EXPORT and IMPORT the Edge configuration

There is no individual import/export Edge button in the NSX manager. Supported procedure is to export the entire NSX manager configuration and import the same and "redeploy" edges.

However Rest API calls exist to import individual parts of the configuration which can be leveraged by users in their automation environment to export/process/re-create the configuration. This is NOT an officially supported procedure.

API TO EXPORT THE EDGE CONFIGURATION

To retrieve configuration information for a single NSX Edge, use the following API call

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

This retrieves most of the Edge configuration however Certificates, local grouping objects and objects defined on NSX manager global scope or vCenter inventory objects are not exported.

The assumption is that the import will be done on the same NSX manager/vCenter and the referenced inventory objects are still prevalent/intact in the system. This includes both NSX and non-NSX objects like logical-switches/virtual-wires, PortGroups, datastores, datacenter names, cluster-ids, vm-names, application-profiles etc referenced in the Edge-Configuration. The assumption is referenced objects defined/referenced on Global scope will still be accessible as well.

Local scope objects are defined on the edge object with Edge-id as scope-id and will need to be backed up separately and recreated if they need to be re-referenced by a brand new Edge with a different Edge-id.

EXPORTING THE CERTIFICATE

GET https://<nsxmgr-ip>/api/2.0/services/truststore/certificate/scope/<Edge-id>

- HOWEVER that API call doesn't backup the private key + passphrase. So can't be used to re-import the cert later.
- Backup the certificates/key/passphrase. This step is done outside of NSX
- Remove the feature configuration using the certificates in the exported edge config (since the certificate is not installed in the Edge yet, the Edge restoration will fail)

EXPORT GROUPING OBJECTS IF THEY EXIST

The assumption is that any grouping objects referenced can be found by the system. If these objects are defined on global scope, Edge needs to be able to access the same. If they are defined on the local (Edge) scope, you will need to export them and process them for import as they are tied to the Edge.

- IPSets
- Application
- ApplicationGroup
- SecurityGroups

Back up grouping objects, such as like IP sets, custom definitions, and applications.

- To back up IP sets that are associated with an NSX Edge, use the REST API call

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

- To back up custom-defined applications for an NSX Edge, use the REST API call

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

- To back up custom-defined application groups for an NSX Edge, use the REST API call

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

API TO CREATE A NEW EDGE BY IMPORTING THE CONFIG

Once the output from the import API is modified, you can do the below

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

```
<edge>
  <appliances>
  <deployAppliances>false</deployAppliances>
  </appliances>
</edge>
```

This will create a new edge without any features.

The edge-id of this brand new edge can be retrieved by looking at the response header(location) of the POST call or by going to the UI and noting the edge-id of the newly create edge. We need to reference this new edge-id in all subsequent PUT calls to the edge or to the edge's local scope.

PROCESS THE EXPORTED DATA TO PREPARE IT FOR IMPORT

Note: The ones in RED are must fix. Others are optional.

- Change the edge.name (to the new name of the edge which was created without features).Duplicate edge names will error out.
- It would be a good practice to either change the fqdn and provide an unique fqdn or not specify it and let the system default it.
- Remove edge.version. Otherwise, the edge will show up with the version supplied + 1 (No functional impact though)
- For distributedRouter : you must remove the interfaces.interface.index which are not unused.
- From appliances.appliance, consider removing the below
    - hostId (if you want the VC to decide the placement based on resourcePoolId and dataStoreId
    - vmFolderId (Check if you want this to be specified or leave it to VC)
- cliSettings : Add the <password>. And fix the <userName> if required.
- Make sure the logical-switches/Distributed virtual portgroups which are referenced in the interface configuration exist.
- From each feature : Remove the <version> field. Otherwise, the edge will show up the feature with the version supplied + 1 (No functional impact though).

- natRules.natRule.ruleTag : If you want the system to generate fresh ruleIds, delete these. If you want to have user-specified ruleIds make sure they are in the range 65537-131072.
- natRules.natRule,ruleId: ruleId uniquely identifies a rule and should be specified only for rules that are being updated. If it is a new rule which is being posted, delete the ruleId
- firewallRules.firewallRule.ruleTag : If you want the system to generate fresh ruleIds, delete these. If you want to have user-specified ruleIds make sure they are in the range 1-65535.
- If users like, NAT/Firewall config can be pushed in a separate api call. Please refer to the API guide for guidance on how to add NAT/firewall config to the Edge.
-
- bgpNeighbour : Add the password if required
- Ipsec.global
    - Change the PSK
    - Certificates
- If the user is using features like DHCP bindings/Load Balancer Pools, make sure the objectswhich are referenced like VM-name etc exist.
- If Edge features are using groupingObjects like IPSet, Application, SecurityGroup, etc, those need to be exported/processed as well if created on Edge scope.
    - If you are creating a new edge from the exported data, it either needs to have access to the existing groupingObjects (objects are created on global scope with inheritanceAllowed=true).

    - If the grouping objects are on local edge scope, they need to be created by the user on the newly created edge scope

      User can do that by creating an edge to obtain the new EdgeId without any configuration/features.

      Then create the Grouping Objects from the processed/exported data on this newly created edge using this edge's scope-id.

      The new Grouping objects may have different Ids to be queried/processed and added on that edge scope.

    - Then post the Edge configuration with the new processed payload which is using the correct objectIds.

Example:
POST https://<nsxmgr-ip>/api/2.0/services/ipset/<scopeId>

Sample:
https://<nsxmgr-ip>/api/2.0/services/ipset/edge-20

```
<ipset>
        <type>
           <typeName>IPSet</typeName>
        </type>
        <description></description>
        <name>ipset1</name>
  <objectTypeName>IPSet</objectTypeName>
        <value>10.30.10.1</value>
    </ipset>
<ipset>
```

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

Note: Bulk posts will not work in a single REST API call. Multiple iterative REST API calls will be needed to post multiple grouping objects.

IMPORTING THE CERTIFICATE

- Create/Import the Edge without the certificate being used in any features.
    - POST https://<nsxmgr-ip>/api/4.0/edges
- Import the certificate in the Edge scope using the API :
    - POST https://<nsxmgr-ip>/api/2.0/services/truststore/certificate/<edge-id>.
    - Get the <certificate-id> information from the HTTP response.
- Use this certificate-id in the edge features.