

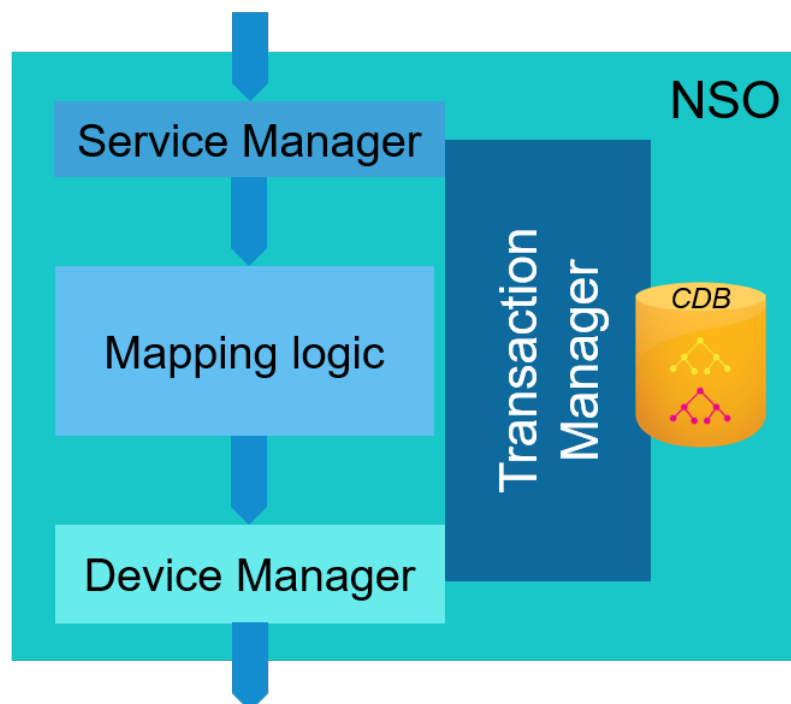
Service Management Tasks

You can use the Cisco NSO CLI or the WebUI to fulfill the service management operations.

Service management consists of many distinct types of actions. In the previous course, you learned many of the tasks when you modeled a service and applied it to the network. Several other tasks are available for performing service management, such as checking the synchronization status, deleting a service, and undeploying a service. The following figure shows how service provisioning includes various components of NSO.

The service management actions consist of the following:

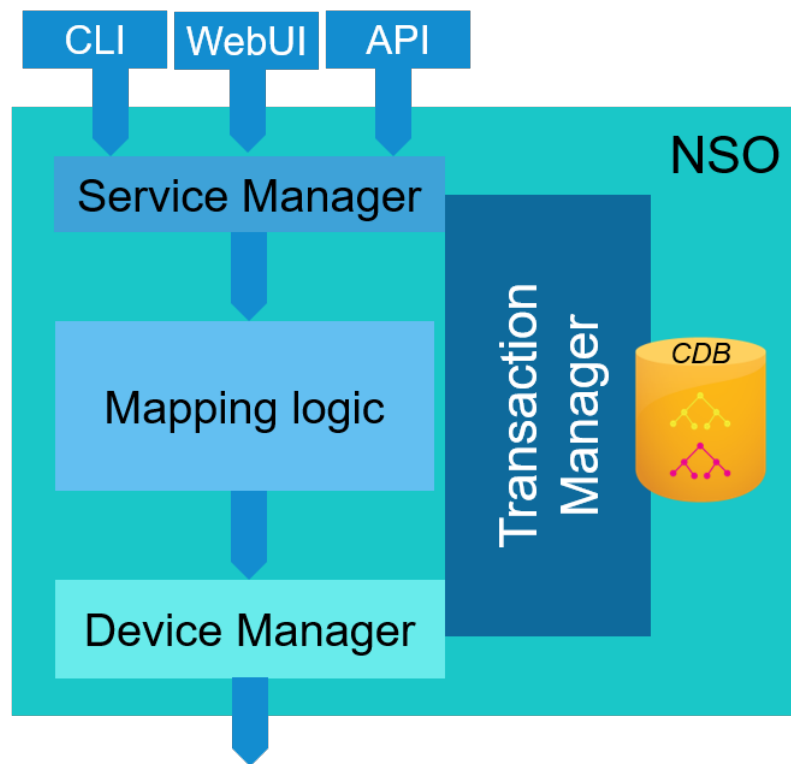
- Service modeling
- Mapping the service to a device model
- Device "effects"
- Service activation
- Service dry run
- Service check-sync
- Service restoration
- Service testing
- Aggregated operational data
- Notifications



