ESC Trunks and VLAN Functionality

ESC Trunks and VLAN Functionality

A VM may have one or more interfaces configured to access networks in a domain. For example eth0 is for data network and eth1 is for management networks.

If a VM needs to connect with multiple networks, then simplify the configuration using OpenStack Trunking.

Note Support for trunks and VLANs has been implemented in ESC 5.8.

In an ESC deployment, a VM group defines a trunk for example:

```
<vm_group>
    <name>...</name>
    <image>...</image>
    <flavor>...</flavor>
    <interfaces>
        <interface>
            <nicid>0</nicid>
            <network>parent-net</network>
        </interface>
    </interfaces>
    <trunks>
        <trunk>
            <name>trunk-name</name>
            <parent_nicid>0</parent_nicid>
            <subports>
                <subport>
                    <name>child-port</name>
                    <network>child-net</network>
                    <segmentation_type>vlan</segmentation_type>
                    <segmentation_id>500</segmentation_id>
                </subport>
            </subports>
        </trunk>
    </trunks>
</vm_group>
```

ESC Trunks and VLAN Functionality

A description of elements under `<trunks>`

Element	Mandatory	Description
trunks	y	More than one trunk may be defined under trunks
trunk	y	Wraps the trunk definition
> name	y	Name of the trunk. This will be the name of the trunk in Openstack
> parent_nicid	y	This specifies the NIC ID as defined under <interfaces>, in this example, 0. The port created for this NIC will be used as the parent port for the trunk.
> subports	y	Wrapper for one or more subport elements
>> subport	y	Wraps the subport definition. At least one subport must be defined
>>> name	y	Name of the subport. This will be the name of the subport in Openstack
>>> network	y	Name or ID of the Openstack network to associate with this port. May be an external or ephemeral network defined for the deployment
>>> segmentation_type	n	Defaults to vlan. See Openstack API documentation for other possible values
>>> segmentation_id	y	The VLAN ID to assign to this subport

Creating trunks:

Submit the deployment XML using either REST or Netconf interface. The trunk is created before ESC deploys the VM.

```
<?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2022-06-08T13:39:14.609+00:00</eventTime>
  <escEvent xmlns="http://www.cisco.com/esc/esc">
    <status>SUCCESS</status>
    <status_code>200</status_code>
   <status_message>Trunk trunk-D120-vm1: CREATE_TRUNK completed successfully</status_message>

    <event>
      <type>CREATE_TRUNK</type>
    </event>
  </escEvent>
</notification>
```

A HTTP callback message:

```
::ffff:127.0.0.1 - - [08/Jun/2022 13:39:09] "POST / HTTP/1.1" 200 2
------------------------------
REQUEST_METHOD:          POST
SERVER_PORT:             9009
PATH_INFO:               /
CONTENT_TYPE:            application/json
HTTP_ESC_TRANSACTION_ID  b9cce743-afd8-4eda-9303-5aaeccf4d400
HTTP_ESC_STATUS_MESSAGE *  Trunk trunk-D120-vm1: CREATE_TRUNK completed successfully
HTTP_ESC_STATUS_CODE     200
DATA:
------------------------------
* <JSON config data>
---------------END DATA---------------
```

Day-zero configuration:

A VM requires network configuration to use the trunk, such as the creation of sub-interfaces in the `/etc/network/interfaces` file or running IP link commands for testing. When ESC deploys the VM, the trunk sub-port information is available to day-zero scripts. The following variables are available in addition to the regular template variables. The index is the zero-based on the subport as defined under the trunk.

- SUBPORT_<index>_VLAN_ID
- SUBPORT_<index>_MAC_ADDRESS
- SUBPORT_<index>_ID
- SUBPORT_<index>_NETWORK
- SUBPORT_<index>_NAME
- SUBPORT_<index>_NETWORK_ID
- SUBPORT_<index>_SEGMENTATION_TYPE

Example config data:

```
<config_data>
    <configuration>
        <dst>--user-data</dst>
        <data>
#cloud-config
hostname: D120-vm2
password: secret
chpasswd: { expire: False }
ssh_pwauth: True
runcmd:
  - [ sh, -xc, "ip link add link ens3 name ens3.$SUBPORT_0_VLAN_ID address
$SUBPORT_0_MAC_ADDRESS type vlan id $SUBPORT_0_VLAN_ID" ]
  - [ sh, -xc, "ip link set dev ens3.$SUBPORT_0_VLAN_ID up" ]
  - [ sh, -xc, "dhclient -v ens3.$SUBPORT_0_VLAN_ID" ]
        </data>
    </configuration>
</config_data>
```

