|  |
| --- |
| Cisco SAE Core Function Pack  User Guide |
|  |
| Version 1.0.0 |
|  |

|  |  |
| --- | --- |
|  |  |
| **SAE Core Function Pack User Guide** | October 2018 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |

Contents

[1 Preface 1](#_Toc528329745)

[1.1 Abstract 1](#_Toc528329746)

[1.2 Purpose of This Document 1](#_Toc528329747)

[1.3 Using This Document 1](#_Toc528329748)

[1.4 Applicability 1](#_Toc528329749)

[1.5 References 1](#_Toc528329750)

[1.6 Terminology 1](#_Toc528329751)

[2 Overview 3](#_Toc528329752)

[2.1 Cisco NSO Introduction 3](#_Toc528329753)

[2.2 Using XML Payloads 4](#_Toc528329754)

[3 SAE Solution Overview 6](#_Toc528329755)

[3.1 Introduction 6](#_Toc528329756)

[3.2 Architecture 6](#_Toc528329758)

[4 SAE Core Function Pack Overview 8](#_Toc528329759)

[4.1 Introduction 8](#_Toc528329760)

[4.2 Architecture 8](#_Toc528329761)

[4.2.1 NSO Packages for SAE 9](#_Toc528329762)

[4.3 Supported Hardware 9](#_Toc528329797)

[4.4 Supported Software 9](#_Toc528329798)

[5 SAE Service Design 11](#_Toc528329799)

[5.1 Introduction 11](#_Toc528329800)

[5.2 Network Planning 12](#_Toc528329807)

[5.2.1 Overview 12](#_Toc528329813)

[5.2.2 Network Infrastructure 12](#_Toc528329814)

[5.2.3 VNFM 17](#_Toc528329815)

[5.2.4 VIM 17](#_Toc528329816)

[5.2.5 Topology 17](#_Toc528329817)

[5.2.6 Packaging, Installation and Versioning 20](#_Toc528329818)

[5.2.7 Troubleshooting and Diagnostics 20](#_Toc528329819)

[5.3 Site Planning 21](#_Toc528329820)

[5.3.1 Overview 21](#_Toc528329826)

[5.3.2 Site 22](#_Toc528329827)

[5.3.3 Prerequisites/Dependencies 25](#_Toc528329828)

[5.3.4 Providers and Tenants 25](#_Toc528329829)

[5.3.5 Prerequisites/Dependencies 25](#_Toc528329830)

[5.3.6 Troubleshooting and Diagnostic 26](#_Toc528329831)

[5.3.7 Catalog 26](#_Toc528329832)

[5.4 Resource Planning 30](#_Toc528329833)

[5.4.1 Overview 30](#_Toc528329839)

[5.4.2 ID and IP Pool 31](#_Toc528329840)

[5.4.3 Usage with Sample Payloads 34](#_Toc528329841)

[5.4.4 Verification 34](#_Toc528329842)

[5.4.5 Troubleshooting and Diagnostic 35](#_Toc528329843)

[5.5 Service Component Planning 36](#_Toc528329845)

[5.5.1 Overview 36](#_Toc528329850)

[5.5.2 External Endpoints 36](#_Toc528329851)

[5.5.3 Virtualized Network Function Descriptor (VNFD) 37](#_Toc528329852)

[5.5.4 NSD 39](#_Toc528329853)

[5.6 Chain Deployment Planning 44](#_Toc528329854)

[5.6.1 Overview 44](#_Toc528329856)

[5.6.2 HA Planning 45](#_Toc528329857)

[5.6.3 Resource Zone Planning 46](#_Toc528329858)

[5.6.4 Affinity Group 49](#_Toc528329859)

[5.6.5 Bandwidth Planning 51](#_Toc528329860)

[5.7 Service Chain Design 56](#_Toc528329861)

[5.7.1 Overview 56](#_Toc528329867)

[5.7.2 EPGW 56](#_Toc528329868)

[5.7.3 Sample Payloads 56](#_Toc528329870)

[6 SAE Service Provisioning 57](#_Toc528329873)

[6.1 Introduction 57](#_Toc528329874)

[6.2 Infrastructure Discovery 57](#_Toc528329880)

[6.2.1 Overview 57](#_Toc528329881)

[6.2.2 Action 59](#_Toc528329882)

[6.2.3 Payload 62](#_Toc528329883)

[6.2.4 Diagnosis 63](#_Toc528329884)

[6.2.5 Troubleshooting 65](#_Toc528329885)

[6.3 EPGW 66](#_Toc528329886)

[6.3.1 Overview 66](#_Toc528329887)

[6.3.2 Sample Payloads 67](#_Toc528329888)

[6.3.3 Verification 67](#_Toc528329890)

[6.3.4 Troubleshooting 67](#_Toc528329891)

[6.4 Service Chain 69](#_Toc528329898)

[6.4.1 Overview 69](#_Toc528329899)

[6.4.2 Prerequisites/Dependencies 71](#_Toc528329900)

[6.4.3 Activate 72](#_Toc528329901)

[6.4.4 Sample Payloads 74](#_Toc528329902)

[6.4.5 Diagnosis 75](#_Toc528329906)

[6.4.6 Troubleshooting 77](#_Toc528329907)

[6.5 Stitching Service Chain 77](#_Toc528329908)

[6.5.1 Overview 77](#_Toc528329909)

[6.5.2 Prerequisites/Dependencies 78](#_Toc528329910)

[6.5.3 Diagnosis 80](#_Toc528329911)

[6.5.4 Troubleshooting 81](#_Toc528329912)

[7 SAE Core FP Features 82](#_Toc528329919)

[7.1 Introduction 82](#_Toc528329920)

[7.2 Service Plan 82](#_Toc528329929)

[7.2.1 Overview 82](#_Toc528329930)

[7.2.2 Sample Payloads 84](#_Toc528329931)

[7.2.3 Diagnosis 86](#_Toc528329933)

[7.2.4 Troubleshooting 86](#_Toc528329934)

[7.3 EPGW HA 87](#_Toc528329946)

[7.3.1 Overview 87](#_Toc528329947)

[7.3.2 Activate 87](#_Toc528329948)

[7.3.3 Modify 87](#_Toc528329949)

[7.3.4 Delete 87](#_Toc528329950)

[7.3.5 Sample Payloads 87](#_Toc528329951)

[7.3.6 Diagnosis 87](#_Toc528329952)

[7.3.7 Troubleshooting 87](#_Toc528329953)

[7.4 Service Chain HA 87](#_Toc528329954)

[7.4.1 Overview 87](#_Toc528329955)

[7.4.2 Activate 87](#_Toc528329956)

[7.4.3 Modify 87](#_Toc528329957)

[7.4.4 Delete 87](#_Toc528329958)

[7.4.5 Sample Payloads 87](#_Toc528329959)

[7.4.6 Diagnosis 87](#_Toc528329960)

[7.4.7 Troubleshooting 87](#_Toc528329961)

[7.5 Inline Service Chain 88](#_Toc528329962)

[7.5.1 Overview 88](#_Toc528329963)

[7.5.2 Activate 88](#_Toc528329964)

[7.5.3 Modify 88](#_Toc528329965)

[7.5.4 Delete 88](#_Toc528329966)

[7.5.5 Sample Payloads 88](#_Toc528329967)

[7.5.6 Diagnosis 88](#_Toc528329968)

[7.6 Controller Based VNF 89](#_Toc528329969)

[7.6.1 ASAv Extension Support 89](#_Toc528329970)

[7.6.2 Usage with Sample Payloads 91](#_Toc528329971)

[7.6.3 Troubleshooting and Diagnostics 93](#_Toc528329978)

[7.6.4 Packaging, Installation and Versioning 93](#_Toc528329979)

[7.6.5 AVI-SE Extension Support 94](#_Toc528329980)

[7.6.6 Prerequisites/Dependencies 96](#_Toc528329981)

[7.6.7 Usage with Sample Payloads 96](#_Toc528329982)

[7.6.8 Verification 96](#_Toc528329983)

[7.6.9 Troubleshooting and Diagnostic 96](#_Toc528329984)

[7.6.10 Packaging, Installation and Versioning 97](#_Toc528329985)

[7.6.11 FTDv Extension Support 97](#_Toc528329986)

[7.7 VNF Replacement 97](#_Toc528329987)

[7.7.1 Overview 97](#_Toc528329988)

[7.7.2 Prerequisites/Dependencies 97](#_Toc528329989)

[7.7.3 Usage with Sample Payloads for Replacing Switch 98](#_Toc528329990)

[7.8 SAE Actions 99](#_Toc528329997)

[7.8.1 Overview 99](#_Toc528329998)

[7.8.2 Site Actions 99](#_Toc528329999)

[7.8.3 Custom Actions 113](#_Toc528330000)

[7.8.4 Prerequisites/Dependencies 115](#_Toc528330001)

[7.8.5 Usage with Sample Payloads 115](#_Toc528330002)

[7.8.6 Cleanup Actions 116](#_Toc528330003)

[8 SAE Models 117](#_Toc528330004)

[8.1 Introduction 117](#_Toc528330005)

[9 Other SAE Features 118](#_Toc528330006)

[9.1 Introduction 118](#_Toc528330007)

[9.2 BGP AS Configuration 119](#_Toc528330011)

[9.2.1 Overview 119](#_Toc528330012)

[9.2.2 Activate 119](#_Toc528330013)

[9.2.3 Modify 119](#_Toc528330014)

[9.2.4 Delete 119](#_Toc528330015)

[9.2.5 Sample Payloads 119](#_Toc528330016)

[9.2.6 Diagnosis 119](#_Toc528330017)

[9.2.7 Troubleshooting 119](#_Toc528330018)

[9.3 Intangible VNF 119](#_Toc528330019)

[9.3.1 Overview 119](#_Toc528330020)

[9.3.2 Activate 119](#_Toc528330021)

[9.3.3 Modify 119](#_Toc528330022)

[9.3.4 Delete 119](#_Toc528330023)

[9.3.5 Sample Payloads 119](#_Toc528330024)

[9.3.6 Diagnosis 119](#_Toc528330025)

[9.3.7 Troubleshooting 119](#_Toc528330026)

[9.4 Spine Leaf Configuration 119](#_Toc528330027)

[9.4.1 Overview 119](#_Toc528330028)

[9.4.2 Activate 120](#_Toc528330029)

[9.4.3 Modify 120](#_Toc528330030)

[9.4.4 Delete 120](#_Toc528330031)

[9.4.5 Sample Payloads 120](#_Toc528330032)

[9.4.6 Diagnosis 120](#_Toc528330033)

[9.4.7 Troubleshooting 120](#_Toc528330034)

[9.5 SAE Add/Delete CSP from infrastructure 120](#_Toc528330035)

[9.5.1 Overview 120](#_Toc528330036)

[9.5.2 Description 120](#_Toc528330037)

[9.5.3 Prerequisites/Dependencies 125](#_Toc528330038)

[9.5.4 Usage with Sample Payloads 125](#_Toc528330039)

[9.5.5 Payloads Validation 129](#_Toc528330040)

[9.5.6 Verification 129](#_Toc528330041)

[9.5.7 Test Case 130](#_Toc528330042)

[9.5.8 Webex Recording 130](#_Toc528330043)

[9.5.9 Troubleshooting and Diagnostic 130](#_Toc528330044)

[9.6 SAE Diagnosis Package 130](#_Toc528330045)

[9.6.1 Overview 130](#_Toc528330046)

[9.6.2 Description 130](#_Toc528330047)

[9.6.3 Yang Model 131](#_Toc528330048)

[9.6.4 Prerequisites/Dependencies 132](#_Toc528330049)

[9.6.5 Usage with Sample Payloads 132](#_Toc528330050)

[9.6.6 Payloads Validation 136](#_Toc528330051)

[9.6.7 Verification 136](#_Toc528330052)

[9.6.8 Troubleshooting and Diagnostic 136](#_Toc528330053)

[9.6.9 Packaging, Install, versioning 136](#_Toc528330054)

[9.6.10 Testcase 136](#_Toc528330055)

[9.6.11 Sources 136](#_Toc528330056)

[9.6.12 Webex Recording 136](#_Toc528330057)

[9.7 SAE ASAv VNF's Extensions Support for HA 136](#_Toc528330058)

[9.7.1 Overview 136](#_Toc528330059)

[9.7.2 Description 136](#_Toc528330060)

[9.7.3 Yang Model 136](#_Toc528330061)

[9.7.4 Prerequisites/Dependencies 138](#_Toc528330062)

[9.7.5 Usage with Sample Payloads 138](#_Toc528330063)

[9.7.6 Payloads Validation 140](#_Toc528330064)

[9.7.7 Verification 140](#_Toc528330065)

[9.7.8 Troubleshooting and Diagnostic 141](#_Toc528330066)

[9.7.9 Packaging, Install, versioning 141](#_Toc528330067)

[9.7.10 Test Case 141](#_Toc528330068)

[9.7.11 Sources 141](#_Toc528330069)

[9.7.12 Webex Recording 142](#_Toc528330070)

[9.8 Transparent and Routed VNFs 142](#_Toc528330071)

[9.8.1 Overview 142](#_Toc528330072)

[9.8.2 Description 142](#_Toc528330073)

[9.8.3 Yang Model 142](#_Toc528330074)

[9.8.4 Prerequisites/Dependencies 143](#_Toc528330075)

[9.8.5 Usage with Sample Payloads 143](#_Toc528330076)

[9.8.6 Allocations in Transparent mode: 143](#_Toc528330077)

[9.8.7 Validations 145](#_Toc528330078)

[9.8.8 Verification 146](#_Toc528330079)

[9.8.9 Assumptions 146](#_Toc528330080)

[9.8.10 Configs 146](#_Toc528330081)

[9.8.11 Troubleshooting and Diagnostic 146](#_Toc528330082)

[9.8.12 Packaging, Install, versioning 146](#_Toc528330083)

[9.8.13 Testcase 147](#_Toc528330084)

[9.8.14 Sources 147](#_Toc528330085)

[9.8.15 Webex Recording 147](#_Toc528330086)

[9.9 Replacement of VNF in existing service [epgw, service-chain] 147](#_Toc528330087)

[9.9.1 Overview 147](#_Toc528330088)

[9.9.2 Prerequisites/Dependencies 147](#_Toc528330089)

[9.9.3 Usage with Sample Payloads for Replacing Switch 147](#_Toc528330090)

[9.9.4 Usage with Sample Payloads for Replacing EPGW 149](#_Toc528330091)

[9.9.5 Usage with Sample Payloads for Replacing VNF in a service chain 150](#_Toc528330092)

[9.9.6 Packaging, Install, versioning 150](#_Toc528330093)

[9.10 SAE Maintenance 151](#_Toc528330094)

[9.10.1 Overview 151](#_Toc528330095)

[9.11 SAE Customization 171](#_Toc528330096)

[9.11.1 Overview 171](#_Toc528330097)

[9.11.2 Custom Template 171](#_Toc528330098)

[9.11.3 Yang Model 171](#_Toc528330099)

[9.11.4 Prerequisites/Dependencies 171](#_Toc528330100)

[9.11.5 Usage with Sample Payloads 172](#_Toc528330101)

[9.11.6 SAE Custom Template Payloads 174](#_Toc528330102)

[9.11.7 Verification 179](#_Toc528330103)

[9.11.8 Troubleshooting and Diagnostic 183](#_Toc528330104)

[9.11.9 Packaging, Install, versioning 183](#_Toc528330105)

[9.11.10 Testcase 184](#_Toc528330106)

[9.11.11 Sources 184](#_Toc528330107)

[9.11.12 Template Overriding 184](#_Toc528330108)

[9.11.13 Yang Model 184](#_Toc528330109)

[9.11.14 Prerequisites/Dependencies 185](#_Toc528330110)

[9.11.15 Usage with Sample Payloads 185](#_Toc528330111)

[9.11.16 Verification 189](#_Toc528330112)

[9.11.17 Troubleshooting and Diagnostic 189](#_Toc528330113)

[9.11.18 Common Issues & Action 189](#_Toc528330114)

[9.11.19 Packaging, Install, versioning 189](#_Toc528330115)

[9.11.20 Testcase 189](#_Toc528330116)

[9.11.21 Sources 190](#_Toc528330117)

[9.12 SAE ITD service 190](#_Toc528330118)

[9.12.1 Overview 190](#_Toc528330119)

[9.12.2 Description 190](#_Toc528330120)

[9.12.3 Yang Model 190](#_Toc528330121)

[9.12.4 Prerequisites/Dependencies 191](#_Toc528330122)

[9.12.5 ITD NSD Diagram 192](#_Toc528330123)

[9.12.6 ITD service Payload 192](#_Toc528330124)

[9.12.7 Payloads Validation 192](#_Toc528330125)

[9.12.8 Verification 193](#_Toc528330126)

[9.12.9 Troubleshooting and Diagnostic 193](#_Toc528330127)

[9.12.10 Packaging, Install, versioning 194](#_Toc528330128)

[9.12.11 Testcase 194](#_Toc528330129)

[9.12.12 Sources 194](#_Toc528330130)

[9.12.13 Webex Recording 194](#_Toc528330131)

[9.13 SAE VNF License deregistration for IOS and ASA 194](#_Toc528330132)

[9.13.1 Overview 194](#_Toc528330133)

[9.13.2 Description 194](#_Toc528330134)

[9.13.3 Yang Model 195](#_Toc528330135)

[9.13.4 Prerequisites/Dependencies 196](#_Toc528330136)

[9.13.5 Usage with Sample Payloads 196](#_Toc528330137)

[9.13.6 Payloads Validation 196](#_Toc528330138)

[9.13.7 Verification 196](#_Toc528330139)

[9.13.8 Troubleshooting and Diagnostic 196](#_Toc528330140)

[9.13.9 Packaging, Install, versioning 196](#_Toc528330141)

[9.13.10 Testcase 196](#_Toc528330142)

[9.13.11 Sources 196](#_Toc528330143)

[9.13.12 Webex Recording 196](#_Toc528330144)

[10 Appendix A. Yang Models and Payloads 197](#_Toc528330145)

[10.1 SAE Model 197](#_Toc528330146)

[10.2 VNF Model 257](#_Toc528330147)

[10.3 NSD Model 277](#_Toc528330148)

[10.4 EPGW Payloads 292](#_Toc528330149)

**List of Tables**

[Table 1 (1.5) – Lists documents and other reference sources containing information that may be essential to understanding topics in this document 8](#_Toc522619960)

[Table 2 (1.6) – Terminology 9](#_Toc522619961)

# Preface

## Abstract

This document serves as the NSO SAE Core Function Pack User Guide containing information about the NSO SAE Core Function Pack, Version 1.0.0.

## Purpose of This Document

This document shows how users can install and configure SAE Core Function Pack for Cisco customers. The target audience is Cisco Advanced Services developers, network, and system engineers who need to deliver SAE functionalities for Cisco customers.

## Using This Document

This document assumes that the reader has a good understanding of NSO and its usage as described in the NSO 4.7 User Guide.

## Applicability

SAE Core Function Pack Version 1.0.0.

## References

Table 1 (1.5) – Lists documents and other reference sources containing information that may be essential to understanding topics in this document

|  |  |  |
| --- | --- | --- |
| No. | Designation | Title |
| 1 |  | SAE Function Pack Installation Guide 1.0.0 |
| 2 |  | NSO 4.7 Installation Guide |
| 3 |  | NSO 4.7 User Guide |
| 4 |  | NSO 4.7 Admin Guide |

## Terminology

Table 2 (1.6) – Terminology

| Term | Definition |
| --- | --- |
| API | Application Programming Interface |
| ASA | Adaptive Security Appliances |
| CDB | Configuration Database |
| CFP | Core Function Pack |
| CLI | Command Line Interface |
| CPE | Customer Premise Equipment |
| ENCS | Enterprise Network Compute System |
| FP | Function Pack |
| IOS | Internetwork Operating System (CISCO) |
| IOSv | Internetwork Operating System virtual (CISCO) |
| ISR | Integrated Services Router |
| NCS | Network Control System |
| NCT | NSO cluster Tools |
| NED | Network Element Driver |
| NETCONF | Network Configuration Protocol |
| NFVIS | Network Function Virtual Infrastructure Software |
| NIC | Network Interface Controller |
| NSO | Network Services Orchestrator |
| PnP | Plug-n-Play |
| REST | Representational State Transfer |
| SR-IOV | Single-Root I/O Virtualization |
| SAE | Secure Agile Exchange |
| VDU | Virtual Deployment Unit |
| VM | Virtual Machine |
| VNF | Virtualized Network Function |
| VNFD | Virtualized Network Function Descriptor |
| vNIC | Virtualized Network Interface Controller |
| WAAS | Wide Area Application Services |
| XML | Extensible Markup Language |
| YANG | Modeling language per RFC6020 |

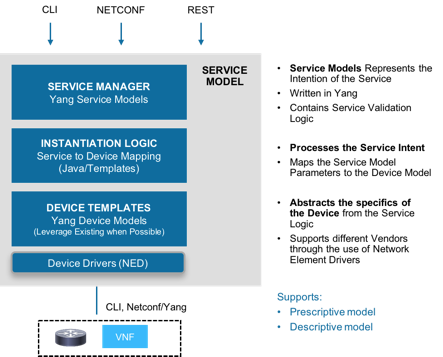
# Overview

The following chapter gives a brief overview of Cisco Network Services Orchestration (NSO).

## Cisco NSO Introduction

Cisco NSO is an evolution of the Tail-f Network Control System (NCS). Tail-f is a company that was acquired by Cisco in 2014. Network engineers use NSO as a central point of management for the entire network, using a network command line interface (CLI). NSO enables service providers to dynamically adopt their orchestration solution with changes in the offered service portfolio. NSO is built on a Model-Driven Architecture which supports the dynamic addition and modification of service definitions. The service models are written in the YANG modeling language (RFC 6020).

Cisco Network Services Orchestrator Layer Architecture.



## Using XML Payloads

Network engineers use NSO as a central point of management for the entire network, using a network command line interface (network CLI). While this guide will illustrate the use cases with the network CLI, it is important to realize that any northbound interface can be used to achieve the same functionality.

Desired service configurations can be written in XML and saved as XML payload files. These can be used to the deploy the services by loading the files through the NSO CLI, or by directly invoking the NETCONF northbound interface. A service can be composed of a single file or multiple files. Service creation and modification operations can be achieved through the XML payload files.

The NETCONF NBI can be invoked from a Linux console and the XML payload can be loaded into NSO:

# netconf-console --port=830 --host=127.0.0.1 -u admin -p cisco123 --edit-config payload.xml

-port as specified for netconf console in ncs.conf

-host is the <NSO IP address> where the service is being created

-u is username of your host

-p is password of your host

To get the config, use the get-config using netconf-console :

# netconf-console --port=830 --host=127.0.0.1 -u admin -p cisco123 –get

NETCONF operations are:

* <get-config>
* <edit-config> (operation="create")
* <edit-config> (operation="replace")
* <edit-config> (operation="merge")
* <edit-config> (operation="delete")

The typical workflow when using the network CLI in NSO is as follows:

* All changes are initially made to a (logical) copy of the NSO database of configurations
* Changes can be viewed and verified by the network operator prior to committing them
* The changes are committed, meaning that the changes are copied to the NSO database and pushed out to the network. Changes that violate integrity constraints or network policies will not be committed. The changes to the devices are done in a holistic distributed atomic transaction, across all devices in parallel

Changes either succeed and remain committed or fail and are rolled back as a whole, returning the entire network to the uncommitted state.

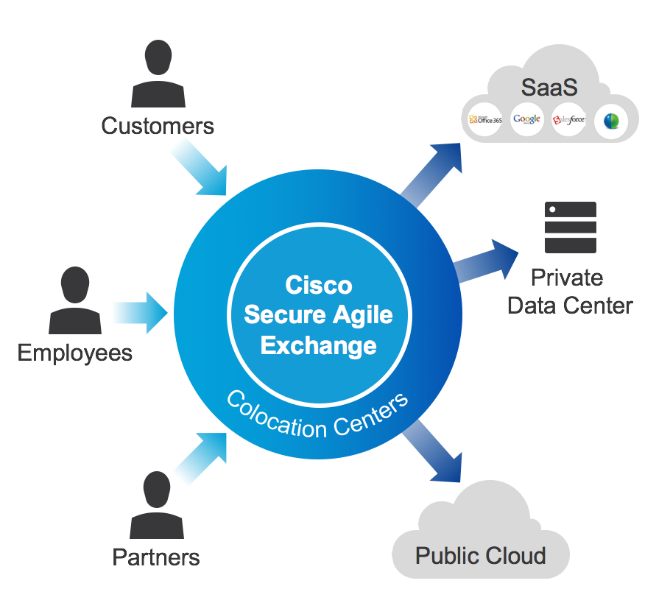
# SAE Solution Overview

The following chapter gives an overview of the SAE solution.

## Introduction

Enterprise is moving from the data center-centric network model to Public Cloud or hybrid models. Usual east-west traffic can be hopped off at a carrier-neutral facility to get better performance from Cloud providers. Company security policy needs to be implemented before traffic is handed over to the Provider network. Hence the centralized Datacenter based DMZ needs to be distributed to many CNF facilities.

Figure 1. Cisco Secure Agile Exchange Overview



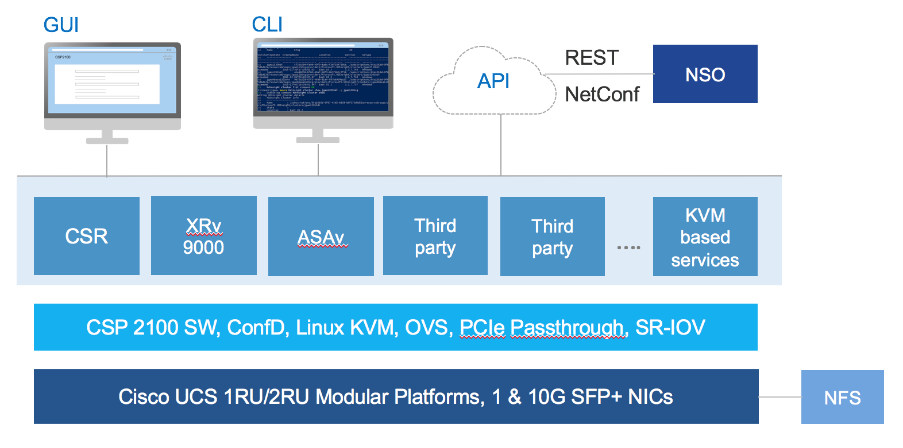
## Architecture

Secure Agile Exchange is a finely crafted and fully designed solution of deploying DMZ in CNF facility. SAE is an open orchestrated NFV platform that does not require the operational complexity of OpenStack or software switching overlays yet delivers on the benefits of virtualization at scale.

The SAE solution involves building a service chain using the following components:

* Cloud Services Platform (CSP) 2100
* Cisco VNFs such as CSR 1000V, ASAv, NGFWv, as well as third-party services such as the Avi Networks load balancer
* Nexus 9000 Series

Figure 2: The SAE Solution



It's important to have solid orchestrater and Automation Engine and flexible circuit consumption when transiting to VNF's. SAE Core function pack is multi-domain orchestration and automation engine to bring up SAE sites. Single SAE application can orchestrate multiple consumers and providers service chains on Multiple CSP, Cisco Nexus 9000 Series Switches.

Service chains are structured string of service nodes or NFV's. Service chains are designed based on business policy for the different type of traffic like Trusted, Untrusted or semi-trusted traffic. Switch redundancy is provided by VPC on Nexus 9000 and NFV redundancy is provided at VNF level. SRIOV on CSP2100 is used in most of the designs for data traffic due to better performance over OVS interface. High-performance Nexus 9000 can be used in standalone mode or Spine/Leaf mode depending on VxLAN requirement.

NSO Core function pack deploy half or full chain depending on the requirements. Multiple Consumer /Provider half service chains can be orchestrated and stitched together. This dynamic and flexible circuit stitching will help to connect the consumer to multiple cloud providers.

Virtualization and automation through NSO SAE Core function pack help to migrate DMZ to co-location. Distributed policy orchestration can be centralized from multidomain Network Service orchestrater helps to reduce the cost of deployment and improve the speed of adoption.

# SAE Core Function Pack Overview

The following chapter gives an overview of the SAE Core Function Pack (FP).

## Introduction

The SAE Core Function Pack (FP) is a collection of multiple re-usable NSO packages organized in layers.

This FP supports on-boarding of both physical and virtual infrastructures. Services can be built on top of the FP using the strongly typed service models. The FP supports pushing Day1 configurations to the VNF via generic key-value pair paradigm. To deploy the VNFs, NSO uses VNF descriptors that are compliant with the ETSI-MANO specifications.

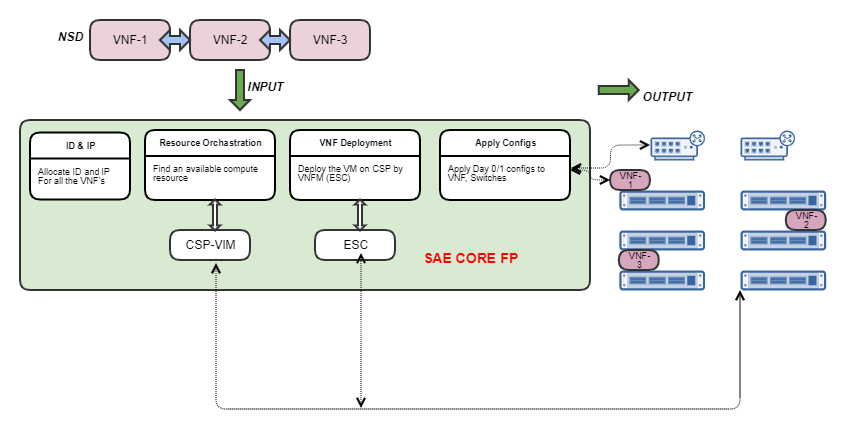
Cisco Secure Agile Exchange Solution enables enterprises to quickly and securely interconnect users to applications by virtualizing the network edge and extending it to colocation centers. Large enterprises can benefit by deploying the solution themselves working with colocation centers like, Barclays or Colo providers like Equinix can deploy SAE in muti-tenant mode and provide SAE services to its customers.

NSO SAE Core Function Pack can help orchestrate service chains on managed hardware.

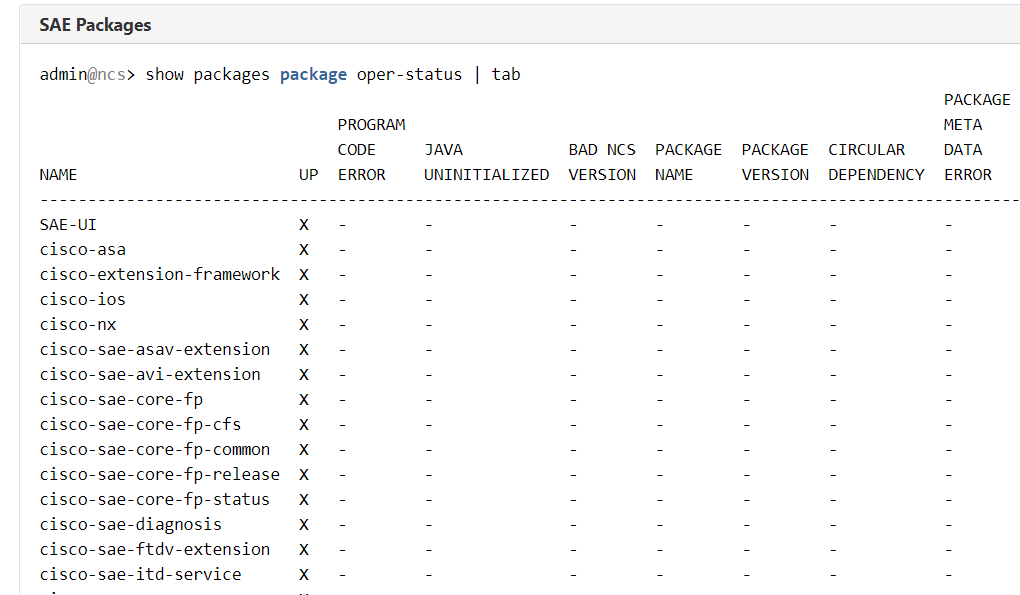
## Architecture

The following diagram illustrates SAE Core Function Pack Architecture.

Figure3: SAE Architecture



### NSO Packages for SAE



## Supported Hardware

The following table lists the hardware that works with SAE.

| **Hardware** | **Type** | **Comments** |
| --- | --- | --- |
| N9K | Switch |  |
| CSP | Compute Node |  |
| ESC | VNFM |  |
| CSP | VIM |  |

## Supported Software

The following table lists VNFs that are supported.

|  |
| --- |
|  |
| **Hardware** | **Type** | **Comments** |
| ASAv |  |  |
| AVI | LoadBalancer |  |
| CSRv |  |  |
| F5 | LoadBalancer |  |
| Fortinet |  |  |
| FTDv |  |  |
| Palo Alto Networks | Firewall |  |

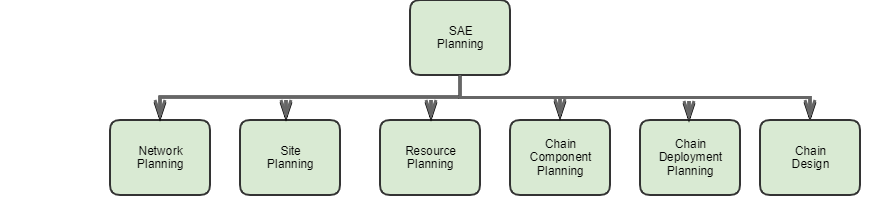
# SAE Service Design

The following chapter gives an explanation of the SAE service design.

## Introduction

An SAE solution consists of multiple components and resources.

Figure 4. SAE Service Design



Each component or resource needs to be planned to create an SAE service.

The table below shows the order in which a service is planned in SAE. Perform each of the following.

| **Component** | **Details** |
| --- | --- |
| Network Planning | The compute nodes, switches, network topology, VNF manager, VIM , etc are decided as part of this. |
| Site Planning | Providers and tenant are required,. Catalog for the given provider / tenant is decided as part of this planning. |
| Resource Planning | The resources required for this service, like internal IP address, VLAN ID, VNID etc. |
| Chain Component Planning | Users need to know what VNF's and NSD's arebeing used for the service. |
| Chain Deployment Planning | This will cover if VNF to be intangible, EPGW in HA or non HA, Service Chain in HA or non HA, etc. |
| Chain Design | The providers and consumers and the service chain connecting both are designed as part of this. |

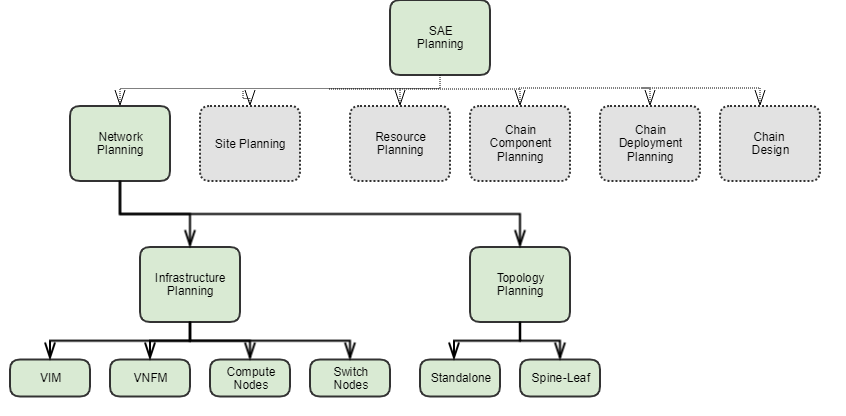
## Network Planning

The following section gives an explanation of SAE network planning.

### Overview

As part of SAE Infrastructure Planning , user need to decide what kind of switches, compute nodes, VIM, topology, VNFM etc.

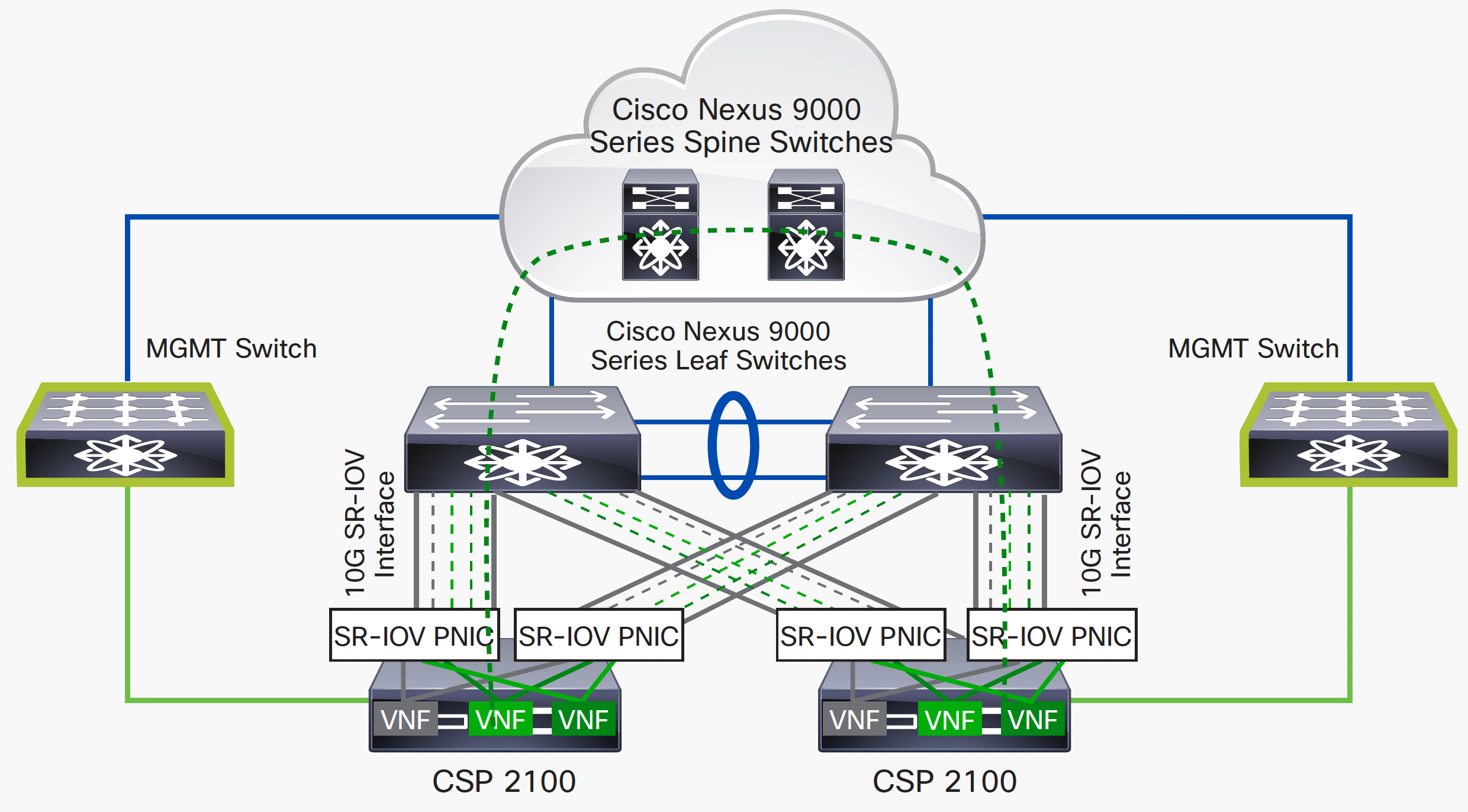
Figure 5. SAE Network Planing



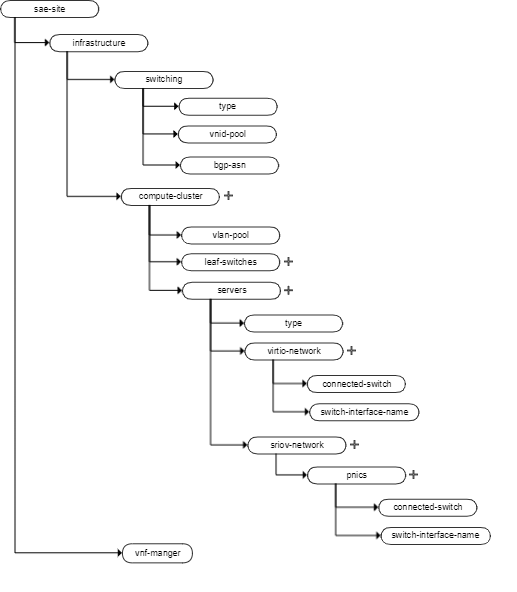
### Network Infrastructure

Infrastructure for the site represents the network topology of servers (CSP) and switches.

Services like gateway-vnf, service-chain, and stitching-service use infrastructure which is defined for the site.



Infrastructure is the topology on which services are created. Infrastructure will have list the list of CSP devices, VNF Manager, switches and their network structure defined. All the services for that particular site will use these infrastructure devices and network and create services based on it. Infrastructure defines the following elements.



