Upgrading Cisco VTS

This chapter provides information about how to upgrade to Cisco VTS 2.6.

Note You can directly upgrade to Cisco VTS 2.6 from Cisco VTS 2.5.2.1 and Cisco VTS 2.5.2. If you are running a version earlier than Cisco VTS2.5.2, you have to upgrade to Cisco VTS 2.5.2, before you upgrade to version 2.6. See *Cisco VTS 2.5.2 Installation Guide* for the procedure to upgrade to Cisco VTS 2.5.2.

Important

You must identify the Service Extension/Device templates that have been impacted due to an NED upgrade, and fix the templates. Also, you must make sure that the templates that have an impact are fixed and migrated to the new NED version prior to starting VTS upgrade. See Migrating Service Extension Templates Before Upgrade, on page 3, for details.

This chapter has the following sections:

Upgrading VTC

Before you upgrade, ensure that:

- Cisco VTS is running version 2.5.2 or 2.5.2.1.
- The admin has taken the backups for Day Zero and Day One configurations for all the switches managed by Cisco VTS.
  See Device documentation for the procedure about how to copy Day Zero configuration locally.

- In an HA set up, HA status is checked on both the VMs. On the Cisco VTS GUI, check HA status under **Administration > High Availability**. Or, you may use the following command:
  ```
  # sudo crm status
  ```
- In an HA setup, both VTCs are online, and one is set as Master and other is set as Slave.
- In an HA setup, *service nso status* of both VTCs is in *active* state.
- In an HA setup, VTS is reachable using the VIP IP address (the IP address used to log in to the Cisco VTS GUI).
- The VTS virtual machines have enough disk-space before starting the upgrade. See Prerequisites chapter for details.
- All the devices in the inventory are reachable and accessible via Cisco VTS. Use the check-sync functionality to make sure all devices are in sync (**Inventory > Network Inventory** GUI).
- For devices that you want to be in *unmanaged* state, you set the devices to *unmanaged* mode:
  ```
  set devices device [device_name] [device_extension]:device-info device-use unmanaged
  commit
  ```
  When a device is specified as *unmanaged*, Cisco VTS will not sync with these devices as part of the upgrade process. Hence, if before upgrade, you use the above command to mark the devices that are not managed by Cisco VTS, then VTS will not sync with these devices and this will not cause a failure during the upgrade.
- Devices are in unlocked state (Check (**Inventory > Network Inventory** GUI).
- You back up the current VTC VMs (Master and Slave) as snapshots which will need to be used to rollback if there is any problem found during the upgrade. See Backing up VTC VMs as Snapshots, on page 5 for details.

Step 1 Setup a writable shared drive, which is accessible to the VTC VM to use a backup drive during the upgrade process. For example:

- On an external server, *create dir extdrive* .
  ```
  root@externalServer:home/admin# mkdir extdrive1
  root@externalServer:home/admin# mkdir extdrive2
  ```
- On an HA setup, on master and slave, mount external location for backup. For example:

```
root@vts1:home/admin# apt-get install sshfs
      root@vts1:home/admin#mkdir extdrive
      root@vts1:home/admin#sshfs root@externalServerIp:/home/admin/extdrive1/ /home/admin/extdrive/

      root@vts2:home/admin# apt-get install sshfs
      root@vts2:home/admin#mkdir extdrive
      root@vts2:home/admin#sshfs root@externalServerIp:/home/admin/extdrive2/ /home/admin/extdrive/
```

Step 2 Mount the upgrade ISO. On an HA setup do this on both Master and Slave:

- Copy the upgrade ISO to any location on VTC VM, for example, to */home/admin*, as root user.
- Mount the VTS upgrade ISO.
  ```
  mount –o loop /home/admin/upgrade.iso /mnt
  ```

Step 3 Run the following command , as root user.

```
show_tech_support -t –a
```

This command backs up log files, including device configuration, and generates a tar file . Copy the tar file outside of the VTC host. This file will be required for troubleshooting purpose. This needs to be done on both VTC nodes in case of an HA setup.

.

Step 4 Run upgrade script. On an HA setup, run the script on both Master and Slave in parallel.

