CHAPTER 6

Service Assurance Administration

The Cisco vMS 2.0 Service Assurance Administration provides information on the Cisco Virtual Managed Services 2.0 Service Assurance component.

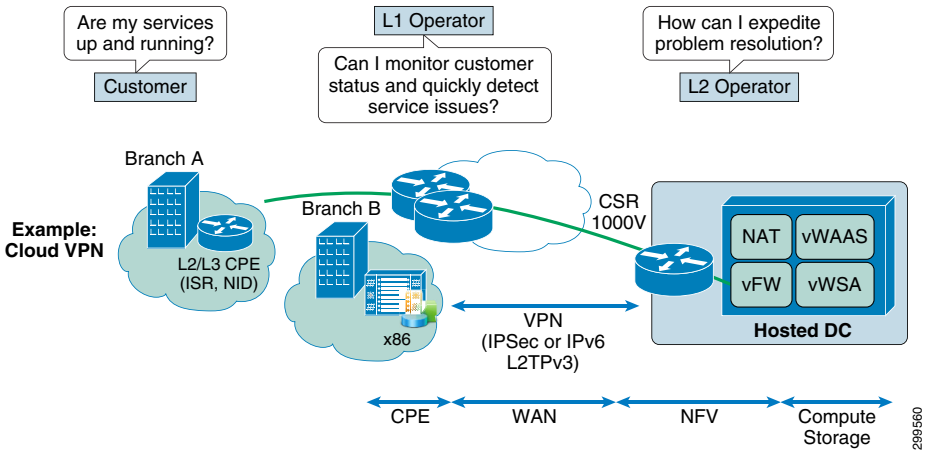
Cisco vMS 2.0 Service Assurance (vMS SA) provides operational insights to both end customers and operations teams to enable delivery of superior service. It provides real-time visibility into service availability and usage attributes to track service status and help operations teams isolate service issues. It empowers operations teams with situation awareness and fault analytics to expedite network problem detection and resolution.

Cisco vMS SA assures customers that their services are up and running and allows network operators to monitor customer status, detect services issues, and facilitate problem resolutions, as defined in Table 6-1 and shown in Figure 6-1.

Table 6-1 Network Operator Level Tasks and Objectives

Operator Level	Task	Objective
L1	View status (Up/Down) of services and overlay components.	Verify if service is up or down and proceed with problem isolation.
L1 and L2	View service component performance (metrics).	Isolate the service component as the cause of the disruption.
L2	View anomalous situations and faults that could be service impacting.	Analyze the issue, pin-point the cause and resolve as a collaborative effort.

Figure 6-1 Service Assurance Customer Administration



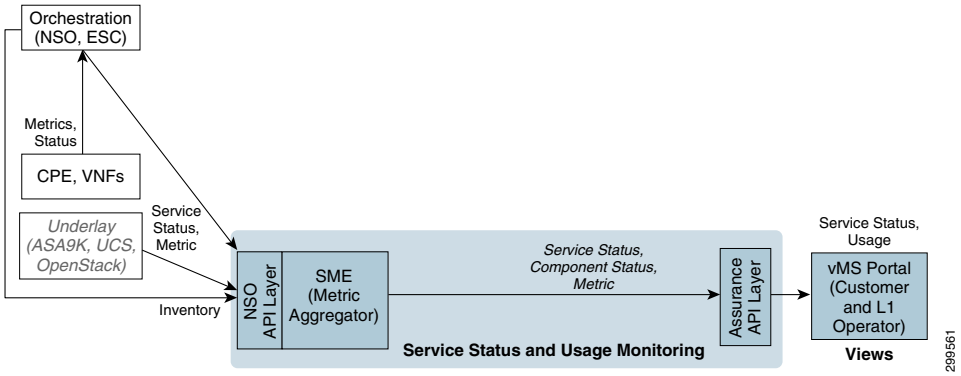
Cisco VMS SA empowers customers by providing real-time service visibility, which results in improved customer experience through rapid detection of service issues. Cisco VMS SA assures service delivery and enables a consistent and reliable user experience. Cisco VMS SA underlying principles include:

- **Cross-Domain and Multi-Vendor**—Provides End-to-end visibility across multiple domains, that is, from customer premise and access device to WAN, NFV, and cloud.
- **Multi-Layer**—Provides correlated views across services and virtual and physical infrastructure layers.
- **Orchestration Integration**—Provisions service assurance with service creation.
- **Self-Healing**—Provides policy-based automation that links visibility and analytics to control and optimization.
- **Out-of-Box Content**—Provides pre-defined content for specific use cases such as cloud VPN.
- **External Integration**—Provides open APIs for OSS integration.

Cisco vMS SA Architecture

The Cisco vMS SA architecture is shown in Figure 6-2.

Figure 6-2 VMS Service Assurance Architecture



Cisco vMS SA assurance capabilities include:

- Service and service components status
- Service usage monitoring

Cisco vMS SA architecture layers include:

- **Managed System Instrumentation**—Monitors the vMS system and services. Each managed component exposes performance, status, and fault data. This data is exposed through syslogs, log files, SNMP, Network Configuration Protocol (NETCONF), APIs, CLI, and other interfaces. The exposed interfaces and data depend on the managed component type. However, vMS generally uses SNMP, NETCONF, and syslog instrumentation.
- **Data Collection and Aggregation**—Collects data metrics using the managed system instrumentation and aggregates it by data type, allowing operators to set thresholds on KPIs to generate alarms. Configuration of desired set of metric and threshold is configured at the service design time and provided in form of templates that get enabled at service on-boarding time. Cisco

vMS SA also collects and aggregates metrics from overlay components such as the Cisco Integrated Services Routers, Cisco Cloud Services Routers, Cisco Adaptive Security Virtual Appliances, and other devices.

- **API**—The API layer provides a single integration point for the vMS Portal. It exposes faults, metrics, logs, and other data related to of tenants and services. Only metrics are exposed. The API layer also obtains inventory data from the orchestrator to provision assurance.

In addition to the data collection, data distribution, analysis, and API layers, Cisco vMS SA also uses the vMS Portal to provide the presentation layer.

Cisco VMS Service Assurance Installation

Cisco VMS Service Assurance is installed during the VMS installation. Refer to Chapter 3, "ESC NCS Installer". Table 6-2 shows the general Cisco VMS SA requirements.

Table 6-2 Cisco VMS Service Assurance Requirements

Pod Size	SME	NSO Interface	SA API
40 Services	1 VM 4 vCPU Cores 8 GB RAM 8 GB Swap 1 x 146 GB SAS Harddrive (OS + SME) 1 x 300 GB SAS harddrive (Backups)	1 VM 4 vCPU Cores 8 GB RAM 8 GB Swap 50 GB HDD	1 VM 4 vCPU Cores 8 GB RAM 8 GB Swap 50 GB HDD
400 Services	2 VMs 6 vCPU Cores 24 GB RAM 12 GB Swap 1 x 3000GB SAS 15K RPM Drive (OS+SME) 1 x 300GB SAS 15K RPM Drive (DB) 1 x 600 GB SAS 10K RPM Drive (Backups)		
1000 Services	2 VM 8 vCPU Cores 64 GB RAM 32 GB Swap 2 x 3000GB SAS 15K RPM Drive (OS+SME) 3 x 300GB SAS 15K RPM Drive (DB) 3 x 300GB SAS 10K RPM Drive (Backups) RAID0		

Verifying the VMS SA Installation

The following procedures verify the installation of Cisco VMS SA component applications.

Verifying SME Installation

Perform the following procedure to verify the installation of SME.

