vnc_api Documentation

Release 0.99

Juniper Networks Inc.

Jan 29, 2019

Contents

Contrail configuration is expressed in terms of objects which have the following characteristics

- Each object is identified by a UUID and a fully qualified name (FQ name).
- **Each object has a parent object (except for top level objects e.g. global-system-config, domain, virtual-machine, etc.)**
  - E.g. a domain object is parent of project objects
- **An object can have any number of child objects**
  - E.g. a project object can have one or more virtual-network objects and/or one or more network-policy objects as children
- **An object can refer to other objects (and conversely, can be referred to by other objects). We say that if an object Ob**
  - E.g. a virtual-network object can refer to one or more network-policy objects. A network-policy object can be referred to by one or more virtual-network objects.
- **There can be metadata attached to the reference between two objects.**
  - E.g. a reference from virtual-network object to network-ipam object can have one or more subnets as metadata on the link
- An object can have any number of property elements. These elements can be of simple types (integer, boolean, string) or complex types that contain other data types.
- There are APIs to create/delete/update/read/list these objects. The list API can take various filters. Read and list APIs can also take the list of fields to be returned.
- It is possible to atomically update a specific field in the object without affecting any other fields. Similarly, it is possible to atomically add or delete a reference without affecting anything else.

CHAPTER 1

Juniper Contrail Configuration API Interfaces

The Juniper Contrail configuration API server enables the manipulation of configuration elements exposed by the Contrail API server. Interaction with the API server is possible using the following interfaces.

- REST interface: This interface can be accessed using a command line tool (e.g. cURL) or through a browser with an extension to parse/send JSON data.
- Contrail Python VNC API: This interface internally uses the same REST API, but provides an easy to use interface in a python client. The API is also available in other languages (e.g. java, go, etc).

This document provides:

- Tutorials for using the REST and library interfaces
- General examples to work on different configuration elements
- Tips to use the system effectively
- Reference to the package, module and classes involved

1.1 REST API Tutorial

This tutorial will detail the steps necessary to create couple of virtual-networks and associate a policy on them such that only http traffic can pass.

1.1.1 Authentication

If the API server is setup to use keystone for authentication (common for OpenStack deployments), then users must first obtain a token using keystone command line client or keystone's REST interface. Then, this token should be sent in the header of the request. If authentication is disabled, then this auth token is not needed. For more information on how to obtain a keystone token, check OpenStack documentation(http://developer.openstack.org/api-guide/quick-start/api-quick-start.html)

1.1.2 Create virtual-network and network-policy objects

To create virtual-networks *vn-red* and *vn-blue* and network-policy *policy-red-blue*:

Request for *vn-blue* create

```
curl -X POST -H "X-Auth-Token: $OS_TOKEN" -H "Content-Type: application/json;␣
↪charset=UTF-8" -d '{"virtual-network": {"parent_type": "project", "fq_name": [
↪"default-domain", "admin", "vn-blue"], "network_ipam_refs": [{"attr": {"ipam_subnets
↪": [{"subnet": {"ip_prefix": "10.1.1.0", "ip_prefix_len": 24}}]}, "to": ["default-
↪domain", "default-project", "default-network-ipam"]}]}}' http://10.84.14.2:8082/
↪virtual-networks
```

Response

```
{"virtual-network": {"fq_name": ["default-domain", "admin", "vn-blue"], "parent_uuid
↪": "df7649a6-3e2c-4982-b0c3-4b5038eef587", "parent_href": "http://10.84.14.2:8082/
↪project/df7649a6-3e2c-4982-b0c3-4b5038eef587", "uuid": "8c84ff8a-30ac-4136-99d9-
↪f0d9662f3eee", "href": "http://10.84.14.2:8082/virtual-network/8c84ff8a-30ac-4136-
↪99d9-f0d9662f3eee", "name": "vn-blue"}}
```

Request for *vn-red* create

```
curl -X POST -H "X-Auth-Token: $OS_TOKEN" -H "Content-Type: application/json;␣
↪charset=UTF-8" -d '{"virtual-network": {"parent_type": "project", "fq_name": [
↪"default-domain", "admin", "vn-red"], "network_ipam_refs": [{"attr": {"ipam_subnets
↪": [{"subnet": {"ip_prefix": "20.1.1.0", "ip_prefix_len": 24}}]}, "to": ["default-
↪domain", "default-project", "default-network-ipam"]}]}}' http://10.84.14.2:8082/
↪virtual-networks
```

Response

```
{"virtual-network": {"fq_name": ["default-domain", "admin", "vn-red"], "parent_uuid":
↪"df7649a6-3e2c-4982-b0c3-4b5038eef587", "parent_href": "http://10.84.14.2:8082/
↪project/df7649a6-3e2c-4982-b0c3-4b5038eef587", "uuid": "47a91732-629b-4cbe-9aa5-
↪45ba4d7b0e99", "href": "http://10.84.14.2:8082/virtual-network/47a91732-629b-4cbe-
↪9aa5-45ba4d7b0e99", "name": "vn-red"}}
```

Request for *policy-red-blue* create

```
curl -X POST -H "X-Auth-Token: $OS_TOKEN" -H "Content-Type: application/json;␣
↪charset=UTF-8" -d '{"network-policy": {"parent_type": "project", "fq_name": [
↪"default-domain", "admin", "policy-red-blue"], "network_policy_entries": {"policy_
↪rule": [{"direction": "<>", "protocol": "tcp", "dst_addresses": [{"virtual_network
↪": "default-domain:admin:vn-blue"}], "dst_ports": [{"start_port": 80, "end_port":␣
↪80}], "action_list": {"simple_action": "pass"}, "src_addresses": [{"virtual_network
↪": "default-domain:admin:vn-red"}], "src_ports": [{"end_port": -1, "start_port": -1}
↪]}] }}}' http://10.84.14.2:8082/network-policys
```