Service Management Options

You have seen that there are two major types of CLI in Cisco NSO to accommodate most network operators who are familiar with the Cisco CLI, Juniper CLI, or both. Additionally, you can manage services through the WebUI that is automatically augmented when you add a service, or you can even further augment it by creating other objects or integrating your own WebUI.

Cisco NSO is also very extensible and can be integrated with other systems when you use a number of the application APIs mentioned earlier. Because all of them are modeled on YANG, no additional coding is required when a new service is added. As shown in the following figure, you can use APIs for interacting with the Service Manager. In addition, it is possible to access device configurations directly and invoke actions on devices by using the Device Manager through those same APIs.

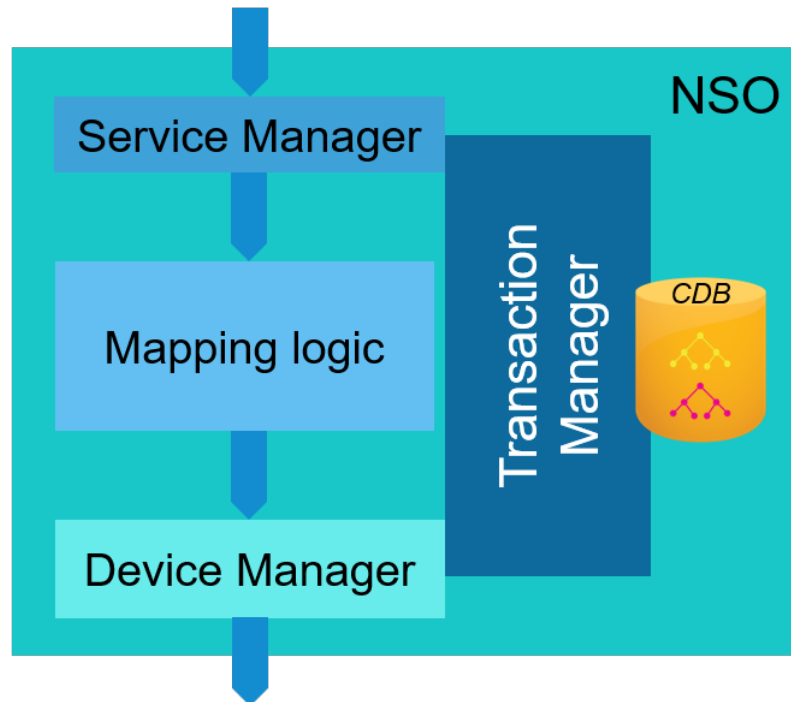


NSO Components and Service Management

The main components of the service management are the same as the main components of the system:

- **Services** are handled by the service manager.
- **Devices** are handled by the device manager.

Additionally, there are many other components that help with the manageability of services (customers, templates, alarms, and other services). As shown in the figure, there is mapping logic connecting service configurations with device configurations. All these tasks are overseen by the transaction manager to ensure an all-or-nothing approach and eliminate the need for manually removing a configuration in the case of a failed commit.