Step 1 Enable SSL.

```
/opt/CSCOppm-gw/bin/ppm ssl enable noprompt
```

Step 2 Update the PAL runtime configuration.

```
/opt/CSCOppm-gw/etc/palRuntime/conf/DeviceCapabilities.xml
<entry name="CLI_NSO_NETCONF_PORT">830</entry>
<entry name="CLI_NSO_NETCONF_PATH">/opt/ncs/ncs-3.4.2.1/bin/netconf-console</entry>
<entry name="CLI_NSO_NETCONF_PORT">830</entry>
<entry name="CLI_NSO_NETCONF_PATH">/opt/ncs/ncs-3.4.2.1/bin/netconf-console</entry>
```

Step 3 Enable the alarm log for logstash.

```
cp /opt/CSCOppm-gw/etc/alarmLogConfig.xml.sample
/opt/CSCOppm-gw/etc/alarmLogConfig.xml
```

Step 4 Enable reports.

```
sme statreps none
sme statreps setstatus "SME System: Server Metrics" enable
sme statreps setstatus "SME System: Data Metrics" enable
sme statreps setstatus "SME System: Poller Metrics" enable
```

Step 5 Restart SME.

Step 6 Discover the NSO device.

```
/opt/CSCOppm-gw/bin/ppm addsnmpcomm -i <nso-ip-address> -c <snmp-community> -P
<snmp-port>
/opt/CSCOppm-gw/bin/ppm addcreds -i <nso-ip-address> -r <connection-protocol> -u
<ssh-username> -p <ssh-password> -n <netconf-username> -e <netconf-password>
/opt/CSCOppm-gw/bin/ppm discover <nso-ip-address>
/opt/CSCOppm-gw/bin/ppm addsnmpcomm -i 128.107.1.37 -c public -P 4000
/opt/CSCOppm-gw/bin/ppm addcreds -i 128.107.1.37 -r SSH_V2 -u root -p Admin-123 -n
admin -e admin
/opt/CSCOppm-gw/bin/ppm discover 128.107.1.37
```

Step 7 Add SNMP credentials and SSH v2 credentials (root username and password for the host machine as primary, NETCONF console username and password as secondary) for the NSO and then discover that device. Note: this assumes that the path to the netconf-console executable and the console port match the CLI_NSO_NETCONF_PATH and CLI_NSO_NETCONF_PORT values in the deployed SME etc/palRuntime/conf/DeviceCapability.xml.

Commands to enable SSL and discover the NSO device.

```
/opt/CSCOppm-gw/bin/ppm ssl enable
/opt/CSCOppm-gw/bin/ppm addsnmpcomm -i <nso-ip-address> -c <snmp-community> -P
<snmp-port>
/opt/CSCOppm-gw/bin/ppm addcreds -i <nso-ip-address> -r <connection-protocol> -u <ssh-username> -p <ssh-password> -n
<netconf-username> -e <netconf-password>
/opt/CSCOppm-gw/bin/ppm discover <nso-ip-address>
/opt/CSCOppm-gw/bin/ppm addsnmpcomm -i 128.107.1.37 -c public -P 4000
/opt/CSCOppm-gw/bin/ppm addcreds -i 128.107.1.37 -r SSH_V2 -u root -p Admin-123 -n
admin -e admin
/opt/CSCOppm-gw/bin/ppm discover 128.107.1.37
```

Verifying the NSO Interface Installation

Perform the following procedure to verify the NSO interface installation.

Step 1 Configure the config file:

```
<install-dir>/CSCOnso-shim/bin/etc/config.json.
```

a. Set the NSO host and credentials.

b. Set the SME host and credentials.

Step 2 Enable SSL.

```
<install-dir>/CSCOnso-shim/bin/nso-shim security ssl enable
```

This command generates a server certificate and enables ssl.

Step 3 Start the service.

```
<install-dir>/CSCOnso-shim/bin/nso-shim start
```

Verifying the Service Assurance API Installation

Perform the following to verify the service assurance API installation.

Step 1 Point your browser to <Tomcat_URL>/assurance-api-1.0/v1.0/report and verify that you see SME reports.

Step 2 Point your browser to <Tomcat_URL>/assurance-api-1.0/v1.0/service/topology and verify that you have services with non-null operational states.