The following table lists the components required for SAE infrastructure.

| **Component Name** | **Details** |
| --- | --- |
| compute cluster | This represents VPC of the site, a site can have multiple VPC. |
| switch type | Type of the switch which is used, possible values are below   * n9k-spine-leaf * n9k-switch-pair |
| vnid-pool | Resource pool id for vnid. |
| bgp-asn | BGP AS number for switches to use |
| leaf-switches | This represents switching devices (example N9K devices), maximum of 2. |
| servers | This represents server devices on which VM(VNF) will be spawned. ( example: CSP devices). |
| servers/type | This represents/refer to the defined type of the device model (pnics , sriov networks, virtio-networks) in sae-catalog |
| vnf-manager | this represents the Manager device for VNF, only one VNF manager is allowed for entire site. VNF manager is repressible for managing deployment and vm states. |
| sriov-network | this represents the sriov network configuration, the name of network should be the pnic id on CSP, which is connected to switch, and switch-interface-name should be the interface on switch, which is connected to the csp pnic. |
| virtio-network | this represents virtio network configuration, the name of the network should be port-channel name on csp , pnics in this model are members of the port-channel on CSP, which are connected to switch, switch interface name should be the port-channel name on switch. |

#### Infrastructure model

admin@ncs% show cisco-sae-core-fp-cfs:sae-site infrastructure  
cisco-sae-core-fp-cfs:sae-site cisco {  
    infrastructure {  
        compute-clusters cluster1 {  
            vlan-pool site-1\_Vlan\_pool;  
            leaf-switches N9k-Leaf-1;  
            leaf-switches N9k-Leaf-2;  
            servers CSP-225 {  
                type csp-type-cluster1;  
                virtio-network TestPC {  
                    pnics enp4s0f0 {  
                        connected-switch      N9k-Leaf-2;  
                        switch-interface-name port-channel121;  
                        bandwidth             1024;  
                    }  
                    pnics enp4s0f1 {  
                        connected-switch      N9k-Leaf-1;  
                        switch-interface-name port-channel121;  
                        bandwidth             1024;  
                    }  
                }  
                sriov-networks enp4s0f2 {  
                    connected-switch      N9k-Leaf-2;  
                    switch-interface-name Ethernet1/3;  
                    bandwidth             1024;  
                }  
                sriov-networks enp4s0f3 {  
                    connected-switch      N9k-Leaf-1;  
                    switch-interface-name Ethernet1/4;  
                    bandwidth             1024;  
                }  
            }  
            servers CSP-226 {  
                type csp-type-cluster1;  
                virtio-network TestPC {  
                    pnics enp4s0f0 {  
                        connected-switch      N9k-Leaf-1;  
                        switch-interface-name port-channel120;  
                        bandwidth             1024;  
                    }  
                    pnics enp4s0f1 {  
                        connected-switch      N9k-Leaf-2;  
                        switch-interface-name port-channel120;  
                        bandwidth             1024;  
                    }  
                }  
                sriov-networks enp4s0f2 {  
                    connected-switch      N9k-Leaf-1;  
                    switch-interface-name Ethernet1/3;  
                    bandwidth             1024;  
                }  
                sriov-networks enp4s0f3 {  
                    connected-switch      N9k-Leaf-2;  
                    switch-interface-name Ethernet1/4;  
                    bandwidth             1024;  
                }  
            }  
        }  
        vnf-manager ESC0 {  
            device-on-boarding nso-unmanaged;  
            vnfm-device-name   ESC0;  
        }  
    }  
}

### VNFM

Configure the SAE Site Infrastructure VNF Manager.

Use the following command syntax to create the SAE Site VNF Manager.

#### Usage

$ ncs\_cli -u admin  
  config  
    set sae-site <site\_name> infrastructure vnf-manager <vnf\_manager\_name> device-on-boarding nso-unmanaged vnfm-device-name <device\_name>  
  commit

#### CLI Example

# Example:  
# Define the management network and sriov data network in catalog according to topology in Figure 1 above  
# Configure compute cluster: CSP servers in cluster, CSP SRIOV pnic connection to N9K interface,  
   
$ ncs\_cli -u admin  
config  
   
set sae-site cisco infrastructure vnf-manager ESC0 device-on-boarding nso-unmanaged vnfm-device-name ESC0  
commit

### VIM

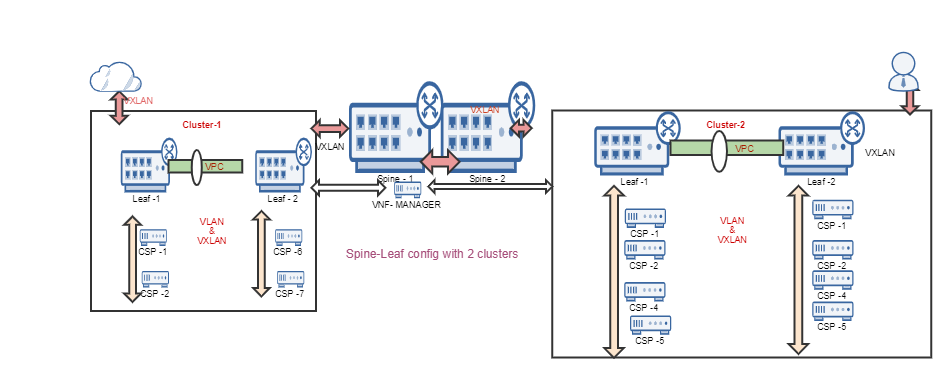
CSP is the compute node and acts as a VIM.

### Topology

Currently SAE supports two types of network topology:

| **Network** | **Details** |
| --- | --- |
| Single Cluster | Standalone with a single VPC Pair. SAE treats this network as a single cluster. So all internal traffic uses VLAN |
| Multi Cluster | Spine Leaf configuration with multiple VPC Pair. Each VPC pair is one cluster.All internal traffic in a cluster user VLAN. Traffic across clusters uses VXLAN via spine switches. |

Figure 6. spine-leaf SAE VPC EPGW Service



#### Standalone Configuration

A typical standalone network infrastructure consists of one VPC pair which includes the following:

| **Component** | **Details** |
| --- | --- |
| N9k Switch | Two 9K switches with VPC configuration |
| CSP | The compute nodes which connects to both the N9k switches |
| ESC | This is the VNFM which manages the VNF and will be running in one of the CSP's |

To create a service in the standalone VPC, the resources listed below are required in SAE core FP.

| **Component** | **Details** |
| --- | --- |
| VLAN pool | This pool will be used to find an available VLAN used for inter VNF communication |
| Internal IP Pool | This pool will be used for allocating IP's for the VNF connection points |
| Management IP Pool | This pool will be used for allocating Management IP's for the VNF |

#### Spine-Leaf Configuration

A typical spine-leaf configuration consists of one than VPC pair. All the resources and hardware required for standalone is applicable for spine-leaf also.

An extra resource required for spine-leaf configuration is the VNID pool listed below:

| **Component** | **Details** |
| --- | --- |
| VNID pool | This is the VNID which will be used for traffic going through spine to reach the other cluster |

#### Standalone commands

// Set the switching type for standalone. default is n9k-switch-pair  
admin@ncs% set sae-site sae-site-1 infrastructure switching type n9k-switch-pair  
   
// Show the switch type  
admin@ncs% show cisco-sae-core-fp-cfs:sae-site sae-site-1 infrastructure switching type  
type n9k-switch-pair;

#### Spine-leaf Commands

// create the vnid pool for spine-leaf

admin@ncs% set resource-pools id-pool sae-vnid-pool range start 5000 end 100000

// show the vnid pool

admin@ncs% show resource-pools id-pool sae-vnid-pool range

start 5000;

end 100000;

// For Spine-leaf set the switch type as n9k-spine-leaf

admin@ncs% set sae-site sae-site-1 infrastructure switching type n9k-spine-leaf

// Show the switch type

admin@ncs% show cisco-sae-core-fp-cfs:sae-site sae-site-1 infrastructure switching type

type n9k-spine-leaf;

// set the vnid pool for spine-leaf. This option will show only for spine leaf switch type

admin@ncs% set sae-site sae-site-1 infrastructure switching vnid-pool sae-vnid-pool

### Packaging, Installation and Versioning

The Infrastructure model is available with following cfp package:

cisco-sae-core-fp-cfs

Packages dependency:

* cisco-sae-core-fp-common
* cisco-sae-core-fp
* core-fp-common

### Troubleshooting and Diagnostics

If you are not sure about the network topology to use, you can run auto discovery, and the topology will be discovered and the infrastructure will be updated.

Please see the section *SAE Auto Discovery* for more information on how to use the feature.

To check whether the network links are connected properly, the following diagnostic commands can be used:

show diagnosis cfp sae-diagnosis infrastructure <CSP> links sriov-network

show diagnosis cfp sae-diagnosis infrastructure <CSP> links virtio-network

For more details on diagnostic commands on infrastructure devices please see the SAE diagnosis package.

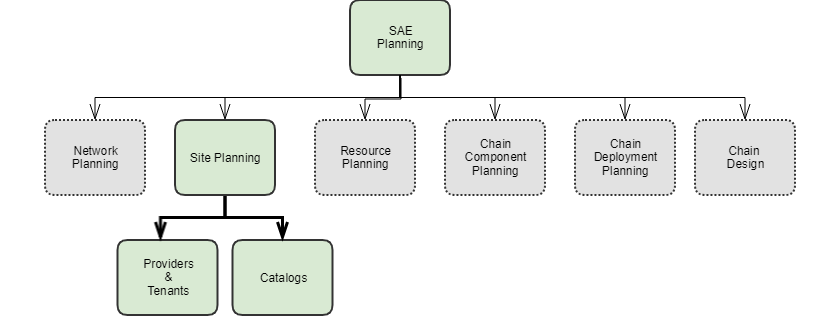
## Site Planning

The following section gives an explanation of SAE site planning.

### Overview

SAE site planning requires some mandatory information.

| **Component** | **Details** |
| --- | --- |
| Site | This represents a logical location / site. All the SAE services will be created under a site. |
| Providers and Tenant | Under Each site there will be multiple providers and each provider can have multiple tenants. |
| Catalogs | Catalog holds information about the type of CSP's. NSD, VND deployment profiles and other details. |



### Site

SAE-site allows the user to define infrastructure and instantiate services on the infrastructure. Infrastructure definition allows users to group N9K devices (that are in vpc pair) and associated CSPs together as a cluster. Such cluster(s) will have one ESC vnf-manager (VNFM) which monitors and maintains VNF lifecycle. SAE-site provides the most flexible way of defining and instantiate services on the infrastructure.

* Gateway VNF can be defined as shared endpoint-gateway, connected to one or more external-end points. Then such shared-gateway can be attached to service-chain(s), allowing one or more service chains to share gateway vnf.
* Create half service chain (consumer/provider half-chain) and attach it with any other half service-chain to form an end-to-end chain. This allows the user to dynamically change the end-to-end by simply changing the stitching.
* Create an end-to-end service chain with no shared endpoint-gateway and no stitching.

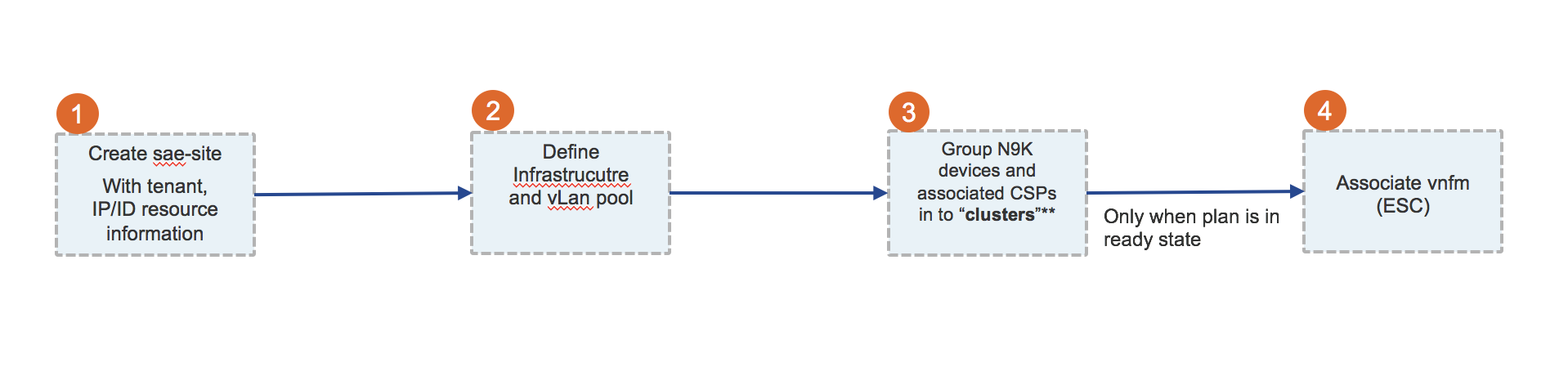
SAE cfp depends on certain validations and assumptions in deployment workflow to provide such flexibility.

**SAE Site create-workflow:**

A typical sae-site create workflow can be categorized in to "Environment creation" and "Service(s) creation":

Stage 1: Environment creation lists the workflow order for on-boarding sae-site infrastructure:

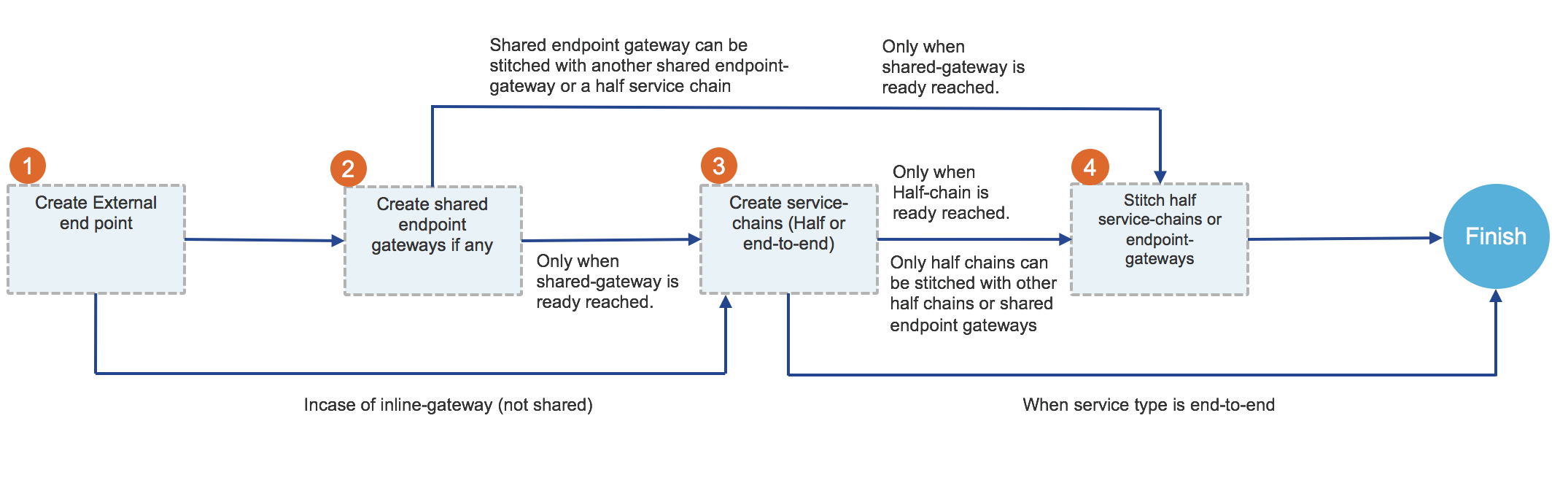
1. Create sae-site with tenant and IP/ID resource information.
2. Define infrastructure, under the sae-site.
3. Add clusters (N9K devices that are in vpc pair and associated CSPs) to the infrastructure
4. Associate vnfm (ESC) with the infrastructure.

Figure 7. Create SAE Site

Stage 2: Service(s) creation lusts the workflow order for creating service chains on the environment.

1. Create external-end-points service connects to.
2. Create any shared endpoint gateways used in the service.
3. Create service-chain.
4. If step 2 doesn't define end-to-end service, a user can create an end-to-end using stitching.

Figure 8. Create Shared Endpoints



**NOTE:** Service means connecting one external-end to another external-end (end-to-end).

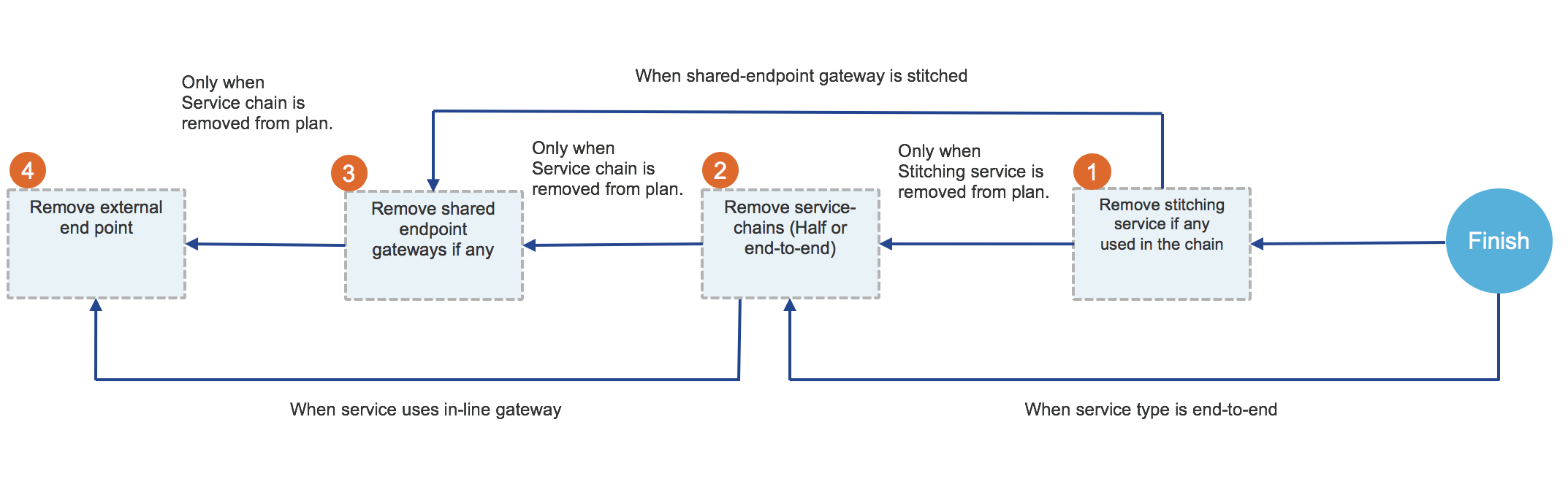
**SAE Site delete-workflow:**

Form create we understand that service(s) is/are dependent on infrastructure. The order of deletion starts from services and ends with infrastructure.

Stage 1: Clean up any services deployed on the infrastructures.

1. Remove stitching service if end-to-end service is formed by stitching two half/endpoint-gateway services.
2. Remove service chain(s) service.
3. Remove any shared-endpoint-gateway.
4. Remove external-end-point definition.

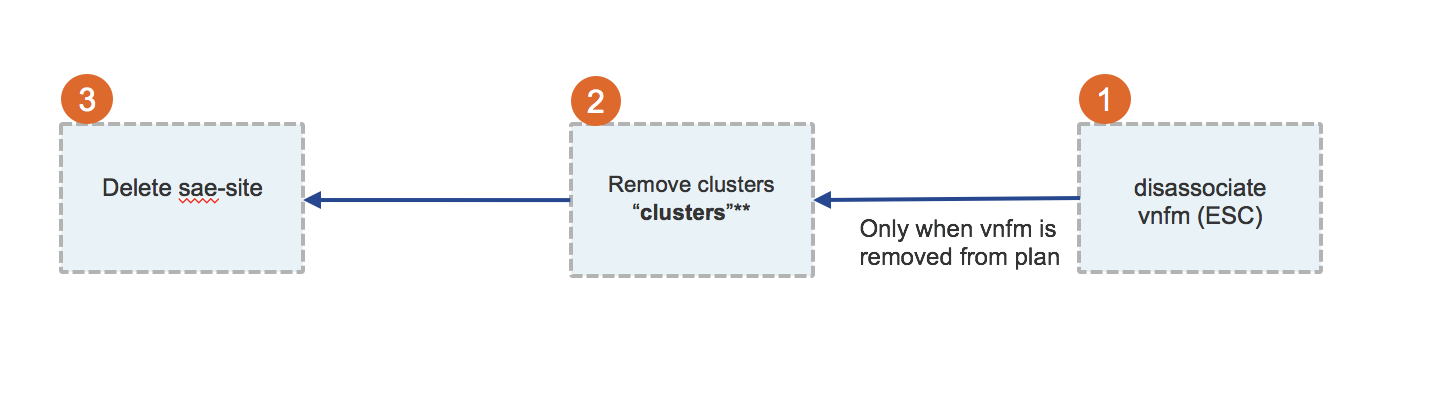
Figure 9. Delete SAEWorkflow



Stage 2: Delete sae-site (only after all the services deployed are deleted)

* Remove vnfm associated with the infrastructure.
* Delete clusters from infrastructure.
* Delete sae-site.

Figure 10. Delete SAE Site



### Prerequisites/Dependencies

Dependencies can again be categorized based on what stage of SAE-site creation we are in.

For infrastructure:

* sae-catalog with CSP deployment details.
* sae-provider/tenant definition.

For service deployment:

* VNFD definition.
* NSD definition.
* VNFD-Deployment definition for VNFs in service (sae-catalog).
* NSD-Deployment definition (sae-catalog).

### Providers and Tenants

The SAE provider and tenant model is provided under the CFS model. There are mandatory inputs required, and you have to set the sae-provider model before creating the sae-site.

**Yang Model**

The SAE-provider is available under cisco-sae-core-fp-cfs module, user has to set provider name & tenant name under this model.

module: cisco-sae-core-fp-cfs  
    +--rw sae-provider\* [name]  
    |  +--rw name                    string  
    |  +--rw sae-provider-catalog    -> /sae-catalog/name  
    |  +--rw sae-tenant\* [name]  
    |  |  +--rw name                  string  
    |  |  +--rw sae-tenant-catalog?   -> /sae-catalog/name

### Prerequisites/Dependencies

Refer to the below prerequisites/dependencies steps to create sae provider and tenants:

* Make sure sae-catalog should be create before create provider & tenant
* For sae-site creation, sae-provider & sae-tenant are mandatory inputs
* user can have multiple sae-sites where sae-provider play key-role to select sae-catalog by providing tenant & provider details
* In sae-provider model, if there is no catalog mentioned under sae-tenant then SAE CFP will select catalog under provided under sae-provider

**Usage with Sample Payloads**

admin@ncs% show cisco-sae-core-fp-cfs:sae-provider | display xml  
<config xmlns="[http://tail-f.com/ns/config/1.0"](http://tail-f.com/ns/config/1.0)>  
  <sae-provider xmlns="[http://com/cisco/nso/sae-core-fp-cfs"](http://com/cisco/nso/sae-core-fp-cfs)>  
    <name>sae-provider-admin</name>  
    <sae-provider-catalog>sae-basic</sae-provider-catalog>  
    <sae-tenant>  
      <name>sae-tenant-admin</name>  
      <sae-tenant-catalog>sae-basic</sae-tenant-catalog>  
    </sae-tenant>  
  </sae-provider>  
</config>

**Payloads Validation**

SAE CFP have some validation as below

* Provider and tenants are mandatory inputs in sae-site & sae-provider model
* Catalog should be mentioned under sae-provider
* If there is no catalog mentioned under sae-tenant then SAE CFP will select catalog under provided under sae-provider

### Troubleshooting and Diagnostic

| **Error** | **Possible Reason** | **Troubleshooting** |
| --- | --- | --- |
| SAE site validation error | sae-catalog is missing in sae-provider model | mention catalog name under sae-provider |

### Catalog

SAE Catalog has main threee list nodes:

1. CSP
   1. CSP node captures the csp network details like virtio, sriov and management network names.
   2. It also captures the details of day1-template to be applied on CSPs
2. VNFD Deployment
3. NSD Deployment

**Yang Model**

1. csp node
   1. CSP node in catalog is used in sae-site infrastructure.
   2. sae-catalog/csp/name referred by sae-site/infrastructure/compute-clusters/servers/type
   3. virtio network added in catalog 'sae-catalog/csp/virtio-networks' are referred in sae-site infrastructure at 'sae-site/infrastructure/compute-clusters/servers/virtio-networks'
   4. sriov network added in catalog 'sae-catalog/csp/sriov-networks' are referred in sae-site infrastructure by 'sae-site/infrastructure/compute-clusters/servers/sriov-networks'
2. vnfd-deployment
3. nsd-deployment

**SAE Catalog Yang Model**

   +--rw sae-catalog\* [name]  
|  +--rw name               string  
|  +--rw var\* [name]  
|  |  +--rw name             variableName  
|  |  +--rw (valtype)?  
|  |     +--:(val)  
|  |     |  +--rw val              string  
|  |     +--:(encrypted-val)  
|  |        +--rw encrypted-val    tailf:aes-cfb-128-encrypted-string  
   
|  +--rw csp\* [name]  
|  |  +--rw name              string  
|  |  +--rw day1-config\* [name]  
|  |  |  +--rw name            string  
|  |  |  +--rw cfg-template    string  
|  |  |  +--rw var\* [name]  
|  |  |     +--rw name             variableName  
|  |  |     +--rw (valtype)?  
|  |  |        +--:(val)  
|  |  |        |  +--rw val              string  
|  |  |        +--:(encrypted-val)  
|  |  |           +--rw encrypted-val    tailf:aes-cfb-128-encrypted-string  
|  |  +--rw virtio-network\* [name]  
|  |  |  +--rw name    network-name  
|  |  +--rw sriov-networks\* [name]  
|  |  |  +--rw name    network-name  
|  |  +--rw mgmt-network?     network-name  
   
|  +--rw nsd-deployment\* [name]  
|  |  +--rw name                     string  
|  |  +--rw nsd?                     -> /nfvo:nfvo/nsd/id  
|  |  +--rw nsd-deployment-flavor    -> deref(../nsd)/../nfvo-rel2:deployment-flavor/id  
|  |  +--rw vnf-deployment\* [vnf-profile]  
|  |     +--rw vnf-profile        -> deref(../../nsd-deployment-flavor)/../nfvo-rel2:vnf-profile/id  
|  |     +--rw vnfd-deployment?   -> ../../../vnfd-deployment/name  
   
|  +--rw vnfd-deployment\* [name]  
|  |  +--rw name                  string  
|  |  +--rw vnfd                  -> /nfvo-rel2:nfvo/vnfd/id  
|  |  +--rw vdu                   -> /nfvo-rel2:nfvo/vnfd[nfvo-rel2:id = current()/../vnfd]/vdu/id  
|  |  +--rw mode?                 enumeration  
|  |  +--rw var\* [name]  
|  |  |  +--rw name             variableName  
|  |  |  +--rw (valtype)?  
|  |  |     +--:(val)  
|  |  |     |  +--rw val              string  
|  |  |     +--:(encrypted-val)  
|  |  |        +--rw encrypted-val    tailf:aes-cfb-128-encrypted-string  
|  |  +--rw intangible?           empty  
|  |  +--rw bootup-time?          int32  
|  |  +--rw recovery-wait-time?   int32  
|  |  +--rw vnf-authgroup?        -> /ncs:devices/authgroups/group/name  
|  |  +--rw vnc-password?         tailf:aes-cfb-128-encrypted-string  
|  |  +--rw ssh-authgroup?        -> /ncs:devices/authgroups/group/name  
|  |  +--rw day0\* [destination]  
|  |  |  +--rw destination    -> /nfvo-rel2:nfvo/vnfd[nfvo-rel2:id = current()/../../cisco-sae-core-fp-cfs:vnfd]/nfvo-rel2:vdu[nfvo-rel2:id = current()/../../cisco-sae-core-fp-cfs:vdu]/nfvo-rel2-esc:day0/destination  
|  |  |  +--rw (source)  
|  |  |  |  +--:(url)  
|  |  |  |  |  +--rw url?           inet:uri  
|  |  |  |  +--:(data)  
|  |  |  |     +--rw data?          tailf:aes-cfb-128-encrypted-string  
|  |  |  +--rw var\* [name]  
|  |  |     +--rw name             variableName  
|  |  |     +--rw (valtype)?  
|  |  |        +--:(val)  
|  |  |        |  +--rw val              string  
|  |  |        +--:(encrypted-val)  
|  |  |           +--rw encrypted-val    tailf:aes-cfb-128-encrypted-string  
|  |  +--rw day1-config\* [name]  
|  |  |  +--rw name            string  
|  |  |  +--rw cfg-template    string  
|  |  |  +--rw var\* [name]  
|  |  |     +--rw name             variableName  
|  |  |     +--rw (valtype)?  
|  |  |        +--:(val)  
|  |  |        |  +--rw val              string  
|  |  |        +--:(encrypted-val)  
|  |  |           +--rw encrypted-val    tailf:aes-cfb-128-encrypted-string  
|  |  +--rw vnf-type?             cisco-extension-framework:extension-group-t  
|  |  +--rw controller?           -> /ncs:devices/device/name  
|  |  +--rw ftdv-settings  
|  |  |  +--rw license\_caps\*   string  
|  |  |  +--rw accessPolicy?   string  
|  |  +--rw asav-ha-settings  
|  |     +--rw ha-role?                     enumeration  
|  |     +--rw failover-connection-point\*   -> deref(../../vnfd)/../nfvo-rel2:vdu/internal-connection-point-descriptor/external-connection-point-descriptor  
|  +--rw timestamp?         string  
|  +---x custom-redeploy  
|     +--ro output  
|        +--ro status?   string  
|        +--ro error?    String

**SAE Catalog: CSP**

<config xmlns="<http://tail-f.com/ns/config/1.0>">  
  <sae-catalog xmlns="<http://com/cisco/nso/sae-core-fp-cfs>">  
    <name>cisco-catalog</name>  
    <csp>  
      <name>[csp-2100-2.2.3.32](https://confluence-eng-sjc1.cisco.com/conf/display/NSOUS/csp-2100-2.2.3.32)</name>  
      <day1-config>  
        <name>ntp</name>  
        <cfg-template>csp-ntp-server</cfg-template>  
        <var>  
          <name>NTP\_SERVER</name>  
          <val>[172.28.112.11](https://confluence-eng-sjc1.cisco.com/conf/display/NSOUS/172.28.112.11)</val>  
        </var>  
      </day1-config>  
      <virtio-network>  
        <name>virt-A</name>  
      </virtio-network>  
      <sriov-networks>  
        <name>sriov-1</name>  
      </sriov-networks>  
      <sriov-networks>  
        <name>sriov-2</name>  
      </sriov-networks>  
      <mgmt-network>mgmt-A</mgmt-network>  
    </csp>  
  </sae-catalog>  
</config>

## Resource Planning

SAE resource planning consists of the following considerations.

### Overview

An SAE core FP Solution is composed of services and multiple resources.

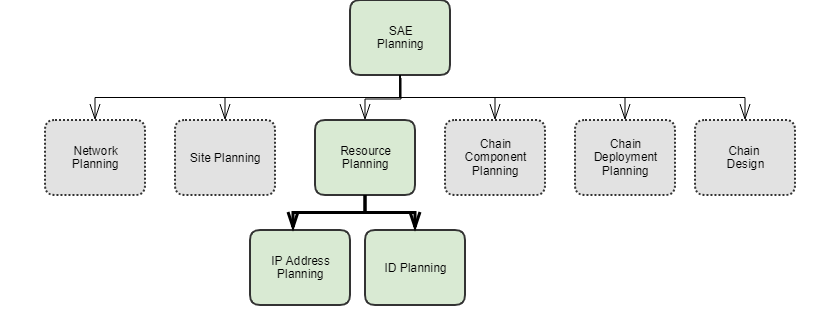
These resources can be managed independently or managed along with SAE itself.

Before creating the service the designer needs to plan and provision the required resources for that service.

The diagram below shows the components as part of resource planning.

This section covers ID and IP pool planning.

Figure 11. Resource Planning



Some of the resources which needs to be designed before using service are:

| **Resource Name** | **Details** |
| --- | --- |
| resource-pools | This is the ID and IP pool generator . |
| resource-zone | This component will group the resources together which can be used for placement. |

Before creating the service user should plan the resources required for the service.

For example, with IP pools if a VNF is deployed in HA mode, then for each connection point an internal IP need to be allocated. That way, the pool can accommodate all the requests.

### ID and IP Pool

NSO has package called *resource-manager* that can be augmented with different resource pools, for example id numbers and ip addresses.

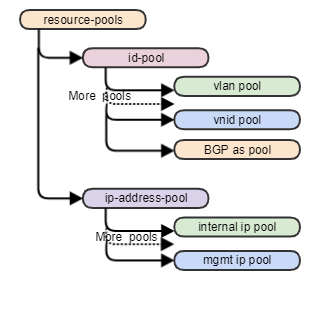
Resource manager includes

* id-pool
* ip-address-pool

The purpose of the pool is generate a unique id for each allocation request. We want to have different allocators for different resources.

| **Pool Name** | **Examples** |
| --- | --- |
| ip-address-pool | * Internal ip pool. This ip address pool can be used for internal ip allocations for data traffic * mgmt ip pool . This ip address pool can be used for management purpose. |
| id-pool | * vlan pool. This is an id pool which can be used for generating and managing unique vlans * vnid pool. This is also an id pool to generate a unique vxlan number |

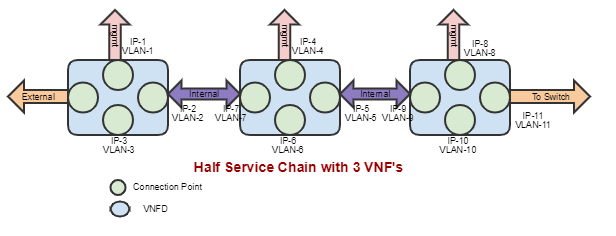
Figure 11. id and ip pools



A sample half service chain with three VNFD's is shown below.

* Each VNF has four connection points.
* One end of NSD is connecting to external entity.
* Other end of NSD is connecting to switch fabric.
* User input is only provided for the external entity VNF.
* External IP and VLAN for one connection point is provided by user.
* For Traffic to continue SAE need to allocate internal IP and internal VLAN for each of the connection point.
* The IP's and VLAN will be generated from the vlan and IP pool.
* For management IP, the IP need to come from a different subnet.

Figure 12. Half Service Chain



So for this NSD IP and vlan is required for a total of 11 connection points.

| **Pool** | **Type** | **Details** |
| --- | --- | --- |
| sae-vlan-pool | ID-pool | 11 VLAN needs to be allocated from the pool |
| mgmt-network-pool | IP-address-pool | 3 mgmt IP need to be allocated from a specific subnet |
| local-network-pool | IP-address-pool | 8 Internal IP need to be allocated from a specific subnet |
| user-as-pool | ID-pool | Pool for the BGP As number. This is optional as there is a default as pool created |
| sae-vnid-pool | ID-pool | If the topology is Spine-Leaf , then user need to create a separate ID pool for VXLAN |
| default-as-pool | ID-pool | This is the default as pool generated by core FP . This will be overridden by custom as pool |

The internal allocations are done automatically by SAE core FP and users doesn't have to make any explicit allocation requests.

Each pool has an allocation list where services are expected to create instances to signal that they request an allocation.

Request parameters are stored in the request container and the allocation response is written in the response container.

### Usage with Sample Payloads

**Create Internal IP Pool**

admin@ncs% set resource-pools ip-address-pool local-network-pool subnet 192.168.2.0 24

**Create Management IP Pool**

admin@ncs% set resource-pools ip-address-pool mgmt-network-pool subnet 10.0.0.0 24

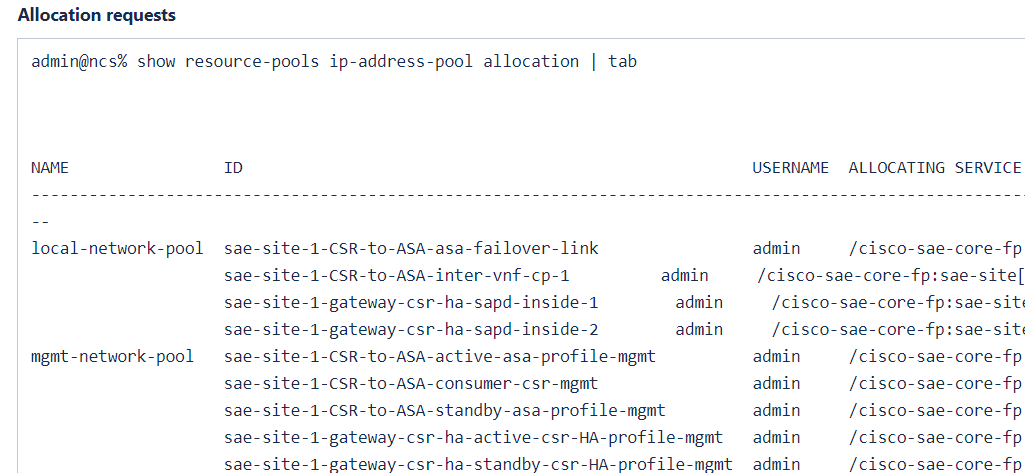
**Show IP Address Pools**

admin@ncs% show resource-pools ip-address-pool subnet  
ip-address-pool local-network-pool {  
    subnet 192.168.2.0 24;  
}  
ip-address-pool mgmt-network-pool {  
    subnet 10.0.0.0 24;  
}

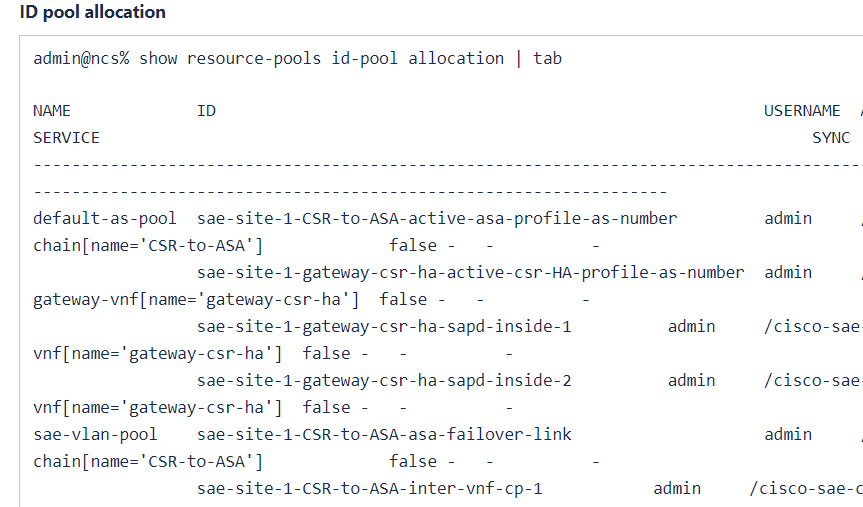
### Verification

The allocation request can be verified by a show command on ip-address-pool and id-pool allocation

**Allocation requests**

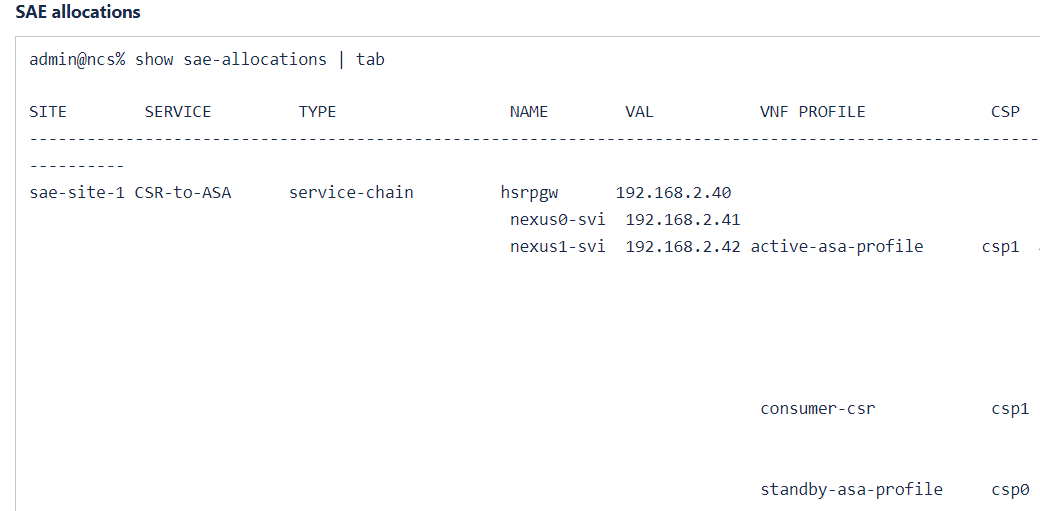
****

**ID pool allocations**

****

The actual IP and VLAN allocated to each connection point can be displayed using show sae-allocations.

**SAE allocations**



### Troubleshooting and Diagnostic

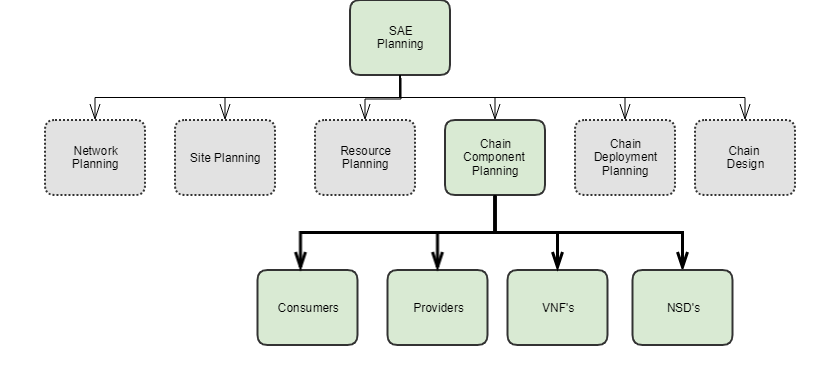
| **Error** | **Possible Reason** |  | **Trouble shooting** |
| --- | --- | --- | --- |
| No IP address available | The IP pool might have exhausted. |  | * Check the subnet in pool. * Check the allocation requests * Make sure the range can support all connection points |
| No ID available | The ID pool might have exhausted. |  | * Check the range in pool. * Check the allocation requests * Make sure the range can support all requests |

## Service Component Planning

SAE service component planning consists of the following considerations.