Response

```
{"network-policy": {"fq_name": ["default-domain", "admin", "policy-red-blue"],
↪"parent_uuid": "df7649a6-3e2c-4982-b0c3-4b5038eef587", "parent_href": "http://10.84.
↪14.2:8082/project/df7649a6-3e2c-4982-b0c3-4b5038eef587", "uuid": "f215a3ec-5cbd-
↪4310-91f4-7bbca52b27bd", "href": "http://10.84.14.2:8082/network-policy/f215a3ec-
↪5cbd-4310-91f4-7bbca52b27bd", "name": "policy-red-blue"}}
```

1.1.3 Update virtual-networks to use the policy

To associate *policy-red-blue* to *vn-red* and *vn-blue* virtual-networks:

Request for *vn-blue* update

```
curl -X PUT -H "X-Auth-Token: $OS_TOKEN" -H "Content-Type: application/json;␣
↪charset=UTF-8" -d '{"virtual-network": {"fq_name": ["default-domain", "admin", "vn-
↪blue"],"network_policy_refs": [{"to": ["default-domain", "admin", "policy-red-blue
↪"], "attr":{"sequence":{"major":0, "minor": 0}}}]}}' http://10.84.14.2:8082/virtual-
↪network/8c84ff8a-30ac-4136-99d9-f0d9662f3eee
```

Response

```
{"virtual-network": {"href": "http://10.84.14.2:8082/virtual-network/8c84ff8a-30ac-
↪4136-99d9-f0d9662f3eee", "uuid": "8c84ff8a-30ac-4136-99d9-f0d9662f3eee"}}
```

Request for *vn-red* update

```
curl -X PUT -H "X-Auth-Token: $OS_TOKEN" -H "Content-Type: application/json;␣
↪charset=UTF-8" -d '{"virtual-network": {"fq_name": ["default-domain", "admin", "vn-
↪red"],"network_policy_refs": [{"to": ["default-domain", "admin", "policy-red-blue"],
↪ "attr":{"sequence":{"major":0, "minor": 0}}}]}}' http://10.84.14.2:8082/virtual-
↪network/47a91732-629b-4cbe-9aa5-45ba4d7b0e99
```

Response

```
{"virtual-network": {"href": "http://10.84.14.2:8082/virtual-network/47a91732-629b-
↪4cbe-9aa5-45ba4d7b0e99", "uuid": "47a91732-629b-4cbe-9aa5-45ba4d7b0e99"}}
```

1.1.4 Read the objects to verify

Request for *vn-blue* read

```
curl -X GET -H "X-Auth-Token: $OS_TOKEN" -H "Content-Type: application/json;␣
↪charset=UTF-8" http://10.84.14.2:8082/virtual-network/8c84ff8a-30ac-4136-99d9-
↪f0d9662f3eee
```

Response

```
{"virtual-network": {"virtual_network_properties": {"network_id": 4, "vxlan_network_
↪identifier": null, "extend_to_external_routers": null}, "fq_name": ["default-domain
↪", "admin", "vn-blue"], "uuid": "8c84ff8a-30ac-4136-99d9-f0d9662f3eee", "access_
↪control_lists": [{"to": ["default-domain", "admin", "vn-blue", "vn-blue"], "href":
↪"http://10.84.14.2:8082/access-control-list/24b9c337-7be8-4883-a9a0-60197edf64e4",
↪"uuid": "24b9c337-7be8-4883-a9a0-60197edf64e4"}], "network_policy_refs": [{"to": [
↪"default-domain", "admin", "policy-red-blue"], "href": "http://10.84.14.2:8082/
↪network-policy/f215a3ec-5cbd-4310-91f4-7bbca52b27bd", "attr": {"sequence": {"major
↪": 0, "minor": 0}}, "uuid": "f215a3ec-5cbd-4310-91f4-7bbca52b27bd"}], "parent_uuid
↪": "df7649a6-3e2c-4982-b0c3-4b5038eef587", "parent_href": "http://10.84.14.2:8082/
↪project/df7649a6-3e2c-4982-b0c3-4b5038eef587", "parent_type": "project", "href":
↪"http://10.84.14.2:8082/virtual-network/8c84ff8a-30ac-4136-99d9-f0d9662f3eee", "id_
↪perms": {"enable": true, "description": null, "created": "2013-09-13T00:26:05.290644
↪", "uuid": {"uuid_mslong": 10125498831222882614, "uuid_lslong":␣
↪11086156774262128366}, "last_modified": "2013-09-13T00:47:41.219833", "permissions
↪": {"owner": "cloud-admin", "owner_access": 7, "other_access": 7, "group": "cloud-
↪admin-group", "group_access": 7}}, "routing_instances": [{"to": ["default-domain",
↪"admin", "vn-blue", "vn-blue"], "href": "http://10.84.14.2:8082/routing-instance/
↪732567fd-8607-4045-b6c0-ff4109d3e0fb", "uuid": "732567fd-8607-4045-b6c0-ff4109d3e0fb
↪"}], "network_ipam_refs": [{"to": ["default-domain", "default-project", "default-
↪network-ipam"], "href": "http://10.84.14.2:8082/network-ipam/a01b486e-2c3e-47df-
↪811c-440e59417ed8", "attr": {"ipam_subnets": [{"subnet": {"ip_prefix": "10.1.1.0",
↪"ip_prefix_len": 24}, "default_gateway": "10.1.1.254"}]}, "uuid": "a01b486e-2c3e-
↪47df-811c-440e59417ed8"}], "name": "vn-blue"}}
```

