orchestration_id`: Orchestration ID.
- `user_id`: ID of the user that created this VM

- `user_name`: Username of the user that created this VM
- `status`: Status of the VM.
- `subnet_ids`: Array of IDs of the subnets that the VM is connected to
- `external_id`: External object ID. Used for integration with third party systems
- `hypervisor_ip`: IP address of the hypervisor that this VM is currently running in

## 322.2 Children

class	fetcher
<i>nuvmresync.NUVMResync</i>	<code>vm_resyncs</code>
<i>numetadata.NUMetadata</i>	<code>metadatas</code>
<i>nualarm.NUAlarm</i>	<code>alarms</code>
<i>nuglobalmetadata.NUGlobalMetadata</i>	<code>global_metadatas</code>
<i>nuvminterface.NUVMInterface</i>	<code>vm_interfaces</code>
<i>nuvrs.NUVRS</i>	<code>vrss</code>
<i>nueventlog.NUEventLog</i>	<code>event_logs</code>

## 322.3 Parents

- *nuqos.NUQOS*
- *nuzone.NUZone*
- *nudomain.NUDomain*
- *nuvport.NUVPort*
- *nusubnet.NUSubnet*
- *nuvrs.NUVRS*
- *nutier.NUTier*
- *nul2domain.NUL2Domain*
- *nume.NUMe*
- *nuegressacltemplate.NUEgressACLTemplate*
- *nuuser.NUUser*
- *nuenterprise.NUEnterprise*
- *nuingressacltemplate.NUIngressACLTemplate*

## NUVMINTERFACE

**`nuvminterface.NUVMInterface(bambou.nurest_object.NUMetaRESTObject, ) :`**

Read only API that can retrieve the VM interface associated with a domain, zone or subnet for mediation created VM's for REST created VM interfaces you need to set the additional proxy header in http request : X-Nuage-ProxyUser value of the header has to be either :1) `enterpriseName@UserName` (example :`bob@Alcatel` Lucent), or 2) external ID of user in VSD, typically is UUID generally decided by the CMS tool in question User needs to have CMS privileges to use proxy user header.

### 323.1 Attributes

- `mac`: MAC address of the interface
- `vmuuid`: UUID of the associated virtual machine
- `ip_address`: IP address of the interface
- `vport_id`: ID of the vport that the interface is attached to
- `vport_name`: Name of the vport that the VM is attached to
- `name`: Device name associated with this interface
- `last_updated_by`: ID of the user who last updated the object.
- `gateway`: Gateway of the subnet that the VM is connected to
- `netmask`: Netmask of the subnet that the VM is attached to
- `network_name`: Name of the network that the VM is attached to
- `tier_id`: ID of the tier that the interface is attached to.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `policy_decision_id`: The policy decision ID for this particular interface
- `domain_id`: ID of the domain that the VM is attached to
- `domain_name`: Name of the domain that the VM is attached to
- `zone_id`: ID of the zone that the interface is attached to
- `zone_name`: Name of the zone that the VM is attached to
- `associated_floating_ip_address`: Floating Ip Address of this network interface eg: 10.1.2.1
- `attached_network_id`: ID of the l2 domain or Subnet that the VM is attached to
- `attached_network_type`: l2 domain or Subnet that the interface is attached to

- `multi_nic_vport_name`: Name of the Multi NIC VPort associated with this VM Interface
- `external_id`: External object ID. Used for integration with third party systems

## 323.2 Children

<b>class</b>	<b>fetcher</b>
<i>nutca.NUTCA</i>	<code>tcas</code>
<i>nuredirectiontarget.NURedirectionTarget</i>	<code>redirection_targets</code>
<i>numetadata.NUMetadata</i>	<code>metadatas</code>
<i>nudhcpoption.NUDHCPOption</i>	<code>dhcp_options</code>
<i>nuglobalmetadata.NUGlobalMetadata</i>	<code>global_metadatas</code>
<i>nupolicydecision.NUPolicyDecision</i>	<code>policy_decisions</code>
<i>nupolicygroup.NUPolicyGroup</i>	<code>policy_groups</code>
<i>nustaticroute.NUStaticRoute</i>	<code>static_routes</code>
<i>nustatistics.NUStatistics</i>	<code>statistics</code>
<i>numulticastchannelmap.NUMultiCastChannelMap</i>	<code>multi_cast_channel_maps</code>
<i>nueventlog.NUEventLog</i>	<code>event_logs</code>