### Overview

Figure 12. SAE Service Component Planning



### External Endpoints

External-end-points represent computer networks that are external to the SAE site. They represent the source or destination of network traffic flowing through the SAE site.

#### Consumers

Consumers are external end points where the traffic is consumed , they represent clients or users of an application, generally networks where policy decisions are made based on the source IP/port information in the IP packet.

#### Providers

Providers are external end points where the traffic is provided. They represent servers of an application, which are generally networks where policy decisions are made based on the destination IP/port information in the IP packet.

### Virtualized Network Function Descriptor (VNFD)

How do you capture VNF requirements and what information should be captured so that a software system can deploy and manage it?

VNFD is the answer to these questions. It defines the structure and the guidelines as to what information should be captured. VNFD is the basic building block required to deploy a VNF based service in SAE. This section explains the VNFD model and how to create a VNFD. However, please note it is very specific to SAE and covers topics related/used in SAE only. For a more detailed explanation of VNFD please refer to ETSI MANO.

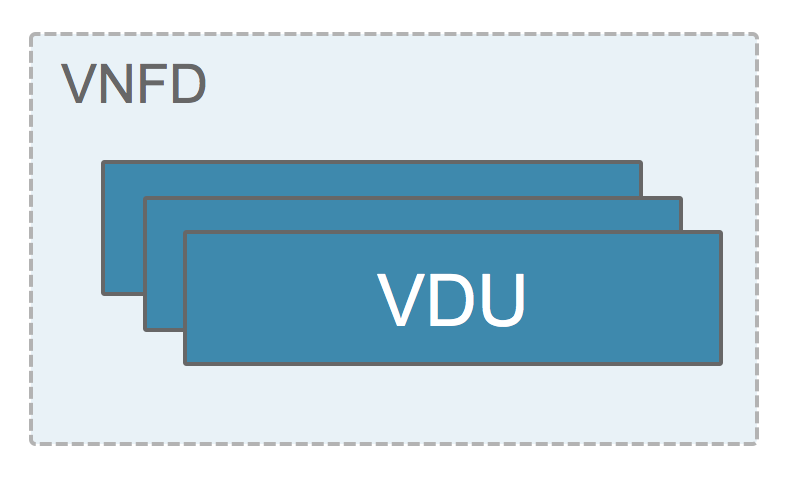
#### Description

A VNF Descriptor (VNFD) is a construct which captures different deployment variations of a VNF. The Virtualization Deployment Unit (VDU) is a deployment template that captures the description of the deployment and operational behavior of a single VNF component, or the entire VNF if it was not componentized in components (this document explains VDUs representing an entire VNF).

A VDU represents a VNF in terms of its requirements:

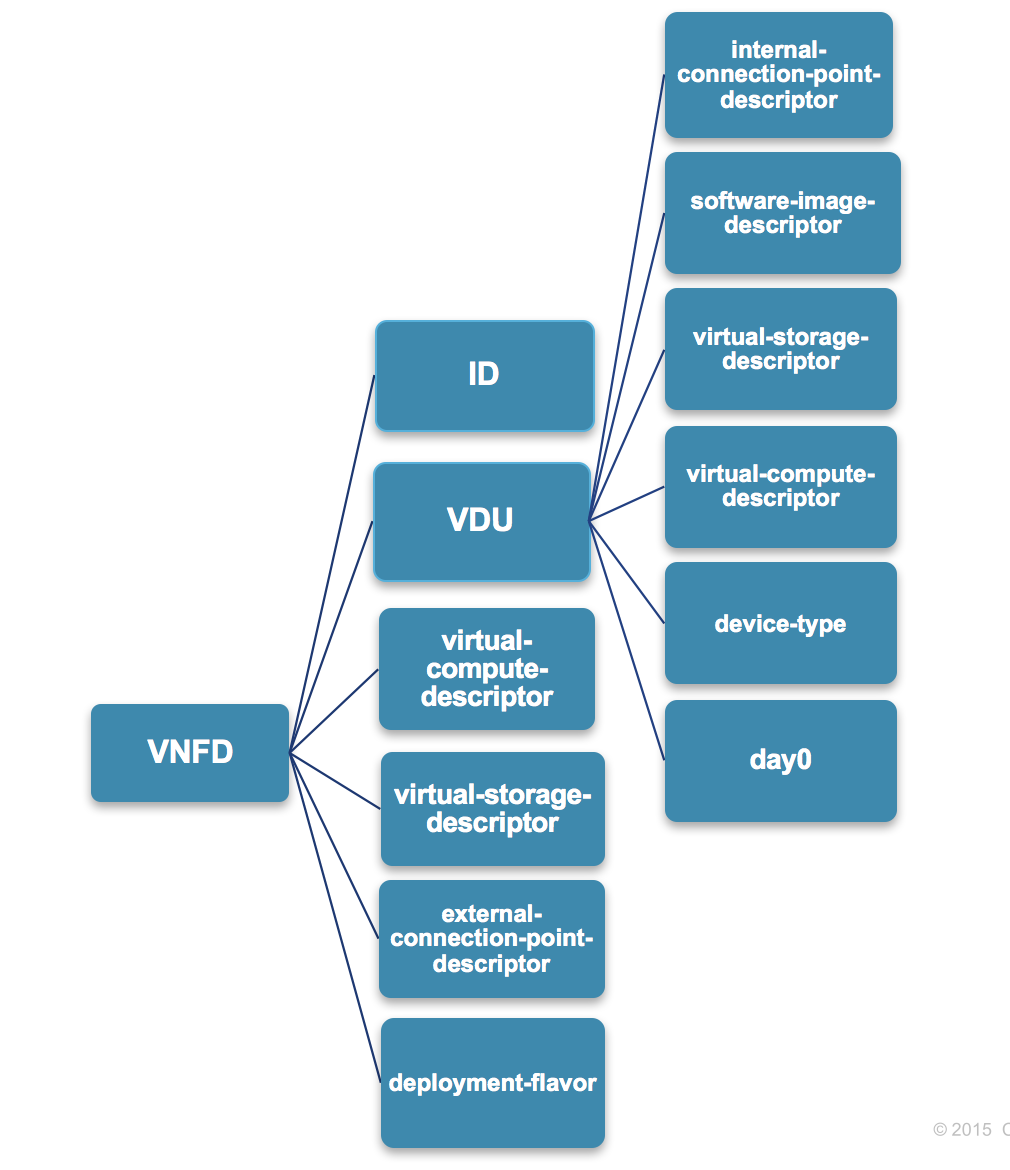
* vNic requirements.
* Virtual compute and storage requirements.
* Software image.
* Day-0 destination
* VNF type (CSR, ASA , and so forth) if managed by NSO.

Figure 13. VDU Construction



The VNFD data structure at a high-level is shown below:

Figure 14. VNFD Data Structure



#### pyang tree representation

vnfd\* [id]

| +--rw id

| +--rw vdu

| | +--rw id

| | +--rw internal-connection-point-descriptor

| | +--rw virtual-compute-descriptor -> ../../virtual-compute-descriptor/id

| | +--rw virtual-storage-descriptor -> ../../virtual-storage-descriptor/id

| | +--rw software-image-descriptor

| | +--rw device-type

| | +--rw day0

| +--rw virtual-compute-descriptor

| +--rw virtual-storage-descriptor

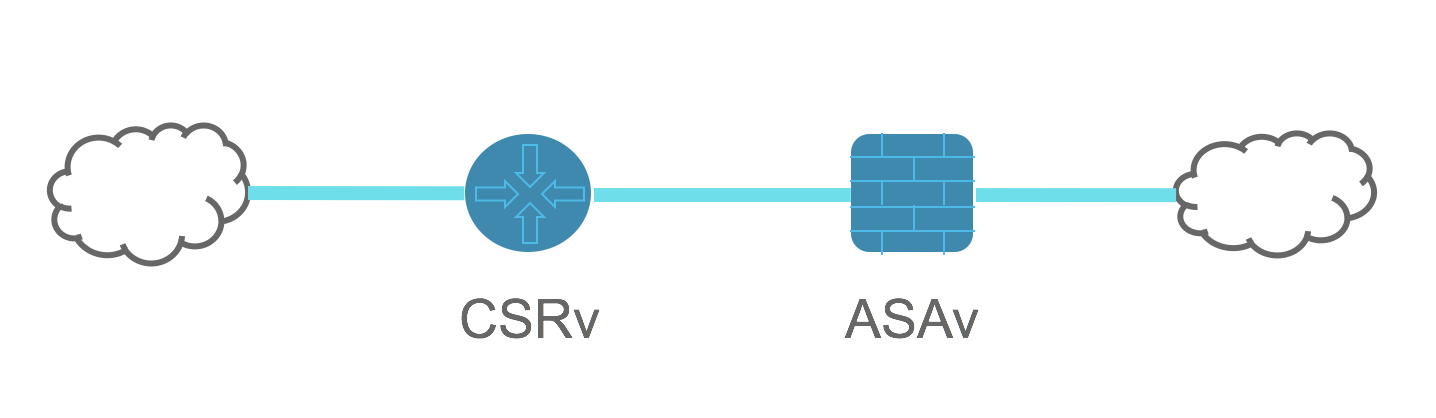
| +--rw external-connection-point-descriptor

| +--rw deployment-flavor

### NSD

The Network Service Descriptor (NSD) is a service deployment template which consists of information used by the SAE Orchestrator. An NS is a composition of VNFs (NF) arranged as a set of vnf-profiles with specified connectivity between them. NS describes a topology of the SAE end-to-end service or a portion (Half-chain) of it by referencing a pool of connection points and service access points and vnf-profiles. As illustrated in the figure below, the description of an NS as used by the SAE CFP to deploy an NS instance.

Figure 15. Deploying an NS Instance



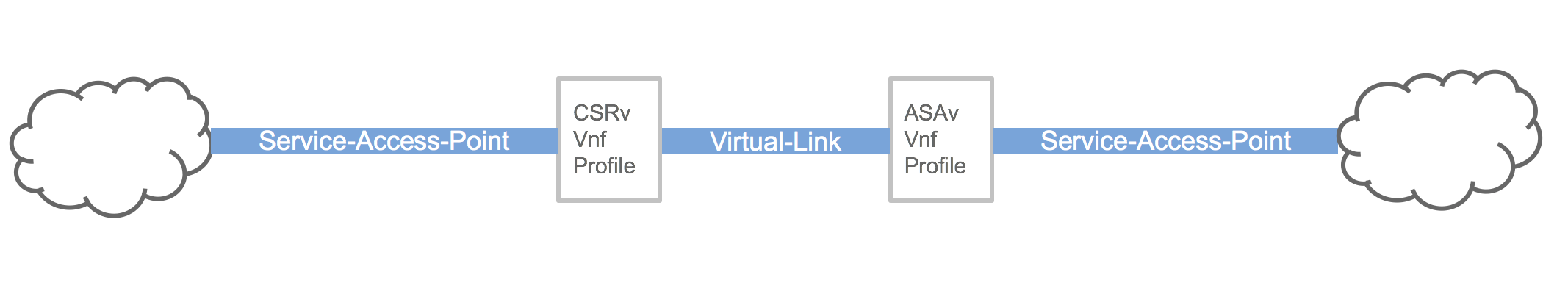
SAE doesn't use all the functions described under the "ETSI Network Service Templates Specification", just the one required to define the plumbing among the VNFs involved in the service definition.

Elements required to create a service topology:

* VNFD deployments used in the service definition. Let’s refer them as "vnf-profile".
* Connectivity between VNFs. Let’s refer them as "virtual-link".
* Connectivity between VNF and the external network. Let’s refer them as "service-access-point-descriptors".

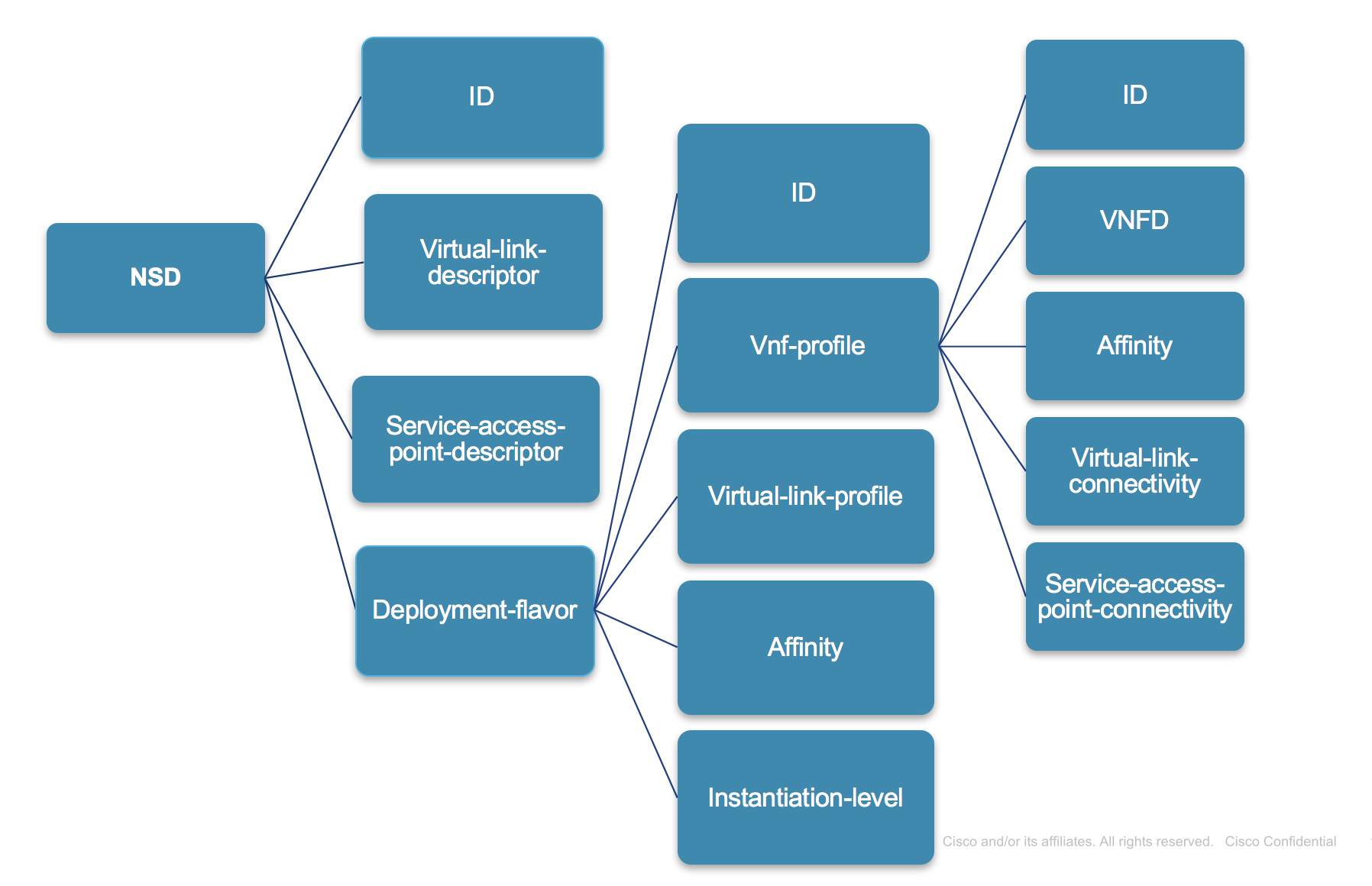
Using these three elements we can represent the service topology defined in figure below:

Figure 16. Service Topology



NSD defines the same elements, NSD data structure at a high-level is as shown in the figure below.

Figure 17. NDS Data Structure



**Pyang tree representation**

+--rw nsd\* [id]  
 | +--rw id string  
 | +--rw vnfd\* [vnfd]  
 | +--rw service-access-point-descriptor\* [id]  
 | +--rw virtual-link-descriptor\* [id]  
 | +--rw deployment-flavor\* [id]  
 | | +--rw id string  
 | | +--rw vnf-profile\* [id]  
 | | | +--rw id string  
 | | | +--rw vnfd -> /nfvo/vnfd/id  
 | | | +--rw flavor -> deref(../vnfd)/../deployment-flavor/id  
 | | | +--rw instantiation-level -> deref(../flavor)/../instantiation-level/id  
 | | | +--rw affinity-or-anti-affinity-group\* [id]  
 | | | +--rw virtual-link-connectivity\* [virtual-link-profile cp]  
 | | | +--rw sapd-connectivity\* [sapd cp]  
 | | +--rw virtual-link-profile\* [virtual-link-descriptor]  
 | | +--rw affinity-or-anti-affinity-group\* [id]  
 | | +--rw instantiation-level\* [id]

Virtual-link-descriptor (VLD) and Service-access-point-descriptor (SAPD) are list elements that represent the connectivity between two entities (either between VNFs or between a VNF and an external network). Deployment-flavor is where a user can define vnfs, their connectivity (the topology) and scaling capabilities.

SAE doesn't support any auto-scaling capabilities and hence most of virtual-link-profile and instantiation-level details are constants. nsd/deployment-flavor/vnf-profile represents a very specific instance of a VNFD and to what links (virtual-link-descriptors/ sapds) it is connected to. When two or more vnf-profiles connect to same link (virtual-link-descriptors) then they are said to be connected. Also, two or more vnf-profiles can participate in affinity policy to state whether all the participating vnfs should be placed on the same CSP or not.

**Virtual-link-descriptor (VLD)**

A VLD enable specifying the link for inter-VNF communication. It captures the type of connectivity, performance characteristics and throughput provided by the link. However, with CSP as VIM, SAE doesn't utilize any of these characteristics. But, since characteristics are mandatory the user is must provide them. A inter-VNF connectivity between CSRv and an ASAv can be represented as below:

<virtual-link-descriptor>  
 <id>INTERVNF\_LINK1</id>  
 <version>1</version>  
 <connectivity-type>  
 <layer-protocol>Ethernet</layer-protocol>  
 </connectivity-type>  
 <deployment-flavor>  
 <id>small</id>  
 <qos>  
 <latency>10</latency>  
 <packet-delay-variation>5</packet-delay-variation>  
 </qos>  
 </deployment-flavor>  
</virtual-link-descriptor>

**Note:** Since SAE 1.0 doesn't use characteristics, All the details on virtual-link-descriptor except the link **<id>** can be constants.

**Service-access-point-descriptor (SAPD)**

An SAPD enables specifying the link connecting VNF and external network. Based on the type of external network SAPD is connecting there are different types of marker tags

* Consumer-endpoint. Marks consumer end of the service chain.
* Provider-endpoint. Marks provider end of the service chain.
* Chain-endpoint. Marks chain-end, in case of half-chain the service ends in N9K and the SAPD should be marked with chain-endpoint.
* No marker. When there is no marking on SAPD, it represent a management connectivity.

These marker tags help SAE CFP in cli auto-complete, resource allocations. An SAPD connecting to a consumer network can be defined as:

<service-access-point-descriptor>  
 <id>FROM\_CONSUMER</id>  
 <consumer-endpoint xmlns="http://com/cisco/nso/sae-core-fp-common"/>  
</service-access-point-descriptor>

and an SAPD representing management network can be defined as:

<service-access-point-descriptor>  
 <id>MANAGEMENT</id>  
</service-access-point-descriptor>

Please note there is no marker tag on management SAPD.

**End-to-End service Vs Half-Chain**

An end-to-end service is one that connects to both provider and consumer networks, and a half-chain is one which connects to only one network (either consumer or provider).

So, an end-to-end service always has at least three SAPDs. One is for management and the other two are for consumer-endpoint and provider-endpoint respectively. A half-chain also has at least three SAPDs. One is for management and the other two are consumer-endpoint/provider-endpoint and chain-endpoint.

**Yang Model**

+--rw nsd\* [id]  
       |  +--rw id                                 string  
       |  +--rw designer?                          string  
       |  +--rw version?                           string  
       |  +--rw name?                              string  
       |  +--rw vnfd\* [vnfd]  
       |  |  +--rw vnfd    -> /nfvo/vnfd/id  
       |  +--rw service-access-point-descriptor\* [id]  
       |  |  +--rw id                         string  
       |  +--rw virtual-link-descriptor\* [id]  
       |  |  +--rw id                   string  
       |  |  +--rw version              string  
       |  |  +--rw connectivity-type  
       |  |  |  +--rw layer-protocol    identityref  
       |  |  +--rw deployment-flavor\* [id]  
       |  |  |  +--rw id                            string  
       |  |  |  +--rw qos  
       |  |  |  |  +--rw latency                   uint32  
       |  |  |  |  +--rw packet-delay-variation    uint32  
       |  +--rw deployment-flavor\* [id]  
       |  |  +--rw id                                 string  
       |  |  +--rw vnf-profile\* [id]  
       |  |  |  +--rw id                                      string  
       |  |  |  +--rw vnfd                                    -> /nfvo/vnfd/id  
       |  |  |  +--rw flavor                                  -> deref(../vnfd)/../deployment-flavor/id  
       |  |  |  +--rw instantiation-level                     -> deref(../flavor)/../instantiation-level/id  
       |  |  |  +--rw min-number-of-instances                 uint16  
       |  |  |  +--rw max-number-of-instances                 uint16  
       |  |  |  +--rw affinity-or-anti-affinity-group\* [id]  
       |  |  |  |  +--rw id    -> ../../../affinity-or-anti-affinity-group/id  
       |  |  |  +--rw virtual-link-connectivity\* [virtual-link-profile cp]  
       |  |  |  |  +--rw virtual-link-profile    -> ../../../virtual-link-profile/virtual-link-descriptor  
       |  |  |  |  +--rw cp                      -> deref(../../vnfd)/../external-connection-point-descriptor/id  
       |  |  |  +--rw sapd-connectivity\* [sapd cp]  
       |  |  |     +--rw sapd    -> ../../../../service-access-point-descriptor/id  
       |  |  |     +--rw cp      -> deref(../../vnfd)/../external-connection-point-descriptor/id  
       |  |  +--rw virtual-link-profile\* [virtual-link-descriptor]  
       |  |  |  +--rw virtual-link-descriptor                 -> ../../../virtual-link-descriptor/id  
       |  |  |  +--rw flavor                                  -> deref(../virtual-link-descriptor)/../deployment-flavor/id  
       |  |  |  +--rw max-bitrate-requirements  
       |  |  |  |  +--rw root    uint64  
       |  |  |  |  +--rw leaf?   uint64  
       |  |  |  +--rw min-bitrate-requirements  
       |  |  |     +--rw root    uint64  
       |  |  |     +--rw leaf?   uint64  
       |  |  +--rw affinity-or-anti-affinity-group\* [id]  
       |  |  |  +--rw id                string  
       |  |  |  +--rw affinity-type     affinity-type  
       |  |  |  +--rw affinity-scope    affinity-scope  
       |  |  +--rw instantiation-level\* [id]  
       |  |  |  +--rw id                      string  
       |  |  |  +--rw description             string  
       |  |  |  +--rw vnf-to-level-mapping\* [vnf-profile]  
       |  |  |  |  +--rw vnf-profile            -> ../../../vnf-profile/id  
       |  |  |  |  +--rw number-of-instances    uint32

## Chain Deployment Planning

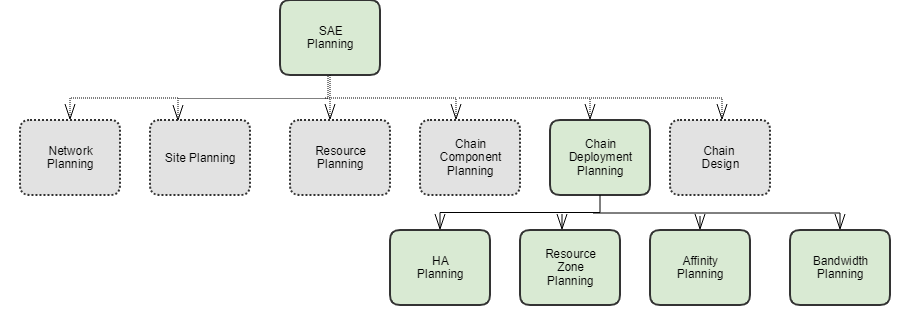
SAE chain deployment planning consists of the following considerations.

### Overview

Once the chain components like VNFDs and NSDs are identified, the designer should plan for chain deployment. Deployment includes some decisions like:

| **Feature** | **Description** |
| --- | --- |
| HA | User need to know if they need HA in EPGW or in Service level. So the NSD's can be designed that way |
| Resource Zone | During deployment the VNF's cane be placed on specific compute nodes by using the resource zones feature |
| Affinity | The Affinity / Anti Affinity feature can be used for VNF and Compute HA. This need to be planned in advance |
| Bandwidth | The bandwidth required for the service needs to be planned before deploying the service. |

Figure 18. Chain Deployment Planning



### HA Planning

The Endpoint gateway is the VNF through which the traffic comes in an SAE topology.

If GW is not in HA, then GW VNF is a single point of failure due to lack of HA. SAE introduces gateway HA using either HSRP or ECMP depending on traffic entry and exit.

The configurations need to be supported as part of day0.

This GW redundancy is supported and tested for CSR, FTDv and ASAv.

For other VNFs, static day0 configuration can define the redundancy configurations.

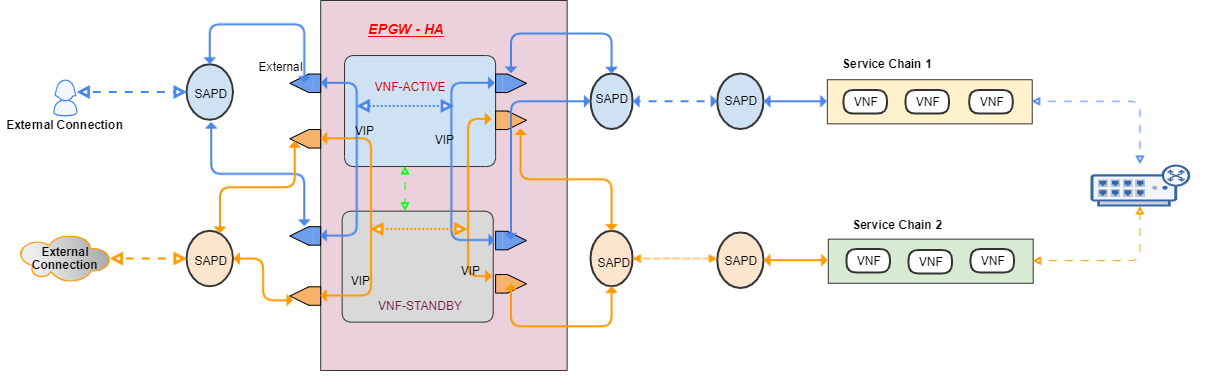
#### Design EPGW with HA

As per the design, active and standby VNFs in case of HA are treated as a service-chain. So EPGW refers to NSD instead of VNFD.

NSD should have a VNF profile created for active and standby VNF along with a failover virtual link.

EPGW is using a NSD deployment name  
list endpoint-gateway-vnf { ..  
    leaf deployment-name {  
    type leafref {  
          path "/cisco-sae-core-fp-cfs:sae-catalog/cisco-sae-core-fp-cfs:nsd-deployment/"  
                +"cisco-sae-core-fp-cfs:name";  
        }  
}

Figure 18. EPGW-HA



### Resource Zone Planning

A resource zone is a logical grouping of a set of compute nodes. The compute node placement needs a zone to work with.

Resource zones can be used to separate compute resources per tenant, per service, per site etc.

SAE uses an nfvo resource-orchestration package to handle the resource zone capability.

Resource orchestration works with a resource-zone to find the available compute node for placement.

For resource placement, a resource zone is mandatory.

You can specify resource zone in the following three places.

| **Resource Zone level** | **Details** |
| --- | --- |
| Service Zone | Optional. If nothing is specified Site zone is used by default. You can create a resource zone and use that in Services. |
| EPGW zone | Optional. You can manually create a resource zone with specific computes and use that zone in EPGW . This is used for EPGW compute selection. |
| Site Zone | This is a default zone. SAE will auto populate a default resource zone with all the compute nodes in the SAE infrastructure for that site.  Auto created default site level zone has naming convention as SAE-DEFAULT-ZONE-<Site Name> |

Resource zone is managed by the nfvo package and can be managed independently.

| **Action** | **Command** | **Details** |
| --- | --- | --- |
| Create | set nfvo resource-orchestration vim-type csp-vim vim VIM\_NAME csp-resource-zone ZONE\_NAME vcompute csp0. | SAE uses csp-vim as VIM and a zone name needs to be provided. |
| View | show nfvo resource-orchestration vim-type csp-vim vim csp-resource-zone. | list all resource zone under this VIM. |

#### Resource Zone Action

// To create resource zone  
admin@ncs% set nfvo resource-orchestration vim-type csp-vim vim SAE-VIM-sae-site-1 csp-resource-zone SAE-DEFAULT-ZONE-sae-site-1 vcompute csp0  
   
   
   
// To view the resource zone  
admin@ncs% show nfvo resource-orchestration vim-type csp-vim vim csp-resource-zone  
vim SAE-VIM-sae-site-1 {  
    csp-resource-zone SAE-DEFAULT-ZONE-sae-site-1 {  
        name     SAE-DEFAULT-ZONE-sae-site-1;  
        state    good;  
        vcompute [ csp0 csp1 csp5 ];  
    }  
}

#### Site level zone

By default a zone is created to be used for all the services in the site.

This default zone is named SAE-DEFAULT-ZONE-name of the site. This zone contains all the CSPs under the site infrastructure.

For example: - SAE-DEFAULT-ZONE-sae-site-1

You can create more resource zones by selecting any CSP's needed.

For example: - SAE-Service-Zone-1 , SAE-EPGW-Zone-1

#### Service level zone

You can specify resource zone under service level. This is optional. If one zone is specified then that zone is used for placement.

If there are no zones specified then the default zone is used.

**Service zone**

// Setting a zone under a specific service  
admin@ncs% set sae-site SITE-NAME service-chain SERVICE-NAME resource-zone ZONE-NAME

**EPGW level zone**

You can specify a resource zone under EPGW level. This is optional. If one zone is specified then that zone is used for placement.

If there are no zones specified then the default zone is used.

**EPGW zone**

// Setting a zone under EPGW  
admin@ncs% set sae-site SITE-NAME endpoint-gateway-vnf EPGW-NAME resource-zone ZONE-NAME

Verify the default zone is populated. If not, then create a new zone.

**Verify zone**

admin@ncs% show nfvo resource-orchestration vim-type csp-vim vim csp-resource-zone  
vim SAE-VIM-sae-site-1 {  
    csp-resource-zone SAE-DEFAULT-ZONE-sae-site-1 {  
        name     SAE-DEFAULT-ZONE-sae-site-1;  
        state    good;  
        vcompute [ csp0 csp1 csp5 ];  
    }  
}

#### Troubleshooting and Diagnostic

| **Action** | **details** |
| --- | --- |
| Placement Error | Make sure at-least one zone is present in resource-orchestration. |
| Remove a faulty CSP | Create a new zone without the faulty CSP and use that in Service chain and EPGW. |

### Affinity Group

SAE RO component is able to enforce the affinity and anti-affinity policy specified in the service chain NSD, and place the appropriate resources (compute, zone, network interface) that are compliant with the policies for each VNF and its network connection points.

The affinity configuration can be placed at different levels of the NSD elements.

The following table show the placement of VNFs that are in the same affinity and anti-affinity group based on the affinity-scope and affinity-type.

|  | **nfvi-node** | **zone** |
| --- | --- | --- |
| affinity | vnfs placed on the same CSP server | vnfs placed on the same Zone |
| anti-affinity | vnfs placed on different CSP server | vnfs placed on the different Zone |

Examples:

**in deployment flavor, defined affinity group:**

<affinity-or-anti-affinity-group>

            <id>anti-group1</id>

            <affinity-type>anti-affinity</affinity-type>

            <affinity-scope>nfvi-node</affinity-scope>

   </affinity-or-anti-affinity-group>

   <affinity-or-anti-affinity-group>

            <id>affinity-group2</id>

            <affinity-type>affinity</affinity-type>

            <affinity-scope>nfvi-node</affinity-scope>

   </affinity-or-anti-affinity-group>

**In vnf-profile ASA:**

<affinity-or-anti-affinity-group>

                <id>anti-group1</id>

 </affinity-or-anti-affinity-group>

 <affinity-or-anti-affinity-group>

                <id>affinity-group2</id>

 </affinity-or-anti-affinity-group>

**In vnf-profile ISR:**

<affinity-or-anti-affinity-group>

                <id>anti-group1</id>

 </affinity-or-anti-affinity-group>

**In vnf-profile WAAS:**

<affinity-or-anti-affinity-group>

          <id>affinity-group2</id>

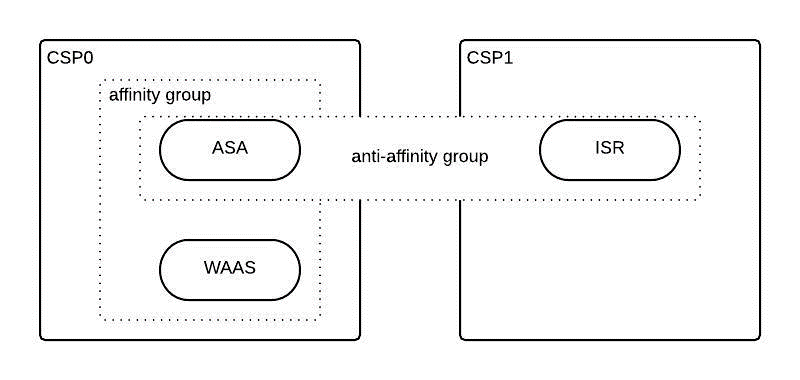
 </affinity-or-anti-affinity-group>

Both ASA and WAAS are in the same affinity-group2, so ASA and WAAS are placed in the same CSP.

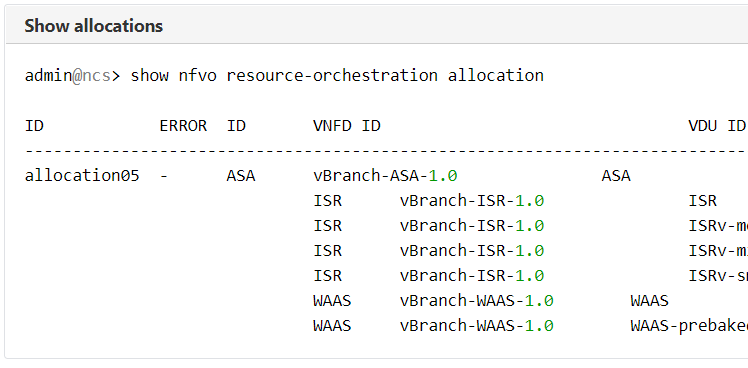
ASA and ISR are in the same anti-group1, so ASA and ISR are placed in different CSPs.

As the diagram shows, ASA is in two different groups at the same time:

Figure 19. Affinity Grouping



The affinity can be verified in the allocation response.



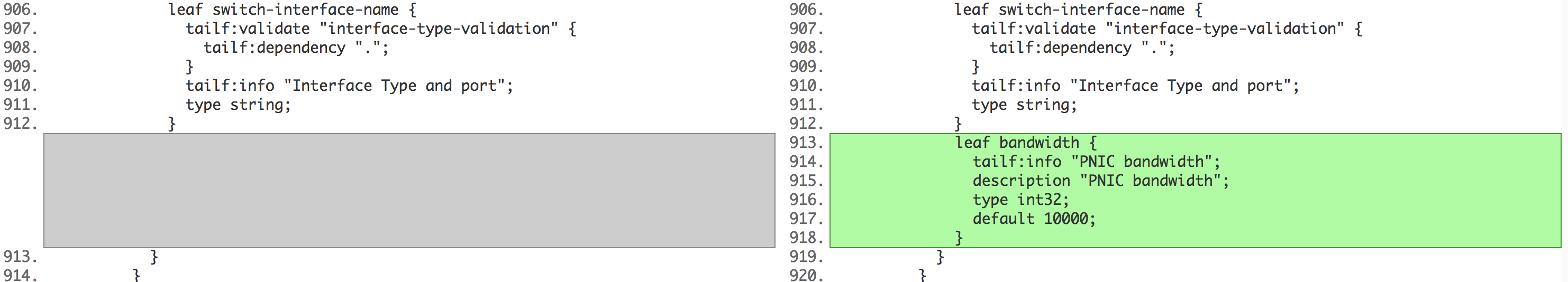
### Bandwidth Planning

Designers should consider bandwidth requirements while designing the service.

Some of the factors which affect bandwidth are sriov, non-sriov pnic selection and pnic-speed.

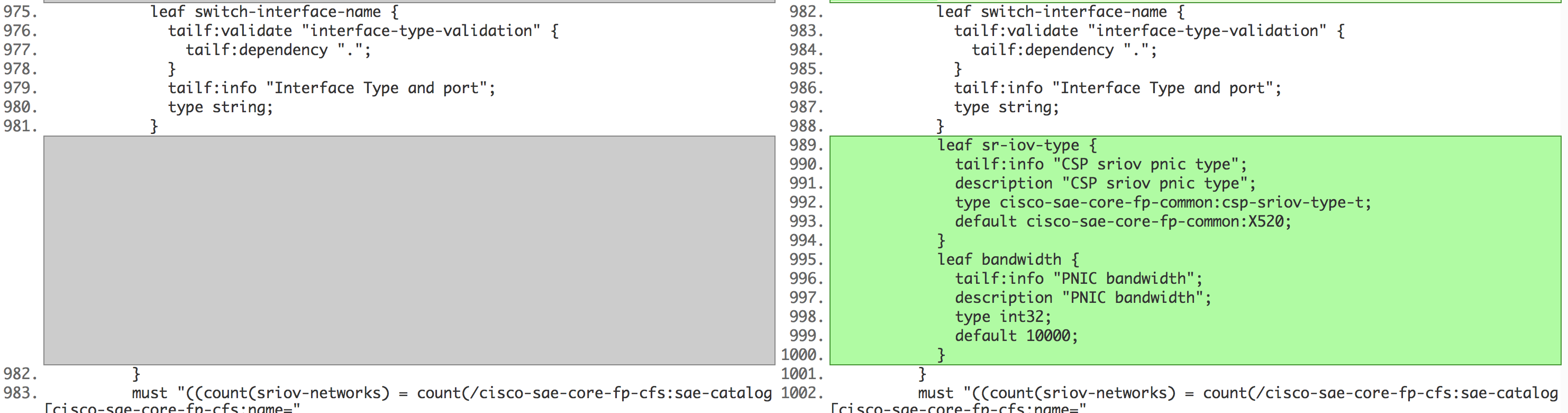
#### SAE infrastructure [virtio-network]

Infrastructure contains the default value for virtio-network bandwidth as '10000' mbps.



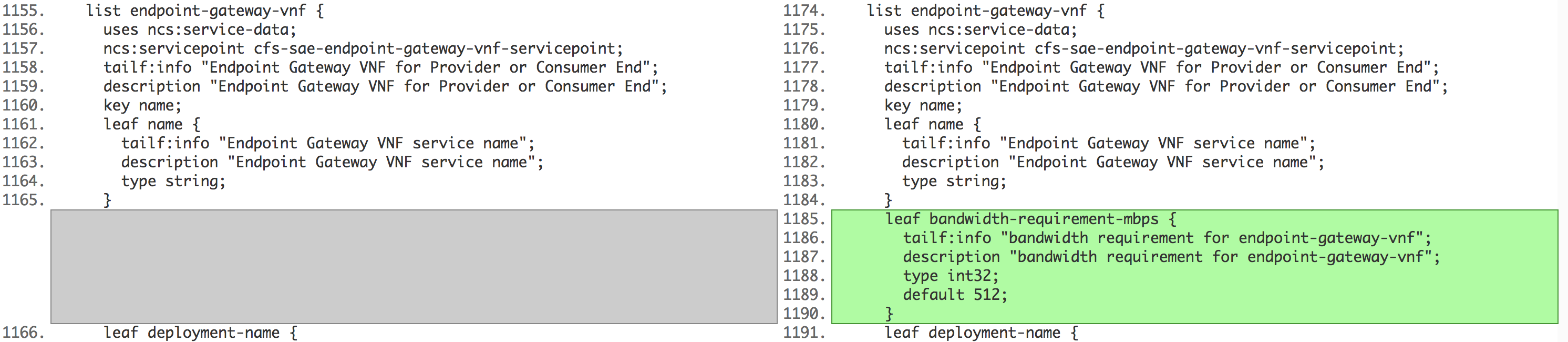
#### SAE infrastructure [sriov-networks]

Infrastructure contains the default value for sriov-network bandwidth as '10000' mbps.