- Change directory to mounted location (*/mnt/*).

  ```
  cd /mnt/upgrades/python
  ```

- Run upgrade script on both Master and Slave, in parallel.

  ```
  nohup python upgrade.py upgrade -ip  <vip_ip> -p <password> -b <backup-dir> &
  ```

  The above command starts upgrade in background. The progress is logged in nohup.log file. Type "tail -f nohup.log " to see the progress of the upgrade.

Note The upgrade script does not do a "sync-to" to the devices unless you use the "-st" flag in the script to explicitly specify that you need the script to do a sync-to operation, as follows:

```
python upgrade.py upgrade -ip <vip-ip> -p <password> -b <backup dir> -st
```

If you have out of band template configuration in Cisco VTS 2.5.2 or 2.5.2.1, follow the procedure detailed in section Preserving Out of Band Template Configuration, on page 6.
If the upgrade fails, you need to perform a rollback. See Performing a Rollback for details.

Step 5 Run the following command , as root user. (Same as Step 3)

```
show_tech_support -t –a
```

This command backs up log files, including device configuration, and generates a tar file . Copy the tar file outside of the VTC host. This file will be required for troubleshooting purpose. This needs to be done on both VTC nodes in case of an HA setup.

Note During the upgrade process, when you do show_tech_support after you run the Upgrade script, L2 High Availability gets broken. If you face this issue, follow the steps listed in the message, and reboot the Master and Slave nodes.

Step 6 Reboot VTC after the upgrade is complete using the reboot command on shell . In case of HA, after the upgrade is complete on both the nodes, reboot each VTC node using the reboot command on shell.

Migrating Service Extension Templates Before Upgrade

This is a pre-upgrade procedure that has to be done before you start the VTS upgrade. You need to identify the impacted Service Extension/Device templates due to NED upgrade and correct them to be migrated to the new NED version.

Step 1 Identify the impacted templates.
Download one of the following *tar* files based on your VTS current version.

Note Some versions of this tool are compatible with only VTS versions from VTS 2.5.2/2.5.2.1 to 2.6.0.

1. After you download the *tar* file, transfer it to the VTC machine that has the customer CDB installed there.
2. Extract the *tar* file via *tar -xvf <filename>*, it will create the directory *vts-launcher*.
3. Go to vts-launcher directory and run *ls -lt* to get the list of files.
4. Identify the file *find-impacted-templates-<version1>-to-<version2>* according to the upgrade path and execute it by logging as *admin* user.
5. If you have impacted templates, you will have a directory created called *templates* that will contain a file per impacted template named *<template name>.impacted.keys*.

   In each file there is a list of key paths that indicate an impacted key in that template.
6. Enter *cli* mode via *ncs_cli -u admin -C*. For each of the templates that was impacted, execute the following CLI to export the template as *json*:

   Note Make sure that the templates directory has the permission set to 777 before running this command.

   ```
   show running-config templates template <template name> | display json | save ./templates/<template
    name>.json
   ```
7. Transfer the *templates* directory to your laptop.

Step 2 Fix the templates. Based on the list of impacted template files, you need to fix each template to be compatible with the target version.
Example for fixing the templates. See for an example.

Note If there is a target VTC VM available, you need to recreate the impacted templates with the same name and export as *json* files by running the following command from the NCS CLI:

```
show running-config templates template <template name> | display json | save ./tmp/<template
 name>.json
```

Step 3 Copy the migrated templates to the VTC machine

1. Create a directory */home/admin/templates-for-migration* in the VTC VM and copy the migrated *json* files to it. During the VTS upgrade, templates in the *json* files will get updated in the VTS database. Make sure that *templates-for-migration* directory with the contents have the correct ownership and permission before starting the upgrade.

   Run the following commands to set the right permission:

   ```
   chown -R admin:vts-admin /home/admin/templates-for-migration
                     chmod -R 777 /home/admin/templates-for-migration
   ```

   Note Only migrated *json* files which need to be imported back to the database should be present in the *templates-for-migration* directory.