(continues on next page)

(continued from previous page)

1.1.5 List the virtual-networks

To list the virtual networks:

Request

```
curl -X GET -H "X-Auth-Token: $OS_TOKEN" -H "Content-Type: application/json;␣
↪charset=UTF-8" http://10.84.14.2:8082/virtual-networks
```

Response

```
{"virtual-networks": [{"href": "http://10.84.14.2:8082/virtual-network/8c84ff8a-30ac-
↪4136-99d9-f0d9662f3eee", "fq_name": ["default-domain", "admin", "vn-blue"], "uuid":
↪"8c84ff8a-30ac-4136-99d9-f0d9662f3eee"}, {"href": "http://10.84.14.2:8082/virtual-
↪network/47a91732-629b-4cbe-9aa5-45ba4d7b0e99", "fq_name": ["default-domain", "admin
↪", "vn-red"], "uuid": "47a91732-629b-4cbe-9aa5-45ba4d7b0e99"}, {"href": "http://10.
↪84.14.2:8082/virtual-network/f423b6c8-deb6-4325-9035-15a8c8bb0a0d", "fq_name": [
↪"default-domain", "default-project", "__link_local__"], "uuid": "f423b6c8-deb6-4325-
↪9035-15a8c8bb0a0d"}, {"href": "http://10.84.14.2:8082/virtual-network/d44a51b0-f2d8-
↪4644-aee0-fe856f970683", "fq_name": ["default-domain", "default-project", "default-
↪virtual-network"], "uuid": "d44a51b0-f2d8-4644-aee0-fe856f970683"}, {"href": "http:/
↪/10.84.14.2:8082/virtual-network/aad9e80a-8638-449f-a484-5d1bfd58065c", "fq_name": [
↪"default-domain", "default-project", "ip-fabric"], "uuid": "aad9e80a-8638-449f-a484-
↪5d1bfd58065c"}]}
```

Request with detail (**WARNING** depending on collection size, this may be inefficient):

```
curl -X GET -H "X-Auth-Token: $OS_TOKEN" -H "Content-Type: application/json;␣
↪charset=UTF-8" http://10.84.14.2:8082/virtual-networks?detail=True
```

Request subset of virtual-networks with detail

```
curl -X GET -H "X-Auth-Token: $OS_TOKEN" -H "Content-Type: application/json;␣
↪charset=UTF-8" "http://10.84.14.2:8082/virtual-networks?obj_uuids=8c84ff8a-30ac-
↪4136-99d9-f0d9662f3eee,47a91732-629b-4cbe-9aa5-45ba4d7b0e99&detail=True"
```

Request virtual-networks anchored by a project(parent):

```
curl -X GET -H "X-Auth-Token: $OS_TOKEN" -H "Content-Type: application/json;␣
↪charset=UTF-8" "http://10.84.14.2:8082/parent_id=83a4bea4-ec45-4670-950c-
↪d7f0f98e0e4f"
```

1.1.6 Delete the objects

To delete the virtual-networks and network-policy objects created:

Request for *vn-red* delete

```
curl -X DELETE -H "X-Auth-Token: $OS_TOKEN" -H "Content-Type: application/json;␣
↪charset=UTF-8" http://10.84.14.2:8082/virtual-network/47a91732-629b-4cbe-9aa5-
↪45ba4d7b0e99
```

Response *None*

1.2 Library API Tutorial

This tutorial will detail the steps necessary to create couple of virtual-networks and associate a policy on them such that only http traffic can pass. The next section generalizes the examples mentioned in this tutorial.

1.2.1 Initialize the library

A single import of the vnc_api module from vnc_api package is sufficient to interact with the configuration API server.

```
>>> from vnc_api import vnc_api
>>> vnc_lib = vnc_api.VncApi(api_server_host='10.84.14.2')
```

1.2.2 Create virtual-network and network-policy objects

To create virtual-networks *vn-red* and *vn-blue* and network-policy *policy-red-blue*

```
>>> vn_blue_obj = vnc_api.VirtualNetwork('vn-blue')
>>> vn_blue_obj.add_network_ipam(vnc_api.NetworkIpam(),
...                              vnc_api.VnSubnetsType([vnc_api.IpamSubnetType(subnet␣
↪= vnc_api.SubnetType('10.1.1.0', 24))]))
>>> vnc_lib.virtual_network_create(vn_blue_obj)
u'57603abb-0089-4a89-b44b-8ca71d4b7826'
```

```
>>> vn_red_obj = vnc_api.VirtualNetwork('vn-red')
>>> vn_red_obj.add_network_ipam(vnc_api.NetworkIpam(),
...                          vnc_api.VnSubnetsType([vnc_api.IpamSubnetType(subnet =␣
↪vnc_api.SubnetType('20.1.1.0', 24))]))
>>> vnc_lib.virtual_network_create(vn_red_obj)
u'5de3af3e-269f-40be-b0f6-69d6bb962a9f'
```

```
>>> policy_obj = vnc_api.NetworkPolicy('policy-red-blue',
...                   network_policy_entries = vnc_api.PolicyEntriesType([vnc_api.
↪PolicyRuleType(direction='<>',
...                       action_list = vnc_api.ActionListType(simple_action='pass'),␣
↪protocol = 'tcp',
...                       src_addresses = [vnc_api.AddressType(virtual_network = vn_
↪blue_obj.get_fq_name_str())],
...                       src_ports = [vnc_api.PortType(-1, -1)],
...                       dst_addresses = [vnc_api.AddressType(virtual_network = vn_
↪red_obj.get_fq_name_str())],
...                       dst_ports = [vnc_api.PortType(80, 80)])]))
>>> vnc_lib.network_policy_create(policy_obj)
u'51388604-c59e-4169-9e0f-39bfcdc603f0'
```