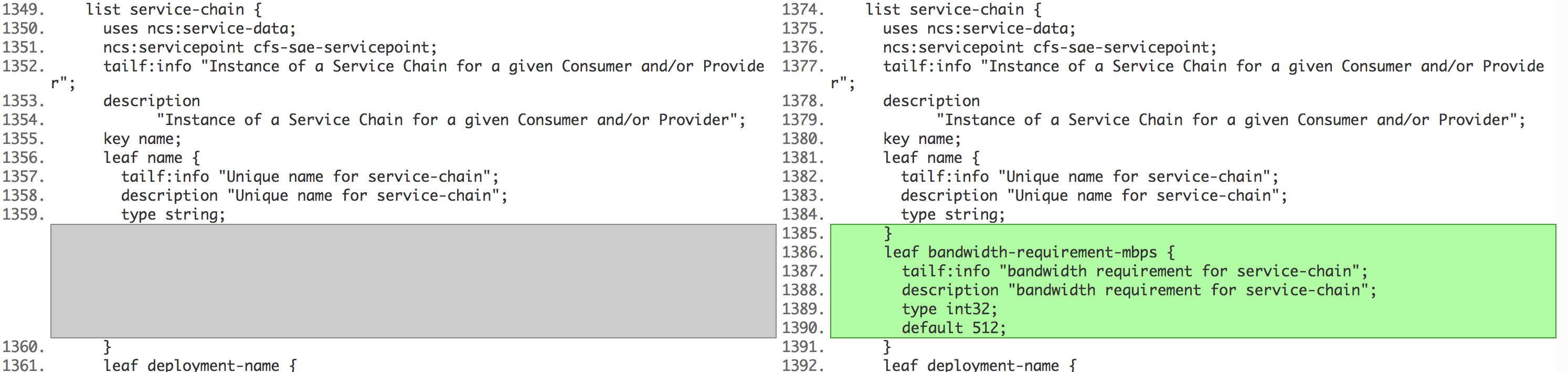
#### SAE endpoint-gateway-vnf

"bandwidth-requirement-mbps" uses a default value of '512' mbps if not changed by the user.



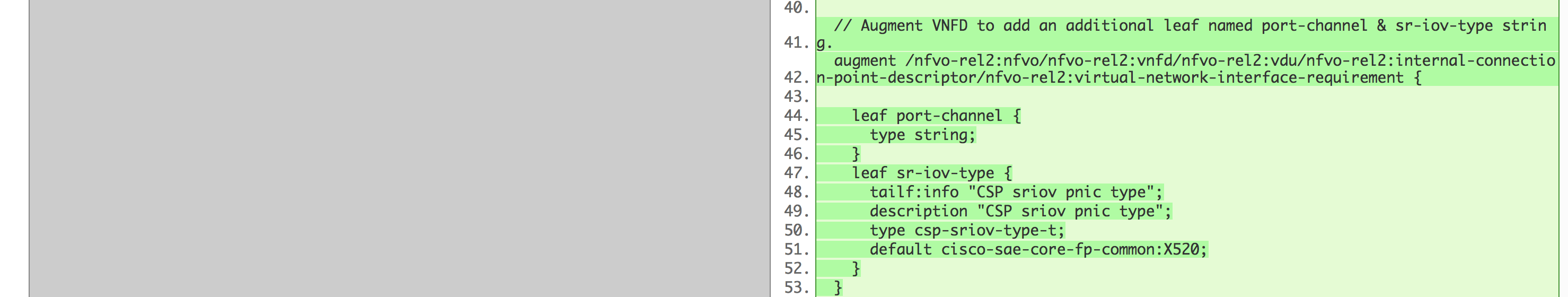
#### SAE service-chain

"bandwidth-requirement-mbps" uses a default value '512' mbps if not changed by the user.



#### NFVD VNFD

The default sr-iov-type is X520.



#### RO placement logic for sriov and virtio selection

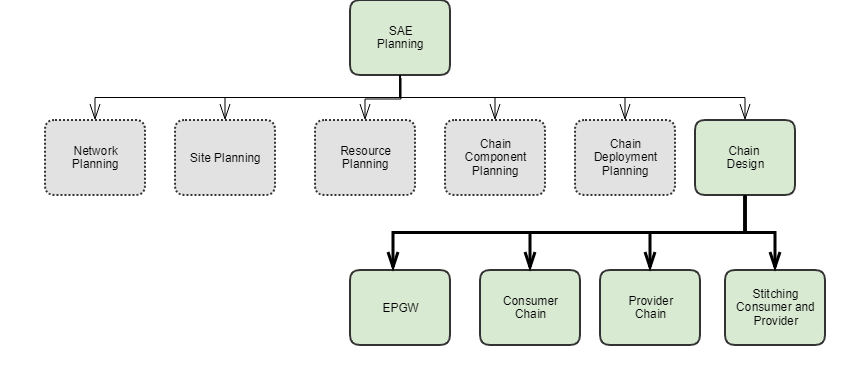
| **Additional leafs in VNFD** | **How RO should use them** |
| --- | --- |
| show nfvo vnfd SAE-CSR vdu CSR  internal-connection-point-descriptor inside {        external-connection-point-descriptor inside;  **bitrate-requirement                  1024;**  } | "bitrate-requirement" in internal-connection-point-descriptor is optional,  IF present in internal-connection-point-descriptor         it overwrites the "bandwidth-requirement-mbps" specified in RO request for         this connection point only.  IF NOT PRESENT         all connection-points are assumed to consume "bandwidth-requirement-mbps"         specified in RO request |
| internal-connection-point-descriptor outside {      external-connection-point-descriptor outside;      layer-protocol                       Ethernet;      interface-id                         2;  } | "virtual-network-interface-requirement" is NOT present, this means this is a virtio port, virtio-network  port-channel should be assigned to this internal-connection-point. and the capacity of both the PNICs in the virtio port-channel must be reduced by the bandwidth required. if bandwidth is insufficient, choose a different CSP, if  unable to find CSP found report error. |
| internal-connection-point-descriptor icp1 {                  external-connection-point-descriptor extra3;                  virtual-network-interface-requirement r1 {  **support-mandatory;**  **requirement SR-IOV;**                  }                  layer-protocol Ethernet    } | SRIOV is mandatory for this connection point, find CSP with SRIOV PNIC that has capacity for the requested bitrate. If no CSP can be found report error. SRIOV-Type is not specified so X520 PNICS can be allocated. |
| internal-connection-point-descriptor icp1 {                  external-connection-point-descriptor extra3;                  virtual-network-interface-requirement r1 {  **support-mandatory;**  **requirement SR-IOV;**  **sr-iov-type 520; // or 710**                  }                  layer-protocol Ethernet    } | Augment: VNFD/VDU/internal-connection-point-descriptor/virtual-network-interface-requirement, add an additional leaf "sr-iov-type" id-ref base can be 520 and 710.  SRIOV is mandatory and Type is specified, find a CSP which has the sriov pnic of the specified type (520 or 710) with enough capacity, if none found report error |
| internal-connection-point-descriptor icp1 {                  external-connection-point-descriptor extra3;                  virtual-network-interface-requirement r1 {  **support-mandatory;**  **requirement SR-IOV;**  **sr-iov-type 520; // or 710**  **port-channel pch1;**                  }                  layer-protocol Ethernet    }  internal-connection-point-descriptor icp2 {                  external-connection-point-descriptor extra3;                  virtual-network-interface-requirement r1 {  **support-mandatory;**  **requirement SR-IOV;**  **sr-iov-type 520; // or 710**  **port-channel pch1;**                  }                  layer-protocol Ethernet    } | VNFD/VDU/internal-connection-point-descriptor/virtual-network-interface-requirement, we have additional leaf named "port-channel"  icp1 and icp2 in addition to needing mandatory SRIOV of type 520 also will be part of the same port-channel "pch1".  i.e  icp1 and icp2 both have to he a PNIC of the specified type and enough capacity  but they cannot be the same. We get 2 different PNICs allocated |

## Service Chain Design

SAE service chain design consists of the following considerations.

### Overview

Figure 19. Service Chain Design



### EPGW

Endpoint gateway is the VNF through which the traffic comes in a SAE topology.

**To spin-up EPGW VNF's in HA/nonHA mode:**

* + 1. Create nsd & vnfd's for active, standby VNF's , if you are using nonHA then specify only one vnf-profile in nsd for epgw
    2. Create nsd-deployment for EPGW in sae-catalog along with vnfd-deployment
    3. Create day0 file for active, standby VNF's, specify it in vnfd-deployment
    4. Create epgw payload sae-site & deploy EPGW in HA mode

### Sample Payloads

To view the sample payloads, see Appendix A, 10.4 EPGW Payloads.

# SAE Service Provisioning

The following chapter gives an explanation of SAE service provisioning.

## Introduction

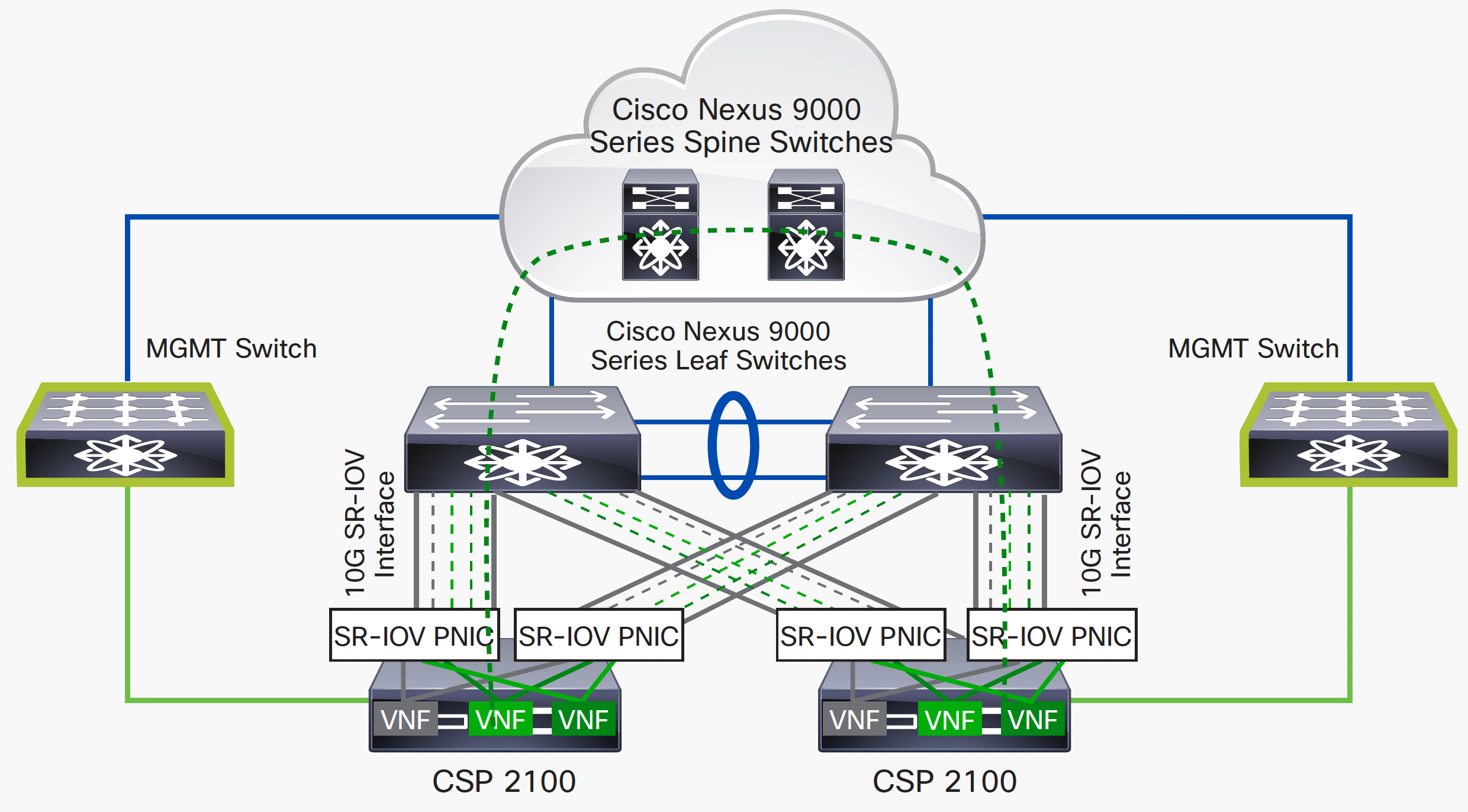
SAE service provisioning consists of the following considerations.

## Infrastructure Discovery

### Overview

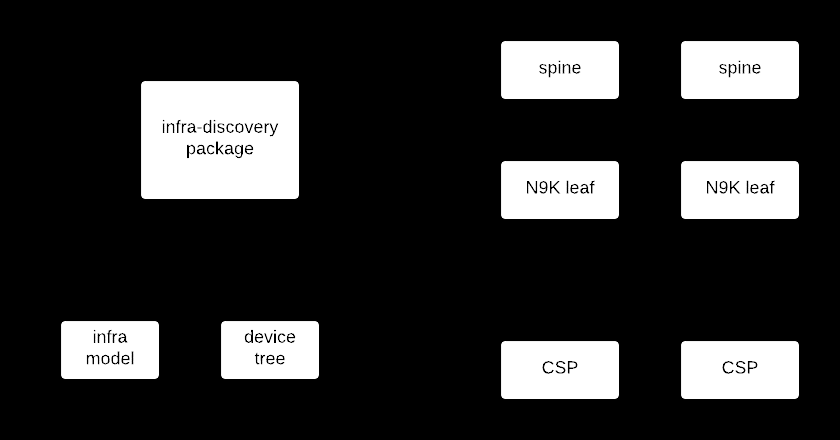
The infra-discovery package is able to crawl the N9K switches in the SAE infrastructure and get toplogy details to onboard CSP devices in NSO and populate the SAE infrastructure model in CDB.

Figure 20. Infrastructure Discovery.



Auto Discovery will onboard devices into devices tree and infrastructure, it can minimize human error, user don't need to type in interface, connected switch, bandwidth etc, significantly decreases infra configuration time.

Figure 21. Workflow



1. Start the infra-discovery-action.
2. infra-discovery invokes NXAPI on IP given to retrieve the LLDP neighbor detail.
3. Infra-discovery crawsl more N9K leaf.
4. Based on the neighbor info, onboard and sync the CSP device.
5. Pull port-channel and Sriov pnic info.
6. Populate the infra-model under existing sae-site/compute-clusters.

#### Prerequisites/Dependencies

1. NSO
   1. NSO 4.7 or newer
      1. Request package (**required on Linux machine**)
      2. SAE 1.0.0 (and above)
2. CSPs
   1. LLDP enabled
   2. Pre-configure port & port-channels
   3. Pre-configure SRIOV
3. N9K
   1. Leaf with CSPs
   2. Spine-Leaf with CSPs (Limited support in 1.0.0)
   3. LLDP feature enabled
   4. NXAPI feature enabled with NXAPI port 443 enabled
   5. Pre-configure Virtio pnics to port-channels (PoCH) and link up SRIOV from CSPs
4. NSO pre-config
   1. SAE Auth-group for device(s)
   2. Vlan resource pools
   3. SAE Catalog
   4. SAE Catalog csp Type
   5. SAE Provider and SAE-Provider-Catalog
   6. SAE Site Infrastructure Compute-Cluster(s) and VLAN Pool
   7. SAE Site Infrastructure switching type
   8. SAE Site Mgmt-vnf-resources; gateway, vlan, and network.
   9. Infra-discovery configFile
   10. Samples

### Action

**CLI commands**

config  
set devices authgroups group switch\_authgroup default-map remote-name admin remote-password AbcD\#  
set devices authgroups group csp\_authgroup default-map remote-name admin remote-password AbcD123\#  
set resource-pools id-pool VLAN\_POOL range start 2 end 1000  
set sae-catalog cisco-catalog csp csp-type-cluster1  
set sae-provider cisco-provider sae-provider-catalog cisco-catalog  
set sae-site cisco sae-provider cisco-provider infrastructure compute-clusters cluster1 vlan-pool VLAN\_POOL  
set sae-site cisco infrastructure switching type n9k-switch-pair  
set sae-site cisco vnf-mgmt-resources vnf-mgmt-gateway 5.5.5.1 vnf-mgmt-vlan 5 vnf-mgmt-netmask 255.255.0.0  
commit  
   
   
admin@ncs% show cisco-sae-core-fp-cfs:sae-site | display xml  
<config xmlns="[http://tail-f.com/ns/config/1.0"](http://tail-f.com/ns/config/1.0)>  
  <sae-site xmlns="[http://com/cisco/nso/sae-core-fp-cfs"](http://com/cisco/nso/sae-core-fp-cfs)>  
    <site>cisco</site>  
    <sae-provider>cisco-provider</sae-provider>  
    <vnf-mgmt-resources>  
      <vnf-mgmt-netmask>255.255.0.0</vnf-mgmt-netmask>  
      <vnf-mgmt-vlan>5</vnf-mgmt-vlan>  
      <vnf-mgmt-gateway>5.5.5.1</vnf-mgmt-gateway>  
    </vnf-mgmt-resources>  
    <infrastructure>  
      <switching>  
        <type xmlns:cisco-sae-core-fp-common="[http://com/cisco/nso/sae-core-fp-common"](http://com/cisco/nso/sae-core-fp-common)>cisco-sae-core-fp-common:n9k-switch-pair</type>  
      </switching>  
      <compute-clusters>  
        <name>cluster1</name>  
        <vlan-pool>VLAN\_POOL</vlan-pool>  
      </compute-clusters>  
    </infrastructure>  
  </sae-site>  
</config>  
[ok][2018-10-17 11:44:00]  
   
[edit]  
   
[edit]  
admin@ncs%

**Sample configFile**

**CLI commands**

$ vi config.cfg  
{  
    "site"              :       "cisco",  
    "cluster"           :       "cluster1",  
    "cspType"           :       "csp-type-cluster1",  
    "nexus\_seed\_address":       "5.5.91.154",  
    "csp\_authgroup"     :       "csp\_authgroup",  
    "n9k\_authgroup"     :       "switch\_authgroup",  
    "rest\_username"     :       "John",  
    "rest\_password"     :       "AbcD#"  
}  
   
Configuration file key  
    "site"                  → SAE site name  
    "cluster"               → compute cluster name from infrastructure  
    "cspType"               → csp type defined from "sae-catalog csp"  
    "nexus\_seed\_address"    → N9K seed IP address, it must be one of the N9K leaf Management IP Address  
    "csp\_authgroup"         → CSP authgroup, be used to connect CSP devices within this cluster by SSH credentials, keeps CSP credentials consistent  
    "n9k\_authgroup"         → N9K authgroup, be used to connect N9K devices by SSH credentials, keep N9K credentials consistent  
    "rest\_username"         → rest username to connect Nexus Seed  
    "rest\_password"             → rest password to connect Nexus Seed  
   
Notes: Keep the key as it is, fill the related values

**Stand alone cluster discovery**

admin@ncs> request discovery\_action discover configFile /home/admin/sae-local-install/config.cfg

Setup configFile as last session mentioned.

**Multi-cluster discovery**

a. Discovery supports multi-cluster. Furthermore, each discovery learns only one cluster at a time and requires separate discovery requests with its respective discovery config.

- To discover multiple leafs, user should create multiple config.cfg, running discovery separately.

b. Auto Discovery supports Spine-leaf scenario, where each switch-pair belongs to a separated cluster.

admin@ncs> request discovery\_action discover configFile /home/admin/sae-local-install/config-cluster1.cfg

admin@ncs> request discovery\_action discover configFile /home/admin/sae-local-install/config-cluster2.cfg

**Discovery Redeployment**

Discovery supports infrastructure update.

a. For example, when a new CSP is added to the existing cluster and Auto Discovery is re-ran, the CSP will be appended into "sae-site/../infrastructure" and devices list.

**Note:** SAE only accepts N9K vPC switch-pairs. If any additional N9K is added to the topology, Auto Discovery will ignore adding this switch to NSO.

b. Deletion of CSP is supported. If any CSP is removed from setup, please make sure there are no VNFs deployed by this CSP, otherwise discovery update would not allow the removal of this CSP.

### Payload

**Yang Model**

module: cisco-sae-core-fp-cfs

+--rw sae-catalog\* [name]  
| +--rw name string  
| +--rw csp\* [name]  
| | +--rw name string  
| | +--rw virtio-network\* [name]  
| | | +--rw name network-name  
| | +--rw sriov-networks\* [name]  
| | | +--rw name network-name  
| | +--rw mgmt-network? network-name  
| +--rw infrastructure  
| | +--rw switching  
| | | +--rw type? cisco-sae-core-fp-common:switching-type-t  
| | +--rw compute-clusters\* [name]  
| | | +--rw name string  
| | | +--rw vlan-pool -> /ralloc:resource-pools/idalloc:id-pool/name  
| | | +--rw leaf-switches\* [name]  
| | | | +--rw name string  
| | | +--rw servers\* [name]  
| | | +--rw name string  
| | | +--rw type -> /sae-catalog/csp/name  
| | | +--rw virtio-network\* [pc-name]  
| | | | +--rw pc-name -> /sae-catalog/csp/virtio-network/name  
| | | | +--rw pnics\* [name]  
| | | | +--rw name string  
| | | | +--rw connected-switch? -> ../../../../leaf-switches/name  
| | | | +--rw switch-interface-name? string  
| | | | +--rw bandwidth? int32  
| | | +--rw sriov-networks\* [name]  
| | | | +--rw name -> /sae-catalog/csp/sriov-networks/name  
| | | | +--rw connected-switch? -> ../../../leaf-switches/name  
| | | | +--rw switch-interface-name? string  
| | | | +--rw sriov-type? cisco-sae-core-fp-common:csp-sriov-type-t  
| | | | +--rw bandwidth? int32

Underscore is a prerequisite before auto-discovery runs. Auto-discovery will fill up bold session.

**Discovery request yang**

**module: infra-discovery**

**+--rw discovery\_action**

**+---x discover**

**|  +---w input**

**|  |  +---w configFile?   string**

**|  |  +---w address?      inet:ip-address**

**|  |  +---w username?     string**

**|  |  +---w password?     string**

**|  +--ro output**

**|     +--ro result?   uint16**

### Diagnosis

Verification

admin@ncs% show cisco-sae-core-fp-cfs:sae-site cisco infrastructure compute-clusters  
compute-clusters cluster1 {  
    vlan-pool My\_Vlan\_Pool;  
    leaf-switches N9k-Leaf-1;  
    leaf-switches N9k-Leaf-2;  
    servers csp-225 {  
        type csp-type-cluster1;  
        virtio-network TestPC {  
            pnics enp4s0f0 {  
                connected-switch      N9k-Leaf-2;  
                switch-interface-name port-channel121;  
                bandwidth             1024;  
            }  
            pnics enp4s0f1 {  
                connected-switch      N9k-Leaf-1;  
                switch-interface-name port-channel121;  
                bandwidth             1024;  
            }  
        }  
        sriov-networks enp4s0f2 {  
            connected-switch      N9k-Leaf-2;  
            switch-interface-name Ethernet1/3;  
            sriov-type            X520;  
            bandwidth             1024;  
        }  
        sriov-networks enp4s0f3 {  
            connected-switch      N9k-Leaf-1;  
            switch-interface-name Ethernet1/4;  
            sriov-type            X520;  
            bandwidth             1024;  
        }  
    }  
    servers csp-226 {  
        type csp-type-cluster1;  
        virtio-network TestPC {  
            pnics enp4s0f0 {  
                connected-switch      N9k-Leaf-1;  
                switch-interface-name port-channel120;  
                bandwidth             1024;  
            }  
            pnics enp4s0f1 {  
                connected-switch      N9k-Leaf-2;  
                switch-interface-name port-channel120;  
                bandwidth             1024;  
            }  
        }  
        sriov-networks enp4s0f2 {  
            connected-switch      N9k-Leaf-1;  
            switch-interface-name Ethernet1/3;  
            sriov-type            X520;  
            bandwidth             1024;  
        }  
        sriov-networks enp4s0f3 {  
            connected-switch      N9k-Leaf-2;  
            switch-interface-name Ethernet1/4;  
            sriov-type            X520;  
            bandwidth             1024;  
        }  
    }  
}

1. Verify that all csp "server" and "leaf-switches" are added to site infrastructure and device list.
2. Verify that "virtio-network" and "sriov-networks" are added to site infrastructure.
3. Verify that "switch-interface-name", "sriov-type" and "bandwidth" are added to site infrastructure.

### Troubleshooting

Infra discovery iog file is generated and appended with all auto-discovery executions.

Location: /var/log/ncs/ncs-python-vm-infra-discovery.log

Search for "CspServers"

CspServers = [{  
    "virtio\_network": {  
        "TestPC": {  
            "enp4s0f0": {  
                "bandwidth": "1G",  
                "switch\_interface\_name": "port-channel120",  
                "connected\_switch": "N9k-Leaf-1"  
            },  
            "enp4s0f1": {  
                "bandwidth": "1G",  
                "switch\_interface\_name": "port-channel120",  
                "connected\_switch": "N9k-Leaf-2"  
            }  
        }  
    },  
    "type": "csp-type-cluster1",  
    "name": "CSP-225",  
    "sriov\_networks": {  
        "enp4s0f2": {  
            "connected\_switch": "N9k-Leaf-1",  
            "bandwidth": "1G",  
            "switch\_interface\_name": "Ethernet1/3",  
            "sriov\_type": "igb"  
        },  
        "enp4s0f3": {  
            "connected\_switch": "N9k-Leaf-2",  
            "bandwidth": "1G",  
            "switch\_interface\_name": "Ethernet1/4",  
            "sriov\_type": "igb"  
        }  
    }  
}, {  
    "virtio\_network": {  
        "TestPC": {  
            "enp4s0f0": {  
                "bandwidth": "1G",  
                "switch\_interface\_name": "port-channel121",  
                "connected\_switch": "N9k-Leaf-2"  
            },  
            "enp4s0f1": {  
                "bandwidth": "1G",  
                "switch\_interface\_name": "port-channel121",  
                "connected\_switch": "N9k-Leaf-1"  
            }  
        }  
    },  
    "type": "csp-type-cluster1",  
    "name": "CSP-226",  
    "sriov\_networks": {  
        "enp4s0f2": {  
            "connected\_switch": "N9k-Leaf-2",  
            "bandwidth": "1G",  
            "switch\_interface\_name": "Ethernet1/3",  
            "sriov\_type": "igb"  
        },  
        "enp4s0f3": {  
            "connected\_switch": "N9k-Leaf-1",  
            "bandwidth": "1G",  
            "switch\_interface\_name": "Ethernet1/4",  
            "sriov\_type": "igb"  
        }  
    }  
}]

1. If 'CspServers' is not present, auto discovery failed before it can retrieve information from N9K and CSPs.
   1. Verify if all connections are correct and if all Prerequisites are set properly.
2. If 'CspServers' is present, check if any element is missing. For example, check if "connected\_switch" is missing from any interface.
   1. If missing, please check physical connections and validate that virtio connections are part of vPC with N9K and linked "UP". If SRIOV, make sure that SRIOV links are "UP" and configured correctly on CSP.
3. Validate Auto-Discovery configFile has the correct key/value pairs for each variable.

## EPGW

### Overview

The endpoint gateway is the VNF through which the traffic comes in a SAE topology.

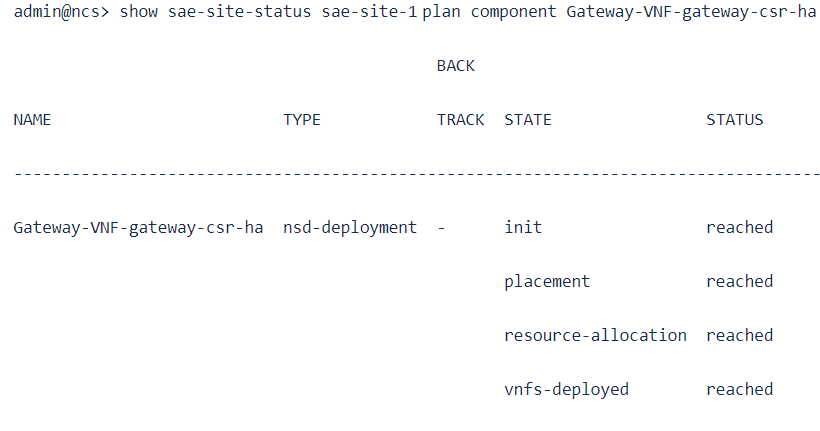
**To spin-up EPGW VNF's in HA/nonHA mode:**

1. Create nsd & vnfd's for active, standby VNF's , if you are using nonHA then specify only one vnf-profile in nsd for epgw
2. Create nsd-deployment for EPGW in sae-catalog along with vnfd-deployment
3. Create day0 file for active, standby VNF's, specify it in vnfd-deployment
4. Create epgw payload sae-site & deploy EPGW in HA mode

### Sample Payloads

To view sample payloads, see Appendix A, EPGW Payloads.

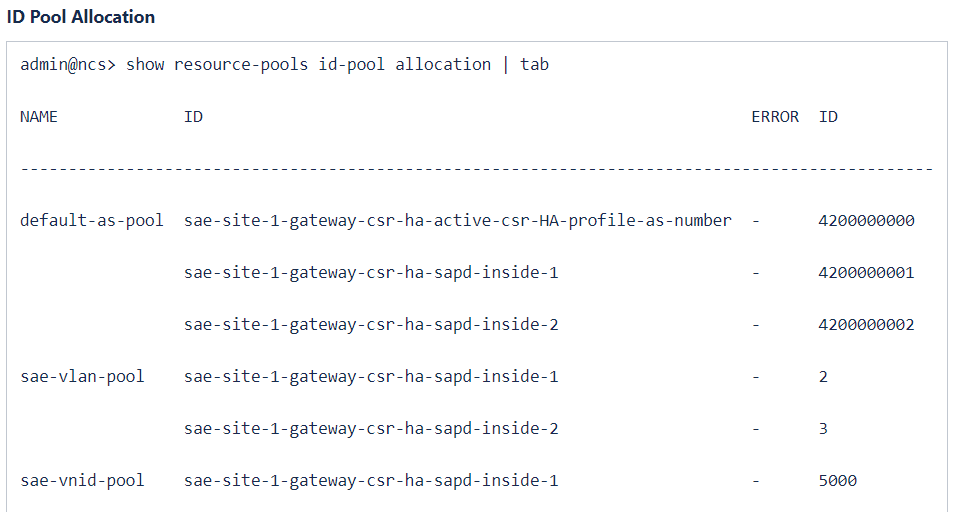
### Verification



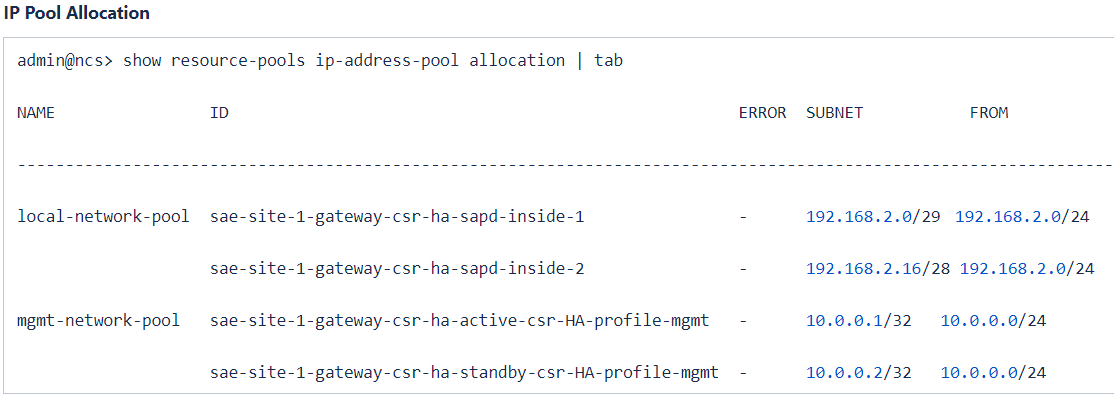
### Troubleshooting

admin@ncs> show endpoint-gateway-association  
   
            ENDPOINT        END                    
   
SAE SITE    GATEWAY         POINT  ACCESS POINT    
   
--------------------------------------------------  
   
sae-site-1  gateway-csr-ha  ep1    sapd-inside-1   
   
                            ep2    sapd-inside-2

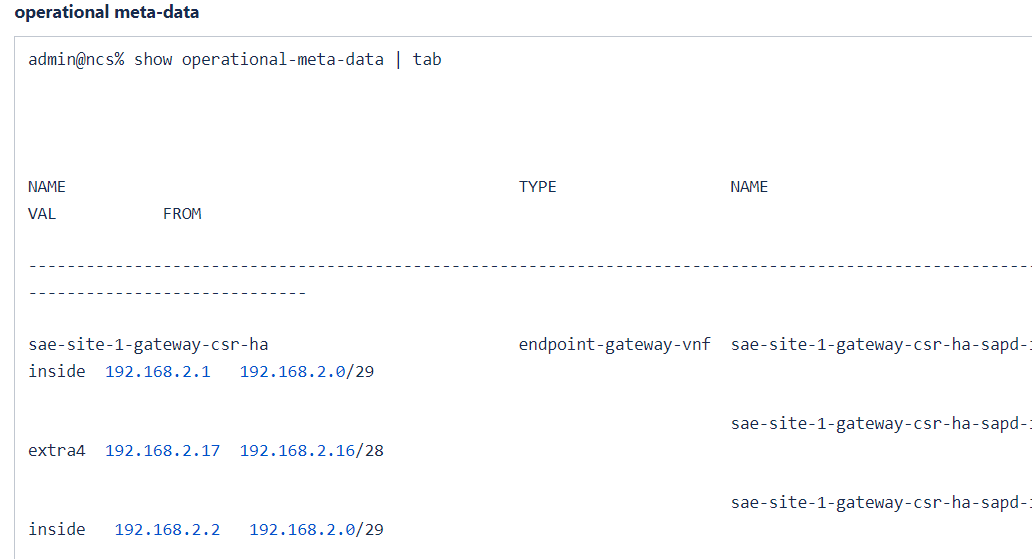
**ID Pool Allocation**



**IP Pool Allocation**



**Operational Meta-Data**



**SAE Allocation**



## Service Chain

### Overview

Half chains can be of two types:

* Consumer half chain: One end of service chain is connected to the consumer end and other end is connected to the fabric (switch) .
* Provider half chain: One end of service chain is connected to the provider end and other end is connected to the fabric (switch).

Depending on how service chain ends are connected to external networks, service chains can be instantiated in either of following ways:

* Inline Service chain: Consumer/Provider end VNF of Service chain is connected to external-end-points.
* Shared Gateway: Consumer/Provider end VNF of Service chain is connected to individual endpoint-gateway, and this endpoint-gateway is connected to external-end-points.

**Yang Model**

1. Service Chain Type: 'type' leaf in yang identifies the type of service. Chain 'type' leaf is an enumeration and can have a value 'consumer', 'provider' or 'end-to-end'.
2. Choice 'gateway-choice' will distinguish a Service Chain as 'Inline Service Chain' or 'Shared Gateway'

Case 'external-go-case': populate the fields of endpoint-gateway for user selection.

Case 'inline-gw-case': will populate the service-access-points of end VNFs for user selection.

'deployment-name' refers to the 'nsd-deployment' in sae-catalog.

**Service-chain Yang Model**

|  +--rw service-chain\* [name]  
|  |  +--rw name                          string  
|  |  +--rw bandwidth-requirement-mbps?   int32  
|  |  +--rw deployment-name               -> /sae-catalog/nsd-deployment/name  
|  |  +--rw type                          enumeration  
|  |  +--rw consumer-end-point\* [external-end-point]  
|  |  |  +--rw external-end-point    -> ../../../external-end-point/name  
|  |  |  +--rw (gateway-choice)?  
|  |  |     +--:(external-gw-case)  
|  |  |     |  +--rw shared-gateway!  
|  |  |     |     +--rw endpoint-gateway-vnf    -> ../../../../endpoint-gateway-vnf/name  
|  |  |     |     +--rw gateway-access-point?   string  
|  |  |     |     +--rw service-access-point?   string  
|  |  |     +--:(inline-gw-case)  
|  |  |        +--rw connectivity\* [service-access-point]  
|  |  |           +--rw service-access-point    string  
|  |  |           +--rw ip\* [address]  
|  |  |              +--rw address        inet:ip-address  
|  |  |              +--rw virtual-ip?    empty  
|  |  |              +--rw vnf-profile?   string  
|  |  +--rw provider-end-point\* [external-end-point]  
|  |  |  +--rw external-end-point    -> ../../../external-end-point/name  
|  |  |  +--rw (gateway-choice)?  
|  |  |     +--:(external-gw-case)  
|  |  |     |  +--rw shared-gateway!  
|  |  |     |     +--rw endpoint-gateway-vnf    -> ../../../../endpoint-gateway-vnf/name  
|  |  |     |     +--rw gateway-access-point?   string  
|  |  |     |     +--rw service-access-point?   string  
|  |  |     +--:(inline-gw-case)  
|  |  |        +--rw connectivity\* [service-access-point]  
|  |  |           +--rw service-access-point    string  
|  |  |           +--rw ip\* [address]  
|  |  |              +--rw address        inet:ip-address  
|  |  |              +--rw virtual-ip?    empty  
|  |  |              +--rw vnf-profile?   string  
|  |  +--rw var\* [name]  
|  |  |  +--rw name             variableName  
|  |  |  +--rw (valtype)?  
|  |  |     +--:(val)  
|  |  |     |  +--rw val              string  
|  |  |     +--:(encrypted-val)  
|  |  |        +--rw encrypted-val    tailf:aes-cfb-128-encrypted-string  
|  |  +--rw resource-zone\*                -> /nfvo-rel2:nfvo/nfvo-rel2-ro:resource-orchestration/vim-type/vim/csp-vim:csp-resource-zone/id  
|  |  +--rw customization  
|  |  |  +--rw vnf-deployment\* [profile]  
|  |  |     +--rw profile    -> deref(../../../deployment-name)/../vnf-deployment/vnf-profile

### Prerequisites/Dependencies

* SAE site infrastructure has been set and 'vnf-manager' is deployed successfully.
* For service chain using a shared gateway, the endpoint-gateway service has been deployed successfully.
* VNFDs describing each VNF has been loaded in NSO.
* NSD describing service chain has been loaded in NSO.
* 'vnfd-deployment' and 'nsd-deployment' have been created under 'see-catalog' which is associated with an sae-site.