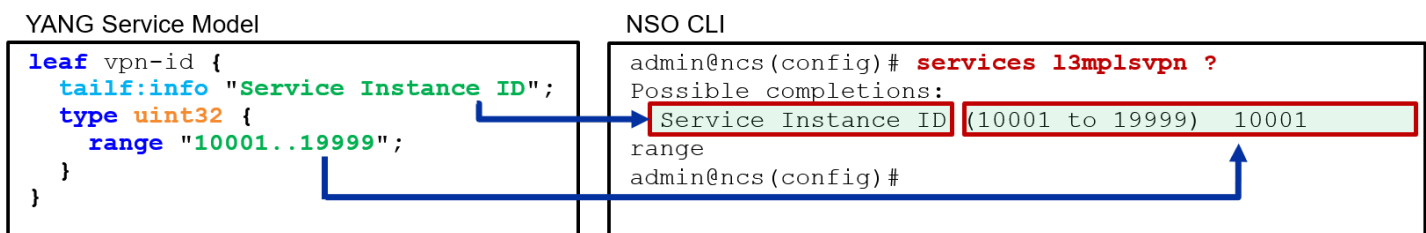


NSO CLI and WebUI for Services

Whenever you design a new service, you will see the service name and parameters become available in the CLI and WebUI. You can also augment the way that the WebUI and CLI provide this information by using many available annotations. For example, you can use a *tailf:info* statement to provide information to users applying the service using the NSO CLI. The following example shows the *tailf:info* statement being used to assist users with what the acceptable "Service Instance ID" range is.

The service management tool has the following characteristics:

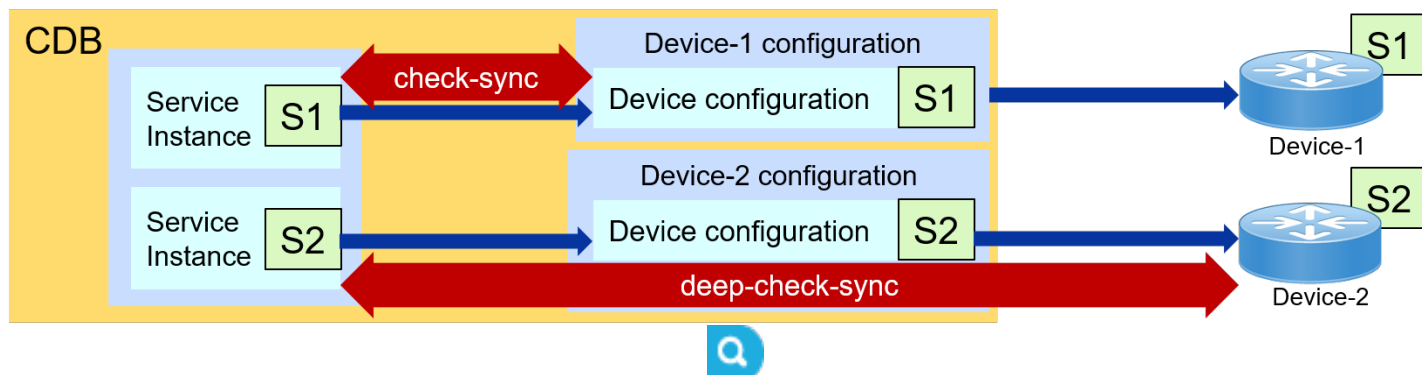
- No extra development is required to provide a CLI or WebUI for services.
- Both the CLI and WebUI options are dynamically created based on service packages.
- YANG annotations can be used to influence the CLI and WebUI (for example, *tailf:info* to provide syntax help).



Service Configuration Maintenance Tools

Service instance configuration can reside in three locations:

- **Device:** The active configuration is on the managed device.
- **Device manager:** The device-specific representation of the configuration is stored in the Cisco NSO configuration database (CDB).
- **Service manager:** The service-model-specific representation of the configuration is stored in the Cisco NSO CDB.



All three locations should be in sync. Out-of-band changes can cause these configurations to become out of synchronization.

To perform an internal sync check between a service configuration and a device-specific configuration because of the service-to-device mapping logic, use the following command:

```
admin@ncs services l3mplsvpn S1 check-sync
```

To extend the sync check to the actual device configuration, use the following command:

```
admin@ncs services l3mplsvpn S2 deep-check-sync
```

This command verifies that the actual devices are configured according to the service. Note that there are out-format options for viewing the diff. Reconcile the diff by using the service redeploy.