1.2.3 Update virtual-networks to use the policy

To associate the *policy-red-blue* with the virtual networks:

```
>>> vn_blue_obj.add_network_policy(policy_obj, vnc_api.VirtualNetworkPolicyType(
                                                  sequence=vnc_api.SequenceType(0,␣
↪0)))
>>> vn_red_obj.add_network_policy(policy_obj, vnc_api.VirtualNetworkPolicyType(
```

(continues on next page)

(continued from previous page)

```
                                                    sequence=vnc_api.SequenceType(0,␣
↪0)))
>>> vnc_lib.virtual_network_update(vn_blue_obj)
u'{"virtual-network": {"href": "http://10.84.14.2:8082/virtual-network/57603abb-0089-
↪4a89-b44b-8ca71d4b7826", "uuid": "57603abb-0089-4a89-b44b-8ca71d4b7826"}}'
>>> vnc_lib.virtual_network_update(vn_red_obj)
u'{"virtual-network": {"href": "http://10.84.14.2:8082/virtual-network/5de3af3e-269f-
↪40be-b0f6-69d6bb962a9f", "uuid": "5de3af3e-269f-40be-b0f6-69d6bb962a9f"}}'
```

1.2.4 Read the objects to verify

An object can be read by using its uuid returned by create...

```
>>> print vnc_lib.virtual_network_read(id = vn_blue_obj.uuid)
```

... or by its fully-qualified name.

```
>>> print vnc_lib.virtual_network_read(fq_name = ['default-domain', 'default-project',
↪ 'vn-blue'])
```

1.2.5 List the virtual-networks

A summary collection of objects can be listed by

```
>>> print vnc_lib.virtual_networks_list()
```

A detailed collection (**Warning** depending on collection size it might transfer a lot of data) of objects can be listed by

```
>>> print vnc_lib.virtual_networks_list(detail=True)
```

A subset collection of objects can be listed by

```
>>> print vnc_lib.virtual_networks_list(
        obj_uuids=['57603abb-0089-4a89-b44b-8ca71d4b7826',
                   '5de3af3e-269f-40be-b0f6-69d6bb962a9f'],
        detail=True)
```

A subset collection of objects anchored by a parent can be listed by

```
>>> print vnc_lib.virtual_networks_list(parent_id='83a4bea4-ec45-4670-950c-
↪d7f0f98e0e4f')
```

1.2.6 Delete the objects

An object can be deleted using its uuid...

```
>>> vnc_lib.virtual_network_delete(id = '57603abb-0089-4a89-b44b-8ca71d4b7826')
```

... or by its fully-qualified name

```
>>> vnc_lib.virtual_network_delete(fq_name = ['default-domain', 'default-project',
↪'vn-blue'])
```

1.3 REST API Details

The configuration API server provides a means of accessing and manipulating configuration elements of the system using HTTP operations on resources represented in JSON.

The configuration element types (also referred to as resource types) have a hierarchical relationship described in vnc_cfg.xsd schema. JSON representation of these objects are what is expected on the wire.

For each resource type, the following APIs are available:

- Create a resource
- Read a resource given its UUID
- Update a resource
- Delete a resource given its UUID
- List resources of given type

In addition, the following APIs are also available:

- Listing all resource types
- Convert FQ name to UUID
- Convert UUID to FQ name
- Add/Delete/Update a reference between two objects

1.3.1 Creating a resource

To create a resource, a `POST` has to be issued on the collection URL. So for a resource of type *example-resource*,

- *METHOD*: POST
- *URL*: http://<ip>:<port>/example_resources/
- *BODY*: JSON representation of example-resource type
- *RESPONSE*: UUID and href of created resource

Example request

```
curl -X POST -H "X-Auth-Token: $OS_TOKEN" -H "Content-Type: application/json;␣
↪charset=UTF-8" -d '{"virtual-network": {"parent_type": "project", "fq_name": [
↪"default-domain", "admin", "vn-blue"], "network_ipam_refs": [{"attr": {"ipam_subnets
↪": [{"subnet": {"ip_prefix": "10.1.1.0", "ip_prefix_len": 24}}]}, "to": ["default-
↪domain", "default-project", "default-network-ipam"]}]}}' http://10.84.14.2:8082/
↪virtual-networks
```

Response

```
{"virtual-network": {"fq_name": ["default-domain", "admin", "vn-blue"], "parent_uuid
↪": "df7649a6-3e2c-4982-b0c3-4b5038eef587", "parent_href": "http://10.84.14.2:8082/
↪project/df7649a6-3e2c-4982-b0c3-4b5038eef587", "uuid": "8c84ff8a-30ac-4136-99d9-
↪f0d9662f3eee", "href": "http://10.84.14.2:8082/virtual-network/8c84ff8a-30ac-4136-
↪99d9-f0d9662f3eee", "name": "vn-blue"}}
```

1.3.2 Reading a resource

To read a resource, a `GET` has to be issued on the resource URL.