### Activate

**Usage with Diagram and Sample Payloads**

**Consumer Half Chain**

Figure 21. Shared Gateway

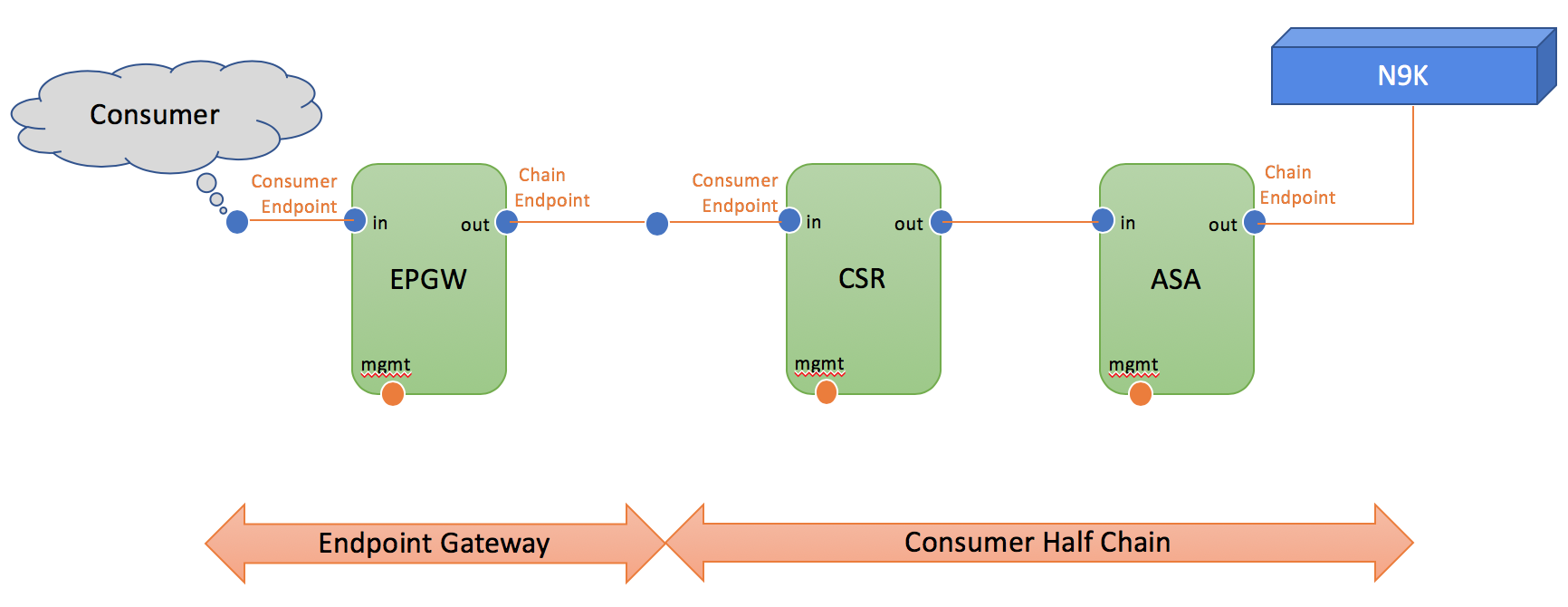
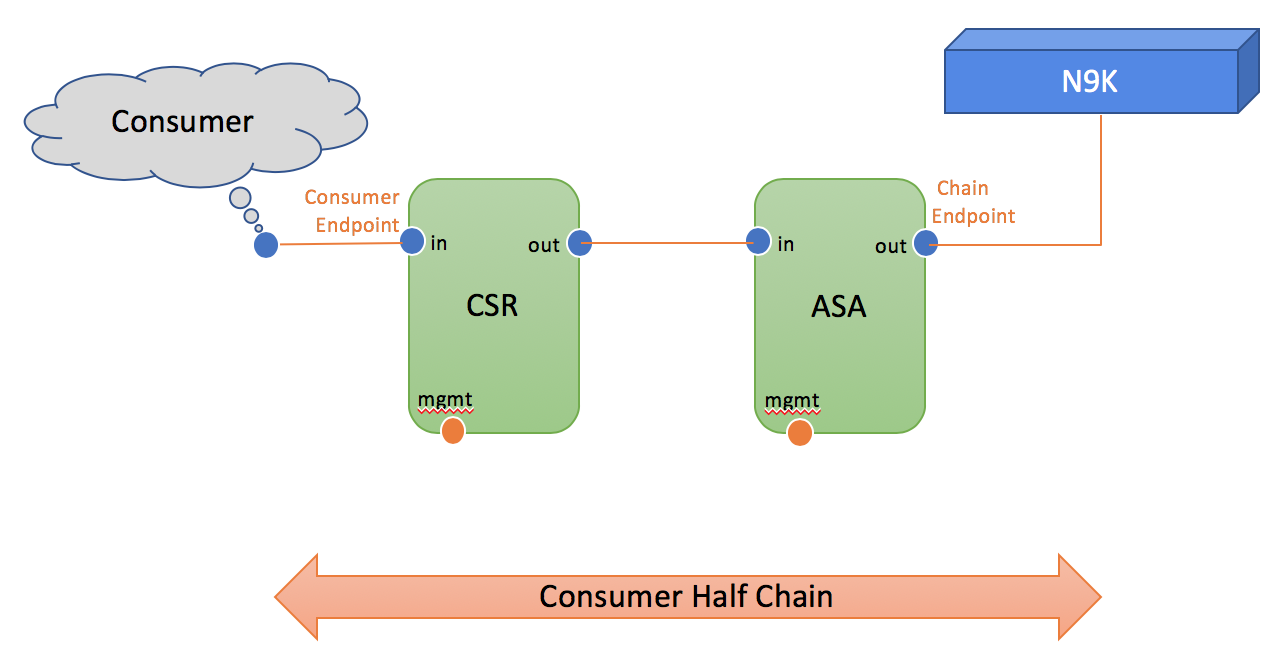


Figure 22. Inline Service Chain 

**Provider Half Chain**

Figure 23. Shared Gateway

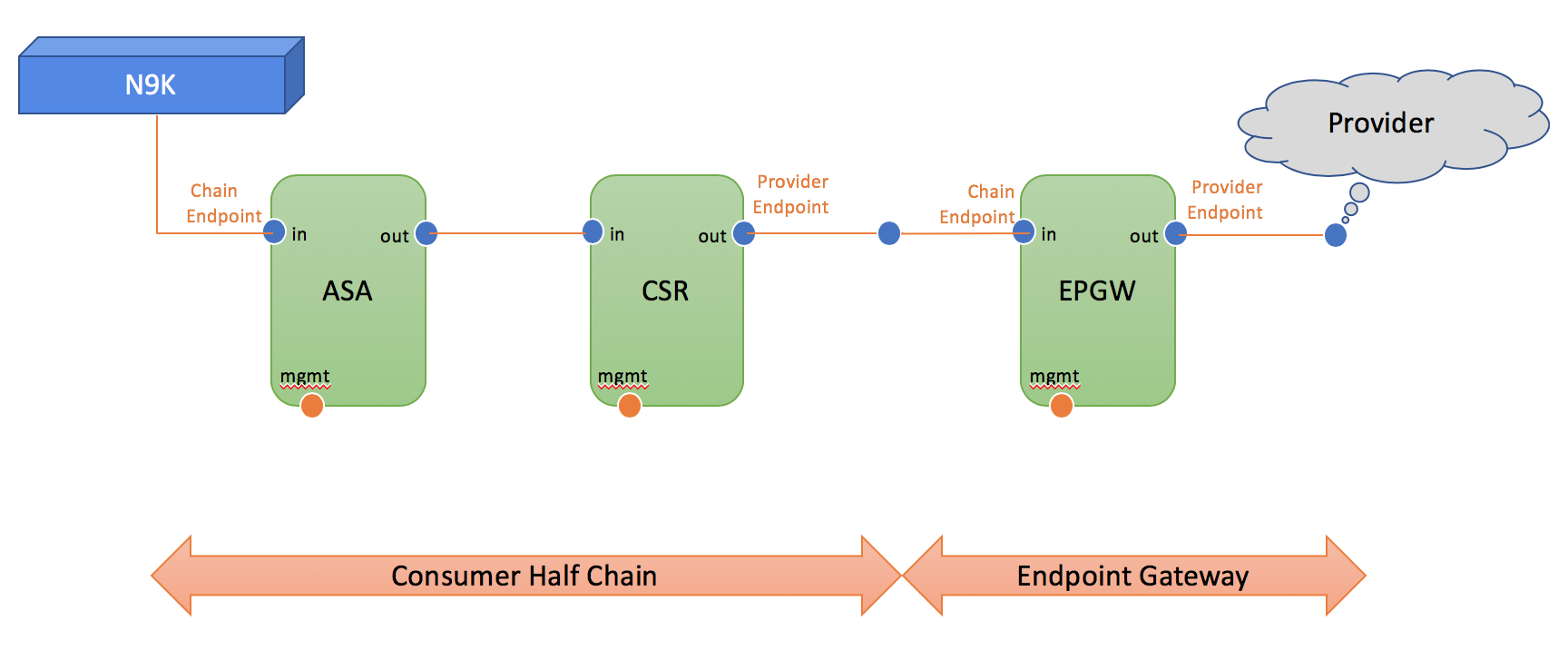
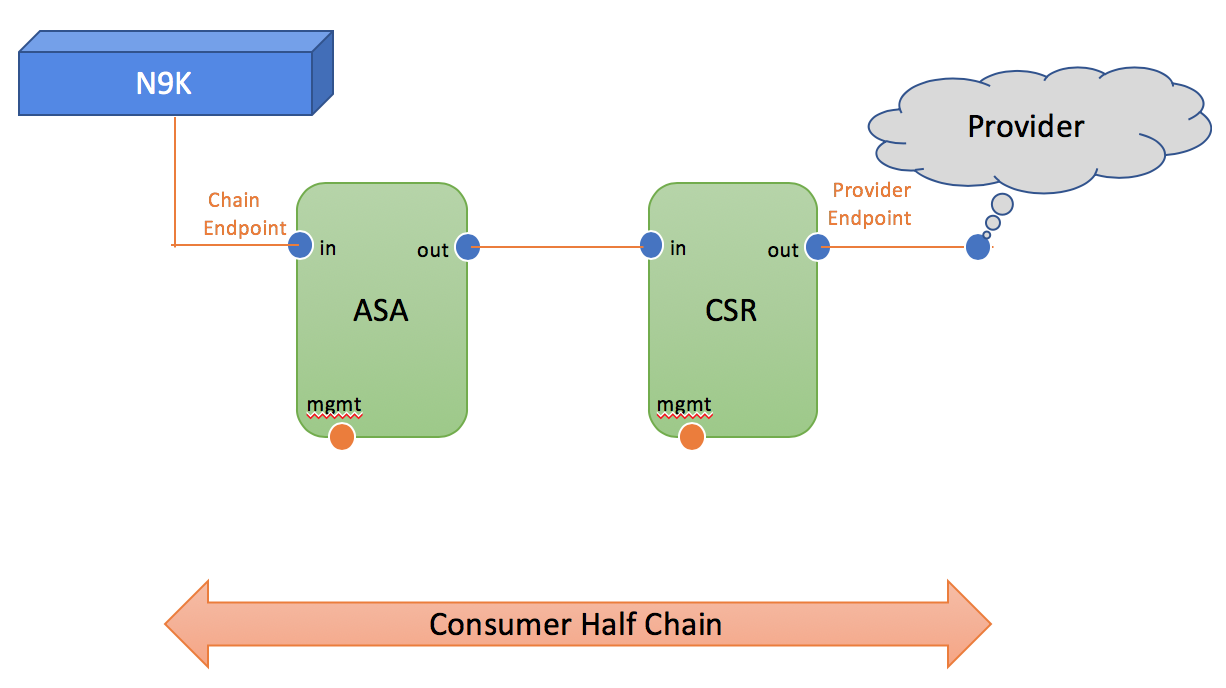


Figure 24. Inline Service Chain



### Sample Payloads

**Consumer Half Chain - Shared Gateway**

<config xmlns="<http://tail-f.com/ns/config/1.0>">  
<sae-site xmlns="<http://com/cisco/nso/sae-core-fp-cfs>">  
  <site>cisco</site>  
  <service-chain>  
    <name>CSR-to-ASA</name>  
    <deployment-name>NSD-CSR-ASA</deployment-name>  
    <type>consumer</type>  
    <consumer-end-point>  
      <external-end-point>ep1</external-end-point>  
      <shared-gateway>  
        <endpoint-gateway-vnf>gateway-vnf-1</endpoint-gateway-vnf>  
      </shared-gateway>  
    </consumer-end-point>  
  </service-chain>  
</sae-site>  
</config>

**Consumer Half Chain - Inline Service Chain**

<config xmlns="<http://tail-f.com/ns/config/1.0>">  
<sae-site xmlns="<http://com/cisco/nso/sae-core-fp-cfs>">  
  <site>cisco</site>  
  <service-chain>  
    <name>CSR-to-ASA-Inline</name>  
    <deployment-name>NSD-CSR-ASA</deployment-name>  
    <type>consumer</type>  
    <consumer-end-point>  
      <external-end-point>ep1</external-end-point>  
      <connectivity>  
        <service-access-point>from-consumer</service-access-point>  
        <ip>  
          <address>[172.28.112.21](https://confluence-eng-sjc1.cisco.com/conf/display/NSOUS/172.28.112.21)</address>  
          <vnf-profile>consumer-csr</vnf-profile>  
        </ip>  
      </connectivity>  
    </consumer-end-point>  
  </service-chain>  
</sae-site>  
</config>

**Provider Half Chain - Shared Gateway**

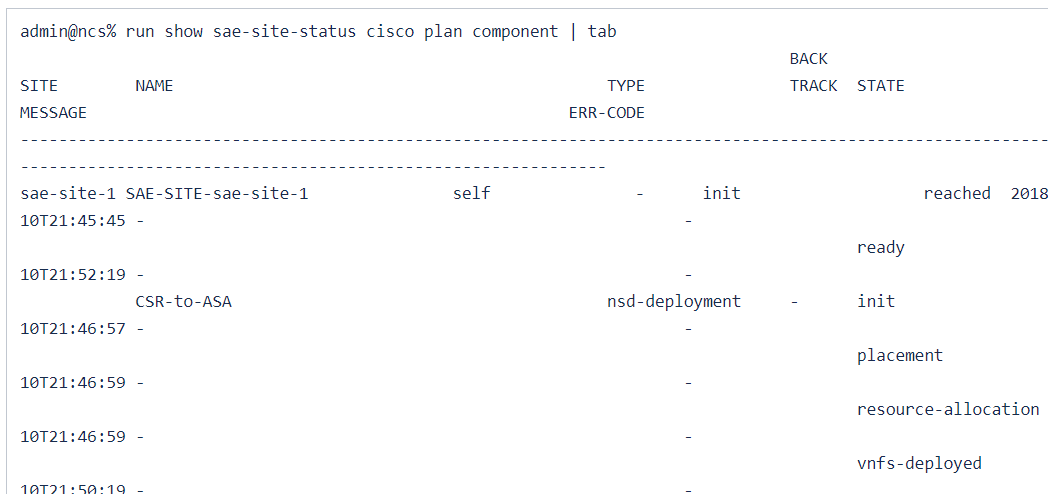
<config xmlns="<http://tail-f.com/ns/config/1.0>">  
    <sae-site xmlns="<http://com/cisco/nso/sae-core-fp-cfs>">  
      <site>cisco</site>  
      <service-chain>  
        <name>CSR-to-ASA-Provider</name>  
        <deployment-name>NSD-CSR-ASA-Provider</deployment-name>  
        <type>provider</type>  
        <provider-end-point>  
          <external-end-point>ep2</external-end-point>  
          <shared-gateway>  
            <endpoint-gateway-vnf>gateway-vnf-2</endpoint-gateway-vnf>  
          </shared-gateway>  
        </provider-end-point>  
      </service-chain>  
    </sae-site>  
</config>

**Provider Half Chain - Inline Service Chain**

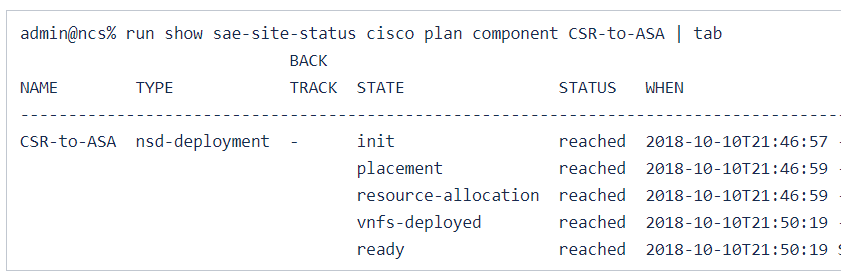
<config xmlns="<http://tail-f.com/ns/config/1.0>">  
<sae-site xmlns="<http://com/cisco/nso/sae-core-fp-cfs>">  
  <site>cisco</site>  
  <service-chain>  
    <name>CSR-to-ASA-Provider-Inline</name>  
    <deployment-name>NSD-CSR-ASA-Provider</deployment-name>  
    <type>provider</type>  
    <provider-end-point>  
      <external-end-point>ep2</external-end-point>  
      <connectivity>  
        <service-access-point>from-provider</service-access-point>  
        <ip>  
          <address>[172.28.112.31](https://confluence-eng-sjc1.cisco.com/conf/display/NSOUS/172.28.112.31)</address>  
          <vnf-profile>provider-csr</vnf-profile>  
        </ip>  
      </connectivity>  
    </provider-end-point>  
  </service-chain>  
</sae-site>  
</config>

### Diagnosis

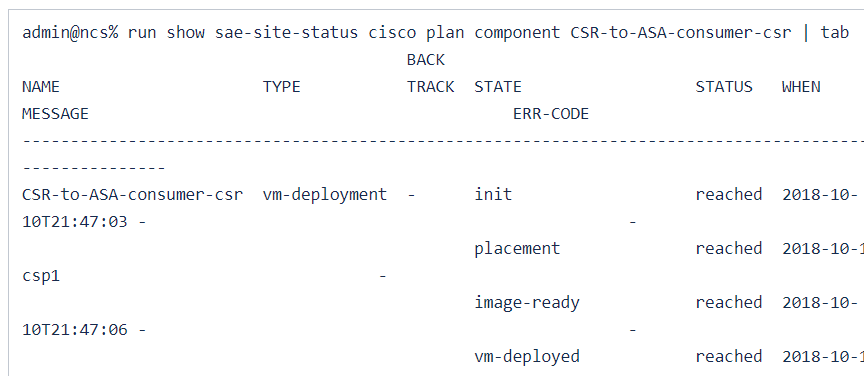
Check the plan for status of service chain and all the VNFs used inside Service Chain:



To check the status of Service Chain, show the plan with component name as service-chain name:



To check the status of any VNF used in the Service Chain, show the plan with the component name of VNF:



### Troubleshooting

Check *ncs-java-vm.log, ncs-python-vm-tailf-etsi-rel2-nfvo-csp.log, ec\_tailf\_nfvo\_csp.log and netconf-<csp\_name>.trace* to troubleshoot any issues while deploying service chains.

## Stitching Service Chain

### Overview

The two ends of Stitching Service Chain can be either of:

* Two Half Service Chains

Or

* One Half Service Chain and Endpoint Gateway

Or

* Two Endpoint Gateways

**Yang Model**

Container 'consumer-end' and 'provider-end' has choice 'half-chain-or-epgw'.

Case 'half-chain': populates the half service chains for user selection.

Case 'shared-gateway': populates the Endpoint Gateways for user selection.

**Service-chain Yang Model**

+--rw sae-site\* [site]     
|  +--rw stitching-service\* [name]  
|  |  +--rw name                        string  
|  |  +--rw consumer-end  
|  |  |  +--rw (half-chain-or-epgw)?  
|  |  |     +--:(half-chain)  
|  |  |     |  +--rw service-chain     -> ../../../service-chain/name  
|  |  |     +--:(epgw)  
|  |  |        +--rw shared-gateway!  
|  |  |           +--rw endpoint-gateway-vnf    -> ../../../../endpoint-gateway-vnf/name  
|  |  |           +--rw external-end-point?     -> deref(../endpoint-gateway-vnf)/../external-end-point/name  
|  |  |           +--rw gateway-access-point?   -> deref(../external-end-point)/../connectivity/next-hop-access-point/gateway-access-point  
|  |  +--rw provider-end  
|  |  |  +--rw (half-chain-or-epgw)?  
|  |  |     +--:(half-chain)  
|  |  |     |  +--rw service-chain     -> ../../../service-chain/name  
|  |  |     +--:(epgw)  
|  |  |        +--rw shared-gateway!  
|  |  |           +--rw endpoint-gateway-vnf    -> ../../../../endpoint-gateway-vnf/name  
|  |  |           +--rw external-end-point?     -> deref(../endpoint-gateway-vnf)/../external-end-point/name  
|  |  |           +--rw gateway-access-point?   -> deref(../external-end-point)/../connectivity/next-hop-access-point/gateway-access-point  
|  |  +--rw var\* [name]  
|  |  |  +--rw name             variableName  
|  |  |  +--rw (valtype)?  
|  |  |     +--:(val)  
|  |  |     |  +--rw val              string  
|  |  |     +--:(encrypted-val)  
|  |  |        +--rw encrypted-val    tailf:aes-cfb-128-encrypted-string

### Prerequisites/Dependencies

SAE Site has been deployed.

Half Service Chain and/or Endpoint Gateway to be used in Stitching Service has been deployed.

**Sample Payloads**

1. Stitching of two half service chains

**Stitching SC to SC**

<config xmlns="<http://tail-f.com/ns/config/1.0>">  
<sae-site xmlns="<http://com/cisco/nso/sae-core-fp-cfs>">  
  <site>cisco</site>  
  <stitching-service>  
    <name>stitch-sc-sc</name>  
    <consumer-end>  
      <service-chain>CSR-to-ASA</service-chain>  
    </consumer-end>  
    <provider-end>  
      <service-chain>CSR-to-ASA-Provider</service-chain>  
    </provider-end>  
  </stitching-service>  
</sae-site>  
</config>

1. Stitching of half service chain with endpoint gateway

**Stitching SC to EPGW**

<config xmlns="<http://tail-f.com/ns/config/1.0>">  
<sae-site xmlns="<http://com/cisco/nso/sae-core-fp-cfs>">  
  <site>cisco</site>  
  <stitching-service>  
    <name>stitch-sc-ep</name>  
    <consumer-end>  
      <service-chain>CSR-to-ASA</service-chain>  
    </consumer-end>  
    <provider-end>  
      <shared-gateway>  
        <endpoint-gateway-vnf>gateway-vnf-1</endpoint-gateway-vnf>  
        <external-end-point>ep1</external-end-point>  
        <gateway-access-point>cp2</gateway-access-point>  
      </shared-gateway>  
    </provider-end>  
  </stitching-service>  
</sae-site>

1. Stitching of endpoint gateway with endpoint gateway

**Stitching EPGW to EPGW**

<config xmlns="<http://tail-f.com/ns/config/1.0>">  
<sae-site xmlns="<http://com/cisco/nso/sae-core-fp-cfs>">  
  <site>cisco</site>  
  <stitching-service>  
    <name>stitch-ep-ep</name>  
    <consumer-end>  
      <shared-gateway>  
        <endpoint-gateway-vnf>gateway-vnf-1</endpoint-gateway-vnf>  
        <external-end-point>ep1</external-end-point>  
        <gateway-access-point>cp2</gateway-access-point>  
      </shared-gateway>  
    </consumer-end>  
    <provider-end>  
      <shared-gateway>  
        <endpoint-gateway-vnf>gateway-vnf-2</endpoint-gateway-vnf>  
        <external-end-point>ep2</external-end-point>  
        <gateway-access-point>cp4</gateway-access-point>  
      </shared-gateway>  
    </provider-end>  
  </stitching-service>  
</sae-site>

### Diagnosis

Check the plan for status of Stitching Service Chain:

**Plan: SC to SC**

admin@ncs% run show sae-site-status cisco plan component stitch-sc-sc | tab  
                              BACK  
NAME          TYPE            TRACK  STATE                  STATUS   WHEN                 MESSAGE                          ERR-CODE  
-------------------------------------------------------------------------------------------------------------------------------------  
stitch-sc-sc  nsd-deployment  -      init                   reached  2018-10-18T23:18:45  -                                -  
                                     switch-configurations  reached  2018-10-18T23:18:48  -                                -  
                                     ready                  reached  2018-10-18T23:18:48  Stitching Service Chain created  -

**Plan: SC to EPWG**

admin@ncs% run show sae-site-status cisco plan component stitch-sc-ep | tab  
                              BACK  
NAME          TYPE            TRACK  STATE                  STATUS   WHEN                 MESSAGE                          ERR-CODE  
-------------------------------------------------------------------------------------------------------------------------------------  
stitch-sc-ep  nsd-deployment  -      init                   reached  2018-10-18T23:06:22  -                                -  
                                     switch-configurations  reached  2018-10-18T23:06:30  -                                -  
                                     ready                  reached  2018-10-18T23:06:30  Stitching Service Chain created

**Plan: EPGW to EPWG**

admin@ncs% run show sae-site-status cisco plan component stitch-ep-ep | tab  
                              BACK  
NAME          TYPE            TRACK  STATE                  STATUS   WHEN                 MESSAGE                          ERR-CODE  
-------------------------------------------------------------------------------------------------------------------------------------  
stitch-ep-ep  nsd-deployment  -      init                   reached  2018-10-18T23:17:10  -                                -  
                                     switch-configurations  reached  2018-10-18T23:17:15  -                                -  
                                     ready                  reached  2018-10-18T23:17:15  Stitching Service Chain created  -

### Troubleshooting

Check *ncs-java-vm.log, ncs-python-vm-tailf-etsi-rel2-nfvo-csp.log, ec\_tailf\_nfvo\_csp.log and netconf-<csp\_name>.trace* to troubleshoot any issues while deploying service chains.

# SAE Core FP Features

The following chapter gives a description of SAE features.

## Introduction

SAE Core FP features are described in the following sections:

## Service Plan

### Overview

The life-cycle of a SAE service is more complex, there is also a need to report the progress of the service in addition to the success or failure of individual transactions.

Plan represents the operational data of the service, covering progress of service and its entities.

**Entity State and Status Interpretation:**

|  | **State** | **Status** | **Comments** |
| --- | --- | --- | --- |
| 1 | init | not-reached | Service creation has not been requested. |
| 2 | init | reached | Service creation has been requested. |
| 3 | ready | not-reached | Service creation has been requested and in one of the intermediate step. |
| 4 | ready | reached | Service creation has been completed. |
| 5 | ready | failed | Service creation has been failed. |
| 6 | placement | not-reached | Placement for VNFs on the CSPs has been requested. |
| 7 | placement | reached | Placement is completed/ VNF is placed on a CSP. |
| 8 | placement | failed | Placement failed/VNF was not able to placed on a CSP. |
| 9 | image-ready | not-reached | image-ready has not reached. |
| 10 | image-ready | reached | image-ready was reached. |
| 11 | image-ready | failed | image-ready was failed because of any reason. |
| 12 | vm-deployed | not-reached | vm was not successfully deployed in ESC. |
| 13 | vm-deployed | reached | vm was deployed in ESC. |
| 14 | vm-deployed | failed | vm deploy was failed on ESC. |
| 15 | switch-configurations | not-reached | Switch configuration was not complete. |
| 16 | switch-configurations | reached | Switch configuration was completed without any NED error. |
| 17 | switch-configurations | failed | Switch configuration failed. |
| 18 | vm-alive | not-reached | vm-alive has not received. |
| 19 | vm-alive | reached | vm-alive from ESC was received. |
| 20 | vm-alive | failed | vm-alive failed from ESC. |
| 21 | day1-configurations | not-reached | If there is no day1 specified then status will be not-reached/ if its not yet configured. |
| 22 | day1-configurations | reached | day1-configuration is applied on VNFs without any NED error. |
| 23 | day1-configurations | failed | application of day1-configuration on VNFs failed. |
| 24 | resource-allocation | not-reached | resource-allocation has not reached yet. |
| 25 | resource-allocation | reached | resource-allocation was reached. |
| 26 | resource-allocation | failed | resource-allocation failed. |
| 27 | vnfs-deployed | not-reached | All vnfs in a service-chain has not been deployed yet. |
| 28 | vnfs-deployed | reached | All vnfs in a service-chain are deployed. |
| 29 | vnfs-deployed | failed | vnf-deployment failed for one or more vnfs in a service-chain. |

### Sample Payloads

How to View Plan

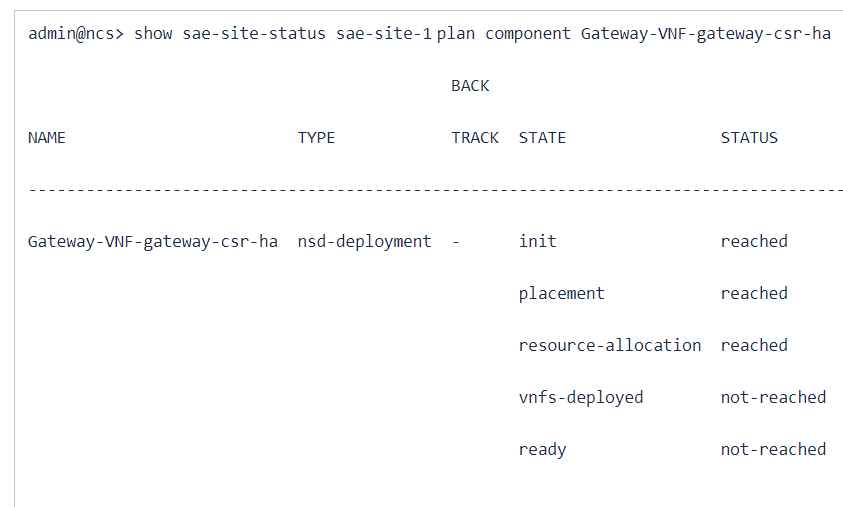
**Plan Component State**

CLI to View all Plan Components State



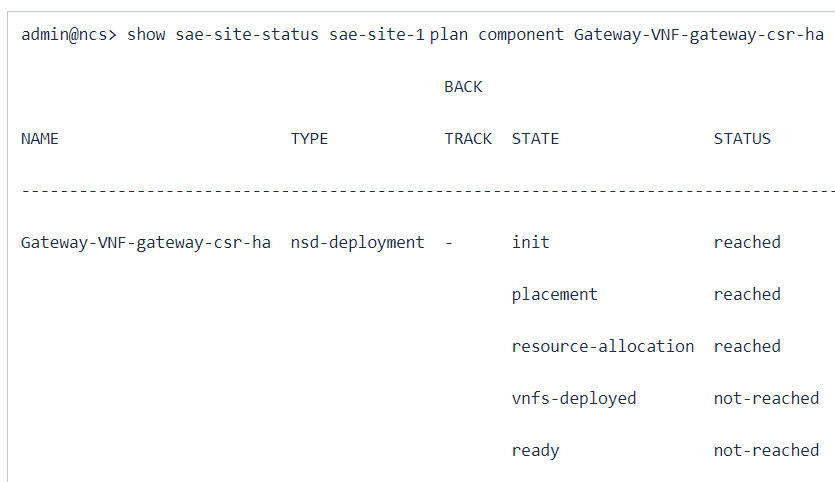
**Plan Component**

CLI to View all Plan Component details



**Plan States**

All the states **init, ready, placement, vm-deployed, switch-configurations,day1-configurations, image-ready,resource-allocation,vnfs-deployed states** are shown in plan component state.



### Diagnosis

### Troubleshooting

## EPGW HA

### Overview

### Activate

### Modify

### Delete

### Sample Payloads

### Diagnosis

### Troubleshooting

## Service Chain HA

### Overview

### Activate

### Modify

### Delete

### Sample Payloads

### Diagnosis

### Troubleshooting

## Inline Service Chain

### Overview

### Activate

### Modify

### Delete

### Sample Payloads

### Diagnosis

Troubleshooting

## Controller Based VNF

### ASAv Extension Support

**Description**

There is an extension adaptor to add support for ASAv HA deployment. SAE CFP contains this extension package for ASAv VNF:

package: cisco-sae-asav-extension

**Yang Model**

There are few model updates to use ASAv extension for HA

**Model updates:**

VNFD-DEPLOYMENT is extended to indicate what "type" a VNF is, and the "HA-settings" can then be tied up with pre extensions to be used during the VNF deployment. Such a definition can be as below:

Under asav-ha-settings,

* Specify role for ASAv vnfd-deployment [active or standby]
* Mention the connection point name as per the ASAv vnfd

**vnfd-deployment payload update**

<config xmlns="<http://tail-f.com/ns/config/1.0>">  
  <sae-catalog xmlns="<http://com/cisco/nso/sae-core-fp-cfs>">  
    <name>sae-basic</name>  
      <vnfd-deployment>  
        <name>Active-ASAV</name>  
        .  
        .  
         <vnf-type xmlns:cisco-sae-asav-extension="<http://com/cisco/nso/sae/extension/asav>">cisco-sae-asav-extension:asav</vnf-type>  
         <asav-ha-settings>  
            <ha-role>active</ha-role>  
            <failover-connection-point>failover</failover-connection-point>  
        </asav-ha-settings>  
        .  
        .  
      </vnfd-deployment>  
   </sae-catalog>  
</config>

**Extension Package Registration :**

ASAv extension registration is completed as part of installation. You should verify extension registration before initiating ASAv HA deployment.

Verifiy using below NSO command:

* admin@ncs% show extension-grouping

**ASAv extension package registration**

<config xmlns="<http://tail-f.com/ns/config/1.0>">  
    <extension-grouping xmlns="<http://com/cisco/nso/sae-core-fp-extension>">  
  <vnf-type xmlns:cisco-sae-asav-extension="<http://com/cisco/nso/sae/extension/asav>">cisco-sae-asav-extension:asav</vnf-type>  
  <create>  
    <pre xmlns:cisco-sae-asav-extension="<http://com/cisco/nso/sae/extension/asav>">cisco-sae-asav-extension:asav-pre-extension</pre>  
  </create>  
</extension-grouping>  
</config>

#### Prerequisites/Dependencies

Refer to the below prerequisite steps to deploy ASAv HA VNF's using SAE CFP.

1. ESC setup : To enable ASAv HA support:
2. Upload the following files to the ESC VM's /var/tmp/ folder:

* asa\_ha\_vmalive.exp
* asa\_ha\_vmalive\_metric.xml
* dmam.py

1. Make the custom metric collector script "asa\_ha\_vmalive.exp" executable. From the ESC VM's console/terminal:

$ chmod +x /var/tmp/asa\_ha\_vmalive.exp

1. Register the custom metric definition with ESC. On an ESC VM terminal (VIM VM console, ssh session, etc), run:

$ /var/tmp/dmam.py create-metric --payload\_xml /var/tmp/asa\_ha\_vmalive\_metric.xml

1. NSO bootstrap details
2. Please make sure ASAv extension package is installed.
3. ASAv extension must be registered, ncs cli to verify:

admin@ncs% show extension

1. ESC device templates should be present in NSO templates list:

admin@ncs% show devices template esc-vmalive-ha-kpi-data-nic-0

1. Update vnfd-deployment in sae-catalog. Refer to the entry below in the vnfd-deployment to activate extension package.

### Usage with Sample Payloads

Sample catalog payload for ASAv Active & Standby vnf

**sae-catalog payload for ASAv**

<config xmlns="<http://tail-f.com/ns/config/1.0>">  
  <sae-catalog xmlns="<http://com/cisco/nso/sae-core-fp-cfs>">  
    <name>cisco-catalog</name>  
    <vnfd-deployment>  
      <name>ASA-Active-Sriov-Dep</name>  
      <vnfd>VNFD\_ASA</vnfd>  
      <vdu>VDU\_ASA\_SRIOV\_HA</vdu>  
      <mode>routed</mode>  
      <bootup-time>600</bootup-time>  
      <recovery-wait-time>90</recovery-wait-time>  
      <vnf-authgroup>{$VNF\_AUTHGROUP}</vnf-authgroup>  
      <day0>  
        <destination>day0-config</destination>  
        <url>{$DAY0\_FILE\_URL}</url>  
        <var>  
          <name>{$DAY0\_CFG\_VAR\_1}</name>  
          <val>{$DAY0\_CFG\_VAR\_1\_VALUE}</val>  
        </var>  
        <var>  
          <name>{$DAY0\_CFG\_VAR\_2}</name>  
          <val>{$DAY0\_CFG\_VAR\_2\_VALUE}</val>  
        </var>  
      </day0>  
      <vnf-type xmlns:cisco-sae-asav-extension="<http://com/cisco/nso/sae/extension/asav>">cisco-sae-asav-extension:asav</vnf-type>  
      <asav-ha-settings>  
        <ha-role>active</ha-role>  
        <failover-connection-point>FAILOVER</failover-connection-point>  
      </asav-ha-settings>  
    </vnfd-deployment>  
    <vnfd-deployment>  
      <name>ASA-Standby-Sriov-Dep</name>  
      <vnfd>VNFD\_ASA</vnfd>  
      <vdu>VDU\_ASA\_SRIOV\_HA</vdu>  
      <mode>routed</mode>  
      <bootup-time>600</bootup-time>  
      <recovery-wait-time>90</recovery-wait-time>  
      <vnf-authgroup>{$VNF\_AUTHGROUP}</vnf-authgroup>  
      <day0>  
        <destination>day0-config</destination>  
        <url>{$DAY0\_FILE\_URL}</url>  
        <var>  
          <name>{$DAY0\_CFG\_VAR\_1}</name>  
          <val>{$DAY0\_CFG\_VAR\_1\_VALUE}</val>  
        </var>  
        <var>  
          <name>{$DAY0\_CFG\_VAR\_2}</name>  
          <val>{$DAY0\_CFG\_VAR\_2\_VALUE}</val>  
        </var>  
      </day0>  
      <vnf-type xmlns:cisco-sae-asav-extension="<http://com/cisco/nso/sae/extension/asav>">cisco-sae-asav-extension:asav</vnf-type>  
      <asav-ha-settings>  
        <ha-role>standby</ha-role>  
        <failover-connection-point>FAILOVER</failover-connection-point>  
      </asav-ha-settings>  
    </vnfd-deployment>  
  </sae-catalog>  
</config>

**Payloads Validation**

We do have some payload validation for ASAv HA vnfd-deployment

* If vnf-type is set as asav, it will enable asav-ha-settings
* asav-ha-settings required both inputs [ha-role & failover-connection-point], it is mandatory input

**Verification**

* Use below steps to make sure ESC metric registration got registered on ESC using below command, it used for monitoring ASAv VNF
* curl -X GET -H "Accept: Application/xml" http://localhost:8080/ESCManager/internal/dynamic\_mapping/metrics | xmllint --format -
* Please day0 config file are correctly provided, refer service deployment in build for reference
* Verify ASAv deployment sae plan status for ready reached status

### Troubleshooting and Diagnostics

| **Error** | **Possible Reason** | **Troubleshooting** |
| --- | --- | --- |
| SAE plan failed to marked ASAv HA vnf ready reached status | * ASAv vnf is reachable from nso * day0 configs are incorrect * ESC failed in deployment | * verify day0 file, is it  correctly configure or not on ASAv vnf * check the reachability from NSO |
| ASAv HA vnf is failing in recovering | N9K configs are missing | verify the failover state on ASAv vnf while recovering |

### Packaging, Installation and Versioning

nfvo       X   -        -              -        -        -        -           -        -      -       
tailf-etsi-rel2-nfvo-csp   X   -        -              -        -        -        -           -        -      -       
   
[ok][2018-10-22 10:53:02]  
admin@ncs> show packages package package-version  
                           PACKAGE   
NAME                       VERSION   
------------------------------------  
SAE-UI                     1.0.0     
avi-vantage                1.4.5     
cisco-asa                  6.3       
cisco-extension-framework  1.0.0     
cisco-fmc                  1.0.4     
cisco-ios                  6.4.1     
cisco-nx                   5.7.2     
cisco-sae-asav-extension   1.0.0     
cisco-sae-avi-extension    1.0.0     
cisco-sae-core-fp          1.0.0     
cisco-sae-core-fp-cfs      1.0.0     
cisco-sae-core-fp-common   1.0.0     
cisco-sae-core-fp-release  1.0.0     
cisco-sae-core-fp-status   1.0.0     
cisco-sae-diagnosis        1.0.0     
cisco-sae-ftdv-extension   1.0.0     
cisco-sae-itd-service      1.0.0     
cisco-waas                 2.0.1     
core-fp-common             1.5.0     
csp                        1.0.0     
csp-vim                    1.0.0     
custom-template-utils      0.1.0     
day1-templates             1.1.0     
diagnosis                  0.1.0     
esc                        4.1.1.6   
infra-discovery            1.0.0     
paloalto-panos\_cli         4.0.1     
resource-manager           3.3.1     
tailf-etsi-rel2-nfvo       3.2.0     
tailf-etsi-rel2-nfvo-csp   1.0.0

### AVI-SE Extension Support

**Description**

We have an extension adaptor to add support for AVI-SE deployment. SAE CFP contains this new extension package for AVI-SE VNF.

package: cisco-sae-avi-extension

AVI-SE deployment requires registration with controller. You have to provide inputs to enable extension while deploying AVI-SE. VNFtype and contoller-ip are required in vnfd-deployment.

For AVI-SE, the extension package takes care of registration during creation, and removal of registration while doing delete.

**Yang Model**

There are few model updates to use AVI-SE extension for deployment:

Model updates:

VNFD-DEPLOYMENT is extended to indicate the "**type**" of VNF and the controller device can then be tied up with pre-extensions to be used during the VNF deployment. Refer to the code block below:

<config xmlns="<http://tail-f.com/ns/config/1.0>">  
  <sae-catalog xmlns="<http://com/cisco/nso/sae-core-fp-cfs>">  
    <name>CISCO-CATALOG</name>  
      <vnfd-deployment>  
        <name>VNFD\_DEPL\_AVISE\_CON</name>  
        <vnfd>VNFD\_AVISE</vnfd>  
        <vdu>VDU\_AVISE</vdu>  
        <intangible/>  
        <var>  
          <name>METRIC\_OCC\_TRUE</name>  
          <val>50</val>  
        </var>  
        <var>  
          <name>METRIC\_VALUE</name>  
          <val>50</val>  
        </var>  
        <var>  
          <name>POLL\_FREQUENCY</name>  
          <val>15</val>  
        </var>  
        .  
        .  
        <day0>  
          <destination>[avi\_meta\_se.yml</destination](https://confluence-eng-sjc1.cisco.com/conf/display/NSOUS/avi_meta_se.yml%3C/destination)>  
          <url>http://9.9.[9.210:](https://confluence-eng-sjc1.cisco.com/conf/9.210:)8080/day0/SJ/[avi\_meta\_se.yml](https://confluence-eng-sjc1.cisco.com/conf/display/NSOUS/avi_meta_se.yml)</url>  
        </day0>  
        <vnf-type xmlns:cisco-sae-avi-extension="<http://com/cisco/nso/sae/extension/avi>">cisco-sae-avi-extension:avi</vnf-type>  
        <controller>AVI-60</controller>  
      </vnfd-deployment>  
  </sae-catalog>  
</config>

**Extension Package Registration:**

ASAv extension registration is completed as part of installation. You should verify extension registration before initiating ASAv HA deployment.

Verfiy using the below NSO command:

* admin@ncs% show extension-grouping

<config xmlns="<http://tail-f.com/ns/config/1.0>">  
  <extension-grouping xmlns="<http://com/cisco/nso/sae-core-fp-extension>">  
    <vnf-type xmlns:cisco-sae-avi-extension="<http://com/cisco/nso/sae/extension/avi>">cisco-sae-avi-extension:avi</vnf-type>  
    <create>  
      <pre xmlns:cisco-sae-avi-extension="<http://com/cisco/nso/sae/extension/avi>">cisco-sae-avi-extension:avi-se-pre-extension</pre>  
      <post xmlns:cisco-sae-avi-extension="<http://com/cisco/nso/sae/extension/avi>">cisco-sae-avi-extension:avi-se-post-extension</post>  
    </create>  
  </extension-grouping>  
</config>

### Prerequisites/Dependencies

Please refer to the prerequisites listed below:

* AVI-SE should be deployed as intangible, as we have no device ned available.
* AVI-SE deployment require higher METRIC\_OCC\_TRUE in vnfd-deployment. Sometimes the controller take longer time in registration. Please refer to the sample vnfd-deployment mentioned above.