Verifying the NSO Integration Layer

Perform the following procedure to verify the NSO integration layer.

Step 1 Verify the service is running: nso-shim status.

Step 2 Verify the service can get the NSO services: nso-shim get services

Step 3 Verify the service can sync with SME: nso-shim syncSME

Step 4 View: http://<sme-host>:4440/ppm/assurance/service/topology/

Step 5 Check the log file:

```
 <install-dir>/CSCOnso-shim/logs/nso-shim.log
{
"rest_bind": "0.0.0.0",
"rest_port": 4450,
"rest_ssl": "false",
"rest_httpauth": "false",
"rest_username": "admin",
"rest_password": "admin",
"ncs_host": "",
```

```
"ncs_port": 2022,
"ncs_username": "admin",
"ncs_password": "admin",
"ncs_notif": "true",
"sme_host": "",
"sme_port": "4440",
"sme_ssl": "false",
"sme_username": "",
"sme_password": "",
"bus_timeout": 180
}
```

Viewing Performance Data on the VMS Portal

After you install and set up Cisco vMS SA, log in to the Cisco vMS portal and verify performance metrics are being received.

To view service metrics:

1. Log in to the Cisco vMS portal using your user credentials.
2. On the left portal window, select Services.

   The Services window displays the services that are running in percentages. The details also include the total number of services, services that are running, and the services that need attention.
3. To filter the service, click the number that appears below the percentage shown against each service.
4. To search for a customer name or service, click Search at the top of the window and enter the search criteria in the search box.

   The list of matching search items is displayed.
5. To view the service status, click a service from the drop-down list against the customer name.

   The service status and service details are displayed in the right pane. You can also monitor monthly usage and performance metrics of the service.
6. To view the monthly usage history of a service, click View History.
7. Click the graphical representation displayed under the Time Frame section to view performance metrics information such as Internet traffic, network traffic, and connected remote access users over a specific time frame. You can also click View History against the Time Frame to view the metrics history for a specific period.

Figure 6-3 shows a sample service status screen.

Figure 6-3 Sample Service Status Screen

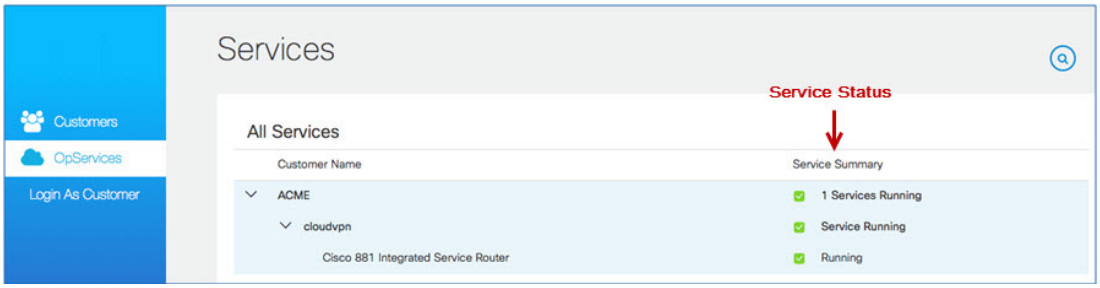
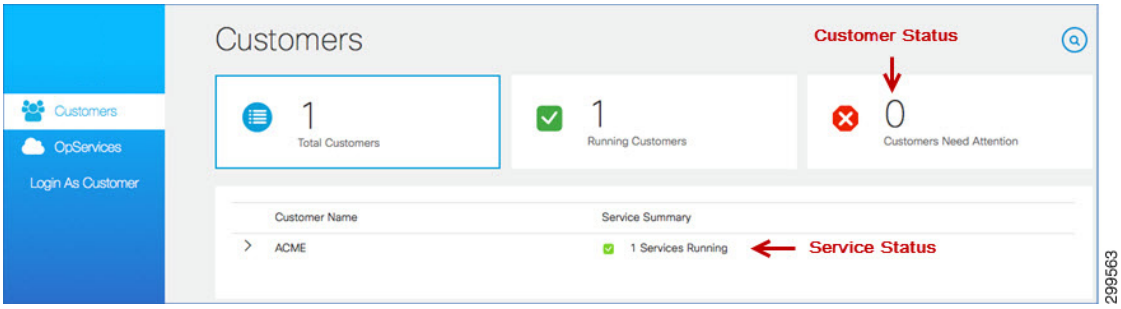


Figure 6-4 shows a sample customer status screen.