- *METHOD*: GET
- *URL*: http://<ip>:<port>/example_resource/<example-resource-uuid>
- *BODY*: None
- *RESPONSE*: JSON representation of the resource

Example request

```
curl -X GET -H "X-Auth-Token: $OS_TOKEN" -H "Content-Type: application/json;␣
↪charset=UTF-8" http://10.84.14.2:8082/virtual-network/8c84ff8a-30ac-4136-99d9-
↪f0d9662f3eee
```

Response

```
{"virtual-network": {"virtual_network_properties": {"network_id": 4, "vxlan_network_
↪identifier": null, "extend_to_external_routers": null}, "fq_name": ["default-domain
↪", "admin", "vn-blue"], "uuid": "8c84ff8a-30ac-4136-99d9-f0d9662f3eee", "access_
↪control_lists": [{"to": ["default-domain", "admin", "vn-blue", "vn-blue"], "href":
↪"http://10.84.14.2:8082/access-control-list/24b9c337-7be8-4883-a9a0-60197edf64e4",
↪"uuid": "24b9c337-7be8-4883-a9a0-60197edf64e4"}], "network_policy_refs": [{"to": [
↪"default-domain", "admin", "policy-red-blue"], "href": "http://10.84.14.2:8082/
↪network-policy/f215a3ec-5cbd-4310-91f4-7bbca52b27bd", "attr": {"sequence": {"major
↪": 0, "minor": 0}}, "uuid": "f215a3ec-5cbd-4310-91f4-7bbca52b27bd"}], "parent_uuid
↪": "df7649a6-3e2c-4982-b0c3-4b5038eef587", "parent_href": "http://10.84.14.2:8082/
↪project/df7649a6-3e2c-4982-b0c3-4b5038eef587", "parent_type": "project", "href":
↪"http://10.84.14.2:8082/virtual-network/8c84ff8a-30ac-4136-99d9-f0d9662f3eee", "id_
↪perms": {"enable": true, "description": null, "created": "2013-09-13T00:26:05.290644
↪", "uuid": {"uuid_mslong": 10125498831222882614, "uuid_lslong":␣
↪11086156774262128366}, "last_modified": "2013-09-13T00:47:41.219833", "permissions
↪": {"owner": "cloud-admin", "owner_access": 7, "other_access": 7, "group": "cloud-
↪admin-group", "group_access": 7}}, "routing_instances": [{"to": ["default-domain",
↪"admin", "vn-blue", "vn-blue"], "href": "http://10.84.14.2:8082/routing-instance/
↪732567fd-8607-4045-b6c0-ff4109d3e0fb", "uuid": "732567fd-8607-4045-b6c0-ff4109d3e0fb
↪"}], "network_ipam_refs": [{"to": ["default-domain", "default-project", "default-
↪network-ipam"], "href": "http://10.84.14.2:8082/network-ipam/a01b486e-2c3e-47df-
↪811c-440e59417ed8", "attr": {"ipam_subnets": [{"subnet": {"ip_prefix": "10.1.1.0",
↪"ip_prefix_len": 24}, "default_gateway": "10.1.1.254"}]}, "uuid": "a01b486e-2c3e-
↪47df-811c-440e59417ed8"}], "name": "vn-blue"}}
```

1.3.3 Updating a resource

To update a resource, a `PUT` has to be issued on the resource URL.

- *METHOD*: PUT
- *URL*: http://<ip>:<port>/example_resource/<example-resource-uuid>
- *BODY*: JSON representation of resource attributes that are changing
- *RESPONSE*: UUID and href of updated resource

References to other resources are specified as a list of dictionaries with "to" and "attr" keys where "to" is the fully-qualified name of the resource being referred to and "attr" is the data associated with the relation (if any).

Example request

```
curl -X PUT -H "X-Auth-Token: $OS_TOKEN" -H "Content-Type: application/json;␣
↪charset=UTF-8" -d '{"virtual-network": {"fq_name": ["default-domain", "admin", "vn-
↪blue"],"network_policy_refs": [{"to": ["default-domain", "admin", "policy-red-blue
↪"], "attr":{"sequence":{"major":0, "minor": 0}}}]}}' http://10.84.14.2:8082/virtual-
↪network/8c84ff8a-30ac-4136-99d9-f0d9662f3eee
```

Response

```
{"virtual-network": {"href": "http://10.84.14.2:8082/virtual-network/8c84ff8a-30ac-
↪4136-99d9-f0d9662f3eee", "uuid": "8c84ff8a-30ac-4136-99d9-f0d9662f3eee"}}
```

1.3.4 Deleting a resource

To delete a resource, a `DELETE` has to be issued on the resource URL

- *METHOD*: DELETE
- *URL*: http://<ip>:<port>/example_resource/<example-resource-uuid>
- *BODY*: None
- *RESPONSE*: None

Example Request

```
curl -X DELETE -H "X-Auth-Token: $OS_TOKEN" -H "Content-Type: application/json;␣
↪charset=UTF-8" http://10.84.14.2:8082/virtual-network/47a91732-629b-4cbe-9aa5-
↪45ba4d7b0e99
```

Response *None*

1.3.5 Listing Resources

To list a set of resources, a `GET` has to be issued on the collection URL with an optional query parameter mentioning the parent resource that contains this collection. If parent resource is not mentioned, a resource named 'default-<parent-type>' is assumed.

