



DeveloperDays
Network Services Orchestrator

Lifecycle NSO

A presentation focused on the IT persona

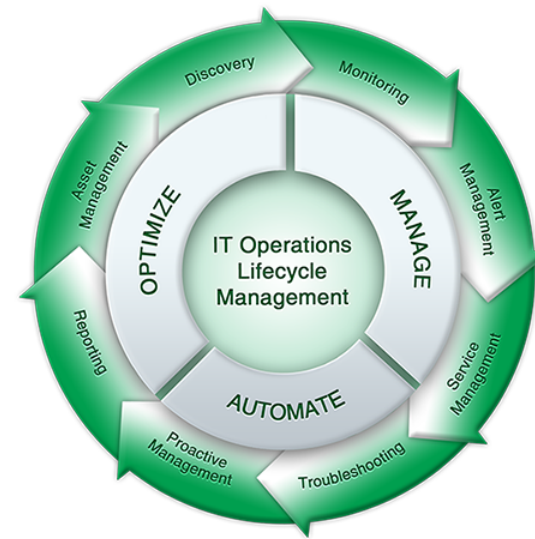
Roque Gagliano
Technical Solutions Architect
19/06/2019

IT is a first class persona for NSO



- NSO is becoming a critical application in our customers' IT environment
- However, most of the IT/Operational knowledge resides on our customers
- KEY is a close collaboration

IT is transforming



IT Asset Management



DevOps for NSO+Packages lifecycle

Today's Agenda

- Before starting
- NSO Installation and requirements
- NSO Security
- NSO Management
- NSO Upgrades

Before starting

© 2019 Cisco and/or its affiliates. All rights reserved. Cisco Public



NSO 5.1.1 Administration Guide

First Published: May 17, 2010

Last Modified: May 20, 2019

Americas Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA

<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Your best friend

NSO Deployment chapter is a must read for IT

We will not cover all elements in this presentation

- NSO Deployment includes a number of best practices with IT-sensitive topics
- IT should read this chapter when preparing a NSO installation
- We will not cover all these point in this presentation

NSO Deployment

- Introduction, page 97
- Initial NSO installation, page 100
- Initial NSO configuration - ncs.conf, page 101
- Setting up AAA, page 103
- Cisco Smart Licensing, page 104
- Global settings and timeouts, page 104
- Enabling SNMP, page 105
- Loading the required NSO packages, page 106
- Preparing the HA of the NSO installation, page 108
- Handling fail-f-hcc HA fallout, page 110
- Preparing the clustering of the NSO installation, page 112
- Testing the cluster configuration, page 113
- NSO system and packages upgrade, page 114
- Log management, page 120
- Monitoring the installation, page 121
- Security considerations, page 121

NSO Installation and requirements

NSO Requirements are changing

- Java Version:
 - Java6 and Java7 are not longer supported by community
- Python Version:
 - Python 2.7 is set to retire in Jan 2020
- We are planning to remove support in new NSO versions

Oracle Java SE Support Roadmap*†				
Release	GA Date	Premier Support Until	Extended Support Until	Sustaining Support
6	December 2006	December 2015	December 2018	Indefinite
7	July 2011	July 2019	July 2022*****	Indefinite
8**	March 2014	March 2022	March 2025	Indefinite
9 (non-LTS)	September 2017	March 2018	Not Available	Indefinite
10 (non-LTS)	March 2018	September 2018	Not Available	Indefinite
11 (LTS)	September 2018	September 2023	September 2026	Indefinite

Python 2.7 will retire in...

0

Years

6

Months

18

Days

7

Hours

46

Minutes

25

Seconds

NSO can be installed in bared-metal, VM, container

- NSO distributed as a signed .bin file
- Can be installed in any Linux system in bared-metal, VM or a **container** in production.
- A Darwin (MAC) version included for testing but not production
- NSO is typically a memory-hungry application with very few process (Erlang, Java or Python VM processes)

What is the best strategy depends on your IT environment.