Edit an existing service instance or create a new one:

```
admin@ncs(config)# services l3mplsvpn 10002 link 3 interface ...
```

Check service synchronization status to mapped device configuration (in CDB):

```
admin@ncs(config)# services l3mplsvpn 10002 check-sync
in-sync true
```

Check service synchronization status to actual device configuration:

```
admin@ncs(config)# services l3mplsvpn 10001 deep-check-sync
in-sync true
```

Redeploy the service instance:

```
admin@ncs(config)# services l3mplsvpn 10001 re-deploy
```

Create devices config from service instance;
send minimum set of changes to devices

Undeploy the service instance:

```
admin@ncs(config)# services l3mplsvpn 10001 un-deploy
```

- Remove devices configuration from devices and CDB
- Keep service instance in CDB

Delete the service instance:

```
admin@ncs(config)# no services l3mplsvpn 10002
```

Remove service instance from CDB and
devices configuration from devices and CDB

The following example illustrates how the previous example can be accomplished using the Juniper style of CLI:

```
admin@ncs% set services l3mplsvpn 10002 link 3 interface ...  
[ok] [2014-11-09 11:39:24]
```

```
admin@ncs% request services l3mplsvpn 10002 check-sync  
in-sync true  
[ok] [2014-11-09 11:40:01]
```

```
admin@ncs% request services l3mplsvpn 10001 re-deploy  
No modifications to commit.  
[ok] [2014-11-09 11:40:14]
```

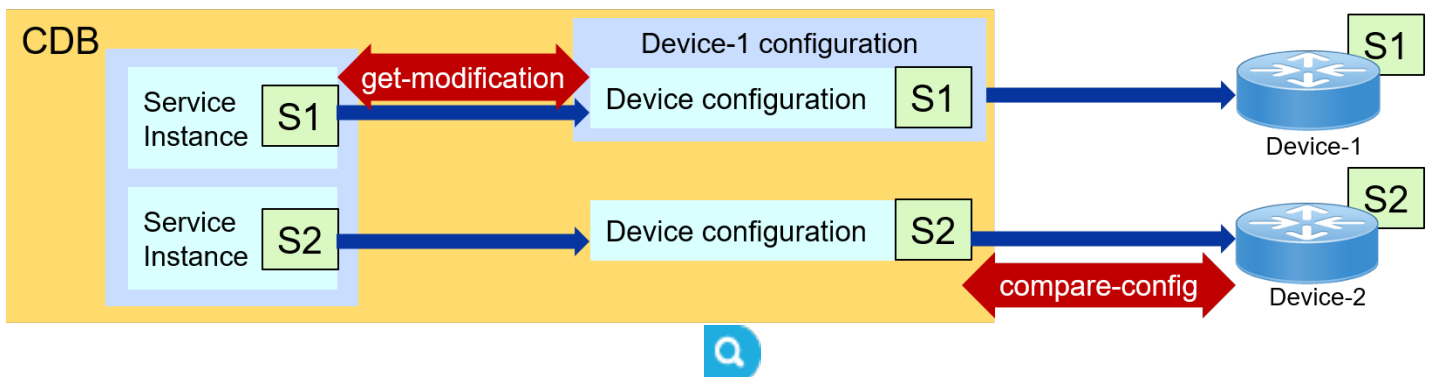
```
admin@ncs% request services l3mplsvpn 10001 un-deploy  
[ok] [2014-11-09 11:40:49]
```

```
admin@ncs% delete services l3mplsvpn 10002  
[ok] [2014-11-09 11:39:13]
```

Comparing Configurations—Service or Device

The following figure illustrates two configuration comparison options:

- Display the device configuration that is created based on a service instance.
- Display the differences between the CDB version of the device configuration and the actual device configuration.



Get device modifications made by a service instance:

```
admin@ncs services l3mplsvpn S1 get-modifications
```

Get discrepancies between the CDB version of a device configuration and the actual device configuration:

```
admin@ncs devices device PE11 compare-config
```

© 2024 Cisco and/or its affiliates. All rights reserved. Printed contents of 00ugmr503zzl5gXtHN5d7