Figure 6-4 Sample Customer Status Screen



For additional information about working with the Cisco vMS Portal, refer to the Cisco Virtual Managed Services 2.0 Portal Administration and Operation Guide, and the Cisco Virtual Managed Services 2.0 Portal Administration and User Guide.

Reference

The following sections provide reference information and commands that you might need to use to install and verify Cisco vMS SA.

NSO Interface

The NSO interface is deployed with SME, NSO, the vMS Service Portal, and the Assurance API. The NSO interface is located between NSO and SME. For each NSO and SME instance, there is one NSO interface, one SME, one Assurance API, and one Service Portal. The NSO interface provides a view of the services provisioned by NSO and configures SME for vMS monitoring.

Note The NSO interface is also called the NSO shim.

Configuring the NSO interface

Perform the following to configure the NSO interface.

Step 1 Configure the NSO and SME credentials.

a. Display the config file in a text editor:

```
<install-dir>/CSCOnso-shim/bin/etc/config.json
```

b. Set the NSO host and credentials.

c. Set the SME host and credentials.

Step 2 Enable SSL.

```
<install-dir>/CSCOnso-shim/bin/nso-shim security ssl enable
```

The command generates a server certificate and enables SSL.

Step 3 Start the service.

```
<install-dir>/CSCOnso-shim/bin/nso-shim start
```

NSO Interface Files

Config File

CSCOnso-shim/etc/config.json

```
{
  "rest_bind": "0.0.0.0",     # nso shim rest server address
  "rest_port": 4450,          # nso shim rest server port
  "rest_ssl": "false",        # nso shim rest server SSL enabled
  "rest_httpauth": "false",   # nso shim rest server HTTP Auth enabled
  "rest_username": "admin",   # nso shim rest server HTTP Auth username
  "rest_password": "admin",   # nso shim rest server HTTP Auth password
  "ncs_host": "",             # nso host address
  "ncs_port": 2022,           # nso host netconf port
  "ncs_username": "admin",    # nso host netconf username
  "ncs_password": "admin",    # nso host netconf password
  "ncs_notif": "true",        # nso notification support enabled
  "sme_host": "",             # sme host address
  "sme_port": "4440",         # sme host rest/soap API port
  "sme_ssl": "false",         # sme host SSL enabled
  "sme_username": "",         # sme host HTTP Auth username
  "sme_password": "",         # sme host HTTP Auth password
  "bus_timeout": 180          # nso shim message bus timeout
}
```

Log Files

/CSCOnso-shim/logs/nso-shim.out—Log file for capturing standard out.

/CSCOnso-shim/logs/nso-shim.log—Regular log file:

- 100 MB max.
- At startup, the old logs are zipped, timestamped, and stored in the logs/backup directory.
- The log level is INFO - which is logging pretty much everything.

Configuration File

CSCOnso-shim/etc/logging.properties

```
handlers= java.util.logging.FileHandler
# LEVELS: SEVERE, WARNING, INFO, CONFIG, FINE, FINER ,FINEST
.level=INFO
java.util.logging.FileHandler.pattern = %h/logs/nso-shim.log
java.util.logging.FileHandler.limit = 100000000
java.util.logging.FileHandler.count = 1
java.util.logging.FileHandler.formatter = java.util.logging.SimpleFormatter
java.util.logging.SimpleFormatter.format=%4$s: %1$tc: %5$s%6$s%n
```