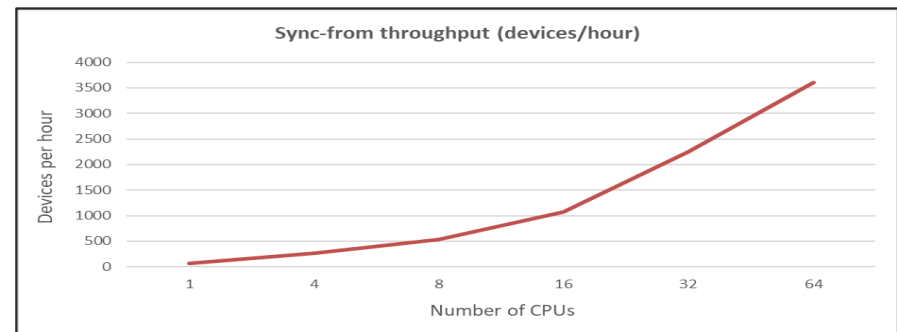
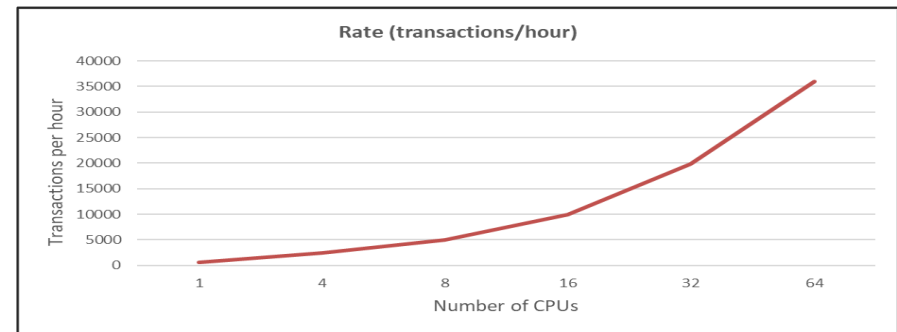
NSO can scale vertically or horizontally but operational costs are different!

NSO Resource Sizing

Still a challenging topic, typically resources are cheap

- What can affect NSO sizing?
 - Use case definition and **implementation**
 - Number, type and behavior of devices
 - Required transaction rate
 - Average size of total configs and diff-set configs
 - Required sync-from frequency
 - Commit queues yes/no

Monitoring NSO resources more important than getting resources right on day 0



Performance testing your installation: create a benchmark and look for introduced changes

NSO High Availability

NSO is active-standby system

- NSO has HA support where an active node synchronizes with one or many standby nodes.
- HA requires same NSO installation (version, packages...)
- HA could be used for disaster recovery
- Automatic Switch in between NSO servers possible but be-aware of split-brain issues!

External HA framework (optional)

NSO tailf-hcc (optional)

HA RC/CLI/JRPC/NC API
Layer 2 and Layer 3 automatic HA switch

NSO infra HA support

HA hardbeats, HA sync, HA Java/Python APIs
Configured in ncs.conf

NSO Installation

What tool to use...follow your IT process, check nct

- You typically want to use your IT process
- A number of examples and best practices shared at the NSO developer hub (including Ansible playbooks)
- Please check the NSO nct tool, it may simplify your task!

```
NCT(1) NCS Manual NCT(1)
NAME
nct - a collection of tools that can be used to install and manage NCS nodes.
INTRODUCTION
A host running NCS is called an NCS node. A host can be either a physical machine or a virtual machine as long as they can be accessed via SSH.
With nct it is possible to install and manage NCS nodes. This assumes that the hosts are running Linux and are accessible via SSH.
The hostsfile
Each NCS node can be operated on independently but the main idea is that nct can operate on a set of NCS nodes. To operate on a set of NCS nodes a, so called, hostsfile is needed. A hostsfile consists of Host entries according to the following example:
{"192.168.23.99", []}.
{"192.168.23.98", []}.
...etc...
Each entry, called a Tuple, begin with a '{' bracket and ends with a '}'.'
Note
Note the ending dot after the '}' bracket!
Each entry consists of a Hostname/IP-Address, enclosed in double quotes and a list of options, where the list begin with a '[' bracket and ends with a corresponding ']' bracket.
In the list of options, information can be given that, in most cases, have a corresponding switch option to the various tool commands. So instead of having to specify the SSH User to every tool command with the '--ssh-user' switch; it can be specified in the hostsfile as:
{"192.168.23.99", [{"ssh_user","user"}]}.
{"192.168.23.98", [{"ssh_user","user"}]}.
...etc...
```