## 323.3 Parents

- *nuzone.NUZone*
- *nudomain.NUDomain*
- *nuvport.NUVPort*
- *nusubnet.NUSubnet*
- *nul2domain.NUL2Domain*
- *nume.NUMe*
- *nuvm.NUVM*



## NUVMRESYNC

`nuvmresync.NUVMResync(bambou.nurest_object.NUMetaRESTObject, ) :`

Provide information about the state of a VM resync request.

### 324.1 Attributes

- `last_request_timestamp`: Time of the last timestamp received
- `last_time_resync_initiated`: Time that the resync was initiated
- `last_updated_by`: ID of the user who last updated the object.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `status`: Status of the resync
- `external_id`: External object ID. Used for integration with third party systems

### 324.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 324.3 Parents

- `nusubnet.NUSubnet`
- `nuvm.NUVM`



## NUVPNCONNECTION

`nuvpnconnection.NUVPNConnection(bambou.nurest_object.NUMetaRESTObject, ) :`

This is the definition of a VPN Connect which holds the PE service association with a DOMAIN.

### 325.1 Attributes

- **name (Mandatory):** Name of the VPNConnect
- **last\_updated\_by:** ID of the user who last updated the object.
- **description:** A description of the VPNConnect
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **associated\_wan\_service\_id:** Associated WAN Service
- **external\_id:** External object ID. Used for integration with third party systems

### 325.2 Children

class	fetcher
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas

### 325.3 Parents

- *nusharednetworkresource.NUSharedNetworkResource*
- *nudomain.NUDomain*
- *nul2domain.NUL2Domain*



## NUVPORT

`nuvport.NUVPort (bambou.nurest_object.NUMetaRESTObject, ) :`

VPorts are a new level in the domain hierarchy, intended to provide more granular configuration than at subnet, and also support a split workflow, where the vPort is configured and associated with a VM port (or gateway port) before the port exists on the hypervisor or gateway.

### 326.1 Attributes

- `vlanid`: associated Vlan of this vport - applicable for type host/bridge
- `name` (**Mandatory**): Name of the vport. Valid characters are alphabets, numbers, space and hyphen( - ).
- `has_attached_interfaces`: Indicates that this vport has attached interfaces
- `last_updated_by`: ID of the user who last updated the object.
- `active`: Indicates if this vport is up or down
- `address_spoofing` (**Mandatory**): Indicates if address spoofing is ENABLED/DISABLED/INHERITED for this vport Possible values are INHERITED, ENABLED, DISABLED, .
- `peer_operational_state`: Operational State of the VPort - RUNNING/SHUTDOWN for peer in mc lag scenarios.
- `description`: Description for this vport
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `domain_id`: ID the Domain associated with the VPort
- `zone_id`: ID the Zone associated with the VPort
- `operational_state`: Operational State of the VPort - RUNNING/SHUTDOWN Possible values are INIT, UP, DOWN, .
- `associated_floating_ip_id`: Id of Floating IP address associated to this vport
- `associated_multicast_channel_map_id`: The ID of the receive Multicast Channel Map this Vport is associated with. This has to be set when enableMultiCast is set to ENABLED
- `associated_send_multicast_channel_map_id`: The ID of the send Multicast Channel Map this Vport is associated with. This has to be set when enableMultiCast is set to ENABLED
- `multi_nic_vport_id`: ID of the Multi NIC VPort associated with the VPort
- `multicast`: Indicates multicast policy on Vport.
- `external_id`: External object ID. Used for integration with third party systems

- **type (Mandatory):** Type of vport - possible values VM/HOST/BRIDGE Possible values are VM, HOST, BRIDGE, .
- **system\_type:** Indicates what system it is.