- *METHOD*: GET
- ***URL*: http://<ip>:<port>/example_resources** http://<ip>:<port>/example_resources?parent_id=<parent_uuid> *OR* http://<ip>:<port>/example_resources?parent_fq_name_str=<parent's fully-qualified name delimited by ':'> *OR* http://<ip>:<port>/example_resources?obj_uuids=<example1_uuid>,<example2_uuid>&detail=True *OR* http://<ip>:<port>/example_resources?back_ref_id=<back_ref_uuid> *OR*
- *BODY*: None
- *RESPONSE*: JSON list of UUID and href of collection if detail not specified, else JSON list of collection dicts

Example request

```
curl -X GET -H "X-Auth-Token: $OS_TOKEN" -H "Content-Type: application/json;␣
↪charset=UTF-8" http://10.84.14.2:8082/virtual-networks
```

Response

```
{"virtual-networks": [{"href": "http://10.84.14.2:8082/virtual-network/8c84ff8a-30ac-
↪4136-99d9-f0d9662f3eee", "fq_name": ["default-domain", "admin", "vn-blue"], "uuid":
↪"8c84ff8a-30ac-4136-99d9-f0d9662f3eee"}, {"href": "http://10.84.14.2:8082/virtual-
↪network/47a91732-629b-4cbe-9aa5-45ba4d7b0e99", "fq_name": ["default-domain", "admin
↪", "vn-red"], "uuid": "47a91732-629b-4cbe-9aa5-45ba4d7b0e99"}, {"href": "http://10.
↪84.14.2:8082/virtual-network/f423b6c8-deb6-4325-9035-15a8c8bb0a0d", "fq_name": [
↪"default-domain", "default-project", "__link_local__"], "uuid": "f423b6c8-deb6-4325
↪9035-15a8c8bb0a0d"}, {"href": "http://10.84.14.2:8082/virtual-network/d44a51b0-f2d8-
↪4644-aee0-fe856f970683", "fq_name": ["default-domain", "default-project", "default-
↪virtual-network"], "uuid": "d44a51b0-f2d8-4644-aee0-fe856f970683"}, {"href": "http:/
```

(continues on next page)

(continued from previous page)

1.3.6 Discovering API server resources

The resources managed by the server can be be obtained at the root URL(home-page).

```
curl http://10.84.14.1:8082/ | python -m json.tool
```

Here is a sample output

```
{
  "href": "http://10.84.14.2:8082",
  "links": [
    {
      "link": {
        "href": "http://10.84.14.2:8082/documentation/index.html",
        "name": "documentation",
        "rel": "documentation"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/config-root",
        "name": "config-root",
        "rel": "root"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/domains",
        "name": "domain",
        "rel": "collection"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/service-instances",
        "name": "service-instance",
        "rel": "collection"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/instance-ips",
        "name": "instance-ip",
        "rel": "collection"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/network-policys",
        "name": "network-policy",
        "rel": "collection"
      }
    },
```

(continues on next page)

(continued from previous page)


```
    {
      "link": {
        "href": "http://10.84.14.2:8082/virtual-DNS-records",
        "name": "virtual-DNS-record",
        "rel": "collection"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/route-targets",
        "name": "route-target",
        "rel": "collection"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/floating-ips",
        "name": "floating-ip",
        "rel": "collection"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/floating-ip-pools",
        "name": "floating-ip-pool",
        "rel": "collection"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/bgp-routers",
        "name": "bgp-router",
        "rel": "collection"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/virtual-routers",
        "name": "virtual-router",
        "rel": "collection"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/global-system-configs",
        "name": "global-system-config",
        "rel": "collection"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/namespaces",
        "name": "namespace",
        "rel": "collection"
      }
    },
    {
```


(continues on next page)

(continued from previous page)


```
      "link": {
        "href": "http://10.84.14.2:8082/provider-attachments",
        "name": "provider-attachment",
        "rel": "collection"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/virtual-DNSs",
        "name": "virtual-DNS",
        "rel": "collection"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/customer-attachments",
        "name": "customer-attachment",
        "rel": "collection"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/virtual-machines",
        "name": "virtual-machine",
        "rel": "collection"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/service-templates",
        "name": "service-template",
        "rel": "collection"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/security-groups",
        "name": "security-group",
        "rel": "collection"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/access-control-lists",
        "name": "access-control-list",
        "rel": "collection"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/network-ipams",
        "name": "network-ipam",
        "rel": "collection"
      }
    },
    {
      "link": {
```


(continues on next page)

(continued from previous page)


```
        "href": "http://10.84.14.2:8082/virtual-networks",
        "name": "virtual-network",
        "rel": "collection"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/projects",
        "name": "project",
        "rel": "collection"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/routing-instances",
        "name": "routing-instance",
        "rel": "collection"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/virtual-machine-interfaces",
        "name": "virtual-machine-interface",
        "rel": "collection"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/domain",
        "name": "domain",
        "rel": "resource-base"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/service-instance",
        "name": "service-instance",
        "rel": "resource-base"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/instance-ip",
        "name": "instance-ip",
        "rel": "resource-base"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/network-policy",
        "name": "network-policy",
        "rel": "resource-base"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/virtual-DNS-record",
```


(continues on next page)

(continued from previous page)


```
        "name": "virtual-DNS-record",
        "rel": "resource-base"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/route-target",
        "name": "route-target",
        "rel": "resource-base"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/floating-ip",
        "name": "floating-ip",
        "rel": "resource-base"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/floating-ip-pool",
        "name": "floating-ip-pool",
        "rel": "resource-base"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/bgp-router",
        "name": "bgp-router",
        "rel": "resource-base"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/virtual-router",
        "name": "virtual-router",
        "rel": "resource-base"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/config-root",
        "name": "config-root",
        "rel": "resource-base"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/global-system-config",
        "name": "global-system-config",
        "rel": "resource-base"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/namespace",
        "name": "namespace",
```


(continues on next page)

(continued from previous page)


```
        "rel": "resource-base"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/provider-attachment",
        "name": "provider-attachment",
        "rel": "resource-base"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/virtual-DNS",
        "name": "virtual-DNS",
        "rel": "resource-base"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/customer-attachment",
        "name": "customer-attachment",
        "rel": "resource-base"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/virtual-machine",
        "name": "virtual-machine",
        "rel": "resource-base"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/service-template",
        "name": "service-template",
        "rel": "resource-base"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/security-group",
        "name": "security-group",
        "rel": "resource-base"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/access-control-list",
        "name": "access-control-list",
        "rel": "resource-base"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/network-ipam",
        "name": "network-ipam",
        "rel": "resource-base"
```