RESTCONF as your RestFull API

- NSO currently has two REST APIs
- All RestFull innovations are only happening in RESTCONF
- All new integrations should be done over RESTCONF
- Working on improving RESTCONF documentation

Only in RESTCONF:

Standard alignment
Token based authentication
Patch-Media-Type
W3C Server-Sent-Events
Get-Schema operation
OpenAPI/Swagger plugin
NSO extension for Query API

NSO Security

NSO follows the Cisco SDL Process

SDL = Secure Development Lifecycle

- The Cisco SDL process covers a number of test required by your security departments
- A number of customers performing penetration test
- In most cases, issues resides on NSO server configuration “security hardening”



[Cisco Trustworthy Solutions](#)

NSO Security Hardening

A number of areas to study in your installation at `ncs.conf` file

- IPC port security:
 - IPC is like a console into NSO
- Authentication:
 - We now include token based Auth.
- Authorization:
 - Via NSO NACM rules
 - System install defaults different from local
- Log management/rotation
- Enabled northbound APIs
- Hosted Linux: permissions, Java memory size and file descriptors
- HA framework configuration (including shared token)

Please review `ncs.conf` MAN page, more options than skeleton file

NSO Security Hardening

A number of areas to study in your installation at ncs.conf file

- Encryption Symmetrical Keys:
 - You should not use the defaults
 - You can now call external script
- SSH/TLS crypto settings:
 - Versions, keys, ciphers, certificates, root certs for crypto management
- HTTPS security headers:
 - Powerful config options

Please review ncs.conf MAN page, more options than skeleton file

NSO Management

NSO Alarms

- NSO Forwards alarms in a variety of ways:
 - SNMP Traps
 - Northbound API Notifications: RESTCONF, NETCONF, JSON-RPC (ncs-alarms stream)
- Alarms are also available in CLI and GUI
- Customers can create their own alarms via the Alarms API (no Python yet)

Alarm Identity abort-error	Initial Perceived Severity major
Description An error happened while aborting or reverting a transaction. Device's configuration is likely to be inconsistent with the NCS CDB.	Recommended Action Inspect the configuration difference with compare-config, resolve conflicts with sync-from or sync-to if any.
Alarm message(s) <ul style="list-style-type: none"> • Device {dev} is locked • Device {dev} is southbound locked • abort error 	
Clear condition(s) If NCS achieves sync with the device, or receives a transaction id for a netconf session towards the device, the alarm is cleared.	
Alarm Identity alarm-type	
Description Base identity for alarm types. A unique identification of the fault, not including the managed object. Alarm types are used to identify if alarms indicate the same problem or not, for lookup into external alarm documentation, etc. Different managed object types and instances can share alarm types. If the same managed object reports the same alarm type, it is to be considered to be the same alarm. The alarm type is a simplification of the different X.733 and 3GPP alarm IRP alarm correlation mechanisms and it allows for hierarchical extensions. A 'specific-problem' can be used in addition to the alarm type in order to have different alarm types based on information not known at design-time, such as values in textual SNMP Notification varbinds.	
Alarm Identity bad-user-input	Initial Perceived Severity critical
Description Invalid input from user. NCS cannot recognize parameters needed to connect to device.	Recommended Action Verify that the user supplied input are correct.
Alarm message(s)	

New! Detailed alarms descriptions with recommended actions in Admin Guide.

NSO Backup and Recovery

- NSO provides a backup and recovery toolkit: ncs-backup
- You should run it every 6/12/24 hours
- Backups != HA but complementary (total disaster, forensic analysis)
- You could use backup data to run batch processes
- Store backups in external storage with backup storage policies

Monitoring NSO

ncs --state

- What to monitor in a NSO server?
 - Watch NSO daemons
 - Memory, CPU and disk (different params)
 - HA mode consistency
 - Loaded packages (are they all up?)
 - Java VMs and Python VMs (all running?)
 - Callpoints: are they all registered?
 - DB Locks: is someone holding the running lock for too long?