## 326.2 Children

<b>class</b>	<b>fetcher</b>
<i>nutca.NUTCA</i>	tcas
<i>nuredirectiontarget.NURedirectionTarget</i>	redirection_targets
<i>numetadata.NUMetadata</i>	metadatas
<i>nuaggregatemetadata.NUAggregateMetadata</i>	aggregate_metadatas
<i>nudhcpoption.NUDHCPOption</i>	dhcp_options
<i>nuvirtualip.NUVirtualIP</i>	virtual_ips
<i>nualarm.NUAlarm</i>	alarms
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuvm.NUVM</i>	vms
<i>nuvminterface.NUVMInterface</i>	vm_interfaces
<i>nupolicygroup.NUPolicyGroup</i>	policy_groups
<i>nuqos.NUQOS</i>	qoss
<i>nuhostinterface.NUHostInterface</i>	host_interfaces
<i>nuvportmirror.NUVPortMirror</i>	vport_mirrors
<i>nubridgeinterface.NUBridgeInterface</i>	bridge_interfaces
<i>nuvrs.NUVRS</i>	vrss
<i>nustatistics.NUStatistics</i>	statistics
<i>nustatisticspolicy.NUStatisticsPolicy</i>	statistics_policies
<i>nueventlog.NUEventLog</i>	event_logs

## 326.3 Parents

- *nuzone.NUZone*
- *nuredirectiontarget.NURedirectionTarget*
- *nudomain.NUDomain*
- *nusubnet.NUSubnet*
- *nupolicygroup.NUPolicyGroup*
- *nuvrs.NUVRS*
- *nutier.NUTier*
- *numultinicvport.NUMultiNICVPort*
- *nul2domain.NUL2Domain*
- *nufloatingip.NUFloatingIp*

## NUVPORTMIRROR

`nuvportmirror.NUVPortMirror(bambou.nurest_object.NUMetaRESTObject,):`

VPort Mirror represents the relationship between a vport and a mirror destination.

### 327.1 Attributes

- `vport_name`: Name of the vport to which the mirror destination is associated with.
- `last_updated_by`: ID of the user who last updated the object.
- `network_name`: Name of the network to which the vport belongs to
- `mirror_destination_id`: Destination ID of the mirror destination object.
- `mirror_destination_name`: Name of the mirror destination
- `mirror_direction`: Describes what type of traffic needs to be mirrored.
- `enterprise_name`: Enterprise to which the vport associated with the mirror destination belongs to.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `domain_name`: Domain name of the vport associated with the mirror destination
- `vport_id`: Id of the vport to which the mirror destination is associated with.
- `attached_network_type`: Type of the network attached - L2/L3
- `external_id`: External object ID. Used for integration with third party systems

### 327.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 327.3 Parents

- `nuvport.NUVPort`
- `numirrordestination.NUMirrorDestination`





## **NUVRS**

`nuvrs.NUVRS(bambou.nurest_object.NUMetaRESTObject, ) :`

System Monitoring details for VRS connected to VSC or HSC

### **328.1 Attributes**

- `jsonrpc_connection_state`: The current JSON RPC connection status.
- `name`: Identifies the entity with a name.
- `management_ip`: The management IP of the VRS entity
- `parent_ids`: Holds VRS controllers ids
- `last_event_name`: The last event name from the hypervisor.
- `last_event_object`: The last event object (including metadata) from the hypervisor.
- `last_event_timestamp`: The last event timestamp from the hypervisor.
- `last_state_change`: Last state change timestamp (in millis).
- `last_updated_by`: ID of the user who last updated the object.
- `db_synced`: Flag to indicate if the ovs database is synced between the NSG pair part of a redundant group
- `address`: The IP of the VRS entity
- `peak_cpuusage`: Peek CPU usage percentage.
- `peak_memory_usage`: Peek memory usage percentage.
- `peer`: The redundant peer id for the current VRS.
- `personality`: VRS personality.
- `description`: Description of the entity.
- `messages`: An array of degraded messages.
- `disks`: Set of disk usage details.
- `cluster_node_role`: Indicate that the controller associated is primary, secondary or unknown.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `location`: Identifies the entity to be associated with a location.
- `role`: Flag to indicate that VRS-G redundancy state (active/standby/standalone). Only applicable for gateways.
- `uptime`: How long the VRS was up.