(continues on next page)

(continued from previous page)


```
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/virtual-network",
        "name": "virtual-network",
        "rel": "resource-base"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/project",
        "name": "project",
        "rel": "resource-base"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/routing-instance",
        "name": "routing-instance",
        "rel": "resource-base"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/virtual-machine-interface",
        "name": "virtual-machine-interface",
        "rel": "resource-base"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/fqname-to-id",
        "name": "name-to-id",
        "rel": "action"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/id-to-fqname",
        "name": "id-to-name",
        "rel": "action"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/useragent-kv",
        "name": "useragent-keyvalue",
        "rel": "action"
      }
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/virtual-network/%s/ip-alloc",
        "name": "virtual-network-ip-alloc",
        "rel": "action"
      }
```


(continues on next page)

(continued from previous page)

```
    },
    {
      "link": {
        "href": "http://10.84.14.2:8082/virtual-network/%s/ip-free",
        "name": "virtual-network-ip-free",
        "rel": "action"
      }
    }
  ]
}
```

1.3.7 Converting FQ name to UUID

To find the UUID of a resource, given its fq name

```
curl -X POST -H "X-Auth-Token: $OS_TOKEN" -H "Content-Type: application/json;␣
↪charset=UTF-8" -d '{"fq_name": ["default-domain", "admin", "vn-blue"], "type":
↪"virtual-network"}' http://10.84.14.2:8082/fqname-to-id
```

Here is a sample output

```
{"uuid": "e3a20048-8cc7-4cff-8c3b-ada61eb822ed"}
```

1.3.8 Converting UUID to FQ name

To find the type and FQ name of a resource, given its UUID

```
curl -X POST -H "X-Auth-Token: $OS_TOKEN" -H "Content-Type: application/json;␣
↪charset=UTF-8" -d '{"uuid": "e3a20048-8cc7-4cff-8c3b-ada61eb822ed"}' http://10.84.
↪14.2:8082/id-to-fqname
```

Here is a sample output

```
{"type": "virtual-network", "fq_name": ["default-domain", "admin", "vn-blue"]}
```

1.3.9 Adding/Deleting/Updating a reference between two objects

To add/delete/update a reference between two objects, you don't need to read and send the entire object. You can atomically update a single reference by using this API. To add or update a reference:

```
curl -X POST -H "X-Auth-Token: $OS_TOKEN" -H "Content-Type: application/json;␣
↪charset=UTF-8" -d '{"operation": "ADD", "uuid": "e3a20048-8cc7-4cff-8c3b-
↪ada61eb822ed", "type": "virtual-network", "ref-type": "network-policy", "ref-uuid":
↪"7810b656-97d9-4c43-94c7-bd52cc4b055d", "attr": {"sequence": {"major": 0, "minor":␣
↪0}}}' http://10.84.14.2:8082/ref-update
```

Note that instead of the ref-uuid, you can also specify ref-fq-name:

```
curl -X POST -H "X-Auth-Token: $OS_TOKEN" -H "Content-Type: application/json;␣
↪charset=UTF-8" -d '{"operation": "ADD", "uuid": "e3a20048-8cc7-4cff-8c3b-
↪ada61eb822ed", "type": "virtual-network", "ref-type": "network-policy", "ref-fq-name
↪": ["default-domain", "default-project", "default-network-policy"], "attr": {
↪"sequence": {"major": 0, "minor": 0}}}' http://10.84.14.2:8082/ref-update
```

(continues on next page)

(continued from previous page)

To delete a reference:

```
curl -X POST -H "X-Auth-Token: $OS_TOKEN" -H "Content-Type: application/json;␣
↪charset=UTF-8" -d '{"operation": "DELETE", "uuid": "e3a20048-8cc7-4cff-8c3b-
↪ada61eb822ed", "type": "virtual-network", "ref-type": "network-policy", "ref-uuid":
↪"7810b656-97d9-4c43-94c7-bd52cc4b055d"}' http://10.84.14.2:8082/ref-update
```

1.4 Library API Details

1.4.1 Connecting to API server

To make any configuration changes, you first need to connect to API server and get a VNC API handle. This can be done by creating a VncApi object by passing all necessary arguments to its constructor. Example:

```
vnc_api = VncApi(username='admin', password=<password>, tenant_name='admin',
              api_server_host='10.84.10.10', api_server_port='8082',
              api_server_url='/', conf_file=<filename>, user_info=None,
              auth_token=None, auth_host=None, auth_port=None,
              auth_protocol = None, auth_url=None, auth_type=None,
              wait_for_connect=False, api_server_use_ssl=False,
              domain_name=None):
```

1.4.2 Introduction

The configuration API library provides a means of accessing and manipulating configuration elements of the system through an object representation. The library API can be classified into two categories:

- those that manipulate an object locally (client-side)
- those that reflect/get an object's content onto/from the configuration API server.

The configuration element types (also referred to as object types) have a hierarchical relationship described in vnc_cfg.xsd schema. The class definitions of all object types are available at `vnc_cfg_api_server.gen.resource_common` module.