Remember! You can retrieve the ncs-status data via Northbound API:

<https://localhost:8888/restconf/data/tailf-ncs-monitoring:ncs-state>

<https://localhost:8888/restconf/data/tailf-ncs:packages>

<https://localhost:8888/restconf/data/tailf-ncs:java-vm>

<https://localhost:8888/restconf/data/tailf-ncs:python-vm>

NSO logging and auditing

Available via Syslog

- NSO provides a number of logs: audit, developer, system (ncs.log), API(s) logs, error, progress tracing, xpath logs, NED traces.
- Some logs could have a performance impacts and should only be enabled for troubleshooting: xpath, NED traces
- Check-out new audit-network-log

```
ROGAGLIA-M-D2QR:simple-mps-vpn roaglia$ tail network-audit.log
snmp-server community cisco_public
END EDIT
<INFO> 17-Jun-2019::18:45:55.473 ROGAGLIA-M-D2QR ncs[23190]: audit_network user: admin/37 thandle 187 hostname ROGAGLIA-M-D2QR device ce2
BEGIN EDIT
snmp-server community cisco_public
END EDIT
<INFO> 17-Jun-2019::18:45:55.473 ROGAGLIA-M-D2QR ncs[23190]: audit_network user: admin/37 thandle 187 hostname ROGAGLIA-M-D2QR device ce0
BEGIN EDIT
snmp-server community cisco_public
END EDIT
ROGAGLIA-M-D2QR:simple-mps-vpn roaglia$ cat network-audit.log
<INFO> 17-Jun-2019::18:45:55.472 ROGAGLIA-M-D2QR ncs[23190]: audit_network user: admin/37 thandle 187 hostname ROGAGLIA-M-D2QR device ce1
BEGIN EDIT
snmp-server community cisco_public
END EDIT
<INFO> 17-Jun-2019::18:45:55.472 ROGAGLIA-M-D2QR ncs[23190]: audit_network user: admin/37 thandle 187 hostname ROGAGLIA-M-D2QR device ce4
BEGIN EDIT
snmp-server community cisco_public
END EDIT
<INFO> 17-Jun-2019::18:45:55.473 ROGAGLIA-M-D2QR ncs[23190]: audit_network user: admin/37 thandle 187 hostname ROGAGLIA-M-D2QR device ce3
BEGIN EDIT
snmp-server community cisco_public
END EDIT
<INFO> 17-Jun-2019::18:45:55.473 ROGAGLIA-M-D2QR ncs[23190]: audit_network user: admin/37 thandle 187 hostname ROGAGLIA-M-D2QR device ce2
BEGIN EDIT
```

NSO Upgrades

- NSO Major Upgrade
- NSO Minor Upgrade
- NSO Package reload
- NSO Java/Python VM redeploy

NSO Major Upgrade

- When upgrading the first or second digit of an NSO version
- Example: 4.2 to 4.4 or 4.7 to 5.1
- Exceptionally, it may required more than one step upgrade
- Requires package recompilation
- Should be done in maintenance windows

NSO Minor Upgrade

- When upgrading the third or fourth digit of an NSO version
- Example: 4.5.0 to 4.5.3 or 5.1.1 to 5.1.2
- Should not require packages recompilation
- Should be done in maintenance windows

NSO Package Reload

- No changes in the NSO runtime environment but only on some of the packages (ex. NEDs, Services, etc.)
- Required for changes in YANG files and/or Templates
- Still requires a maintenance windows as NSO (even in HA) will not be available for a number of minutes
- Cannot be done “per-package”

NSO Java/Python VM redeploy

- No changes in the NSO runtime environment but only on some of the packages (ex. NEDs, Services, etc.)
- Only valid when changes in Java/Python VMs (your code)
- Could be done without maintenance windows as package will be unable for some seconds/minutes if northbound system can re-try
- Can be performed “per-package”

General notes on NSO Upgrades

- Remember than HA, requires same NSO version and same packages versions.
- Do not forget to backup all your instances of NSO
- During upgrades is when all your investment in testing pays-off! It is fundamental
- We would love you to perform a major upgrade once a year

Conclusions and call for action

- IT is an important persona for NSO
- We are investing on the NSO Application lifecycle
- You are running NSO everyday, **we want to hear from you!**