- `product_version`: Product version supported by this entity.
- `is_resilient`: Flag to indicate that the VRS is part of a redundant group.
- `status`: Computed status of the entity.
- `multi_nic_vport_enabled`: VRS is in Multi-NIC VPORT Mode
- `number_of_bridge_interfaces`: Number of bridge interfaces defined in this VRS.
- `number_of_host_interfaces`: Number of host interfaces defined in this VRS.
- `number_of_virtual_machines`: Number of VMs defined in this VRS.
- `current_cpuusage`: Current CPU usage percentage.
- `current_memory_usage`: Current memory usage percentage.
- `average_cpuusage`: Average CPU usage percentage.
- `average_memory_usage`: Average memory usage percentage.
- `external_id`: External object ID. Used for integration with third party systems
- `dynamic`: Flag to indicate it is dynamically configured or not.
- `hypervisor_connection_state`: The VRS connection state with the hypervisor.
- `hypervisor_identifier`: The hypervisor IP (or name) associated with the VRS.
- `hypervisor_name`: The hypervisor name associated with the VRS.
- `hypervisor_type`: The hypervisor type associated with the VRS.

## 328.2 Children

<b>class</b>	<b>fetcher</b>
<i>numetadata.NUMetadata</i>	metadatas
<i>nualarm.NUAlarm</i>	alarms
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuvm.NUVM</i>	vms
<i>nujob.NUJob</i>	jobs
<i>numonitoringport.NUMonitoringPort</i>	monitoring_ports
<i>nuvport.NUVPort</i>	vports
<i>nuhsc.NUHSC</i>	hscs
<i>nuvsc.NUVSC</i>	vscs
<i>numultinicvport.NUMultiNICVPort</i>	multi_nic_vports
<i>nueventlog.NUEventLog</i>	event_logs

## 328.3 Parents

- *nuvport.NUVPort*
- *nuhsc.NUHSC*
- *nuvsc.NUVSC*
- *nuvm.NUVM*

## NUVRSADDRESSRANGE

`nuvrsaddressrange.NUVRSAddressRange(bambou.nurest_object.NUMetaRESTObject, ) :`

This is the definition of a Address Range associated with a VRS.

### 329.1 Attributes

- `last_updated_by`: ID of the user who last updated the object.
- `max_address` (**Mandatory**): Highest address in the address range
- `min_address` (**Mandatory**): Lowest address in the address range
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `external_id`: External object ID. Used for integration with third party systems

### 329.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 329.3 Parents

- `nuvcentervrsconfig.NUVCenterVRSCfg`
- `nuvcenterhypervisor.NUVCenterHypervisor`
- `nuvcenterdatacenter.NUVCenterDataCenter`
- `nuvcentercluster.NUVCenterCluster`
- `nuvcenter.NUVCenter`



## NUVSC

`nuvsc.NUVSC(bambou.nurest_object.NUMetaRESTObject, ) :`

System Monitoring details for VSC.

### 330.1 Attributes

- `name`: Identifies the entity with a name.
- `management_ip`: The management IP of the VSC/HSC entity
- `last_state_change`: Last state change timestamp (in millis).
- `last_updated_by`: ID of the user who last updated the object.
- `address`: The IP of the VRS entity
- `peak_cpuusage`: Peek CPU usage percentage.
- `peak_memory_usage`: Peek memory usage percentage.
- `description`: Description of the entity.
- `messages`: An array of degraded messages.
- `disks`: Set of disk usage details.
- `already_marked_for_unavailable`: Flag to indicate that it is already marked a unavailable.
- `unavailable_timestamp`: The duration the controller is unavailable (in millis).
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `location`: Identifies the entity to be associated with a location.
- `product_version`: Product version supported by this entity.
- `vsds`: A collection of VSD id(s) which are identified by this controller.
- `status`: Computed status of the entity.
- `current_cpuusage`: Current CPU usage percentage.
- `current_memory_usage`: Current memory usage percentage.
- `average_cpuusage`: Average CPU usage percentage.
- `average_memory_usage`: Average memory usage percentage.
- `external_id`: External object ID. Used for integration with third party systems

## 330.2 Children

<b>class</b>	<b>fetcher</b>
<i>numetadata.NUMetadata</i>	metadatas
<i>nubgppeer.NUBGPPeer</i>	bgp_peers
<i>nualarm.NUAlarm</i>	alarms
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nujob.NUJob</i>	jobs
<i>numonitoringport.NUMonitoringPort</i>	monitoring_ports
<i>nuvrs.NUVRS</i>	vrss
<i>nueventlog.NUEventLog</i>	event_logs

## 330.3 Parents

- *nuvsp.NUVSP*
- *nuvrs.NUVRS*

**nuvsd.NUVSD (bambou.nurest\_object.NUMetaRESTObject, ) :**

System Monitoring details for VSD.