All objects have:

- a fully-qualified name which is an array of strings representing ancestor names from root
- an id-perms property which provide unix file-like permissions for the owner, group and others
- zero or more properties which represent information relevant only to the object
- zero or more references to other objects
- zero or more back references from other objects (computed automatically by the server)
- **methods**
  - to construct an instance

    ```
    >>> example_obj = ExampleType('name', <property_name> = <property_
    ↪value>)
    ```

  - accessors for properties

```
>>> print example_obj.get_<property_name>()
>>> example_obj.set_<property_name>(<property-value>)
```

- accessors for referenced objectes

```
>>> print example_obj.get_<reference_type>_refs()
>>> example_obj.add_<reference_type>(reference_obj)
```

- getters for back-references (i.e. objects referring to this)

```
>>> print example_obj.get_<reference_type>_back_refs()
```

- getters for children objects

```
>>> print example_obj.get_<child_type>s()
```

These methods do not communicate with the API server

The main library class VncApi insert-ref-here has methods for every object type to:

- create an object

```
>>> vnc_lib.example_type_create(example_obj)
```

- read an object

```
>>> example_obj = vnc_lib.example_type_read(id = <example-uuid>)
>>> example_obj = vnc_lib.example_type_read(fq_name = ['example-root', ..
↪. ,'example-parent', 'example-name'])
```

- update an object

```
>>> vnc_lib.example_type_update(example_obj)
```

- delete an object

```
>>> vnc_lib.example_type_delete(id = <example-uuid>)
>>> vnc_lib.example_type_delete(fq_name = ['example-root', ... ,'example-
↪parent', 'example-name'])
```

- list objects (an optional delete=True parameter returns detail instead of summary)

```
>>> vnc_lib.example_types_list(parent_id = <parent-uuid>)
>>> vnc_lib.example_types_list(parent_fq_name = ['example-root', ... ,
↪'example-parent'])
>>> vnc_lib.example_types_list(obj_uuids = [<example-obj1-uuid>,
↪<example-obj2-uuid>])
>>> vnc_lib.example_types_list(back_ref_id = <back-ref-uuid>)
```

These CRUD methods communicate with the API server.

The API server for most objects acts purely as a data store. However properties for some object types are allocated by the API server itself. These include

- default gateway in case of subnet (if not specified by user)
- ip address for instance-ip objects (if not specified by user)

1.4.3 Exceptions

Errors from API server (http status codes and response content) are translated to exception objects and raised. The different types of exceptions are defined at cfgm_common.exceptions module

1.4.4 Tips

Online documentation of the vnc_api module can be found by:

```
>>> help(vnc_api)
Help on module vnc_api.vnc_api in vnc_api:
NAME
    vnc_api.vnc_api
FILE
    /usr/lib/python2.7/site-packages/vnc_api/vnc_api.py
DESCRIPTION
    This is the main module in VNS Config API library. It handles connection to API␣
↪server,
    exposes configuration elements as objects and allows for manipulating objects␣
↪locally
    and updating API server
CLASSES
    vnc_api.gen.vnc_api_client_gen.VncApiClientGen(__builtin__.object)
        VncApi
    class VncApi(vnc_api.gen.vnc_api_client_gen.VncApiClientGen)
     |  Method resolution order:
     |      VncApi
     |      vnc_api.gen.vnc_api_client_gen.VncApiClientGen
     |      __builtin__.object
     |
     |  Methods defined here:
     |
     |  __init__(self, username=None, password=None, tenant_name=None, api_server_
↪host=None, api_server_port=None, api_server_url=None)
     |
```

Online documentation of object methods etc. with standard `dir` and `help` python commands

```
>>> dir(vn_blue_obj)
['__class__', '__delattr__', '__dict__', '__doc__', '__format__', '__getattribute__',
↪'__hash__', '__init__', '__module__', '__new__', '__reduce__', '__reduce_ex__', '__
↪repr__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', '__weakref__',
↪'_type', 'add_network_ipam', 'add_network_policy', 'del_network_ipam', 'del_network_
↪policy', 'dump', 'factory', 'fq_name', 'from_fq_name', 'get_access_control_lists',
↪'get_floating_ip_pools', 'get_fq_name', 'get_fq_name_str', 'get_id_perms', 'get_
↪instance_ip_back_refs', 'get_network_ipam_refs', 'get_network_policy_refs', 'get_
↪parent_fq_name', 'get_parent_fq_name_str', 'get_project_back_refs', 'get_route_
↪target_list', 'get_routing_instances', 'get_type', 'get_virtual_machine_interface_
↪back_refs', 'name', 'network_ipam_refs', 'network_policy_refs', 'parent_name', 'set_
↪id_perms', 'set_network_ipam', 'set_network_ipam_list', 'set_network_policy', 'set_
↪network_policy_list', 'set_route_target_list', 'uuid']
```

```
>>> help(vn_blue_obj)
Help on VirtualNetwork in module vnc_api.gen.resource_common object:
class VirtualNetwork(__builtin__.object)
 |  Represents virtual-network configuration representation.
```

(continues on next page)

(continued from previous page)

```
 |
 |  Child of :class:`.Project` object
 |
 |  Properties of:
 |      * route-target-list (:class:`.RouteTargetList` type)
 |      * id-perms (:class:`.IdPermsType` type)
 |
 |  References to:
 |      * list of (:class:`.NetworkIpam` object, :class:`.VnSubnetsType` attribute)
 |      * list of (:class:`.NetworkPolicy` object, :class:`.VirtualNetworkPolicyType`␣
↪attribute)
 |
 |  Referred by:
 |      * list of :class:`.Project` objects
 |      * list of :class:`.VirtualMachineInterface` objects
 |      * list of :class:`.InstanceIp` objects
 |
```

1.5 Library API reference

1.5.1 `resource_common` Module

1.5.2 `resource_xsd` Module

CHAPTER 2

Indices and tables

- genindex
- modindex
- search