Example —Fixing Templates Before Upgrade

Say after executing the script "**./find-impacted-templates-252-260.sh**" there was one file in my templates directory called *sample1.impacted.keys* and I already exported the template using the above show command, hence we have also a file called *sample1.json* with the old content of the template.

Suppose the "sample1.impacted.keys" has the following entry:

```
config/nx:router/ospf{}/area{}/range{}/mask
```

This means that in the upgrade-to version the *mask* attribute was either altered or deleted, now you need to figure out what was the change. To do that we open the target <schema name>.txt under the schemas directory and seek the path up until what was change, in this example we seek the string " config/nx:router/ospf{}/area{}/range".

Backing up VTC VMs as Snapshots

Saving VTC snapshots involves:

- On vCenter—Need to be done for all VTC VMs (Master and Slave):

  1. Power Off the VTC VM (recommended)
  2. Right click on the VTC VM, select **Snapshot**, and then select **Take Snapshot...**.
  3. Enter Name and Description for snapshot and click **Ok**.
  4. Power On the VTC VM.

- On OpenStack—Need to be done for all VTC VMs (Master and Slave):

  1. Shutdown the VTC VMs to take snapshot using virsh save utility. VTC VMs will no longer be available in running state.

     Do virsh list, which shows the VTC domain ID, name, and status. Use Domain ID to save VTC VMs.

     ```
     root@vts-controller-110 ]# virsh list
      Id          Name             State
     ------------------------------------
     236          VTC1             running
     237          VTC2             running
     virsh save <id> <file>
     ```

     For example:

     ```
      virsh save <VTC Domain ID> <file>
     virsh save 236 vtc1.txt
                              virsh save 237 vtc2.txt
     ```

  2. Take vtc.qcow2 image backup which was used to bring up Master and Slave.

     ```
      tar –cvf vtc1.qcow2.tar vtc1.qcow2
             tar –cvf vtc2.qcow2.tar vtc2.qcow2
     ```

  3. Copy tar images to external drive (vtc1.qcow2.tar ,vtc2.qcow2.tar are VTC snapshots, which will be used to rollback).
  4. Restore VTC VMs which will bring VTC VMs back to running state.

     ```
     virsh restore vtc1.txt
              virsh restore vtc2.txt
     ```

  5. Verify if Master and Slave are up and running in HA mode. Verify GUI login using VIP IP.

Preserving Out of Band Template Configuration

If you have out of band template configuration in Cisco VTS 2.5.2 or 2.5.2.1 and want to upgrade to 2.6, do the following to ensure that the out of band template configuration is preserved after you upgrade to 2.6 without any interruption to the data plane.

Step 1 Upgrade to VTS 2.6 without doing a sync-to.

```
cd /mnt/upgrades/python
python upgrade.py upgrade -ip <vip-ip> -p <password> -b <backup dir>
```

Step 2 Run sync-to dry-run.

```
cd /mnt/upgrades/python/scripts
       ./sync_to_dry_run.script
```

Step 3 Check /opt/vts/run/upgrade/ folder with files having non-zero size.

Step 4 If there are files with non-zero size, then Southbound lock all the devices.

```
cd /mnt/upgrades/python/scripts
       ./southbound_lock_managed_devices.script
```

Step 5 Create templates that contain the out of band configuration and apply the templates. Configuration with - sign will be removed from device configuration. Configuration with + sign will be added to device configuration.

Step 6 Unlock all the devices.

```
cd /mnt/upgrades/python/scripts
      ./unlock_managed_devices.script
```

Step 7 Do a sync-to to all the devices.

```
cd /mnt/upgrades/python/scripts
      ./synch_to.script
```

Upgrading VTSR

To upgrade VTSR VM, do the following:

Step 1 Generate new VTSR ISO before upgrading to new VTSR.

Step 2 Delete the existing VTSR VM and bring up the new VM using the new image. See Installing VTSR for details.

Step 3 After the VTSR VM comes up, do a sync-to operation in order to sync the configuration from the VTC.

Upgrading VTF

VTF has to be uninstalled and installed after the VTC upgrade.