### **331.1 Attributes**

- **url:** An optional web url for management.
- **name:** Identifies the entity with a name.
- **management\_ip:** An optional management IP to log into this component.
- **last\_state\_change:** Last state change timestamp (in millis).
- **last\_updated\_by:** ID of the user who last updated the object.
- **address:** An optional IP to access this component.
- **peak\_cpuusage:** Peek CPU usage percentage.
- **peak\_memory\_usage:** Peek memory usage percentage.
- **peer\_addresses:** A comma separated list of peer addresses, if it is in cluster mode.
- **description:** Description of the entity.
- **messages:** An array of degraded messages.
- **disks:** Set of disk usage details.
- **already\_marked\_for\_unavailable:** Flag to indicate that it is already marked a unavailable.
- **unavailable\_timestamp:** The duration the controller is unavailable (in millis).
- **entity\_scope:** Specify if scope of entity is Data center or Enterprise level
- **location:** Identifies the entity to be associated with a location.
- **mode:** Standalone or cluster mode.
- **product\_version:** Product version supported by this entity.
- **status:** Computed status of the entity.
- **current\_cpuusage:** Current CPU usage percentage.
- **current\_memory\_usage:** Current memory usage percentage.
- **average\_cpuusage:** Average CPU usage percentage.
- **average\_memory\_usage:** Average memory usage percentage.

- `external_id`: External object ID. Used for integration with third party systems

## 331.2 Children

<b>class</b>	<b>fetcher</b>
<i>numetadata.NUMetadata</i>	metadatas
<i>nualarm.NUAlarm</i>	alarms
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nujob.NUJob</i>	jobs
<i>nuvsdcomponent.NUVSDComponent</i>	vsd_components
<i>nueventlog.NUEventLog</i>	event_logs

## 331.3 Parents

- *nuvsp.NUVSP*



## NUVSDCOMPONENT

`nuvsdcomponent.NUVSDComponent (bambou.nurest_object.NUMetaRESTObject, ) :`

System Monitoring details for components of VSD system.

### 332.1 Attributes

- `name`: Identifies the entity with a name.
- `management_ip`: An optional management IP to log into this component.
- `address`: An optional IP to access this component.
- `description`: Description of the entity.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `location`: Identifies the entity to be associated with a location.
- `product_version`: Product version supported by this entity.
- `status`: Current status of the entity. Possible values are UP, DOWN, ADMIN\_DOWN, .
- `external_id`: External object ID. Used for integration with third party systems
- `type`: Type of the component.

### 332.2 Children

class	fetcher
<code>numetadata.NUMetadata</code>	<code>metadatas</code>
<code>nuglobalmetadata.NUGlobalMetadata</code>	<code>global_metadatas</code>

### 332.3 Parents

- `nuvsd.NUVSD`



## NUVSGREDUNDANTPORT

`nuvsgredundantport.NUVsgRedundantPort(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents a redundant Port under a particular gateway object or redundant group object.

### 333.1 Attributes

- `vlan_range`: VLAN Range of the Port. Format must conform to a-b,c,d-f where a,b,c,d,f are integers between 0 and 4095.
- `name` (**Mandatory**): Name of the Port
- `last_updated_by`: ID of the user who last updated the object.
- `permitted_action`: The permitted action to USE/EXTEND this Gateway.
- `description`: A description of the Port
- `physical_name` (**Mandatory**): Identifier of the Port
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `port_peer1_id`: The master gateway peer port id.
- `port_peer2_id`: The slave gateway peer port id.
- `port_type` (**Mandatory**): Type of the Port.
- `use_user_mnemonic`: determines whether to use user mnemonic of the Port
- `user_mnemonic`: user mnemonic of the Port
- `associated_egress_qos_policy_id`: ID of the Egress QOS Policy associated with this Vlan.
- `status`: Status of the port.
- `external_id`: External object ID. Used for integration with third party systems