### Usage with Sample Payloads

Sample payload are available at the below URL:

https://confluence-eng-sjc1.cisco.com/conf/pages/viewpage.action?pageId=79598815

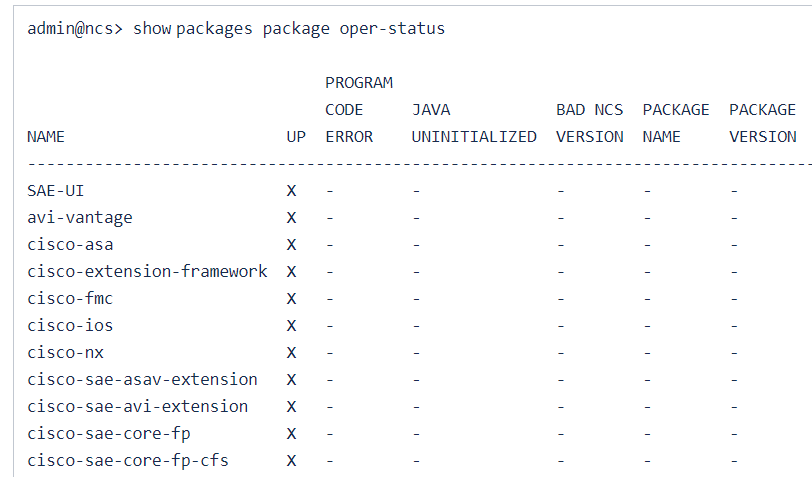
### Verification

Capture verification steps if available, capture test scenario as well

### Troubleshooting and Diagnostic

| **Error** | **Possible Reason** | **Troubleshooting** |
| --- | --- | --- |
| SAE plan failed to marked ASAv HA vnf ready reached status | * ASAv vnf is reachable from nso * day0 configs are incorrect * ESC failed in deployment | * verify day0 file, is it  correctly configure or not on ASAv vnf * check the reachability from NSO |
| ASAv HA vnf is failing in recovering | N9K configs are missing | verify the failover state on ASAv vnf while recovering |

### Packaging, Installation and Versioning



### FTDv Extension Support

## VNF Replacement

### Overview

SAE core FP provides capabilities to replace a VNF in an existing service for generally maintenance purpose.

### Prerequisites/Dependencies

Existing VNF or Service Chain

### Usage with Sample Payloads for Replacing Switch

As far as there is one leaf-switch, in a standalone mode if at all the 2nd leaf-switch goes on a maintenance mode, things will work as it is.

Services might have affected even though we have a different switch in DCI mode. Remember SRIOV ports are connected to only one switch.

**Procedure:**

1. Southbound lock N9K.

**admin@ncs% set devices device nexus0 state admin-state southbound-locked  
[ok][2018-10-22 10:34:06]  
  
[edit]  
admin@ncs% commit  
Commit complete.  
[ok][2018-10-22 10:34:09]  
  
[edit]  
admin@ncs%**

1. Replace the N9K.
2. Southbound unlock N9K.

**admin@ncs% set devices device nexus0 state admin-state unlocked  
[ok][2018-10-22 10:34:31]  
  
[edit]  
admin@ncs% commit  
Commit complete.  
[ok][2018-10-22 10:34:32]  
  
[edit]  
admin@ncs%**

1. Fetch ssh-host-keys.

**admin@ncs% request devices fetch-ssh-host-keys device [ nexus0 ]  
fetch-result {  
    device nexus0  
    result unchanged  
    fingerprint {  
        algorithm ssh-rsa  
        value 1d:c4:44:34:25:0d:71:84:be:a2:9b:1d:0f:12:d8:4e  
    }  
}  
[ok][2018-10-22 10:30:38]**

1. Connect to N9K

**admin@ncs% request devices connect device [ nexus  
Possible completions:  
  nexus0  nexus1  nexus2  nexus3  
admin@ncs% request devices connect device [ nexus0 ]  
connect-result {  
    device nexus0  
    result true  
    info (admin) Connected to nexus0 -**[**127.0.0.1:10028**](http://127.0.0.1:10028/) **}  
[ok][2018-10-22 10:30:48]**

1. Sync-to the configs (Assuming new N9K has all the day -1 configuration required or NSO knows all the day -1 configuration via sync-from done on earlier device).

**admin@ncs% request devices sync-to device [ nexus0 ]  
sync-result {  
    device nexus0  
    result true  
}  
[ok][2018-10-22 10:31:01]**

## SAE Actions

The following sections describes various types of SAE actions.

### Overview

The document covers different action commands available as ***sae-site-actions*** and ***sae-custom actions***, their usage and purpose.

### Site Actions

There are three sae-site-actions commands available.

**recover-vnf-on-csp**

When a VNF deployment on a CSP goes wrong for any reason, you can recover that VNF on that CSP or any other CSP available, using this action command. This action command deletes the faulty VNF on the CSP and deploys a new VNF with same characteristics (determined by NSD and VNFD) on that CSP or any other available CSP.

The selection of CSP is determined through resources available to deploy the VNF in CSPs and affinity-or-anti-affinity-group of VNF's NSD (Please refer NSD documents for more details). CSP selection is handled by resource-orchestration module and recovery action is handled by csp-vim module in SAE.

**services-on-csp**

User can check all the service deployment request in a particular CSP device, using this action command.

**undep-redep-services-on-csp**

You can undeploy-redeploy services deployed in a particular CSP using this action command. This action command fetches all the service deployment request exists in a CSP device and executes custom-undeploy-redeploy on those services.

If a service's VNFs are spread over multiple CSP devices, then this action command on a particular CSP, which is part of the service will custom-redeploy-undeploy the whole service. So the VNF's part of the service but deployed on other CSPs are goes through undeploy-redeploy process.

**Yang Model**

+--rw sae-site-actions  
    |  +---x services-on-csp  
    |  |  +---w input  
    |  |  |  +---w csp-device    string  
    |  |  +--ro output  
    |  |     +--ro status    string  
    |  +---x undep-redep-services-on-csp  
    |  |  +---w input  
    |  |  |  +---w csp-device    string  
    |  |  +--ro output  
    |  |     +--ro status    string  
    |  +---x recover-vnf-on-csp  
    |     +---w input  
    |     |  +---w sae-site                        -> /sae-site/site  
    |     |  +---w (service-name)?  
    |     |  |  +--:(half-chain)  
    |     |  |  |  +---w service-chain?                  -> deref(../sae-site)/../service-chain/name  
    |     |  |  +--:(epgw-name)  
    |     |  |     +---w endpoint-gateway-vnf-service?   -> /sae-site[cisco-sae-core-fp-cfs:site=current()/../sae-site]/endpoint-gateway-vnf/name  
    |     |  +---w vnf-profile?                    -> /nfvo-rel2:nfvo/nsd/deployment-flavor/vnf-profile/id  
    |     +--ro output  
    |        +--ro status    string

#### Prerequisites/Dependencies

SAE services must be deployed.

#### Sample Payloads

**recover-vnf-on-csp**

request sae-site-actions recover-vnf-on-csp sae-site $SAE\_SITE endpoint-gateway-vnf-service $EPGW vnf-profile $VNF\_PROFILE

| **Variable Name** | **Variable Description** |
| --- | --- |
| SAE\_SITE | sae-site name. Mandatory. |
| EPGW | End point gateway vnf service name. Mandatory |
| SRV\_CHN | Service chain name. Mandatory |
| VNF\_PROFILE | Vnf profile used in provided end point gateway/service chain. Mandatory |

**Sample Output**

admin@ncs> request sae-site-actions recover-vnf-on-csp sae-site sae-site-1 endpoint-gateway-vnf-service gateway-vnf-1 vnf-profile SAE-EP-GW-CSR-profile  
status : Service VNFs re-deployed

**services-on-csp**

request sae-site-actions services-on-csp csp-device $CSP\_DEVICE\_NAME

**Sample Output**

admin@ncs> request sae-site-actions services-on-csp csp-device csp0  
status : List of Services on CSP  
sae-site[name='sae-site-1'] endpoint-gateway-vnf[name='gateway-vnf-1']  
sae-site[name='sae-site-1'] service-chain[name='CSR-to-ASA']  
sae-site[name='sae-site-1'] service-chain[name='CSR-to-ASA-Provider']  
   
   
admin@ncs> request sae-site-actions services-on-csp csp-device csp1  
status : List of Services on CSP  
sae-site[name='sae-site-1'] service-chain[name='CSR-to-ASA']  
sae-site[name='sae-site-1'] service-chain[name='CSR-to-ASA-Provider']

**undep-redep-services-on-csp**

request sae-site-actions undep-redep-services-on-csp csp-device $CSP\_DEVICE\_NAME

**Sample Output**

admin@ncs> request sae-site-actions undep-redep-services-on-csp csp-device csp0

status :

Starting undeploy & redeploy all Services on CSP device: csp0

sae-site[name='sae-site-1'] service-chain[name='CSR-to-ASA']

sae-site[name='sae-site-1'] service-chain[name='CSR-to-ASA-Provider']

#### Verification

**recover-vnf-on-csp**

1. Check plan of ***vnf-profile***, which is intended to be recovered on CSP. Check for ***message*** of ***placement*** state of the component. The message may have some thing like "***Selected csp: <csp-device-name>***".

admin@ncs> show sae-site-status sae-site-1 plan component Gateway-VNF-gateway-vnf-1-SAE-EP-GW-CSR-profile  
                                                                BACK  
NAME                                             TYPE           TRACK  STATE                  STATUS   WHEN                 MESSAGE                                            ERR-CODE  
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------  
Gateway-VNF-gateway-vnf-1-SAE-EP-GW-CSR-profile  vm-deployment  -      init                   reached  2018-10-08T20:40:30  -                                                  -  
                                                                       placement              reached  2018-10-08T20:40:30  Selected csp: csp0                                 -  
                                                                       image-ready            reached  2018-10-08T20:40:30  -                                                  -  
                                                                       vm-deployed            reached  2018-10-08T20:40:31  VNF is deployed                                    -  
                                                                       switch-configurations  reached  2018-10-08T20:40:32  -                                                  -  
                                                                       vm-alive               reached  2018-10-08T20:40:37  -                                                  -  
                                                                       day1-configurations    reached  2018-10-08T20:40:37  -                                                  -  
                                                                       ready                  reached  2018-10-08T20:40:37  VNF configured with vnfd-deployment->day1-config.  –

1. Execute the command for the ***vnf-profile***. Check Sample payloads section of this document.

admin@ncs> request sae-site-actions recover-vnf-on-csp sae-site sae-site-1 endpoint-gateway-vnf-service gateway-vnf-1 vnf-profile SAE-EP-GW-CSR-profile  
status : Service VNFs re-deployed

1. Again check the plan like in step-1 after awhile. The message may contain a different CSP device name after successful execution of the command and all states of the vnf-profile will be in reached state.

admin@ncs> show sae-site-status sae-site-1 plan component Gateway-VNF-gateway-vnf-1-SAE-EP-GW-CSR-profile  
                                                                BACK  
NAME                                             TYPE           TRACK  STATE                  STATUS   WHEN                 MESSAGE                                            ERR-CODE  
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------  
Gateway-VNF-gateway-vnf-1-SAE-EP-GW-CSR-profile  vm-deployment  -      init                   reached  2018-10-08T20:50:10  -                                                  -  
                                                                       placement              reached  2018-10-08T20:50:10  Selected csp: csp1                                 -  
                                                                       image-ready            reached  2018-10-08T20:50:10  -                                                  -  
                                                                       vm-deployed            reached  2018-10-08T20:50:11  VNF is deployed                                    -  
                                                                       switch-configurations  reached  2018-10-08T20:50:11  -                                                  -  
                                                                       vm-alive               reached  2018-10-08T20:50:17  -                                                  -  
                                                                       day1-configurations    reached  2018-10-08T20:50:17  -                                                  -  
                                                                       ready                  reached  2018-10-08T20:50:17  VNF configured with vnfd-deployment->day1-config.  -

**services-on-csp**

1. Check plan and figure out how many vnf-profiles are on a particular csp device. Check message of placement state of the vnf-profiles and check for message like "***Selected csp: <csp-device-name>***".
2. Check in plan the vnf-profiles belong to which services.
3. Check, whether the services determined in step-2 is matching with output of the action command.

**Example:**

admin@ncs> show sae-site-status sae-site-1 plan | tab | nomore  
                                                                    BACK  
NAME                                             TYPE               TRACK  STATE                  STATUS   WHEN                 MESSAGE                                                  ERR-CODE  
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------  
SAE-SITE-sae-site-1                              self               -      init                   reached  2018-10-08T17:51:20  -                                                        -  
                                                                           ready                  reached  2018-10-08T20:50:27  -                                                        -  
csp0                                             config-deployment  -      init                   reached  2018-10-08T17:51:20  -                                                        -  
                                                                           ready                  reached  2018-10-08T17:51:20  CSP Day1 Config Applied                                  -  
csp0-ntp                                         config-deployment  -      init                   reached  2018-10-08T17:51:20  -                                                        -  
                                                                           ready                  reached  2018-10-08T17:51:20  Template Applied Successfully                            -  
csp1                                             config-deployment  -      init                   reached  2018-10-08T17:51:20  -                                                        -  
                                                                           ready                  reached  2018-10-08T17:51:21  CSP Day1 Config Applied                                  -  
csp1-ntp                                         config-deployment  -      init                   reached  2018-10-08T17:51:20  -                                                        -  
                                                                           ready                  reached  2018-10-08T17:51:21  Template Applied Successfully                            -  
VNFM-ESC-unmanaged-trial                         vm-deployment      -      init                   reached  2018-10-08T17:51:30  -                                                        -  
                                                                           ready                  reached  2018-10-08T17:51:30  Netconf subscription, tenant & csp vim settings created  -  
Gateway-VNF-gateway-vnf-1                        nsd-deployment     -      init                   reached  2018-10-08T18:45:34  -                                                        -  
                                                                           placement              reached  2018-10-08T18:45:36  -                                                        -  
                                                                           resource-allocation    reached  2018-10-08T18:45:36  -                                                        -  
                                                                           vnfs-deployed          reached  2018-10-08T20:50:17  -                                                        -  
                                                                           ready                  reached  2018-10-08T20:50:17  EPGW service created                                     -  
Gateway-VNF-gateway-vnf-1-SAE-EP-GW-CSR-profile  vm-deployment      -      init                   reached  2018-10-08T20:50:10  -                                                        -  
                                                                           placement              reached  2018-10-08T20:50:10  Selected csp: csp1                                       -  
                                                                           image-ready            reached  2018-10-08T20:50:10  -                                                        -  
                                                                           vm-deployed            reached  2018-10-08T20:50:11  VNF is deployed                                          -  
                                                                           switch-configurations  reached  2018-10-08T20:50:11  -                                                        -  
                                                                           vm-alive               reached  2018-10-08T20:50:17  -                                                        -  
                                                                           day1-configurations    reached  2018-10-08T20:50:17  -                                                        -  
                                                                           ready                  reached  2018-10-08T20:50:17  VNF configured with vnfd-deployment->day1-config.        -  
CSR-to-ASA                                       nsd-deployment     -      init                   reached  2018-10-08T20:32:50  -                                                        -  
                                                                           placement              reached  2018-10-08T20:32:51  -                                                        -  
                                                                           resource-allocation    reached  2018-10-08T20:32:52  -                                                        -  
                                                                           vnfs-deployed          reached  2018-10-08T20:36:02  -                                                        -  
                                                                           ready                  reached  2018-10-08T20:36:02  Service Chain created                                    -  
CSR-to-ASA-active-asa-profile                    vm-deployment      -      init                   reached  2018-10-08T20:32:54  -                                                        -  
                                                                           placement              reached  2018-10-08T20:32:54  Selected csp: csp1                                       -  
                                                                           image-ready            reached  2018-10-08T20:32:59  -                                                        -  
                                                                           vm-deployed            reached  2018-10-08T20:34:56  VNF is deployed                                          -  
                                                                           switch-configurations  reached  2018-10-08T20:34:56  -                                                        -  
                                                                           vm-alive               reached  2018-10-08T20:35:21  -                                                        -  
                                                                           day1-configurations    reached  2018-10-08T20:35:22  -                                                        -  
                                                                           ready                  reached  2018-10-08T20:35:22  VNF configured with vnfd-deployment->day1-config.        -  
CSR-to-ASA-consumer-csr                          vm-deployment      -      init                   reached  2018-10-08T20:32:54  -                                                        -  
                                                                           placement              reached  2018-10-08T20:32:54  Selected csp: csp1                                       -  
                                                                           image-ready            reached  2018-10-08T20:32:57  -                                                        -  
                                                                           vm-deployed            reached  2018-10-08T20:34:08  VNF is deployed                                          -  
                                                                           switch-configurations  reached  2018-10-08T20:34:09  -                                                        -  
                                                                           vm-alive               reached  2018-10-08T20:34:29  -                                                        -  
                                                                           day1-configurations    reached  2018-10-08T20:34:30  -                                                        -  
                                                                           ready                  reached  2018-10-08T20:34:30  VNF configured with vnfd-deployment->day1-config.        -  
CSR-to-ASA-standby-asa-profile                   vm-deployment      -      init                   reached  2018-10-08T20:32:54  -                                                        -  
                                                                           placement              reached  2018-10-08T20:32:54  Selected csp: csp0                                       -  
                                                                           image-ready            reached  2018-10-08T20:32:56  -                                                        -  
                                                                           vm-deployed            reached  2018-10-08T20:35:47  VNF is deployed                                          -  
                                                                           switch-configurations  reached  2018-10-08T20:35:51  -                                                        -  
                                                                           vm-alive               reached  2018-10-08T20:35:59  -                                                        -  
                                                                           day1-configurations    reached  2018-10-08T20:36:02  -                                                        -  
                                                                           ready                  reached  2018-10-08T20:36:02  VNF configured with vnfd-deployment->day1-config.        -  
CSR-to-ASA-Provider                              nsd-deployment     -      init                   reached  2018-10-08T20:33:16  -                                                        -  
                                                                           placement              reached  2018-10-08T20:33:19  -                                                        -  
                                                                           resource-allocation    reached  2018-10-08T20:33:20  -                                                        -  
                                                                           vnfs-deployed          reached  2018-10-08T20:37:27  -                                                        -  
                                                                           ready                  reached  2018-10-08T20:37:27  Service Chain created                                    -  
CSR-to-ASA-Provider-active-asa-profile           vm-deployment      -      init                   reached  2018-10-08T20:33:31  -                                                        -  
                                                                           placement              reached  2018-10-08T20:33:31  Selected csp: csp0                                       -  
                                                                           image-ready            reached  2018-10-08T20:33:35  -                                                        -  
                                                                           vm-deployed            reached  2018-10-08T20:36:52  VNF is deployed                                          -  
                                                                           switch-configurations  reached  2018-10-08T20:36:52  -                                                        -  
                                                                           vm-alive               reached  2018-10-08T20:37:02  -                                                        -  
                                                                           day1-configurations    reached  2018-10-08T20:37:02  -                                                        -  
                                                                           ready                  reached  2018-10-08T20:37:02  VNF configured with vnfd-deployment->day1-config.        -  
CSR-to-ASA-Provider-provider-csr                 vm-deployment      -      init                   reached  2018-10-08T20:33:31  -                                                        -  
                                                                           placement              reached  2018-10-08T20:33:31  Selected csp: csp0                                       -  
                                                                           image-ready            reached  2018-10-08T20:33:34  -                                                        -  
                                                                           vm-deployed            reached  2018-10-08T20:36:25  VNF is deployed                                          -  
                                                                           switch-configurations  reached  2018-10-08T20:36:26  -                                                        -  
                                                                           vm-alive               reached  2018-10-08T20:36:35  -                                                        -  
                                                                           day1-configurations    reached  2018-10-08T20:36:36  -                                                        -  
                                                                           ready                  reached  2018-10-08T20:36:36  VNF configured with vnfd-deployment->day1-config.        -  
CSR-to-ASA-Provider-standby-asa-profile          vm-deployment      -      init                   reached  2018-10-08T20:33:31  -                                                        -  
                                                                           placement              reached  2018-10-08T20:33:31  Selected csp: csp1                                       -  
                                                                           image-ready            reached  2018-10-08T20:33:33  -                                                        -  
                                                                           vm-deployed            reached  2018-10-08T20:37:15  VNF is deployed                                          -  
                                                                           switch-configurations  reached  2018-10-08T20:37:18  -                                                        -  
                                                                           vm-alive               reached  2018-10-08T20:37:24  -                                                        -  
                                                                           day1-configurations    reached  2018-10-08T20:37:27  -                                                        -  
                                                                           ready                  reached  2018-10-08T20:37:27  VNF configured with vnfd-deployment->day1-config.        -  
stitch                                           nsd-deployment     -      init                   reached  2018-10-08T17:57:14  -                                                        -  
                                                                           switch-configurations  reached  2018-10-08T17:57:14  -                                                        -  
                                                                           ready                  reached  2018-10-08T17:57:14  Stitching Service Chain created

From above plan, we figured out Gateway-VNF-gateway-vnf-1-SAE-EP-GW-CSR-profile, CSR-to-ASA-active-asa-profile, CSR-to-ASA-consumer-csr and CSR-to-ASA-Provider-standby-asa-profile are on CSP device csp1.

Gateway-VNF-gateway-vnf-1-SAE-EP-GW-CSR-profile belongs to end point gateway vnf service gateway-vnf-1 (Ignoring prefix Gateway-VNF- in plan).

CSR-to-ASA-active-asa-profile and CSR-to-ASA-consumer-csr belong to service chain CSR-to-ASA.

CSR-to-ASA-Provider-standby-asa-profile belongs to service chain CSR-to-ASA-Provider.

So on csp1 device there should be 3 services deployed.

Now if we execute the action command, we should see these 3 services.

admin@ncs> request sae-site-actions services-on-csp csp-device csp1  
status : List of Services on CSP  
sae-site[name='sae-site-1'] endpoint-gateway-vnf[name='gateway-vnf-1']  
sae-site[name='sae-site-1'] service-chain[name='CSR-to-ASA']  
sae-site[name='sae-site-1'] service-chain[name='CSR-to-ASA-Provider']

**undep-redep-services-on-csp**

1. Execute services-on-csp action command as above to check what are all the services deployed on a particular CSP device.
2. Execute the action command.
3. Check the plan to observe the services deployed on that CSP device goes away from plan (due to un-deploy) and re-appear after sometime (due to re-deploy).

### Custom Actions

**Description**

Due to the multi-layer architecture of SAE core function pack, some of the NSO provided action commands like re-deploy, undeploy-redeploy executed on SAE services directly do not work.

If SAE core function pack does not allow update request, there are some custom actions added in SAE to enable redeploy or undeploy-redeploy of services. The custom actions added at various SAE services are:

1. custom-redeploy: Re-deploys the service. This custom action command is available at

* sae-site
* sae-site/endpoint-gateway-vnf
* sae-site/service-chain
* sae-site/stitching-service
* sae-site/infrastructure/compute-clusters/leaf-switches
* sae-site/infrastructure/compute-clusters/servers
* sae-site/infrastructure/vnf-manager

1. custom-undeploy-redeploy: Undeploy-redploy the service. This custom action command is available at

* sae-site/endpoint-gateway-vnf
* sae-site/service-chain

**Yang Model**

module: cisco-sae-core-fp-cfs  
    +--rw sae-site\* [site]  
    |  +--rw site                    service-name  
    |  +--rw infrastructure  
    |  |  +--rw compute-clusters\* [name]  
    |  |  |  +--rw name             string  
    |  |  |  +--rw leaf-switches\* [name]  
    |  |  |  |  +--rw name               string  
    |  |  |  |  +--rw timestamp?         string  
    |  |  |  |  +---x custom-redeploy  
    |  |  |  |     +--ro output  
    |  |  |  |        +--ro status?   string  
    |  |  |  |        +--ro error?    string  
    |  |  |  +--rw servers\* [name]  
    |  |  |     +--rw name                        string  
    |  |  |     +--rw timestamp?                  string  
    |  |  |     +---x custom-redeploy  
    |  |  |        +--ro output  
    |  |  |           +--ro status?   string  
    |  |  |           +--ro error?    string  
    |  |  +--rw vnf-manager\* [name]  
    |  |     +--rw name                        string  
    |  |     +--rw timestamp?                  string  
    |  |     +---x custom-redeploy  
    |  |        +--ro output  
    |  |           +--ro status?   string  
    |  |           +--ro error?    string  
    |  +--rw endpoint-gateway-vnf\* [name]  
    |  |  +--rw name                          string  
    |  |  +--rw timestamp?                    string  
    |  |  +---x custom-redeploy  
    |  |  |  +--ro output  
    |  |  |     +--ro status?   string  
    |  |  |     +--ro error?    string  
    |  |  +---x custom-undeploy-redeploy  
    |  |     +--ro output  
    |  |        +--ro status    string  
    |  +--rw service-chain\* [name]  
    |  |  +--rw name                          string  
    |  |  +--rw timestamp?                    string  
    |  |  +---x custom-redeploy  
    |  |  |  +--ro output  
    |  |  |     +--ro status?   string  
    |  |  |     +--ro error?    string  
    |  |  +---x custom-undeploy-redeploy  
    |  |     +--ro output  
    |  |        +--ro status    string  
    |  +--rw stitching-service\* [name]  
    |  |  +--rw name                        string  
    |  |  +--rw timestamp?                  string  
    |  |  +---x custom-redeploy  
    |  |     +--ro output  
    |  |        +--ro status?   string  
    |  |        +--ro error?    string  
    |  +--rw timestamp?              string  
    |  +---x custom-redeploy  
    |     +--ro output  
    |        +--ro status?   string  
    |        +--ro error?    String

### Prerequisites/Dependencies

SAE services must be deployed.

### Usage with Sample Payloads

**custom-redeploy**

admin@ncs> request sae-site $SAE\_SITE\_NAME custom-redeploy  
admin@ncs> request sae-site $SAE\_SITE\_NAME endpoint-gateway-vnf $EPGW\_NAME custom-redeploy  
admin@ncs> request sae-site $SAE\_SITE\_NAME service-chain $SRVC\_CHAIN\_NAME custom-redeploy  
admin@ncs> request sae-site $SAE\_SITE\_NAME stitching-service $STITCH\_SRVC\_NAME custom-redeploy  
admin@ncs> request sae-site $SAE\_SITE\_NAME infrastructure compute-clusters $CLUSTER\_NAME leaf-switches $SWITCH\_DEVICE\_NAME custom-redeploy  
admin@ncs> request sae-site $SAE\_SITE\_NAME infrastructure compute-clusters $CLUSTER\_NAME servers $CSP\_DEVICE\_NAME custom-redeploy  
admin@ncs> request sae-site $SAE\_SITE\_NAME infrastructure vnf-manager $VNFM\_SERVICE\_NAME custom-redeploy

**custom-undeploy-redeploy**

admin@ncs> request sae-site $SAE\_SITE\_NAME endpoint-gateway-vnf $EPGW\_NAME custom-undeploy-redeploy  
admin@ncs> request sae-site $SAE\_SITE\_NAME service-chain $SRVC\_CHAIN\_NAME custom-undeploy-redeploy

| **Variable Name** | **Variable Description** |
| --- | --- |
| SAE\_SITE\_NAME | sae-site service name |
| EPGW\_NAME | endpoint-gateway-vnf service name |
| SRVC\_CHAIN\_NAME | service-chain service name |
| STITCH\_SRVC\_NAME | stitching-service service name |
| CLUSTER\_NAME | compute-clusters name |
| SWITCH\_DEVICE\_NAME | leaf-switches device name |
| CSP\_DEVICE\_NAME | servers CSP device name |
| VNFM\_SERVICE\_NAME | vnf-manager service name |

### Cleanup Actions

# SAE Models

The following chapter shows some SAE models.

## Introduction

Yang is the primary data modeling language used by NSO. So all SAE data structures including configuration and operational data is modeled in yang.

These models align with the ETSI MANO Standards.

More details about ETSI MANO can be found here

https://www.etsi.org/deliver/etsi\_gs/NFV-MAN/001\_099/001/01.01.01\_60/gs\_NFV-MAN001v010101p.pdf

SAE models include the following:

* SAE Model: Please see, Appendix A. Yang Models and Payloads, section 10.1.
* VNF Model: Please see, Appendix A. Yang Models and Payloads, section 10.2.
* NSD Model: Please see, Appendix A. Yang Models and Payloads, section 10.3.

# Other SAE Features

The following chapter shows some other features found in SAE.

## Introduction

These other SAE features include the capabilities for diagnosis, customization, maintenance, and so forth.

## BGP AS Configuration

### Overview

### Activate

### Modify

### Delete

### Sample Payloads

### Diagnosis

### Troubleshooting

## Intangible VNF

### Overview

### Activate

### Modify

### Delete

### Sample Payloads

### Diagnosis

### Troubleshooting

## Spine Leaf Configuration

### Overview

### Activate

### Modify

### Delete

### Sample Payloads

### Diagnosis

### Troubleshooting

## SAE Add/Delete CSP from infrastructure

### Overview

SAE supports resizing CSPs under infrastructure on existing compute clusters.

### Description

Assume that there are VNFs growing on existing CSPs. For bringing up more VNFs, you need additional CSPs to be involved to increase available resources. Under such circumstances, SAE supports addition of more CSPs into SAE.

Removing CSPs is also supported.

**Yang Model**

list sae-site {

  key site;

  leaf site {

    type service-name;

  }

  container infrastructure {

    ...

    container switching {

      leaf type {

        type cisco-sae-core-fp-common:switching-type-t;

        default cisco-sae-core-fp-common:n9k-switch-pair;

      }

      leaf vnid-pool {

        type leafref {

          path "/ralloc:resource-pools/idalloc:id-pool/idalloc:name";

        }

        mandatory true;

      }

      ...

    }

    list compute-clusters {

      ...

      key name;

      leaf name {

        type string;

      }

      leaf vlan-pool {

        type leafref {

          path "/ralloc:resource-pools/idalloc:id-pool/idalloc:name";

        }

        mandatory true;

      }

      list leaf-switches {

        tailf:info "List of leaf Switches";

        max-elements 2;

        key name;

        ordered-by user;

        leaf name {

          type string;

        }

        ...

      }

      list servers {

        key name;

        leaf name {

          tailf:info "Name of CSP device";

          description "Name of CSP device";

          type string;

        }

        leaf type {

          type leafref {

            path "/cisco-sae-core-fp-cfs:sae-catalog/cisco-sae-core-fp-cfs:csp/"

            + "cisco-sae-core-fp-cfs:name";

          }

          mandatory true;

        }

        list virtio-network {

          key pc-name;

          leaf pc-name {

            type leafref {

              path "/cisco-sae-core-fp-cfs:sae-catalog/cisco-sae-core-fp-cfs:csp/"

              + "cisco-sae-core-fp-cfs:virtio-network/cisco-sae-core-fp-cfs:name";

            }

          }

          list pnics {

            max-elements 2;

            key name;

            leaf name {

              type string;

            }

            leaf connected-switch {

              type leafref {

                path "../../../../leaf-switches/name";

              }

            }

            leaf switch-interface-name {

              type string;

            }

            leaf bandwidth {

              type int32;

              default 10000;

            }

          }

        }

        list sriov-networks {

          key name;

          ordered-by user;

          leaf name {

            type leafref {

              path "/cisco-sae-core-fp-cfs:sae-catalog/cisco-sae-core-fp-cfs:csp/"

              + "cisco-sae-core-fp-cfs:sriov-networks/cisco-sae-core-fp-cfs:name";

            }

          }

          leaf connected-switch {

            type leafref {

              path "../../../leaf-switches/name";

            }

          }

          leaf switch-interface-name {

            type string;

          }

          leaf sriov-type {

            type cisco-sae-core-fp-common:csp-sriov-type-t;

            default cisco-sae-core-fp-common:X520;

          }

          leaf bandwidth {

            type int32;

            default 10000;

          }

        }

      }

    }

  }

}

**Module**

module: cisco-sae-core-fp-cfs

    ...

    +--rw sae-site\* [site]

    |  +--rw site                    service-name

    |  ...

    |  +--rw infrastructure

    |  |  +--rw switching

|  |  |  +--rw type?        cisco-sae-core-fp-common:switching-type-t

|  |  |  +--rw vnid-pool    -> /ralloc:resource-pools/idalloc:id-pool/name

|  |  +--rw compute-clusters\* [name]

    |  |  |  +--rw name             string

    |  |  |  +--rw vlan-pool        -> /ralloc:resource-pools/idalloc:id-pool/name

    |  |  |  +--rw leaf-switches\* [name]

    |  |  |  |  +--rw name               string

    |  |  |  +--rw servers\* [name]

    |  |  |     +--rw name                        string

    |  |  |     +--rw type                        -> /sae-catalog/csp/name

    |  |  |     +--rw virtio-network\* [pc-name]

    |  |  |     |  +--rw pc-name    -> /sae-catalog/csp/virtio-network/name

    |  |  |     |  +--rw pnics\* [name]

    |  |  |     |     +--rw name                     string

    |  |  |     |     +--rw connected-switch?        -> ../../../../leaf-switches/name

    |  |  |     |     +--rw switch-interface-name?   string

    |  |  |     |     +--rw bandwidth?               int32

    |  |  |     +--rw sriov-networks\* [name]

    |  |  |     |  +--rw name                     -> /sae-catalog/csp/sriov-networks/name

    |  |  |     |  +--rw connected-switch?        -> ../../../leaf-switches/name

    |  |  |     |  +--rw switch-interface-name?   string

    |  |  |     |  +--rw sriov-type?              cisco-sae-core-fp-common:csp-sriov-type-t

    |  |  |     |  +--rw bandwidth?               int32

    +--rw sae-catalog\* [name]

    |  +--rw name               string

|  +--rw csp\* [name]

    |  |  +--rw name              string

|  |  +--rw virtio-network\* [name]

    |  |  |  +--rw name    network-name

    |  |  +--rw sriov-networks\* [name]

    |  |  |  +--rw name    network-name

    |  |  +--rw mgmt-network?     network-name

### Prerequisites/Dependencies

Capture any dependencies on other components/libraries. Capture prerequisites. For example, certain dependent components to be installed or data to be populated before this component can be used.

Minimum Prerequisites are required to deploy a SAE service using NSO.

Addition of CSPs:

* New CSP has mirror SRIOV, Port-channel. The new CSP should has the same SRIOV, and Port-channel as the rest CSPs in cluster.

Deletion of CSPs:

* To remove CSP from current setup, you need to make sure that no VNF is deployed on this CSP, otherwise, SAE will not allow user to remove it

### Usage with Sample Payloads

#### Adding a CSP

* 1. Onboard CSP

**set devices device csp5 address 172.23.80.150  
set devices device csp5 port 830  
set devices device csp5 ssh host-key ssh-  
set devices device csp5 authgroup csp\_authgroup  
set devices device csp5 device-type netconf  
set devices device csp5 device-type netconf ned-id netconf  
set devices device csp5 trace pretty  
set devices device csp5 state admin-state unlocked**

**commit**

**admin@ncs> request devices device csp5 ssh fetch-host-keys**

**result unchanged**

**fingerprint {**

**algorithm ssh-rsa**

**value bd:28:de:b8:f2:93:95:35:75:95:4c:44:0a:7f:2a:19**

**}**

**admin@ncs> request devices device csp5 sync-from**

**result true**

1. Add into infrastructure:

**set sae-site cisco infrastructure compute-clusters cluster-1 servers csp5 type csp-type-cluster1 virtio-network virt-A pnics enp10s0f0 connected-switch nexus0 switch-interface-name port-channel2**

**set sae-site cisco infrastructure compute-clusters cluster-1 servers csp5 virtio-network virt-A pnics enp10s0f1 connected-switch nexus1 switch-interface-name port-channel3**

**set sae-site cisco infrastructure compute-clusters cluster-1 servers csp5 sriov-networks sriov-1 connected-switch nexus0 switch-interface-name Ethernet1/1**

**set sae-site cisco infrastructure compute-clusters cluster-1 servers csp5 sriov-networks sriov-2 connected-switch nexus1 switch-interface-name Ethernet1/3**

**show | compare**

**cisco-sae-core-fp-cfs:sae-site cisco {**

**infrastructure {**

**compute-clusters cluster-1 {**

**+            servers csp5 {**

**+                type csp-type-cluster1;**

**+                virtio-network virt-A {**

**+                    pnics enp10s0f0 {**

**+                        connected-switch nexus0;**

**+                        switch-interface-name port-channel7;**

**+                    }**

**+                    pnics enp10s0f1 {**

**+                        connected-switch nexus1;**

**+                        switch-interface-name port-channel8;**

**+                    }**

**+                }**

**+                sriov-networks sriov-1 {**

**+                    connected-switch nexus0;**

**+                    switch-interface-name Ethernet1/7;**

**+                }**

**+                sriov-networks sriov-2 {**

**+                    connected-switch nexus1;**

**+                    switch-interface-name Ethernet1/8;**

**+                }**

**+            }**

**}**

**}**

**}**

**commit**

Make sure "sriov-networks" and "virtio-network" from the CSP is identical to other CSPs in the same cluster.

**admin@ncs% show cisco-sae-core-fp-cfs:sae-site cisco infrastructure compute-clusters cluster-1   
vlan-pool sae-vlan-pool;  
leaf-switches nexus0;  
leaf-switches nexus1;  
servers csp-225 {  
type csp-type-cluster1;  
virtio-network virt-A {  
pnics enp10s0f0 {  
connected-switch nexus0;  
switch-interface-name port-channel2;  
}  
pnics enp10s0f1 {  
connected-switch nexus1;  
switch-interface-name port-channel3;  
}  
}  
sriov-networks sriov-1 {  
connected-switch nexus0;  
switch-interface-name Ethernet1/1;  
}  
sriov-networks sriov-2 {  
connected-switch nexus1;  
switch-interface-name Ethernet1/3;  
}  
var NTP\_SERVER {  
val 10.10.0.0;  
}  
}  
servers csp-226 {  
type csp-type-cluster1;  
virtio-network virt-A {  
pnics enp10s0f0 {  
connected-switch nexus0;  
switch-interface-name port-channel4;  
}  
pnics enp10s0f1 {  
connected-switch nexus1;  
switch-interface-name port-channel5;  
}  
}  
sriov-networks sriov-1 {  
connected-switch nexus0;  
switch-interface-name Ethernet1/4;  
}  
sriov-networks sriov-2 {  
connected-switch nexus1;  
switch-interface-name Ethernet1/5;  
}  
}  
servers csp5 {  
type csp-type-cluster1;  
virtio-network virt-A {  
pnics enp10s0f0 {  
connected-switch nexus0;  
switch-interface-name port-channel7;  
}  
pnics enp10s0f1 {  
connected-switch nexus1;  
switch-interface-name port-channel8;  
}  
}  
sriov-networks sriov-1 {  
connected-switch nexus0;  
switch-interface-name Ethernet1/7;  
}  
sriov-networks sriov-2 {  
connected-switch nexus1;  
switch-interface-name Ethernet1/8;  
}  
}**

#### Deleting CSPs

Delete commands

**delete sae-site sae-site-1 infrastructure compute-clusters cluster-1 servers csp5**

**commit**

SAE-cleanup

**request sae-cleanup csp\_cleanup compute-clusters cluster-1 csp-device csp5 sae-site sae-site-1**

**########################**

**#        Warning       #**

**########################**

**You are about to forcefully clean up a compute intent**

**This could southbound lock CSP device momentarily, affecting**

**any actively deploying service(s) on this site**

**Also, this might leave CSP and NSO out-of-sync**

**Are you sure, you want to proceed? [no,yes]**

Auto discovery, Please reference to Auto discovery redeployment

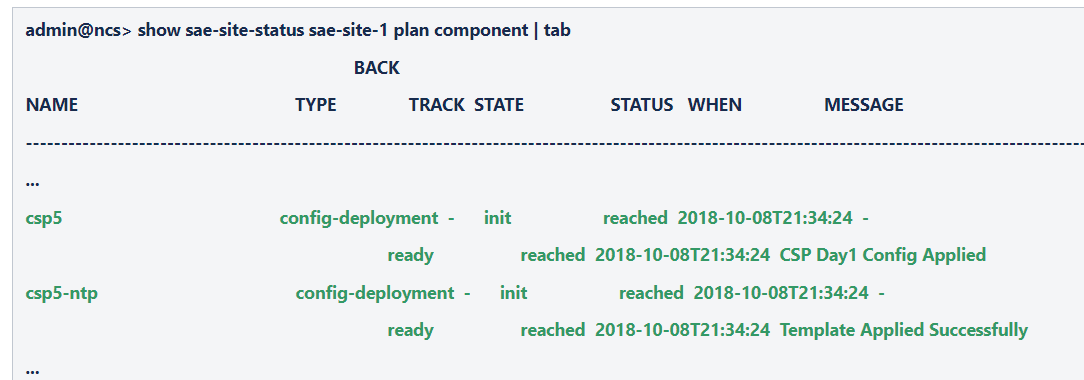
### Payloads Validation

Please reference CSP onboarding.

### Verification

Addition CSP

After Add



### Test Case

Pending

### Webex Recording

Pending

### Troubleshooting and Diagnostic

Check the Error Code

Check logs under path "/var/log/ncs/....."

## SAE Diagnosis Package

### Overview

This section explains the usage of diagnostic commands that are available when the packages "diagnosis" and "cisco-sae-diganosis" were installed along with SAE core function pack in NSO.

### Description

The diagnosis package provides commands to verify/check the following information from NSO, including commands to check links like sriov and virtio.