Command Usage

```
nso-shim {start | stop | restart | status}
nso-shim {get services [service] | search [device]]}
nso-shim {get devices [device]}
nso-shim {syncSME [service]}
nso-shim {security ssl [enable | disable | status]
```

```
nso-shim {security httpauth [enable | disable | status]
```

Examples

```
nso-shim get services
nso-shim get services SKT-ACME-VIPS-Medium
nso-shim get services search cpe-CPE_SKT_ACME_V_2

nso-shim get devices
nso-shim get devices cpe-CPE_SKT_ACME_V_2

nso-shim syncSME
nso-shim syncSME SKT-ACME-VIPS-Medium
(hidden option) -d enables debug mode
nso-shim -d start
nso-shim -d get services
```

Command Descriptions

- **nso-shim start**—Starts the nso shim. A rest API is presented at the configured bind address and port.

  (hidden) 'nso-shim -d start'—The -d option will start the nso shim is debug mode. Java socket attach debug is enabled on port 44750.
- **nso-shim stop**—Stops the nso shim. If the nso shim fails to stop within 60 seconds after receiving a regular kill signal, a thread dump logged, and the process is forcefully stopped.
- **nso-shim restart**—Restarts the nso shim.
- **nso-shim status**—Displays the status of the nso shim. If a PID for the nso shim is found by grepping the process list, the output of this command indicates that the nso shim is running.
- **nso-shim get services**—Lists the service instances configured in NSO. Example:

  ```
  nso-shim get services
  ```

  ### Corresponding REST API

  ```
  http://0.0.0.0:4450/services?&refresh=true&outputType=json
  ```

  ### Example Output

  ```
  Response Code: 200
  Response:
  [ {
    "virto" : "SKT-CVPN2",
    "provider" : "SKT",
    "tenant" : "coke",
    "devices" : [ "SKT-SKT-CVPN2-ASA-esc-zg", "SKT-SKT-CVPN2-CSR-esc-zg",
         "SKT-SKT-  CVPN2-WSA-esc-zg", "cpe-FTX12347", "cpe-FTX12348" ]
    }, {
    "virto" : "SKT-CVPN1",
    "provider" : "SKT",
    "tenant" : "coke",
    "devices" : [ "SKT-SKT-CVPN1-ASA-esc-vz", "SKT-SKT-CVPN1-CSR-esc-vz",
         "SKT-SKT-CVPN1-WSA-esc-vz", "cpe-FTX12345", "cpe-FTX12346" ]
  } ]
  ```

- **nso-shim get services [service]**—Lists the service for the given service name. Example:

  ```
  nso-shim get services SKT-CVPN1
  ```

Corresponding REST API

```
http://0.0.0.0:4450/services/SKT-CVPN1?&refresh=true&outputType=json
```

Example Output

```
Response Code: 200
Response:
{
  "virto" : "SKT-CVPN1",
  "provider" : "SKT",
  "tenant" : "coke",
  "devices" : [ "SKT-SKT-CVPN1-ASA-esc-vz", "SKT-SKT-CVPN1-CSR-esc-vz",
      "SKT-SKT-CVPN1-WSA-esc-vz", "cpe-FTX12345", "cpe-FTX12346" ]
}
```

- **nso-shim get services search [device]**—Lists the service for the given device name. Example:

```
nso-shim get services search SKT-SKT-CVPN1-WSA-esc-vz
Corresponding REST API
http://0.0.0.0:4450/services?search=SKT-SKT-CVPN1-WSA-esc-vz&refresh=true&outputType=j
son
Example output:
Response Code: 200
Response:
{
  "virto" : "SKT-CVPN1",
  "provider" : "SKT",
  "tenant" : "coke",
  "devices" : [ "SKT-SKT-CVPN1-ASA-esc-vz", "SKT-SKT-CVPN1-CSR-esc-vz",
       "SKT-SKT-CVPN1-WSA-esc-vz", "cpe-FTX12345", "cpe-FTX12346" ]
}
```

- **nso-shim get devices**—Lists the devices configured in NSO. The list is limited to devices supported by SME. Example:

```
nso-shim get devices
```