## 333.2 Children

<b>class</b>	<b>fetcher</b>
<i>nupermision.NUPermission</i>	permissions
<i>numetadata.NUMetadata</i>	metadatas
<i>nuvlan.NUVLAN</i>	vlangs
<i>nualarm.NUAlarm</i>	alarms
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuenterprisepermission.NUEnterprisePermission</i>	enterprise_permissions

## 333.3 Parents

- *nuredundancygroup.NURedundancyGroup*
- *nuredundancygroup.NURedundancyGroup*

## NUVSP

`nuvsp.NUVSP (bambou.nurest_object.NUMetaRESTObject, ) :`

System Monitoring details for VSP.

### 334.1 Attributes

- `name`: Name of the VSP
- `last_updated_by`: ID of the user who last updated the object.
- `description`: Description of the VSP
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `location`: Installed location of the VSP product
- `product_version`: Product version number for VSP
- `external_id`: External object ID. Used for integration with third party systems

### 334.2 Children

<b>class</b>	<b>fetcher</b>
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuhsc.NUHSC</i>	hscs
<i>nuvsc.NUVSC</i>	vscs
<i>nuvsd.NUVSD</i>	vsds
<i>nueventlog.NUEventLog</i>	event_logs

### 334.3 Parents

- *nume.NUMe*



## NUWANSERVICE

`nuwanservice.NUWANService(bambou.nurest_object.NUMetaRESTObject, ) :`

Represents a WAN Service Object.

### 335.1 Attributes

- `wan_service_identifier`: Identifier of the WAN Service
- `irb_enabled`: Determines whether Integrated Routing and Bridging is enabled on the WAN Service
- `name` (**Mandatory**): Name of the WAN Service
- `last_updated_by`: ID of the user who last updated the object.
- `permitted_action`: The permitted action to USE/EXTEND this Gateway.
- `service_policy`: Name of 7X50 Policy associated with the service
- `service_type` (**Mandatory**): Type of the service.
- `description`: A description of the WAN Service
- `vn_id`: VNID of the BackHaul Subnet of L3Domain /L2Domain to which this WANService is associated
- `enterprise_name`: The associated enterprise name.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `domain_name`: The associated domain name.
- `config_type`: Type of the CONFIG.
- `orphan`: Indicates if this WAN Service is orphan or not.
- `use_user_mnemonic`: Determines whether to use user mnemonic of the WAN Service
- `user_mnemonic`: user mnemonic of the WAN Service
- `associated_domain_id`: ID of the entity to which the WAN Service is attached to. This could be ID DOMAIN/L2DOMAIN
- `associated_vpn_connect_id`: The associated vpn connect ID.
- `tunnel_type`: Type of the tunnel.
- `external_id`: External object ID. Used for integration with third party systems
- `external_route_target`: Route target associated with the WAN. It is an optional parameter that can be provided by the user

## 335.2 Children

<b>class</b>	<b>fetcher</b>
<i>nupermision.NUPermission</i>	permissions
<i>numetadata.NUMetadata</i>	metadatas
<i>nualarm.NUAlarm</i>	alarms
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuenterpriasepermission.NUEnterprisePermission</i>	enterprise_permissions
<i>nueventlog.NUEventLog</i>	event_logs

## 335.3 Parents

- *nuredundancygroup.NURedundancyGroup*
- *nuautodiscoveredgateway.NUAutoDiscoveredGateway*
- *nugateway.NUGateway*



## NUZONE

`nuzone.NUZone(bambou.nurest_object.NUMetaRESTObject, ) :`

The zone is a collection of subnets attached to a domain. The zone concept enables the definition of policies for collections of subnets.