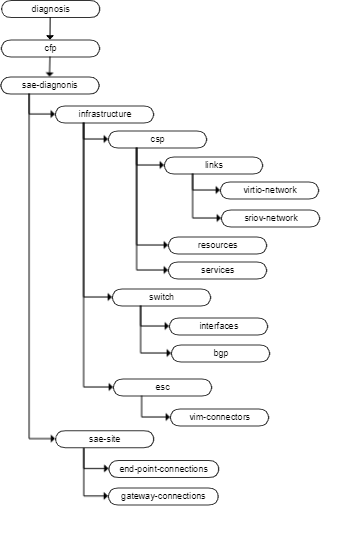
It also provide commands to show information on servers, switches and VNF manager, where you can check the resource information, Service details on CSP device. On switch, shows bgp information and interface info. with esc, displays vim-connectors information.

On sae-site, you can see the destination end point along with the services which connects both source and destination endpoints and also you can see the list of end points connected to selected gateway vnf.

| **show Command** | **Description** |
| --- | --- |
| show diagnosis cfp sae-diagnosis infrastructure csp {$CSP} links sriov-network | Checks and displays connections status of sriov network links for selected CSP device |
| show diagnosis cfp sae-diagnosis infrastructure csp {$CSP} links virtio-network | Checks and displays connection info and status of virtio network links for selected CSP device. |
| show diagnosis cfp sae-diagnosis infrastructure csp {$CSP} resources | Displays resource information for selected csp device |
| show diagnosis cfp sae-diagnosis infrastructure csp {$CSP} services {$SERVICE\_NAME\_ON\_CSP} | Displays the information for selected service on selected csp device |
| show diagnosis cfp sae-diagnosis infrastructure switch {$SWITCH} interfaces | Displays interface information for selected switch |
| show diagnosis cfp sae-diagnosis infrastructure switch {$SWITCH} bgp | Displays BGP information of selected switch |
| show diagnosis cfp sae-diagnosis infrastructure esc {$ESC} vim-connections | Displays VIM connection status for selected ESC device |
| show diagnosis cfp sae-diagnosis sae-site {$sae-site} end-point-connections | Displays destination endpoint along with the services connected to it for selected endpoint of for all. |
| show diagnosis cfp sae-diagnosis sae-site {$SAE\_SITE} gateway-connections | Displays connected endpoints for the gate ways or for selected gateway |

### Yang Model

Pictorial representation of the Model



### Prerequisites/Dependencies

These commands do not work with netsim and only works with real devices. Infrastructure commands only work if infrastructure is defined under sae-site.

### Usage with Sample Payloads

CSP commands in infrastructure

**diagnostics on csp to check sriov links**

admin@ncs> show diagnosis cfp cisco-sae-diagnosis infrastructure csp csp-88-153 links sriov-network

SWITCH

PNIC CONNECTED INTERFACE CSP LLDP LINK

CSP PNIC SWITCH NAME PORT ID STATUS REASON

----------------------------------------------------------------------------

enp4s0f2 N9k-Leaf-2-PRE\_CICD Ethernet1/3 Ethernet1/3 CONNECTED -

enp4s0f3 N9k-Leaf-1-PRE\_CICD Ethernet1/4 Ethernet1/4 CONNECTED –

**virtio network check**

admin@ncs> show diagnosis cfp cisco-sae-diagnosis infrastructure csp csp-88-153 links virtio-network

SWITCH

CSP SWITCH PORT

PORT CSP PNIC SWITCH PORT INTERFACE CHANNEL CSP PCH CSP LLDP

CSP PNIC CHANNEL SWITCH CONNECTED CONNECTED CHANNEL CONNECTED STATE STATE PORT ID STATUS REASON

------------------------------------------------------------------------------------------------------------------------------------------------

enp4s0f0 TestPC N9k-Leaf-2-PRE\_CICD enp4s0f0, enp4s0f1 port-channel121 Ethernet1/1 UP CREATED, UP Ethernet1/1 CONNECTED -

enp4s0f1 TestPC N9k-Leaf-1-PRE\_CICD enp4s0f0, enp4s0f1 port-channel121 Ethernet1/2 UP CREATED, UP Ethernet1/2 CONNECTED -

**show csp resources command**

admin@ncs> show diagnosis cfp cisco-sae-diagnosis infrastructure csp csp-88-153 resources

resources num\_cpus\_free 14

resources num\_cpus\_total 14

resources percent\_cpu\_idle 99.2

resources ram\_free\_mb 112186

resources ram\_total\_mb 112186

resources disk\_space\_free\_gb 2037.0

resources disk\_space\_total\_gb 2037

resources mgmt\_pnic Mgmtpc

resources mgmt\_pnic\_mode shared

resources mgmt\_vlan 5

resources last\_change\_time "Tue Aug 14 17:06:23 EDT 2018"

**show the details of service on csp**

admin@ncs> show diagnosis cfp cisco-sae-diagnosis infrastructure csp csp-91-152 service ESC-104

service ESC-104

summary uuid 190cc040-de30-4810-b7fd-aafa2938b102

summary memory 8192

summary numcpu 4

summary macid 11

summary disk\_size 100.0

summary disk\_type virtio

summary iso\_name ESC-4\_3\_0\_68.qcow2

summary power on

summary state deployed

summary cpu\_load 0

summary disk\_used\_mb 2048.0

summary memory\_used\_kb 69020

summary day0\_filename user-data-104

VF

PASSTHROUGH MGMT NETWORK NETWORK

NIC MAC ADDRESS VLAN NATIVE TAGGED TYPE MODE MODEL VNIC NAME NAME

---------------------------------------------------------------------------------------------------

0 02:35:05:5B:98:0B 5 - false access none virtio - Mgmtpc none

**BGP info on switch**

admin@ncs> show diagnosis cfp cisco-sae-diagnosis infrastructure switch N9k-Leaf-1-PRE\_CICD bgp

VRF

LOCAL   VRF ROUTER

VRF     AS  ID

---------------------------

default 10  [120.120](https://confluence-eng-sjc1.cisco.com/conf/display/NSOUS/120.120).120.1

**sae-site gateway-connections**

admin@ncs> show diagnosis cfp cisco-sae-diagnosis sae-site sae-site-1 end-point-connections

                      DESTINATION                 SOURCE SERVICE       STICHING  DESTINATION SERVICE  DESTINATION

SOURCE END POINT      END POINT    SOURCE GW      CHAIN                SERVICE   CHAIN                GW

---------------------------------------------------------------------------------------------------------------------

ep1                   ep2          gateway-vnf-1  CSR-to-ASA           stitch    CSR-to-ASA-Provider  gateway-vnf-1

ep2                   ep1          gateway-vnf-1  CSR-to-ASA-Provider  stitch    CSR-to-ASA           gateway-vnf-1

ep3

ep4

provider-end-point-1

consumer-end-point-1

**sae-site end-point-connections**

admin@ncs> show diagnosis cfp cisco-sae-diagnosis sae-site sae-site-1 end-point-connections

                      DESTINATION                 SOURCE SERVICE       STICHING  DESTINATION SERVICE  DESTINATION

SOURCE END POINT      END POINT    SOURCE GW      CHAIN                SERVICE   CHAIN                GW

---------------------------------------------------------------------------------------------------------------------

ep1                   ep2          gateway-vnf-1  CSR-to-ASA           stitch    CSR-to-ASA-Provider  gateway-vnf-1

ep2                   ep1          gateway-vnf-1  CSR-to-ASA-Provider  stitch    CSR-to-ASA           gateway-vnf-1

ep3

ep4

provider-end-point-1

consumer-end-point-1

### Payloads Validation

### Verification

### Troubleshooting and Diagnostic

### Packaging, Install, versioning

### Testcase

### Sources

### Webex Recording

## SAE ASAv VNF's Extensions Support for HA

### Overview

This section captures details about SAE- ASAv HA support.

### Description

We have an extension adaptor to add support for ASAv HA deployment, SAE CFP contains extension package for ASAv VNF

package: cisco-sae-asav-extension

### Yang Model

There are few model updates to use ASAv extension for HA

**Model updates:**

VNFD-DEPLOYMENT is extended to indicate what "type" a VNF is, and the "HA-settings" can then be tied up with pre extensions to be used during the VNF deployment. Such a definition can be as below under asav-ha-settings:

* + - Specify role for ASAv vnfd-deployment [active or standby]
    - Mention the connection point name as per the ASAv vnfd

**vnfd-deployment payload update**

<config xmlns="<http://tail-f.com/ns/config/1.0>">

  <sae-catalog xmlns="http://com/cisco/nso/sae-core-fp-cfs">

    <name>sae-basic</name>

      <vnfd-deployment>

        <name>Active-ASAV</name>

        .

        .

         <vnf-type xmlns:cisco-sae-asav-extension="http://com/cisco/nso/sae/extension/asav">cisco-sae-asav-extension:asav</vnf-type>

         <asav-ha-settings>

            <ha-role>active</ha-role>

            <failover-connection-point>failover</failover-connection-point>

        </asav-ha-settings>

        .

        .

      </vnfd-deployment>

   </sae-catalog>

</config>

**Extension Package Registration :**

ASAv extension registration is completed as part of installation, it is good to verify extension registration before initiating ASAv HA deployment

Verifiy using below NSO command

* + 1. admin@ncs% show extension-grouping

**ASAv extension package registration**

<config xmlns="<http://tail-f.com/ns/config/1.0>">

    <extension-grouping xmlns="http://com/cisco/nso/sae-core-fp-extension">

  <vnf-type xmlns:cisco-sae-asav-extension="http://com/cisco/nso/sae/extension/asav">cisco-sae-asav-extension:asav</vnf-type>

  <create>

    <pre xmlns:cisco-sae-asav-extension="http://com/cisco/nso/sae/extension/asav">cisco-sae-asav-extension:asav-pre-extension</pre>

  </create>

</extension-grouping>

</config>

### Prerequisites/Dependencies

Refer below Prerequisites steps to deploy ASAv HA VNF's using SAE CFP

ESC setup : To enable ASAv HA support

* + 1. Upload the following files to the ESC VM's /var/tmp/ folder.
       1. asa\_ha\_vmalive\_metric.xml
       2. dmam.py
       3. asa\_ha\_vmalive.exp
    2. Make the custom metric collector script "asa\_ha\_vmalive.exp" executable. From the ESC VM's console/terminal:

$ chmod +x /var/tmp/asa\_ha\_vmalive.exp

* + 1. Register the custom metric definition with ESC. On an ESC VM terminal (VIM VM console, ssh session, etc), run:

$ /var/tmp/dmam.py create-metric --payload\_xml /var/tmp/asa\_ha\_vmalive\_metric.xml

NSO bootstrap details:

* + 1. Please make sure ASAv extension package is installed
    2. ASAv extension must be registered, ncs cli to verify

admin@ncs% show extension

* + 1. ESC device templates should be present in NSO templates list

admin@ncs% show devices template esc-vmalive-ha-kpi-data-nic-0

Update vnfd-deployment in sae-catalog, refer below entry in vnfd-deployment to activate extension package.

### Usage with Sample Payloads

Sample catalog payload for ASAv Active & Standby vnf

**sae-catalog payload for ASAv**

<config xmlns="<http://tail-f.com/ns/config/1.0>">

  <sae-catalog xmlns="http://com/cisco/nso/sae-core-fp-cfs">

    <name>cisco-catalog</name>

    <vnfd-deployment>

      <name>ASA-Active-Sriov-Dep</name>

      <vnfd>VNFD\_ASA</vnfd>

      <vdu>VDU\_ASA\_SRIOV\_HA</vdu>

      <mode>routed</mode>

      <bootup-time>600</bootup-time>

      <recovery-wait-time>90</recovery-wait-time>

      <vnf-authgroup>{$VNF\_AUTHGROUP}</vnf-authgroup>

      <day0>

        <destination>day0-config</destination>

        <url>{$DAY0\_FILE\_URL}</url>

        <var>

          <name>{$DAY0\_CFG\_VAR\_1}</name>

          <val>{$DAY0\_CFG\_VAR\_1\_VALUE}</val>

        </var>

        <var>

          <name>{$DAY0\_CFG\_VAR\_2}</name>

          <val>{$DAY0\_CFG\_VAR\_2\_VALUE}</val>

        </var>

      </day0>

      <vnf-type xmlns:cisco-sae-asav-extension="http://com/cisco/nso/sae/extension/asav">cisco-sae-asav-extension:asav</vnf-type>

      <asav-ha-settings>

        <ha-role>active</ha-role>

        <failover-connection-point>FAILOVER</failover-connection-point>

      </asav-ha-settings>

    </vnfd-deployment>

    <vnfd-deployment>

      <name>ASA-Standby-Sriov-Dep</name>

      <vnfd>VNFD\_ASA</vnfd>

      <vdu>VDU\_ASA\_SRIOV\_HA</vdu>

      <mode>routed</mode>

      <bootup-time>600</bootup-time>

      <recovery-wait-time>90</recovery-wait-time>

      <vnf-authgroup>{$VNF\_AUTHGROUP}</vnf-authgroup>

      <day0>

        <destination>day0-config</destination>

        <url>{$DAY0\_FILE\_URL}</url>

        <var>

          <name>{$DAY0\_CFG\_VAR\_1}</name>

          <val>{$DAY0\_CFG\_VAR\_1\_VALUE}</val>

        </var>

        <var>

          <name>{$DAY0\_CFG\_VAR\_2}</name>

          <val>{$DAY0\_CFG\_VAR\_2\_VALUE}</val>

        </var>

      </day0>

      <vnf-type xmlns:cisco-sae-asav-extension="http://com/cisco/nso/sae/extension/asav">cisco-sae-asav-extension:asav</vnf-type>

      <asav-ha-settings>

        <ha-role>standby</ha-role>

        <failover-connection-point>FAILOVER</failover-connection-point>

      </asav-ha-settings>

    </vnfd-deployment>

  </sae-catalog>

</config>

### Payloads Validation

We do have some payload validation for ASAv HA vnfd-deployment

* + 1. If vnf-type is set as asav, it will enable asav-ha-settings
    2. asav-ha-settings required both inputs [ha-role & failover-connection-point], it is mandatory input

### Verification

Use below steps to make sure ESC metric registration got registered on ESC using below command, it used for monitoring ASAv VNF

curl -X GET -H "Accept: Application/xml" http://localhost:8080/ESCManager/internal/dynamic\_mapping/metrics | xmllint --format -

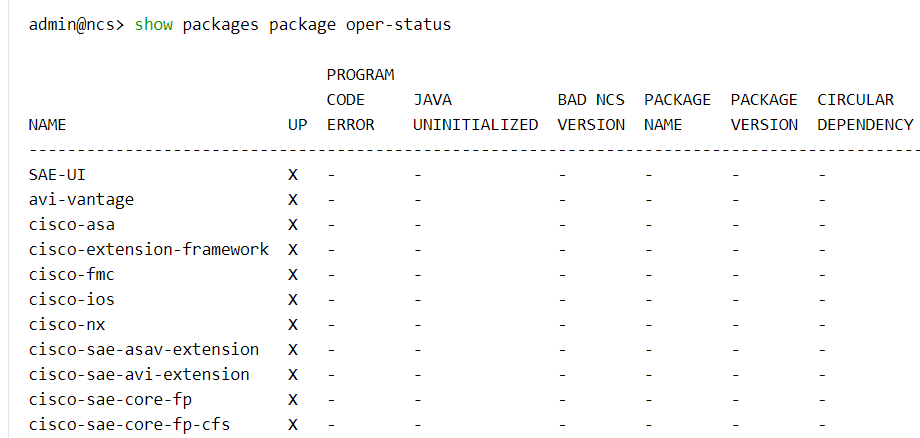
Please day0 config file are correctly provided, refer service deployment in build for reference

Verify ASAv deployment sae plan status for ready reached status

### Troubleshooting and Diagnostic

| **Error** | **Possible Reason** | **Troubleshooting** |
| --- | --- | --- |
| SAE plan failed to marked ASAv HA vnf ready reached status | * ASAv vnf is reachable from nso * day0 configs are incorrect * ESC failed in deployment | * verify day0 file, is it  correctly configure or not on ASAv vnf * check the reachability from NSO |
| ASAv HA vnf is failing in recovering | N9K configs are missing | verify the failover state on ASAv vnf while recovering |

### Packaging, Install, versioning



### Test Case

Capture automation testcase which covering this feature

### Sources

ASAv extension support added in below package

package: cisco-sae-asav-extension

### Webex Recording

Go over the above wiki and sample usage on netsim/real setup in the webex recording.

## Transparent and Routed VNFs

### Overview

The document covers the setup of VNF modes: Transparent and Routed.

### Description

Service chain VNFs can be in Transparent mode or Routed mode.

VNFs in HA can also be configured in Transparent mode.

### Yang Model

Leaf 'mode' under 'vnfd-deployment' allows to set mode as Transparent or Routed.

Leaf 'mode' is an enumeration with possible values: 'transparent' or 'routed'. If not set specifically default value is 'routed'

**Yang: vnfd-deployment:**

|  +--rw vnfd-deployment\* [name]

|  |  +--rw name                  string

|  |  +--rw vnfd                  -> /nfvo-rel2:nfvo/vnfd/id

|  |  +--rw vdu                   -> /nfvo-rel2:nfvo/vnfd[nfvo-rel2:id = current()/../vnfd]/vdu/id

|  |  +--rw mode?                 enumeration

|  |  +--rw var\* [name]

|  |  |  +--rw name             variableName

|  |  |  +--rw (valtype)?

|  |  |     +--:(val)

|  |  |     |  +--rw val              string

|  |  |     +--:(encrypted-val)

|  |  |        +--rw encrypted-val    tailf:aes-cfb-128-encrypted-string

|  |  +--rw intangible?           empty

|  |  +--rw bootup-time?          int32

|  |  +--rw recovery-wait-time?   int32

|  |  +--rw vnf-authgroup?        -> /ncs:devices/authgroups/group/name

|  |  +--rw ssh-authgroup?        -> /ncs:devices/authgroups/group/name

|  |  +--rw day0\* [destination]

|  |  |  +--rw destination    -> /nfvo-rel2:nfvo/vnfd[nfvo-rel2:id =

### Prerequisites/Dependencies

SAE services must be deployed.

### Usage with Sample Payloads

VNF mode can be set in vnfd-deployment under sae-catalog.

**VNF Mode:**

<config xmlns="<http://tail-f.com/ns/config/1.0>">

  <sae-catalog xmlns="http://com/cisco/nso/sae-core-fp-cfs">

    <name>providercatA</name>

    ...

    <vnfd-deployment>

      <name>ASA-DEP-T</name>

      <vnfd>SAE-ASA</vnfd>

      <mode>transparent</mode>

        ...

      <day0>

        ...

      </day0>

      <day1-config>

        ...

      </day1-config>

    </vnfd-deployment>

    ...

  </sae-catalog>

</config>

### Allocations in Transparent mode:

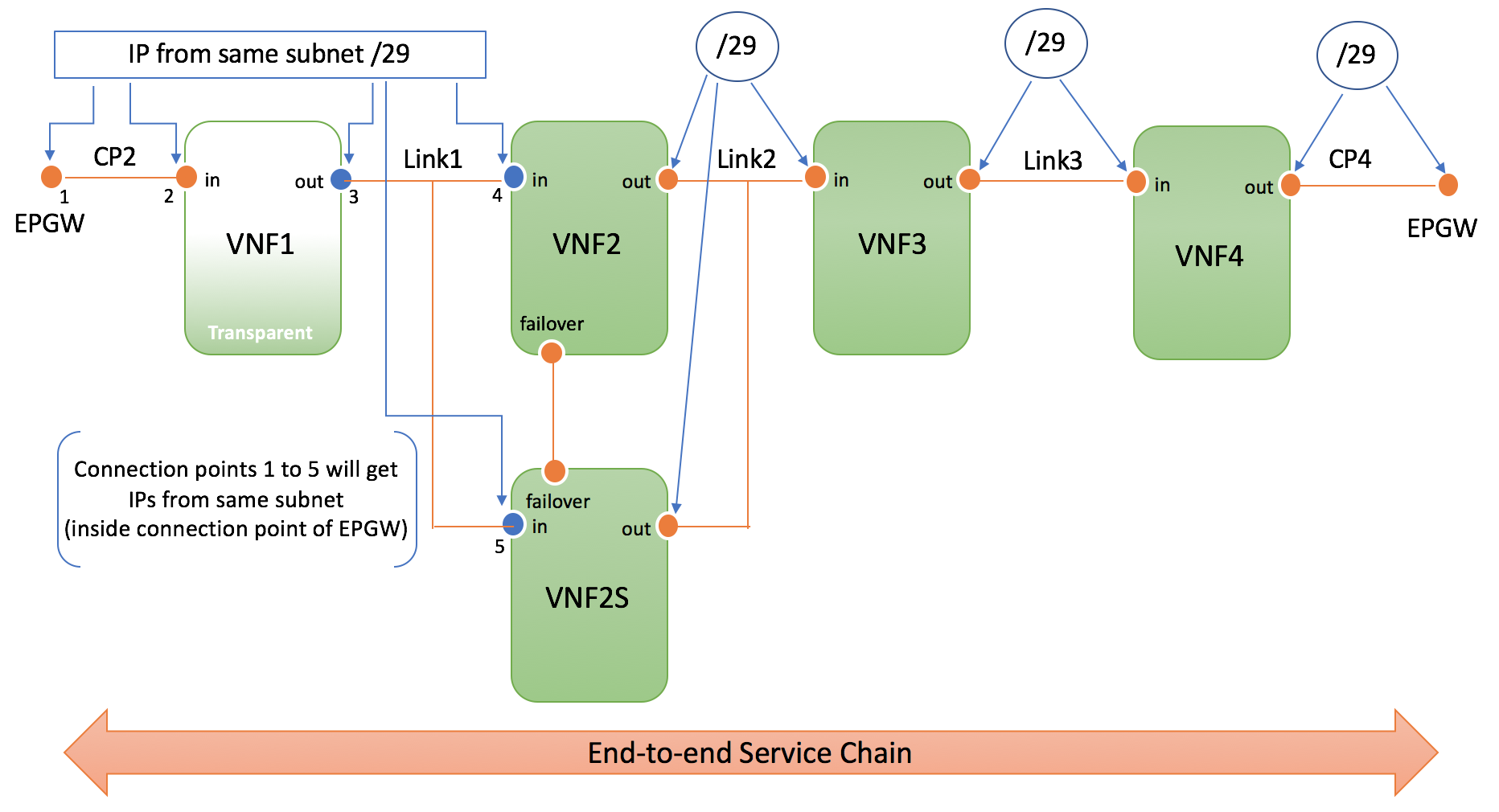
VLAN allocation will be done for all links same as Routed mode VNF.

No IP allocations will be done for virtual-links of Transparent mode VNF except for management like and failover link.

Example 1: Transparent VNF in end-to-end service chain.

* VNF1 is transparent, no IP allocation will happen for virtual-link Link1.
* Connection points 1 to 5 will get IPs from same subnet, allocated at 'inside' connection point of endpoint gateway.

Figure 21. End-toend Service Chain



Example 2: Transparent VNF in half service chain.

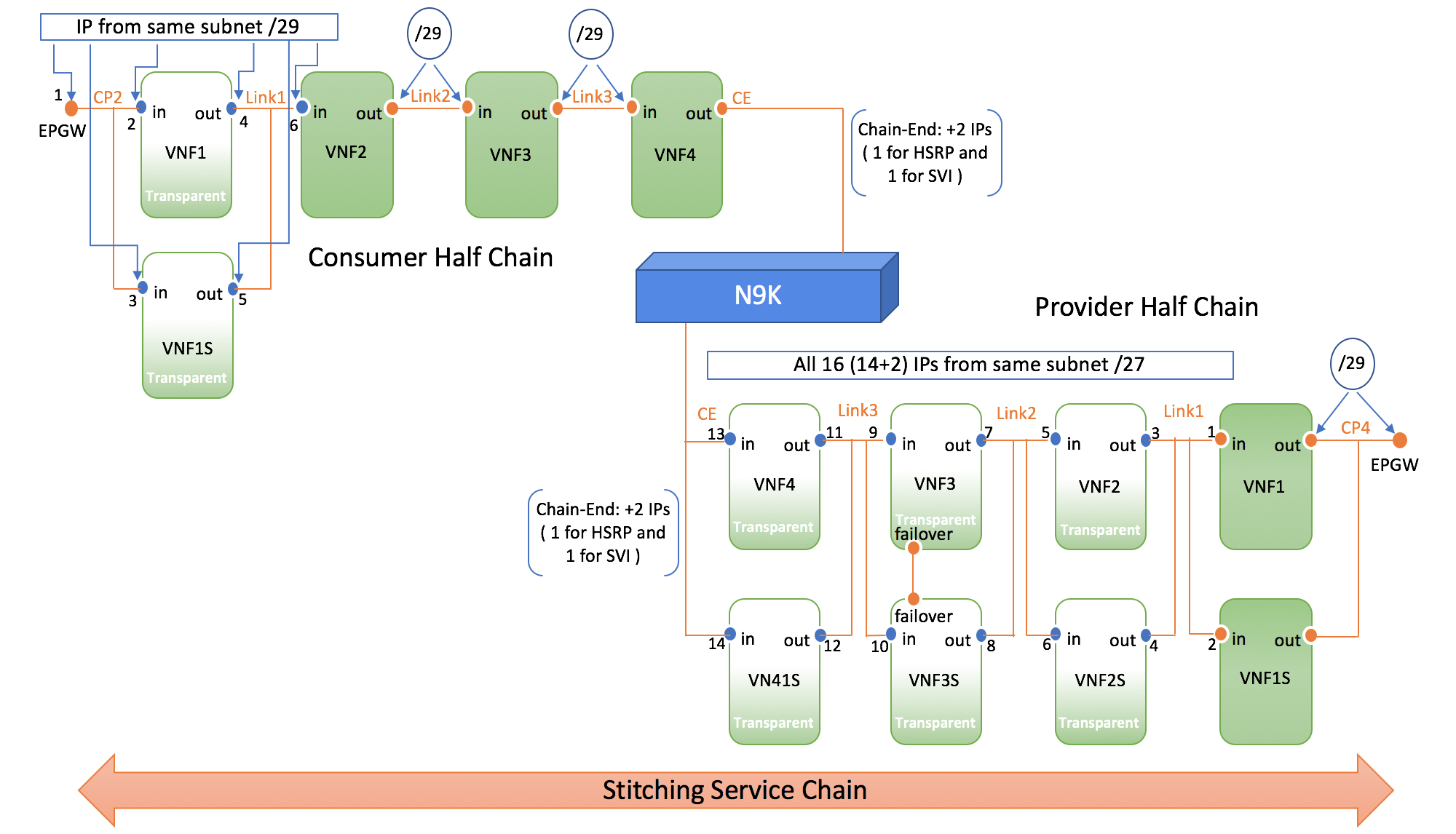
Consumer Half Chain

* VNF1 and VNF1s are in HA and transparent, so no IP allocation will happen for virtual-link Link1.
* Connection points 1 to 6 will get IPs from same subnet, allocated at 'inside' connection point of endpoint gateway.

Provider Half Chain

* VNF2, VNF2s, VNF3, VNF3s, VNF4 and VNF4s are in HA and transparent, so no IP allocation will happen for virtual-links Link2, Link3 and CE.
* Connection points 1 to 14 and 2 additional IPs (for chain-end) i.e total 16 IPs will be allocated from same subnet, which is subnet allocated for Link1.
* SAE CFP code will consider the number of IPs required to calculate the subnet prior to allocation; IP allocation will use /27 subnet on Link1
* Failover link and Management links will have separate allocation even for transparent vnf

Figure 22. Stitching Service Chain



### Validations

EndPoint Gateway can not be transparent.

Service Chain:

* All VNFs of a service chain can not be transparent.
* Shared Gateway Service Chain: If first or last VNF is Transparent, and default subnet value of Endpoint Gateway is /29, max 6 IP addresses can be allocated from the subnet.

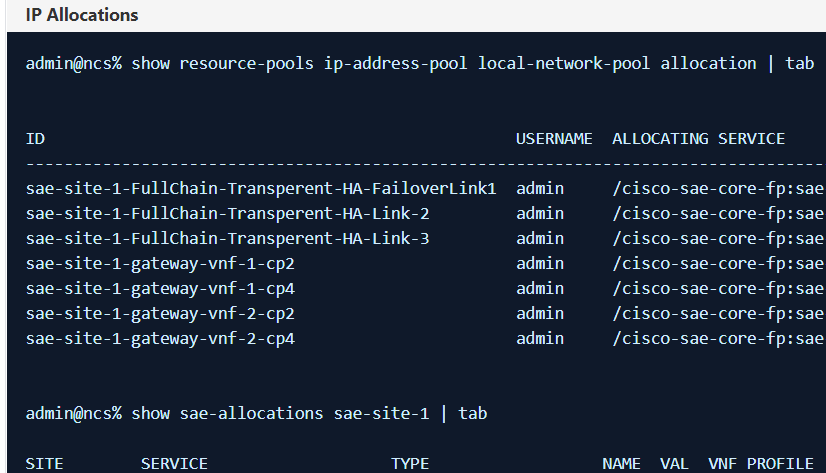
If more connection points are configured to use 'inside' IP Pool of EPGW, the IP pool will run out of IP addresses. An error will occur during allocation.

To avoid this issue, provide appropriate subnet size to Endpoint Gateway VNF during its deployment.

* Inline Service Chain: First and Last VNF of inline service chain can not be transparent.

### Verification

Verify the ip allocations happened on the links of a service chain.



### Assumptions

If VNFs are in HA, both has to be of same mode, either Routed or Transparent.

### Configs

Cisco VNFs need to configure BVI interface for transparent mode VNF.

A IP address is required to configure BVI interface. Ycan use any IP address allocated on 'in' or 'out' connection points.

### Troubleshooting and Diagnostic

Check ncs-java-vm.log to troubleshoot any issues while deploying service chains.

### Packaging, Install, versioning

Capture the packaging, install details. List the packages, including dependent packages and their versions. snapshot of show package version, package contents etc

### Testcase

Capture automation testcase which covering this feature

### Sources

Check vnfd-deployment in cisco-sae-core-fp-cfs.yang model.

Check below methods in NSDeployment.java

manageTransparentVnf and hasTransperentVnf methods

Check below methods in VirtualLink.java

allocateIP and checkForSubnetResourceReady methods

### Webex Recording

Go over the above wiki and sample usage on netsim/real setup in the webex recording.

## Replacement of VNF in existing service [epgw, service-chain]

### Overview

This document details how to replace a VNF in an existing service for generally maintainance purpose. We will explain how to replace a switch in a service.

### Prerequisites/Dependencies

Existing VNF or Service Chain.

### Usage with Sample Payloads for Replacing Switch

As long as there is one leaf-switch in a standalone mode, if the second leaf-switch goes on maintenance mode, things will still work as it is.

Services might be affected even though we have a different switch in DCI mode. Remember SRIOV ports are connected to only one switch.

Procedure:

1. southbound lock N9K.

admin@ncs% set devices device nexus0 state admin-state southbound-locked  
[ok][2018-10-22 10:34:06]  
  
[edit]  
admin@ncs% commit  
Commit complete.  
[ok][2018-10-22 10:34:09]  
  
[edit]  
admin@ncs%

1. Replace N9K.
2. Southbound unlock N9K.

admin@ncs% set devices device nexus0 state admin-state unlocked[ok][2018-10-22 10:34:31]  
  
[edit]  
admin@ncs% commit  
Commit complete.  
[ok][2018-10-22 10:34:32]  
  
[edit]  
admin@ncs%

1. Fetch ssh-host-keys.

admin@ncs% request devices fetch-ssh-host-keys device [ nexus0 ]  
fetch-result {  
    device nexus0  
    result unchanged  
    fingerprint {  
        algorithm ssh-rsa  
        value 1d:c4:44:34:25:0d:71:84:be:a2:9b:1d:0f:12:d8:4e  
    }  
}  
[ok][2018-10-22 10:30:38]

1. Connect to N9K

admin@ncs% request devices connect device [ nexus  
Possible completions:  
  nexus0  nexus1  nexus2  nexus3  
admin@ncs% request devices connect device [ nexus0 ]  
connect-result {  
    device nexus0  
    result true  
    info (admin) Connected to nexus0 - [127.0.0.1:10028](http://127.0.0.1:10028/)  
}  
[ok][2018-10-22 10:30:48]

1. Sync-to the configs (Assuming new N9K has all the day -1 configuration required or NSO knows all the day -1 configuration via sync-from done on earlier device).

admin@ncs% request devices sync-to device [ nexus0 ]  
sync-result {  
    device nexus0  
    result true  
}  
[ok][2018-10-22 10:31:01]

### Usage with Sample Payloads for Replacing EPGW

If EPGW is to be replaced in a service chain then running services will be impacted as you will need to delete and re-create the service chain.

Also if there is any service using endpoint gateway, SAE CFP won't allow to delete, so both services are be deleted and recreated.

Note: We can not delete endpoint-gateway service if service-chains are using it. You have to delete all the service chains using the EPGW.

* + 1. Delete all the service chains using the EPGW.
    2. Delete the endpoint-gateway service.

admin@ncs% delete sae-site sae-site-1 endpoint-gateway-vnf gateway-vnf-1

[ok][2018-10-22 17:02:50]

[edit]

admin@ncs% commit

Commit complete.

* 1. Create VNFD for the new vnf : Refer to SAE VNFD Model and Creation.
  2. Add the deployment in catalog.
  3. Create new endpoint-gateway service with new VNF : Refer to EPGW → Instantiation and HA support.
  4. Recreate all the service chains using the EPGW.

### Usage with Sample Payloads for Replacing VNF in a service chain

In a service-chain, if one VNF is to be replaced by another VNF

1. Delete the service-chain

admin@ncs% delete sae-site sae-site-1 service-chain CSR-to-ASA

[ok][2018-10-22 17:02:50]

[edit]

admin@ncs% commit

Commit complete.

1. Create VNFD for the new vnf : Refer to SAE VNFD Model and Creation.
2. Add the vnfd-deployment in catalog.
3. Update NSD to use new VNFD : Refer SAE NSD Model and Creation.
4. Update nsd-deployment.
5. Create new service-chain with new NSD :
   * + For end to end service chain : SAE End to End chain Instantiation.
     + For half service chain : SAE Half Service Chain Instantiation.

### Packaging, Install, versioning

Package: cisco-sae-core-fp-cfs , cisco-sae-core-fp and NEDs for the VNF concerned

Version: 1.0.0

Install: Installed through SAE installer.

## SAE Maintenance

### Overview

Various SAE Maintenance options are available from NSO Core FP.

|  |
| --- |
| **#** | **Which component is under maintenance** | **Reason behind maintenance** | **What is the impact** | **User Options** | **What actions/steps can be taken** | **Commands** | **Northbound Notification** |
| **1** | **CSP →**  **========** | Out of the number of CSPs in a given compute-cluster, some CSPs are in maintenance mode Cleanup CSP , cleanup VNF’s and/or configs.  CSP cabling needs to be updated | Running services are NOT impacted. | SAE CFP should not use this CSP for further service/VNF deployment. | Below step is if you want to move the running services to other CSPs which are not in maintenance -  1.southbound lock the CSP  2. RO should not take into account the CSP which is SBlocked. [Taken care]  3. When up, do unlock  4. Perform SAE Actions to recover VNF on CSP | 1. set devices device <csp-device-name> state admin-state southbound-locked 2. admin@ncs> request sae-site-actions  Possible completions: recover-vnf-on-csp - redeploy vnf from one csp to another csp services-on-csp - List services on CSP device undep-redep-services-on-csp - undep-redep services on CSP device 3. After maintenance mode on CSP is completed , request devices device <csp-device-name> state admin-state unlocked 4. request devices device sync-from <csp-device-name>  5. Update sae-site infrastructure with CSP connections | SAE-Plan -status will be updated updated |
|  | 1. CSP is dead/CSP is not reachable   2. CSP needs to be moved from one rack to another | 1. Running services are impacted. | 1. Replace the CSP and move the services/VNF’s on to the new CSP 2. Move the services/VNF’s on to  other CSP | After VNF is created by SAE, there could be one or more day-2 services on top of SAE that configures the VNF. So, when a VNF is lost because of various reasons, the first option should be to recover to the last known state. We should follow the below destructive (meaning recovering VNF to day-0 state) only when that fails.  1. Get the list of services on the dead CSP  2. Now recover vnf on that CSP  3. Remove the csp from sae infrastructure if there is no other service attached to it.  4. Add a new CSP to NSO device tree , perform fetch ssh keys, connect and sync-from . Add CSP to the SAE-site Infrastructure.  5. Perform custom redeploy of service on ESC .  6. Ensure Plan status is updated and verify ESC has the VIM connectors in reached state.  7. Continue to deploy new services. | 1.request sae-site-actions services-on-csp csp-device <name-of-the-dead-csp>  2. admin@ncs> request sae-site-actions recover-vnf-on-csp  Possible completions: endpoint-gateway-vnf-service - Name of Endpoint Gateway VNF service sae-site - Name of SAE Site service-chain - Name of service chain vnf-profile - Name of vnf-profile to be redeployed.  3. delete sae-site <sae-site-name> infrastructure compute-clusters <cluster-name> servers <csp-device-name>  4. request sae-site <sae-site-name> service-chain <service-chain-name> custom-undeploy-redeploy  or  request sae-site <sae-site-name> stitching-service <stitching-service-name> custom-undeploy-redeploy  5.run show sae-site-status sae-site-1 plan | tab  request sae-site-actions undep-redep-services-on-csp csp-device <name-of-csp-device>  6 [optional](https://confluence-eng-sjc1.cisco.com/conf/pages/createpage.action?spaceKey=NSOUS&title=optional&linkCreation=true&fromPageId=81957378) - After adding new CSP device.  request devices fetch-ssh-keys device [ <new csp> ]  request devices connect device [ new csp]  request devices sync-from device [ new csp]  Add CPE to SAE Site and perform  request sae-site <site-name> infrastructure vnfm <vnfm-name> custom-redeploy. | SAE-Plan -status will be updated updated |
|  | CSP needs hardware maintenance – add more CPU/Memory/cards etc  or  CSP Firmware or Image Upgrade | Running services will get impacted. | User needs an option to migrate the VNF’s/services into other CSP OR gracefully shutdown the service (all VNF’s in the service) and bring them up again after the maintenance window. | 1. Upgrade or change on CSP is not recommended if VNF is using it.   2. Southbound lock the CSP.  3. Power off the VNF on CSP - using esc commands.  4.Make the card change/ upgrade image on CSP.  5. Power on the VNF on CSP - using esc commands.  6. Unlock the CSP and perform a sync-from | esc provides host of commands to "disable-monitoring", "enable-monitoring", "stop", "start", "reboot"  ex: request devices device esc0 rpc rpc-vmAction vmAction actionType DISABLE\_MONITOR vmName <VNF-NAME-AS-KNOWN-BY-ESC>  to stop a VNF we need to  1) disable-monitoring  2) stop vnf  similarly to start a VNF we need to  1) start vnf  2) enable-monitroing  We can use esc commands and provide a wrapper command that helps user to stop and start VNF. |  |
| **2** | VNFM ->  ======  The unmanaged VNFM is in maintenance mode in a site | VNFM down/VNFM is powered off | Running services are impacted from monitoring prospective. And new deployment requests will be impacted as there is no VNFM available to handle the requests. | We should block any service deployments for the affected sae-site. | The way it should be done from NSOs prospective is :  1) block any service deployments for the affected sae-site  2) bring back ESC VM to a state that is known to NSO.  3) fetch ssh-host-keys  4) connect to the ESC VM  5) un-block affected sae-site for deployments. | 1. set devices devices <ESC> state admin state south bound locked. 2. Perform Maintenance on ESC 3. set devices devices <ESC> state admin state unlocked. | Perform ESC backup and restore |
|  | Services are down as ESC is unable to connect to CSP and VNF ,attempts recovery of VNF | Running services are impacted only from monitoring prospective. And new deployment requests will be impacted as there is no VNFM available to handle the requests. | Restart the services | * sudo escadm stop * sudo rm -rf /opt/cisco/esc/esc\_database * sudo escadm reload * sudo escadm start   ESC will take few minutes to get up & running, User can validate health using below command   * sudo health.sh  From NSOs prospective procedure is :  1) block any service deployments for the affected sae-site 2) Fix the services in ESC (out of NSOs context). 3) un-block affected sae-site for deployments. | * sudo escadm stop * sudo rm -rf /opt/cisco/esc/esc\_database * sudo escadm reload * sudo escadm start | NSO Alarm about device down |
|  | VNFM in maintenance | Running services are impacted from monitoring prospective. And new deployment requests will be impacted as there is no VNFM available to handle the requests. | Block any service deployments for the affected sae-site. | From NSOs prospective procedure is :  1) block any service deployments for the affected sae-site 2) Carry out maintenance activity on ESC (out of NSOs context). 3) un-block affected sae-site for deployments. | 1. set devices devices <ESC> state admin state south bound locked. 2. Perform Maintenance on ESC 3. set devices devices <ESC> state admin state unlocked. |  |
|  | ESC DB is not in sync with NSO.  It contains failed deployment |  |  | NSO to sync-from ESC | 1. request devices device <ESC >sync-from |  |
| **3** | Switches→  ========  One of the leaf-switches in a site is in maintenance mode | 1. Switch down – replace | Running services are impacted. |  | As far as there is one leaf-switch, in a standalone mode if at all the 2nd leaf-switch goes on a maintenance mode, things will work as it is.  Services might have affected even though we have a different switch in DCI mode. Remember SRIOV ports are connected to only one switch.  Procedure:  1) southbound lock N9K.  2) replace N9K.  3) unlock N9K.  4) fetch ssh-host-keys.  5) connect to N9K  6) sync-to the configs (Assuming new N9K has all the day -1 configuration required or NSO knows all the day -1 configuration via sync-from done on earlier device) | 1. set devices device <N9k> state admin state southbound locked 2. set devices device < new N9k> state admin state unlocked. 3. request devices fetch-ssh-keys device <newN9k> 4. request devices connect device <new n9k> 5. request devices sync-to device < new N9k> 6. request devices sync-from device < new N9k> |  |
|  | 1. Change switch with new better Switch or model change or OS Upgrade, change some hardware | Running services are impacted. |  | 1) southbound lock N9K.  2) replace N9K.  3) unlock N9K.  4) fetch ssh-host-keys.  5) connect to N9K  6) sync-to the configs (Assuming new N9K has all the day -1 configuration required or NSO knows all the day -1 configuration via sync-from done on earlier device) | 1. set devices device <N9k> state admin state southbound locked 2. Upgrade image on the switch. ( N9k Leaf node commands) 3. set devices device <N9k> state admin state unlocked. 4. request devices fetch-ssh-keys device <N9k> 5. request devices sync-to device <N9k-2> 6. request devices sync-from <N9k-2> 7. Perform steps 1- 6 for other N9k leaf node. |  |
| **4.** | service-chain→  ========  In a service-chain, if one VNF is to be replaced by another VNF | VNFs in chain are to be rebooted  s]Service-chain in maintenance | Running services are impacted. | Gracefully do not allow any updates on it | esc provides host of commands to "disable-monitoring", "enable-monitoring", "stop", "start", "reboot"  to repoot a VNF we need to issue request devices device esc0 rpc rpc-vmAction vmAction actionType REBOOT vmName <VNF-NAME-AS-KNOWN-BY-ESC>  But, these commands can be wrapped in SAE-CFPs own commands for a better experiance. |  |  |
| **5.** | VNF Upgrade from one version to other (Image upgrade) |  | Running services are impacted. |  | 1.southbound lock the VNF.  2. Take back-up of VNF configs  2.Login to the VNF  3.Copy image on VNF  4.Change boot parameter  5.Reload  6. unlock the VNF  7. fetch-ssh-host keys.  8. connect to VNF.  9. sync-to (during the time of upgrade if we have any day-2 services trying to configure VNFs)  or .  Redeploy the VNF with new image   1. Update the vnfd software-image descriptor with teh new image and perform "custom-undeploy-redeploy" , simulates delete and create of endpoint-gateway. | set nfvo vnfd <VNFD> vdu <VDU> software image-descriptor image <name>  file <path> |  |
| VNF is powered off |  | Running services are impacted | ESC runs recovery of the VM and VM comes up.  But Plan is not notified of recovery it stays in reached condition always which may be enhanced. |  |  |  |
| Update the VNFD with some changes in say storage, RAM or CPU |  | Running services are impacted. |  |  |  |  |
| VNF replacement by deleting existing VNF |  | Running services are impacted. | Update only one VNF in the chain. | Delete VNF Info from NFVO.  Redeploy Chain. | Add new action to handle this scenario.  [US11747](https://rally1.rallydev.com/#/153700652268ud/detail/userstory/220977332252?fdp=true) |  |
| **6** | NSO is in maintenance mode | NSO is not reachable | Running services are not impacted. NSO is just a config engine, actual VNFs are running on CSP. | In NSO HA, standby NSO will have the cdb backup, when the master HA comes up , the cdb can be replaced.  For standalone case we do not have anything available as of now. |  |  |  |
|  | NSO patch/upgrade, NED patch/upgrade, CFP patch/upgrade | Running services should not be impacted if there is not a hard model change | 1. Stop NCS. 2. take backup 3. patch/upgrade ncs   restore NED: ======   1. Confirm that existing templates work as it was which means the change was backward compartible and no template change is needed. 2. This is only when new command support is needed to be used in custom templates 3. delete the softlink in /var/opt/ncs/packages for ned 4. put the new ned in /var/opt/ncs/packages   5. ncs restart with package reload CFP Patch =========   1. Stop ncs 2. delete the softlink in /var/opt/ncs/packages for the package 3. put the new package in /var/opt/ncs/packages 4. ncs restart with package reload | Upgrade Steps for NSO HA   1. Disable HA between Slave nodes and Master Nodes. 2. Upgrade image on all other Slave nodes e.g. NSO 4.7 to 4.7.1 3. Connect Slave nodes back to master node 4. Configure slave node as acting master node. 5. Upgrade original master node e.g. NSO 4.7 to 4.7.1. 6. Configure HA to add it to acting master node as another slave node. 7. Configure this node as master node. | 1. On Slave node perform: request ha commands de-activate 2. Upgrade NSO image in slave node 3. On the slave node perform : request ha commands activate 4. On Master node , perform :  request ha commands de-activate , Slave node will take role of master (acting) 5. Perform the upgrade . 6. On Original Master node perform request ha commands activate 7. Ensure the node is acting as slave . |  |
| **7** | Resource Pools exhaused | Resource Pools exhausted. | New Services failed to deploy | Resource pools to be added | 1. Delete failed service. 2. Add resource pools . 3. Create new service | 1. delete sae-site <site-name> service <service-name> 2. set resource pools id-pool <id-pool> or set resource pools ip-pool <ip-pool> 3. set sae-site <site-name> service <service-name> |  |

## SAE Customization

### Overview

This section is about

* + - Customizing configs on switches, CSPs and VNFs after SAE deployment through applying custom templates.
    - Customizing SAE deployments through template overriding feature.