See Installing VTF on OpenStack and Installing VTF on vCenterfor details about VTF installation and uninstallation.

Note In case of upgrade, user cannot use the GUI to install VTF again after uninstallation, if there are any workloads (ports) attached.

Upgrading J-Driver

To upgrade J-Driver, do the following:

Step 1 Stop neutron service.
```
#systemctl stop neutron-server
```

Step 2 Drop the Journaling table:

- Connect to MySQLL
- Find what is the correct DB.
- Select DB displayed.
- Use <db_name> (db_name is the name displayed in the table as the output from the previous command)
- Drop the Journaling tables
```
drop table ciscocontroller_maintenance;
> drop table ciscocontrollerjournal;
```

Step 3 Upgrade the RPM— either manually or using Anisble.

Step 4 Start neutron service again.
```
#systemctl stop neutron-server
```

Post Upgrade Considerations

This section certain important points you need to consider after you upgrade to Cisco VTS 2.6

- After upgrade, run `chown -R nso:vts-log /opt/vts/log/nso` once on the VTS Slave. This is required so that the ssh user has access to nso logs.
- Upgrade from Cisco VTS 2.5.2 to Cisco VTS 2.6—Impact on SRIOV ports:

  OpenStack behavior for SRIOV ports is similar to that of OVS ports in that SRIOV ports, by default, get associated with tenant's default Security Group.

  When SRIOV ports get migrated from Cisco VTS 2.5.2 to Cisco VTS 2.6, Cisco VTS removes any Security Groups associated with them as they do not serve any purpose anyways.

  You must edit these SRIOV ports and associate them with either 'no security groups' or with a security group that does not use 'remote-sg'.

If above action is not performed, any subsequent SRIOV ports updates from OpenStack would get rejected as Cisco VTS does not allow SRIOV ports to get associated with Security Groups containing remote-sg.

- After upgrade from Cisco VTS 2.5.2 to Cisco VTS 2.6.0, fabric static routes (for an overlay Router) which are designated for selective devices will not be device specific anymore. They will be applied to all network devices that have the overlay networks associated to the Router. As such, after upgrade, many devices will be going out-of-sync because of this and you have to decide if you want these static routes on those devices.
- After upgrade you need to go to Administration > Virtual Machine Manager page, and edit each OpenStack VMM and edit each Neutron Server and save. This is required to update J-Driver plugin.
- After upgrade you need to go to Inventory > Host Inventory page, and edit each host with OVS virtual switch type to trigger reinstallation of Host Agent.

Important See Upgrade Behavior for Security Groups, on page 8 for important information related to Security Groups upgrade behavior.

Upgrade Behavior for Security Groups

Eventhough Security Groups were unsupported in versions earlier than Cisco VTS 2.6, the Operator could still use OpenStack to create Security Groups and associate them with VMs. Cisco VTS, in earlier releases, would never register any Security Group event, and hence no Security Group event would flow into the VTS database. But when a Port is associated with an Security Group, OpenStack embeds the entire payload of the corresponding Security Group into Port payload. In version 2.5.2, Cisco VTS used to consume this payload as-is and store in its database. In a way, Cisco VTS was aware of the Security Group contents even in version 2.5.2. But this content would not reflect the latest updated version of the respective Security Group because Cisco VTS did not register for OpenStack Security Group events.

Upgrade from Cisco VTS version 2.5.2 to 2.6, fetches Security Group contents that were already present in its Port database, and creates corresponding Security Groups and Rules in its database. Upgrade process triggers a 'redeploy' of Security Group services post data migration. This may cause out-of-sync VTSRs if there are any VTF eligible Security Groups. If the deployment only has 'default' Security Groups with remote-sg rule then none of these Security Groups are considered eligible; hence none get pushed to VTSR.

We recommend that you update the security groups from the OpenStack backend CLI to trigger Security Group updates from OpenStack Controller towards VTC, post upgrade.