Querying trunks:

Using the REST or netconf API the deployment data of the trunk and subports is available.

```
<trunks>
    <trunk>
        <port_id>2f1dc1e6-a90a-4568-8c9e-965fda6c0cfb</port_id>
        <parent_nicid>0</parent_nicid>
        <id>13485eec-c0e0-41d0-b32e-b95ecd23ecef</id>
        <name>trunk-D120-vm1</name>
        <subports>
            <name>child-port-D120-vm1</name>
            <network>child-D120-net</network>
            <mac_address>fa:16:3e:72:cd:8d</mac_address>
            <segmentation_type>vlan</segmentation_type>
            <segmentation_id>120</segmentation_id>
            <id>1b6a0601-3281-4950-baca-f485470f74e3</id>
        </subports>
    </trunk>
</trunks>
```

Deleting the trunk:

Once the VM undeploys, the trunks and subports are deleted. A netconf message is posted for each trunk.

```
<?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2022-06-08T13:24:47.834+00:00</eventTime>
  <escEvent xmlns="http://www.cisco.com/esc/esc">
    <status>SUCCESS</status>
    <status_code>200</status_code>
   <status_message>Trunk trunk-D120-vm2: DELETE_TRUNK completed successfully</status_message>

    <event>
      <type>DELETE_TRUNK</type>
    </event>
  </escEvent>
</notification>
```

A HTTP callback event:

```
::ffff:127.0.0.1 - - [08/Jun/2022 13:24:47] "POST / HTTP/1.1" 200 2
------------------------------
REQUEST_METHOD:          POST
SERVER_PORT:             9009
PATH_INFO:               /
CONTENT_TYPE:            application/xml
HTTP_ESC_TRANSACTION_ID  be42f1d2-a792-4516-91dc-583bdcd28c55
HTTP_ESC_STATUS_MESSAGE *  Trunk trunk-D120-vm2: DELETE_TRUNK completed successfully
HTTP_ESC_STATUS_CODE     200
DATA:
------------------------------
* <?xml version="1.0" encoding="UTF-8"?><!-- trunk details -->
---------------END DATA---------------
```

Modifying the trunk:

The trunk itself is not modified, but the subports can be added or removed and modified.

Submitting an HTTP PUT REST changes the trunk subports as required. Replace the Subports with those in the PUT request:

- Existing subports NOT in the requests are removed from the VM
- Existing subports in the request remain and keep their IDs, MAC address
- Adds new subports in the request.

```
curl -X PUT "http://localhost:8080/ESCManager/v0/deployments/D120" \
    -H "Callback: http://localhost:9009" \
    -H "Callback-ESC-Events: http://localhost:9009" \
    -H "Content-Type: application/xml" \
    -d "<esc_datamodel xmlns=\"http://www.cisco.com/esc/esc\"> ..."
```

Using Netconf `edit-config`

In a Netconf request, the rules are slightly different.

- Ignores the Existing Subports NOT in the request and continues unchanged.
- Adds new subports in the request.
- Removes the subports and annotates with nc:operation='delete' For example:

```
<subport nc:operation='delete'>
    <name>child-port-D120-vm1</name>
    <network>child-D120-net</network>
    <segmentation_type>vlan</segmentation_type>
    <segmentation_id>120</segmentation_id>
</subport>
```

Using `edit-config` (REST API)

Using the internal REST API, submit the Netconf payload

```
curl -X POST --location "http://localhost:8080/ESCManager/internal/conf/edit-config" \
    -H "Callback: http://localhost:9009" \
    -H "Callback-ESC-Events: http://localhost:9009" \
    -H "Content-Type: application/xml" \
    -d "<esc_datamodel
    xmlns=\"http://www.cisco.com/esc/esc\"
    xmlns:nc=\"urn:ietf:params:xml:ns:netconf:base:1.0\">
    ..."
```

Limitations

- Openstack does not allow adding a trunk to a deployed VM. It is possible to attach a secondary interface and use the port as the parent for the trunk
- Does not support scaling of a VM group.
- Does not support specifying a MAC address for the subport.

ESC Trunks and VLAN Functionality