### 336.1 Attributes

- `ip_type`: IPv4 or IPv6
- `maintenance_mode`: Indicates if the Zone is accepting VM activation requests.
- `name` (**Mandatory**): Name of the current entity (Zone or zone template or subnet etc..) Valid characters are alphabets, numbers, space and hyphen ( - ).
- `last_updated_by`: ID of the user who last updated the object.
- `address`: IP address of the subnet defined. In case of zone, this is an optional field for and allows users to allocate an IP address range to a zone. The VSD will auto-assign IP addresses to subnets from this range if a specific IP address is not defined for the subnet
- `template_id`: The ID of the template that this zone was derived from
- `description`: A description of the zone
- `netmask`: Netmask of the subnet defined
- `encryption`: Determines whether or not IPSEC is enabled.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `policy_group_id`: PG ID for the subnet. This is unique per domain and will be in the range 1-4095
- `associated_application_id`: The associated application ID.
- `associated_application_object_id`: The associated application object ID.
- `associated_application_object_type`: The associated application object type. Refer to API section for supported types.
- `associated_multicast_channel_map_id`: The ID of the Multi Cast Channel Map this zone/zone template is associated with. This has to be set when `enableMultiCast` is set to `ENABLED`
- `public_zone`: If a zone is marked as public, then it is lined to the public network associated with this data center
- `multicast`: Indicates multicast policy on zone.

- `number_of_hosts_in_subnets`: Number of hosts in each of the subnets that can be created under a zone and are auto-assigned IP addresses
- `external_id`: External object ID. Used for integration with third party systems

## 336.2 Children

<b>class</b>	<b>fetcher</b>
<i>nutca.NUTCA</i>	<code>tcas</code>
<i>nupermision.NUPermission</i>	<code>permissions</code>
<i>numetadata.NUMetadata</i>	<code>metadatas</code>
<i>nudhcpoption.NUDHCPOption</i>	<code>dhcp_options</code>
<i>nuglobalmetadata.NUGlobalMetadata</i>	<code>global_metadatas</code>
<i>nuvm.NUVM</i>	<code>vms</code>
<i>nuvminterface.NUVMInterface</i>	<code>vm_interfaces</code>
<i>nuqos.NUQOS</i>	<code>qoss</code>
<i>nuvport.NUVPort</i>	<code>vports</code>
<i>nugroup.NUGroup</i>	<code>groups</code>
<i>nustatistics.NUStatistics</i>	<code>statistics</code>
<i>nustatisticspolicy.NUStatisticsPolicy</i>	<code>statistics_policies</code>
<i>nusubnet.NUSubnet</i>	<code>subnets</code>
<i>nueventlog.NUEventLog</i>	<code>event_logs</code>

## 336.3 Parents

- *nudomain.NUDomain*
- *nume.NUMe*

## NUZONETEMPLATE

`nuzonetemplate.NUZoneTemplate(bambou.nurest_object.NUMetaRESTObject, ) :`

As in domains and subnets, zones are derived from templates. This object provides the definition of the template.

### 337.1 Attributes

- `ip_type`: IPv4 or IPv6(only IPv4 is supported in R1.0) Possible values are IPV4, IPV6, .
- `name` (**Mandatory**): Name of the current entity(Zone or zone template or subnet etc..) Valid characters are alphabets, numbers, space and hyphen( - ).
- `last_updated_by`: ID of the user who last updated the object.
- `address`: IP address of the subnet defined. In case of zone, this is an optional field for and allows users to allocate an IP address range to a zone. The VSD will auto-assign IP addresses to subnets from this range if a specific IP address is not defined for the subnet
- `description`: A description of the Zone template
- `netmask`: Netmask of the subnet defined
- `encryption`: Determines whether or not IPSEC is enabled.
- `entity_scope`: Specify if scope of entity is Data center or Enterprise level
- `associated_multicast_channel_map_id`: The ID of the Multi Cast Channel Map this zone/zone template is associated with. This has to be set when enableMultiCast is set to ENABLED
- `public_zone`: Identifies if the zone is a public zone, in which case any subnets associated with this zone are actually connected to the public subnet of the data center
- `multicast`: Indicates multicast policy on zone template.
- `number_of_hosts_in_subnets`: Number of hosts in the subnets where IP addresses are automatically assigned from the zone IP pool
- `external_id`: External object ID. Used for integration with third party systems

## 337.2 Children

<b>class</b>	<b>fetcher</b>
<i>numetadata.NUMetadata</i>	metadatas
<i>nuglobalmetadata.NUGlobalMetadata</i>	global_metadatas
<i>nuqos.NUQOS</i>	qoss
<i>nusubnettemplate.NUSubnetTemplate</i>	subnet_templates
<i>nueventlog.NUEventLog</i>	event_logs

## 337.3 Parents

- *nudomaintemplate.NUDomainTemplate*