To update default Security Groups's description field:

```
[root@overcloud-controller-0 ~]# source overcloudrc
[root@overcloud-controller-0 ~]# openstack
(openstack) security group list
(openstack) security group set --description "update trigger"
038f2c29-7c66-4119-a4fc-62c826e08223
```

To update the default Security Group's description field: for OSPD Setup execute CLI from OSPD Director.

```
[stack@ospd-director ~]$ source overcloudrc
[stack@ospd-director ~]$ openstack
(openstack) security group set <default-sg-id> --project-id <project-id> --description
"Updated default SG description"
(openstack)
```

Doing this ensures that the VTC is consistent with OpenStack's Security Groups.

Migrating Ports from Cisco VTS 2.5.2 to Cisco VTS 2.6

All OVS ports get migrated to Cisco VTS 2.6, as-is. Non OVS Ports-SRIOV, Baremetal, and VTF Ports on the other hand go through a scrubbing process where the Security Group list of each of these ports is set to empty. This scrubbing process is required to ensure that no underlying traffic flows are affected due to premature application of Security Groups that come from version 2.5.2. It also helps to avoid pushing not-yet-fully vetted security configuration towards ToRs, thus avoiding traffic disruptions due to upgrade.

Post migration, the Operator is expected to associate each of these non OVS Ports with Security Groups that reflect Operator's vetted security intent.

	Ports in 2.5.2 (Before Upgrade)	Ports in 2.6 (After Upgrade)	Ports in 2.6 (Operator-created new SGs Specifically for non-OVS Ports)

Openstack OVS Ports	Port P1 {
	Network: ...
	...
	Security-Groups {
	default {
	Ingress Rules,
	Egress Rules
	}
	custom-sg1 {
	Ingress Rules,
	Egress Rules
	}
	}
	}
		Port P1 {
		Network: ...
		...
		Security-Groups {
		default,
		custom-sg1
		}
		}
		Note In Cisco VTS 2.6, ports contain just the Security Group references.
			Port P1 {
			Network: ...
			...
			Security-Groups {
			default,
			custom-sg1
			}
			}
			Note No change is required for OVS Ports as the security for these ports is already fully realized by OpenStack.

Non OVS Ports (SRIOV, BM, VTF)	Port P1 {
	Network: ...
	...
	Security-Groups {
	default {
	Ingress Rules,
	Egress Rules
	}
	custom-sg1 {
	Ingress Rules,
	Egress Rules
	}
	}
	}
		Port P1 {
		Network: ...
		...
		Security-Groups { }
		}
		Note All Security Groups are cleaned up from the Port because realizing not-so-fully vetted Security Groups can result in traffic disruption.	

	Ports in 2.5.2 (Before Upgrade)	Ports in 2.6 (After Upgrade)	Ports in 2.6 (Operator-created new SGs Specifically for non-OVS Ports)

			Port P1 {
			Network: ...
			...
			Security-Groups {
			default,
			custom-default-with-no-remote-sg,
			custom-sg2-no-remote-sg
			}
			}
			Note In this scenario, the Operator created two new Security Groups—One corresponding to each Security Group that was earlier associated with this port:
			• custom-default-with-no-remote-sg—Operator created this Security Group to reflect most of 'default' Security Group intent by replacing remote-sg rules with corresponding remote-ip-prefix based rules.
			• custom-sg2-with-no-remote-sg —Operator created this Security Group to reflect most of 'custom-sg1' Security Group intent by replacing remote-sg rules with corresponding remote-ip-prefix based rules.
			Also, the Operator has reassociated P1 with 'default' fully knowing that this Security Group intent will not be realized. This is because it allows OVS Ports to continue to identify non-OVS Ports through this Security Group association. Thus OVS Ports can continue to communicate with non-OVS Ports without any explicit changes to their Security Groups (default and custom-sg1, in this scenario).

Changes To OpenStack Settings Through Upgrade

Security Groups in OpenStack can be realized by either of the below firewall drivers:

- OVSHybridIptablesFirewallDriver

- Openvswitch

Openvswitch does not support OVS trunk ports in Newton release.

We recommend that for firewall settings you use OVSHybridIptablesFirewallDriver. When Cisco VTS is upgraded to version 2.6, following settings automatically take effect if the Operator is deploying VTS Plugin via OpenStackPlatform director. If not, settings in column OpenStack Settings Post 2.6 Upgrade that enable Security have to be manually done.