Corresponding REST API

```
http://0.0.0.0:4450/devices?&refresh=true&outputType=json
```

Example Output

```
Response Code: 200
Response:
[ {
"name" : "cpe-FTX12348",
"virto" : "SKT-CVPN2",
"provider" : "SKT",
"tenant" : "coke",
"type" : "cpe",
"address" : "127.0.0.1"
}, {
"name" : "SKT-SKT-CVPN1-ASA-esc-vz",
"virto" : "SKT-CVPN1",
"provider" : "SKT",
"tenant" : "coke",
"type" : "vFirewall",
"address" : "127.0.0.1"
}, {
"name" : "cpe-FTX12345",
"virto" : "SKT-CVPN1",
"provider" : "SKT",
"tenant" : "coke",
```

```
"type" : "cpe",
"address" : "127.0.0.1"
}, {

.
.

} ]
```

- **nso-shim get devices [device]**—Lists the device for the given device name. Example:

```
nso-shim get devices SKT-SKT-CVPN1-ASA-esc-vz
```

Corresponding REST API

```
http://0.0.0.0:4450/devices/SKT-SKT-CVPN1-ASA-esc-vz?&refresh=true&outputType=json
```

Example Output

```
Response Code: 200
Response:
{
"name" : "SKT-SKT-CVPN1-ASA-esc-vz",
"virto" : "SKT-CVPN1",
"provider" : "SKT",
"tenant" : "coke",
"type" : "vFirewall",
"address" : "127.0.0.1"
}
```

- **nso-shim syncSME**—Configures SME to monitor the services provisioned by NSO.
  - The SME global 5 minute and 1 minute report interval settings are enabled.
  - The NSO reports are enabled for all intervals. The report category is:
  - Reports -> Orchestration -> Network Services Orchestration
  - The NSO threshold definitions are created.

    TCA Name: Symphony_NSO_Node_Oper_State
  - KPI Report: NSO Node Operational State

    TCA Name: Symphony_NSO_Oper_State;
  - KPI Report: NSO Operational State
  - The Assurance API service topology templates are installed.
  - The Assurance API service meter templates are installed.

Example

```
nso-shim syncSME
```

Corresponding REST API

```
http://0.0.0.0:4450/syncSME?&dryRun=false
```

Example Output

```
Response Code: 200
Response:
Subscribed: cloudvpn-notif:4b95a99e428a40389bc10c8ffaa335e1
SME Sync finished for:
SKT-CVPN2
SKT-CVPN1
```

(optional) When dryRun is enabled, the SME configuration is logged and not installed.

- **nso-shim syncSME [service]**—Configures SME to monitor the service for the given service name. Assures the assurance API service topology and service meter templates are installed.

Example

```
nso-shim syncSME SKT-CVPN1
```

Corresponding REST API

```
http://0.0.0.0:4450/syncSME/SKT-CVPN1?&dryRun=false
```

Example Output

```
Response Code: 200
Response:
SME Sync finished for:
SKT-CVPN1
```

(optional) When dryRun is enabled, the SME configuration is logged and not installed.

SME Synchronization

The NSO interface attempts to connect to the NSO NETCONF interface every 60 seconds. After a connection to NSO is established, the NSO interface attempts to connect to SME. After connections to both SME and NSO are established, the NSO interface configures SME to monitor the services that are provisioned by NSO.

Upon receipt of a cloudvpn-provisioned notification, the NSO interface calls back to NSO to retrieve the service details of the service name given in the notification, then configures SME to monitor the service.

If the NETCONF connection is terminated, the NSO interface attempts to reconnect every 60 seconds. After the connection is reestablished, the NSO interface performs a full synchronization with SME. If an SME error prevents a full synchronization of NSO services, the NSO interface backs off to prevent overloading NSO with data requests. After several consecutive failures, the time between attempts will be more than one hour. In this case, resolve the issue with SME and then manually synchronize the services with the **nso-shim syncSME** command.