	OpenStack Setting before Upgrade to 2.6	**OpenStack Settings Post 2.6 Upgrade that enable Security Groups**

Scenario-1:
OS Deployment has Security Groups Enabled
	Controllers: /etc/neutron/plugin.ini
	`enable_security_group = true`
	`firewall_driver = openvswitch`
	Note The above two settings were being ignored by VTS Plugin before to 2.6; So the value of these setting do not matter. Regardless of above settings, VM Vif type is always set to plug tap interfaces directly into integration bridge.
	Computes: openvswitch_agent.ini
	`firewall_driver = openvswitch`
	Note The above setting on the compute is essential to enable firewall on VM tap interfaces that plug directly into integration bridge.
		Controllers: /etc/neutron/plugin.ini
		`firewall_driver = neutron.agent.linux.iptables_firewall.OVSHybridIptablesFirewallDriver`
		`enable_security_group = true`
		Note VTS Plugin automatically sets the VM Vif type to reflect above 2 settings. In the above case, since HybridIptablesFirewall is being used, the VM's vif will connect via Linux Bridges to OVS Integration bridge on the compute. Above settings take effect for new VMs only. VMs spun up in 2.5.2 continue to plug directly into integration bridge. For Security Groups to take effect for existing VMs, set the below setting on the computes to use Openvswitch firewall driver.
		Computes: openvswitch_agent.ini
		Settings on the compute do not matter as long as Controller has the above settings.

Scenario-2:
OS Deployment has Security Groups Disabled
	Controllers: /etc/neutron/plugin.ini
	`#enable_security_group = true`
	(The above setting was being ignored by VTS Plugin before to 2.6; So the value of this setting does not really matter)
	Computes: openvswitch_agent.ini
	`#firewall_driver =`
		Same as above.

Performing a Rollback

The following sections describes the procedure to roll back to the Cisco VTS version from which you upgraded.

- Performing a Rollback on vCenter

Performing a Rollback on OpenStack

Do the following to rollback to the Cisco VTS version from which you upgraded. This should be done for all the VTC VMs (Master and Slave).

Step 1 On the controller, do *virsh list*.

```
root@vts-controller-110 ]# virsh list
 Id          Name             State
-------------------------------------
236          VTC1             running
237          VTC2             running
```

Step 2 Virsh destroy already existing VTC VMs (Master and Slave).

```
virsh destroy <id>
```

Step 3 Copy *vtc1.qcow2.tar* and *vtc2.qcow2.tar* from external drive to the controller.

Step 4 Untar *vtc1.qcow2.tar* and *vtc2.qcow2.tar*

```
untar –xvf vtc1.qcow2.tar
        untar –xvf vtc2.qcow2.tar
```

Step 5 Create Master and Slave VTC (virsh create utility) using *vtc.xml* file which points to the location of *qcow* images that is untarred in the above step.

Note Create the Master VTC first, wait for two to three minutes, and then create the Slave VTC.

Step 6 Verify if Master and Slave are up and running in HA mode. Verify GUI log in using VIP IP.

Note Make sure that the *service nso status* of both VTCs is in *active* state.

In case nso status is in *inactive* state then kill and recreate that VTC. Then reverify if Master and Slave are up and running in HA mode. Verify GUI log in using VIP IP. Also, make sure that service nso status of both VTC is currently in *active* state.

Step 7 Manually reregister the VMM and Host Agent from VTS GUI.

Performing a Rollback on vCenter

Do the following to rollback to the Cisco VTS version from which you upgraded. This should be done for all the VTC VMs (Master and Slave).

Step 1 Power Off the VTC VM (recommended)

Step 2 Right click on VTC VM and select **Snapshot**, and then **Snapshot Manager...**.

Step 3 Select **Snapshot** and click **Go to**. Click **Close** to close the screen.

Step 4 Power On the VTC VM.

Step 5 Verify if HA is up and running. Verify GUI log in using VIP IP.

Step 6 Manually reregister VMM from VTS GUI.