### Custom Template

A package "custom-template-utils" has been integrated to SAE core function pack to enable custom template feature in SAE.

* + - While customizing devices/VNFs through custom templates, it is expected that the user must know the effect/affect of customizing the devices/VNFs before/after service deployment.
    - User takes all responsibility and owns issues, due to conflicting configs pushed to devices/VNFs through custom templates before/after service deployment.
    - User must take care while customizing devices/VNFs before service deployment and must know about all the configs get applied to devices/VNFs through service deployment, because, before service deployment, the configs pushed through custom templates may make the future configs conflicting, which are going to be applied through service deployment.

### Yang Model

custom-template-utils package introduces model that exposes custom template functionality. grouping template-hook is used in SAE models to support for custom templates from north bound.

### Prerequisites/Dependencies

Template naming restriction: Name of custom templates must start with either "ct-" or "CT-".

Variable naming restriction: The device name variable in config templates must be either "DEVICE\_NAME" or "DEVICE". These two variable name are reserved only for device name in templates and hence must not be used for anything else in a template.

Template creation: Create template (config/device) considering above mentioned restrictions. Follow NSO documentation for template creation.

Load custom templates to NSO: Load the created custom templates to NSO. Device templates are directly load merged in ncs cli. Config templates can be kept under day1-templates/templates directory and reload packages to load the custom config templates to NSO.

### Usage with Sample Payloads

**Custom Template Generic Sample Payloads**

Please refer below custom-template generic payload depending on various usage scenarios. These payloads can be incorporated into SAE payloads to apply custom templates in SAE.

Apply custom template to a device single time:

To apply custom templates, the name of the template and its variables name and value for 1 iteration have to be set in the model.

| **Without variable** | **With variables** |
| --- | --- |
| | **Cli** | **XML** | | --- | --- | | |  | | --- | | custom-template {$CUSTOM\_TEMPLATE\_NAME} {    iteration 1;  } | | |  | | --- | | <custom-template>    <name>{$CUSTOM\_TEMPLATE\_NAME}</name>    <iteration>      <number>1</number>    </iteration>  </custom-template> | | | | **Cli** | **XML** | | --- | --- | | |  | | --- | | custom-template {$CUSTOM\_TEMPLATE\_NAME} {    iteration 1 {      variable {$FIRST\_VARIABLE} {        value {$FIRST\_VARIABLE\_VALUE};      }      variable {$SECOND\_VARIABLE} {        value {$SECOND\_VARIABLE\_VALUE};      }    }  } | | |  | | --- | | <custom-template>    <name>{$CUSTOM\_TEMPLATE\_NAME}</name>    <itertion>      <number>1</number>      <variable>        <name>{$FIRST\_VARIABLE}</name>        <value>{$FIRST\_VARIABLE\_VALUE}</value>      </variable>      <variable>        <name>{$SECOND\_VARIABLE}</name>        <value>{$SECOND\_VARIABLE\_VALUE}</value>      </variable>    </iteration>  </custom-template> | | |

Apply custom template to a device multiple times:

To apply custom template multiple times, the name of the template and its variables name and value for n iteration have to be set in the model. Depending on how many iterations of variables defined, the custom template will get applied that many times.

| **Cli** | **XML** |
| --- | --- |
| |  | | --- | | custom-template {$CUSTOM\_TEMPLATE\_NAME} {    iteration {$ITERATION\_NUMBER} {      variable {$FIRST\_VARIABLE} {        value {$FIRST\_VARIABLE\_VALUE};      }      variable {$SECOND\_VARIABLE} {        value {$SECOND\_VARIABLE\_VALUE};      }    }  } | | |  | | --- | | <custom-template>    <name>{$CUSTOM\_TEMPLATE\_NAME}</name>    <iteration>      <number>{$ITERATION\_NUMBER}</number>      <variable>        <name>{$FIRST\_VARIABLE}</name>        <value>{$FIRST\_VARIABLE\_VALUE}</value>      </variable>      <variable>        <name>{$SECOND\_VARIABLE}</name>        <value>{$SECOND\_VARIABLE\_VALUE}</value>      </variable>    </iteration>  </custom-template> | |

The value of variable ITERATION\_NUMBER ranges from 1 to 65,535. So for example, if for a custom template 3 iteration defined with each iteration having all template variables with different values, then template will be applied 3 times.

Pass xpath reference as a value of a custom template variable

xpath reference path can be set as a value to a variable through setting xpath-value leaf. The value of xpath provided will be calculated before applying the template.

| **Apply single time** | **Apply multiple times** |
| --- | --- |
| | **Cli** | **XML** | | --- | --- | | |  | | --- | | custom-template {$CUSTOM\_TEMPLATE\_NAME} {    iteration 1 {      variable {$FIRST\_VARIABLE} {        xpath-value {$FIRST\_VARIABLE\_XPATH\_VALUE};      }      variable {$SECOND\_VARIABLE} {        value {$SECOND\_VARIABLE\_VALUE};      }    }  } | | |  | | --- | | <custom-template>    <name>{$CUSTOM\_TEMPLATE\_NAME}</name>    <iteration>      <number>1</number>      <variable>        <name>{$FIRST\_VARIABLE}</name>        <xpath-value>{$FIRST\_VARIABLE\_XPATH\_VALUE}</xpath-value>      </variable>      <variable>        <name>{$SECOND\_VARIABLE}</name>        <value>{$SECOND\_VARIABLE\_VALUE}</value>      </variable>    </iteration>  </custom-template> | | | | **Cli** | **XML** | | --- | --- | | |  | | --- | | custom-template {$CUSTOM\_TEMPLATE\_NAME} {    iteration {$ITERATION\_NUMBER} {      variable {$FIRST\_VARIABLE} {        value {$FIRST\_VARIABLE\_VALUE};      }      variable {$SECOND\_VARIABLE} {        xpath-value {$SECOND\_VARIABLE\_XPATH\_VALUE};      }    }  } | | |  | | --- | | <custom-template>    <name>{$CUSTOM\_TEMPLATE\_NAME}</name>    <iteration>      <number>{$ITERATION\_NUMBER}</number>      <variable>        <name>{$FIRST\_VARIABLE}</name>        <xpath-value>{$FIRST\_VARIABLE\_VALUE}</xpath-value>      </variable>      <variable>        <name>{$SECOND\_VARIABLE}</name>        <xpath-value>{$SECOND\_VARIABLE\_XPATH\_VALUE}</xpath-value>      </variable>    </iteration>  </custom-template> | | |

**Turn ON/OFF custom-templates**

Custom-template feature can be turned ON/OFF through flag "apply-custom-template" value true/false respectively. Default value is true.

admin@ncs% set apply-custom-template <true/false>

[ok]

[edit]

admin@ncs% commit

ON/OFF custom-template Expand source

### SAE Custom Template Payloads

Custom templates can be applied to Switches, CSPs and VNFs in SAE core function pack. To apply the custom templates, user has to populate the custom-template model in various sae-site services through above mentioned generic custom-template payloads and commit.

Before applying customization to any device, please make sure the device is ready and in nso device tree and the services to which the device belong are also deployed and ready through plan or oper data.

**Apply custom templates to leaf switches**

Populate custom-template model inside /sae-site{site-name}/infrastructure/compute-clusters{cluster-name}/leaf-switches{switch-name}.

| **Cli** | **Netconf/REST** |
| --- | --- |
| |  | | --- | | sae-site {$SITE\_NAME} {    infrastructure {      compute-clusters {$CLUSTER\_NAME} {        leaf-switches {$SWITCH\_NAME} {          # custom template sample cli pay-load goes here          custom-template {$CUSTOM\_TEMPLATE\_NAME} {            .            .            .          }        }      }    }  } | | |  | | --- | | <config xmlns="<http://tail-f.com/ns/config/1.0>">    <sae-site xmlns="http://com/cisco/nso/sae-core-fp-cfs">      <site>{$SITE\_NAME}</site>      <infrastructure>        <compute-clusters>          <name>{$CLUSTER\_NAME}</name>          <leaf-switches>            <name>{$SWITCH\_NAME}</name>              <!--custom template sample xml pay-load goes here-->            <custom-template>              .              .              .            </custom-template>            </leaf-switches>        </compute-clusters>      </infrastructure>    </sae-site>  </config> | |

Load merge payload like above sample payload and commit to apply the custom templates to leaf switch.

Apply custom templates to CSPs

Populate custom-template model inside /sae-site{site-name}/infrastructure/compute-clusters{cluster-name}/servers{CSP-name}.

| **Cli** | **Netconf/REST** |
| --- | --- |
| Collapse source   |  | | --- | | sae-site {$SITE\_NAME} {    infrastructure {      compute-clusters {$CLUSTER\_NAME} {        servers {$CSP\_NAME} {          # custom template sample cli pay-load goes here          custom-template {$CUSTOM\_TEMPLATE\_NAME} {            .            .            .          }        }      }    }  } | | Collapse source   |  | | --- | | <config xmlns="<http://tail-f.com/ns/config/1.0>">    <sae-site xmlns="http://com/cisco/nso/sae-core-fp-cfs">      <site>{$SITE\_NAME}</site>      <infrastructure>        <compute-clusters>          <name>{$CLUSTER\_NAME}</name>          <servers>            <name>{$CSP\_NAME}</name>              <!--custom template sample xml pay-load goes here-->            <custom-template>              .              .              .            </custom-template>            </servers>        </compute-clusters>      </infrastructure>    </sae-site>  </config> | |

Load merge payload like above sample payload and commit to apply custom templates to CSP.

Apply custom templates to end point gateway VNFs

Populate custom-template model inside /sae-site{site-name}/endpoint-gateway-vnf{epgw-name}/customization/vnf-deployment{vnf-profile-name}.

| **Cli** | **Netconf/REST** |
| --- | --- |
| |  | | --- | | sae-site {$SITE\_NAME} {    endpoint-gateway-vnf {$EPGW\_NAME} {      customization {        vnf-deployment {$VNF\_PROFILE\_NAME} {          # custom template sample cli pay-load goes here          custom-template {$CUSTOM\_TEMPLATE\_NAME} {            .            .            .          }        }      }    }  } | | |  | | --- | | <config xmlns="<http://tail-f.com/ns/config/1.0>">    <sae-site xmlns="http://com/cisco/nso/sae-core-fp-cfs">      <site>{$SITE\_NAME}</site>      <endpoint-gateway-vnf>        <name>{$EPGW\_NAME}</name>        <customization>          <vnf-deployment>            <profile>{$VNF\_PROFILE\_NAME}</profile>              <!--custom template sample xml pay-load goes here-->            <custom-template>              .              .              .            </custom-template>            </vnf-deployment>        </customization>      </endpoint-gateway-vnf>    </sae-site>  </config> | |

Load merge payload like above sample payload and commit to apply custom templates to end point gateway VNFs. The VNFs are determined through the VNF\_PROFILE\_NAME provided through yang model.

Apply custom templates to service chain VNFs

Populate custom-template model inside /sae-site{site-name}/service-chain{srvc-chain-name}/customization/vnf-deployment{vnf-profile-name}.

| **Cli** | **Netconf/REST** |
| --- | --- |
| |  | | --- | | sae-site {$SITE\_NAME} {    service-chain {$SRVC\_CHAIN\_NAME} {      customization {        vnf-deployment {$VNF\_PROFILE\_NAME} {          # custom template sample cli pay-load goes here          custom-template {$CUSTOM\_TEMPLATE\_NAME} {            .            .            .          }        }      }    }  } | | |  | | --- | | <config xmlns="<http://tail-f.com/ns/config/1.0>">    <sae-site xmlns="http://com/cisco/nso/sae-core-fp-cfs">      <site>{$SITE\_NAME}</site>      <service-chain>        <name>{$SRVC\_CHAIN\_NAME}</name>        <customization>          <vnf-deployment>            <profile>{$VNF\_PROFILE\_NAME}</profile>              <!--custom template sample xml pay-load goes here-->            <custom-template>              .              .              .            </custom-template>            </vnf-deployment>        </customization>      </service-chain>    </sae-site>  </config> | |

Load merge payload like above sample payload and commit to apply custom templates to service chain VNFs. The VNFs are determined through the VNF\_PROFILE\_NAME provided through yang model.

**Pass SAE xpath reference to value of a custom-template variable**

Same as mentioned above. Check pass xpath reference as a value of a custom template variable. Please check below example:

**Pass xpath-value**

admin@ncs% show cisco-sae-core-fp-cfs:sae-site sae-site-1 infrastructure compute-clusters cluster-1 servers csp0 var NTP\_SERVER val | display xpath

/cisco-sae-core-fp-cfs:sae-site[site='sae-site-1']/infrastructure/compute-clusters[name='cluster-1']/servers[name='csp0']/var[name='NTP\_SERVER']/val [10.10](https://confluence-eng-sjc1.cisco.com/conf/display/NSOUS/10.10).0.0

[ok][2018-05-16 11:01:15]

[edit]

admin@ncs% set sae-site sae-site-1 infrastructure compute-clusters cluster-1 servers csp1 custom-template CT-csp\_ntp iteration 1 variable NTP\_IP xpath-value /cisco-sae-core-fp-cfs:sae-site[site='sae-site-1']/infrastructure/compute-clusters[name='cluster-1']/servers[name='csp0']/var[name='NTP\_SERVER']/val

[ok][2018-05-16 11:02:05]

[edit]

admin@ncs% show | compare

 cisco-sae-core-fp-cfs:sae-site sae-site-1 {

     infrastructure {

         compute-clusters cluster-1 {

             servers csp1 {

+                # first

+                custom-template CT-csp\_ntp {

+                    variables NTP\_IP {

+                        xpath-value /cisco-sae-core-fp-cfs:sae-site[site='sae-site-1']/infrastructure/compute-clusters[name='cluster-1']/servers[name='csp0']/var[name='NTP\_SERVER']/val;

+                    }

+                }

             }

         }

     }

 }

[ok][2018-05-16 11:02:13]

[edit]

admin@ncs% commit

Commit complete.

[ok][2018-05-16 11:02:25]

**Rollback SAE customization through custom-templates**

Turn OFF custom-template feature by setting flag "apply-custom-template" to false.

custom-redeploy sae-site to rollback all SAE customization through applying custom-templates altogether .

### Verification

Check custom templates loaded to NSO

Press tab after custom-template to see the loaded custom templates in ncs\_cli. Please refer below example:

**Check custom templates**

admin@ncs% set sae-site sae-site-1 infrastructure compute-clusters cluster-1 leaf-switches nexus0 custom-template

Possible completions:

  CT-csp\_ntp  CT-nexus\_itd\_device\_group\_active  CT-nexus\_itd\_device\_group\_standby  ct-csp\_clock  ct-nexus\_feature\_itd  ct-nexus\_itd\_service

**Check variables of a custom template**

Press tab after custom-template <template-name> variables in ncs\_cli to see the variables defined in the custom template. Please refer below example:

**Check variables**

admin@ncs% set sae-site sae-site-1 infrastructure compute-clusters cluster-1 leaf-switches nexus0 custom-template ct-nexus\_itd\_service iteration 1 variable ?

Possible completions:

  ITD\_DEVICE\_GROUP  ITD\_SERVICE\_NAME  ITD\_SRVC\_INGRESS\_INTF  ITD\_SRVC\_LB\_BUCKET  ITD\_SRVC\_PEER\_SRVC  ITD\_SRVC\_VIRTUAL\_IP  ITD\_SRVC\_VIRTUAL\_MASK  ITD\_SRVC\_VRF

**Check customization notifications**

Notifications are sent when custom templates are APPLIED, UPDATED, DELETED and FAILED. The command to check notification is

**Check notification**

admin@ncs> show notification stream customization-notif

Please refer below sample output of above command:

| **Status** | **Description** |
| --- | --- |
| APPLIED | When a custom template is applied to device. |
| UPDATED | When a custom template is updated. It can be a actual update to the template or just a service re-deploy. |
| DELETED | When a custom template is deleted. Custom template can also be deleted when service is re-deployed after setting **apply-custom-template** flag to **false**. |
| FAILED | When a custom template is failed to apply to a device, may be due to missing variable value or incompatible value or wrong template etc.. |
| NO\_OPERATION | There will be no operation on template, if there is any *CREATE/UPDATE/DELETE* operation on template when **apply-custom-template** flag is **false**. This notification will be sent for all the templates defined under list custom-template for a device. |

Please refer below sample output of above command:

**Notification sample output**

admin@ncs> show notification stream customization-notif

notification {

    eventTime 2018-06-26T21:08:[26.975](https://confluence-eng-sjc1.cisco.com/conf/display/NSOUS/26.975)+00:00

    customization-notif {

        name ct-hosttemplate\_asa

        device asa0

        status DELETED

        message apply-custom-template flag is false

    }

}

notification {

    eventTime 2018-06-26T21:10:[23.271](https://confluence-eng-sjc1.cisco.com/conf/display/NSOUS/23.271)+00:00

    customization-notif {

        name ct-interfacetemplate\_cpe

        device cpe0

        status APPLIED

        message APPLY SUCCESS

    }

}

notification {

    eventTime 2018-06-26T21:21:[14.579](https://confluence-eng-sjc1.cisco.com/conf/display/NSOUS/14.579)+00:00

    customization-notif {

        name ct-usertemplate\_asa

        device cpe1

        status FAILED

        message Error applying device template: ct-usertemplate\_asa

    }

}

notification {

    eventTime 2018-06-26T21:23:[28.823](https://confluence-eng-sjc1.cisco.com/conf/display/NSOUS/28.823)+00:00

    customization-notif {

        name CT-asa\_login

        device asa0

        status APPLIED

        message UPDATE SUCCESS

    }

}

notification {

    eventTime 2018-06-26T21:23:[29.718](https://confluence-eng-sjc1.cisco.com/conf/display/NSOUS/29.718)+00:00

    customization-notif {

        name CT-login

        device cpe0

        status DELETED

        message DELETE SUCCESS

    }

}

notification {

    eventTime 2018-06-29T19:50:[17.848](https://confluence-eng-sjc1.cisco.com/conf/display/NSOUS/17.848)+00:00

    customization-notif {

        name CT-asa\_interface

        device asa0

        status NO\_OPERATION

        message apply-custom-template flag is false

    }

}

**Check custom templates apply success status**

Custom templates successful apply status on devices can be checked by executing "show custom-template-status".

This command only displays the custom templates applied successfully to devices. To check for failed or deleted status, check customization notifications.

Please refer below sample output:

**Sample output of custom-template-status**

admin@ncs> show custom-template-status

NAME                              DEVICE  STATUS   MESSAGE

------------------------------------------------------------

CT-csp\_ntp                        csp2    APPLIED  SUCCESS

CT-csp\_ntp                        csp3    APPLIED  SUCCESS

CT-csp\_ntp                        csp4    APPLIED  SUCCESS

ct-csp\_clock                      csp0    APPLIED  SUCCESS

ct-csp\_clock                      csp1    APPLIED  SUCCESS

ct-nexus\_feature\_itd              nexus0  APPLIED  SUCCESS

ct-nexus\_feature\_itd              nexus1  APPLIED  SUCCESS

ct-nexus\_itd\_service              nexus0  APPLIED  SUCCESS

CT-nexus\_itd\_device\_group\_active  nexus0  APPLIED  SUCCESS

### Troubleshooting and Diagnostic

Check customization notifications to see for custom template apply failure reason message.

Check ncs-java-vm.log and device and VNF's NED traces for errors.

#### Common Issues and Actions

Missing variable value: Variable is not provided in payload or through SAE internal VNF variables. Variable name mismatch due to typo. Provide correct variable in payload.

Incompatible value: The value provided for a variable is rejected by NED yang validation/restrictions, as NED knows the value provided will be rejected by the device/VNF. Provide compatible value for variables in payload.

External NED error: Device/VNF is throwing the error for the config applied through a custom template. Verify on the device/VNF, why it is rejecting the config pushed by custom template. Delete the custom-template payload. Update the template as per the correct config, load the template to nso and re-apply the template.

Internal error while applying template: Any issues, which also include the above mentioned issues while applying a device template used as a custom template. Check for the actual issue and take corrective action.

### Packaging, Install, versioning

Package: custom-template-utils

Version: 0.1.0

Install: Installed through SAE installer.

### Testcase

Capture automation testcase which covering this feature

### Sources

Please refer yang model custom-template-hook.yang

Please refer CustomTemplateUtils.java for various API methods exposed for integration.

Please refer cisco-sae-core-fp-cfs.yang to look for grouping template-hook call from custom-template-hook.yang

Please refer ctCompletion method of SaeDeviceCliCompletionCallback.java which calls API methods provided in CustomTemplateUtils.java to populate template name and variable name cli completion.

Please refer applyCustomTemplates method of SaeUtils.java which calls API method provided in CustomTemplateUtils.java to apply custom template.

Please refer README to for any other feature and integration information.

### Template Overriding

**Description**

When default configuration pushed via the config template do not fit a given requirement or deployment, now it is possible to replace default config template with an alternative one through this template overriding feature.

This feature can only be used for config templates used inside various services in SAE and not for any custom templates.

### Yang Model

**Model: core-fp-common-template-mapper**

module: core-fp-common-template-mapper

    +--rw override-template

       +--rw ignore-override-settings?   boolean

       +--rw service-template-mapper\* [service-template]

       |  +--rw service-template     string

       |  +--rw override-template?   string

       +--rw per-provider\* [provider]

          +--rw provider                    string

          +--rw ignore-override-settings?   boolean

          +--rw service-template-mapper\* [service-template]

          |  +--rw service-template     string

          |  +--rw override-template?   string

          +--rw per-tenant\* [tenant]

             +--rw tenant                      string

             +--rw ignore-override-settings?   boolean

             +--rw service-template-mapper\* [service-template]

                +--rw service-template     string

                +--rw override-template?   string

### Prerequisites/Dependencies

* + - You must know what are all the config template exists in SAE core function pack packages, which are used to apply configs to switches and VNFs.
    - You must know the template names, content of each template like all the variables used in a template and the configs applied through a template.
    - You must know the values passed for each variable for the templates through code and how many times a template applied etc..
    - You must know where the templates are applied, like whether in compute cluster service, end-pont-gateway, service-chain or stitching-service etc.
    - You must know how to create and load config templates to NSO. Please refer NSO documentation.
    - The alternate template must be a config template and must have same or lesser template variables as the original template and must be loaded to NSO.
    - The configs pushed through alternate template to a device/VNF must be acceptable by device/VNF considering the fact that the values of the variables in alternate template will still be the same as for the original template passed through the source code.

### Usage with Sample Payloads

**Populate the yang model**

Please refer below to populate the model to maintain the template mappings at global, provider and tenant level.

| **Commands** | **Cli** | **Netconf/REST** |
| --- | --- | --- |
| **Global flag**   |  | | --- | | set override-template ignore-override-settings {$FLAG} | | **Global flag**   |  | | --- | | override-template {    ignore-override-settings {$FLAG};  } | | **Global flag**   |  | | --- | | <config xmlns="<http://tail-f.com/ns/config/1.0>">    <override-template xmlns="http://com/cisco/cfp-common-temp-mapper">      <ignore-override-settings>{$FLAG}</ignore-override-settings>    </override-template>  </config> | |
| **Provider flag**   |  | | --- | | set override-template per-provider {$PROVIDER} ignore-override-settings {$FLAG} | | **Provider flag**   |  | | --- | | override-template {    per-provider {$PROVIDER} {      ignore-override-settings {$FLAG};    }  } | | **Provider flag**   |  | | --- | | <config xmlns="<http://tail-f.com/ns/config/1.0>">    <override-template xmlns="http://com/cisco/cfp-common-temp-mapper">      <per-provider>        <provider>{$PROVIDER}</provider>        <ignore-override-settings>{$FLAG}</ignore-override-settings>      </per-provider>    </override-template>  </config> | |
| **Tenant flag**   |  | | --- | | set override-template per-provider {$PROVIDER} per-tenant {$TENANT} ignore-override-settings {$FLAG} | | **Tenant flag**   |  | | --- | | override-template {    per-provider {$PROVIDER} {      per-tenant {$TENANT} {        ignore-override-settings {$FLAG};      }    }  } | | **Tenant flag**   |  | | --- | | <config xmlns="<http://tail-f.com/ns/config/1.0>">    <override-template xmlns="http://com/cisco/cfp-common-temp-mapper">      <per-provider>        <provider>{$PROVIDER}</provider>        <per-tenant>          <tenant>{$TENANT}</tenant>          <ignore-override-settings>{$FLAG}</ignore-override-settings>        </per-tenant>      </per-provider>    </override-template>  </config> | |

| **Variable Name** | **Variable Description** |
| --- | --- |
| ORIGINAL\_TEMPLATE\_NAME | The name of the original config template, which need to be overridden. |
| ALTERNATE\_TEMPLATE\_NAME | The name of the alternate template, which will be applied instead of the original template. |
| PROVIDER | The name of the provider for which template mapping is to be maintained. |
| TENANT | The name of the tenant under a provider, for which template mapping is to be maintained. |

#### Deploy/Custom-redeploy services

After populating the model, deploy the services if services are not deployed. While deploying the services alternative templates will be selected to apply instead of original template, whose mappings are populated by previous step.

If the services are already deployed, then custom-redeploy specific service, in which the specific original template is applied. So that, during re-deployment, alternate template will be selected to apply for that original template.

#### Turn OFF/ON template overriding feature

Set the flag ignore-override-settings to true to turn OFF and to false to turn ON the feature under override-template at global level. Same flag is also present under provider and tenant. So we can turn off/on feature at provider and tenant level also. Default value of this flag is false.

\

| **Commands** | **Cli** | **Netconf/REST** |
| --- | --- | --- |
| **Global flag** Collapse source   |  | | --- | | set override-template ignore-override-settings {$FLAG} | | **Global flag** Collapse source   |  | | --- | | override-template {    ignore-override-settings {$FLAG};  } | | **Global flag** Collapse source   |  | | --- | | <config xmlns="<http://tail-f.com/ns/config/1.0>">    <override-template xmlns="http://com/cisco/cfp-common-temp-mapper">      <ignore-override-settings>{$FLAG}</ignore-override-settings>    </override-template>  </config> | |
| **Provider flag** Collapse source   |  | | --- | | set override-template per-provider {$PROVIDER} ignore-override-settings {$FLAG} | | **Provider flag** Collapse source   |  | | --- | | override-template {    per-provider {$PROVIDER} {      ignore-override-settings {$FLAG};    }  } | | **Provider flag** Collapse source   |  | | --- | | <config xmlns="<http://tail-f.com/ns/config/1.0>">    <override-template xmlns="http://com/cisco/cfp-common-temp-mapper">      <per-provider>        <provider>{$PROVIDER}</provider>        <ignore-override-settings>{$FLAG}</ignore-override-settings>      </per-provider>    </override-template>  </config> | |
| **Tenant flag** Collapse source   |  | | --- | | set override-template per-provider {$PROVIDER} per-tenant {$TENANT} ignore-override-settings {$FLAG} | | **Tenant flag** Collapse source   |  | | --- | | override-template {    per-provider {$PROVIDER} {      per-tenant {$TENANT} {        ignore-override-settings {$FLAG};      }    }  } | | **Tenant flag** Collapse source   |  | | --- | | <config xmlns="<http://tail-f.com/ns/config/1.0>">    <override-template xmlns="http://com/cisco/cfp-common-temp-mapper">      <per-provider>        <provider>{$PROVIDER}</provider>        <per-tenant>          <tenant>{$TENANT}</tenant>          <ignore-override-settings>{$FLAG}</ignore-override-settings>        </per-tenant>      </per-provider>    </override-template>  </config> | |

| **Variable Name** | **Variable Description** |
| --- | --- |
| FLAG | The value of flag. **True to turn off and false to turn on.** |
| PROVIDER | The name of the provider for which the feature is to be turned off/on. |
| TENANT | The name of the tenant under a provider, for which the feature is to be turned off/on. |

#### Rollback to original configs

First turn OFF the template override feature.

custom-redeploy SAE services to rollback configs applied through alternate templates and original configs through original templates will be applied.

### Verification

After sae-site-staus plan is reached successfully, login to devices or VNFs and check for configs and verify if the configs pushed through alternate templates are correct.

### Troubleshooting and Diagnostic

Check sae-site-status plan message to see for error message.

Check ncs-java-vm.log and device and VNF's NED traces for errors.

### Common Issues & Action

**Missing variable value:** Variable name mismatch due to typo or More variable present in alternate template as compared to original template. Remove the variable from alternate template and put hard coded value or change code to pass value for the new variables, reload the packages and redeploy(custom-redeploy, custom-undeploy-redeploy etc..) affected services.

**Incompatible value:** The value provided for a variable is rejected by NED yang validation/restrictions, as NED knows the value provided will be rejected by the device/VNF. Remove the variable from alternate template and put hard coded value or change code to pass compatible value for the variable for original as well as alternate template, reload the packages and redeploy(custom-redeploy, custom-undeploy-redeploy etc..) affected services.

**External NED error:** Device/VNF is throwing the error for the config applied through a alternate template. Verify on the device/VNF, why it is rejecting the config pushed by alternate template. Update the alternate template as per the correct config, reload the packages and redeploy(custom-redeploy, custom-undeploy-redeploy etc..) affected services.

### Packaging, Install, versioning

Package: core-fp-common

Version: 1.3.0

Install: Installed through SAE installer.

### Testcase

Capture automation testcase which covering this feature

### Sources

* + - Please refer yang model core-fp-common-template-mapper.yang
    - findMappingTemplate method of TemplateFinderFunction.java will fetch the alternate template name for the original template name passed to it. If there is no alternate template defined or if the alternate template is not loaded to NSO, or if the feature is turned off then the same original template name is returned. Provider and tenant information passed to this method is also taken into consideration while fetching the alternate template name if the alternate template is defined for a particular provider and tenant.
    - getTemplate method of BaseDeviceConfig.java calls the findMappingTemplate method of TemplateFinderFunction.java

## SAE ITD service

### Overview

This page captures the ITD service support details in SAE CFP

### Description

we have another ITD service on top of saes-ite, which will allow user to enable ITD for specified VNF.

This ITD service will take inputs of existing service-chain to apply ITD switch configs, it enables ITD for specified VNF's .

This service is not be part of SAE models, This separate itd service will enable ITD independently for service chain

### Yang Model

**ITD Yang tree**

module: sae-itd-service

    +--rw sae-itd-service\* [name]

       +--rw name                        string

       +--rw site-name                   -> /cisco-sae-core-fp-cfs:sae-site/site

       +--rw service-name                -> deref(../site-name)/../cisco-sae-core-fp-cfs:service-chain/name

       +--rw pre-itd-sapd!

       |  +--rw pre-itd-access-point?       string

       |  +--rw pre-itd-vnf-access-point?   string

       +--rw post-itd-sapd!

          +--rw post-itd-access-point?       string

          +--rw post-itd-vnf-access-point?   string

| **Name** | **Description** |
| --- | --- |
| site-name, service-name | its leafref to the existing sae-site & service, once user select the site name, it'll show you servies where its applicable  currently itd is supported only for end-to-end chain type |
| sae-itd-service name | Name of ITD service, supported length is 28 char |
| post-itd-sapd   * post-itd-access-point * post-itd-vnf-access-point | User has to select correct SAPD name as below, refer NSD diagram  here post-itd means, user has select sapd's which are available after itd-vnf   * post-itd-access-point - vnfs which are connected in post part& not used to load balance traffic * post-itd-vnf-access-point - used to load balance traffic |
| pre-itd-sapd   * pre-itd-access-point * pre-itd-vnf-access-point | User has to select correct SAPD name as below, refer NSD diagram  here pre-itd means, user has select sapd's which are available before itd-vnf   * pre-itd-access-point - vnfs which are connected in pre part & not used to load balance traffic * pre-itd-vnf-access-point - used to load balance traffic |

### Prerequisites/Dependencies

* + - ITD service support is only for service-chain of type end-to-end
    - ITD VNF should be present in middle as displayed in service-chain NSD picture
    - User has to specify pre-itd & post-itd sapd endpoints,
    - Service-chain should be ready reached in plan before creating ITD service
    - ITD yang supports autocompletion, it shows suggested possible completion while using set command
    - refer sample NSD picture, we have added new SAPD type for ITD [pre-itd-endpoint, post-itd-endpoint]

### ITD NSD Diagram

Figure 22. ITD NSD Diagram



### ITD service Payload

**payload sample**

sae-itd-service sae-itd-service {

    site-name    sae-site-1;

    service-name CSR-to-ASA;

    pre-itd-sapd {

        pre-itd-access-point     inter-vnf-cp-3;

        pre-itd-vnf-access-point inter-vnf-cp-1;

    }

    post-itd-sapd {

        post-itd-access-point     inter-vnf-cp-4;

        post-itd-vnf-access-point inter-vnf-cp-2;

    }

}

### Payloads Validation

We do have few validation for ITD service payload:

* + - All ITD service model inputs are mandatory.
    - ITD yang supports autocompletion, it shows suggested possible completion while using set command to create ITD service.
    - ITD service name can not exceed 28 characters, its supports this name pattern "[a-zA-Z0-9\\.\\-]
    - User is not allowed to create ITD service on already used service-name where ITD is already enabled.
    - SAPD inputs for example, pre-itd-sapd and post-itd-sapd are mandatory for ITD service.

### Verification

User can verify ITD service status using below CLI

**CLI**

admin@ncs% run show sae-itd-service-status

NAME      STATUS   MESSAGE

-----------------------------------------------------

itdchain  Success  ITD configs applied successfully

### Troubleshooting and Diagnostic

| **Error** | **Possible Reason** | **Troubleshooting** |
| --- | --- | --- |
| commit aborted, for example:  admin@ncs% comm  Aborted: Exception in callback: Failed to create sae service/sae-itd-service:sae-itd-service{itdsrv} | wrong itd payload inputs | * + Check for ITD service payload   + Verify NSD as per NSD diagram shown above, make sure it contain all itd-endpoints   + check ncs-java-vm logs for Error message |
| Cisco nexus ned error | Dependent configs/features are not enabled | Check the nexus ned logs for error, it'll show the dependencies & missing features  delete & recreate the itd service after resolving configs dependencies |

### Packaging, Install, versioning

Package name: cisco-sae-itd-service

### Testcase

Capture automation testcase which covering this feature

### Sources

github URL: https://wwwin-github.cisco.com/nso-function-pack/sae/tree/develop/packages/cisco-sae-itd-service

### Webex Recording

Go over the above wiki and sample usage on netsim/real setup in the webex recording.

## SAE VNF License deregistration for IOS and ASA

### Overview

SAE core function pack have the support for VNF license de registration (on IOS and ASA).

### Description

SAE CFP has added support for license de registration, as part of this framework, we can develop different license handlers (each specialized in a particular device type or a group of device types).

After defining VNFDs, a user has to define a mapping between the VNFD/VDU and the license handler which is appropriate for that VNF.

<config xmlns="<http://tail-f.com/ns/config/1.0>">

<vnf-licensing xmlns="http://com/cisco/nso/sae-core-fp-common">

<vnf-profiles>

<vnfd>SAE-CSR</vnfd>

<vdu>CSR</vdu>

<handler-id>cisco-smart-license</handler-id>

</vnf-profiles>

</vnf-licensing>

</config>

Configuring license on VNF is still part of Day-0/Day-1. However, SAE CFP will handle the de-registration according to the mapping provided.

Currently, SAE cfp has one handler "cisco-smart-license" that can deregister licenses on IOS and ASA VNFs.

"what-is" support to see what license handlers are available and what it does:

admin@ncs% show vnf-licensing what-is

what-is cisco-smart-license {

description "Deregisters cisco smart licenses configured on IOS and ASA vnf devices. Issues \" license smart deregister\" towards the vnfs. If a vnf is not is not reachable or if the vnf-deployment is in a failure state, the handler will silently skip the deregistration operation. For any other failure cases, VNF will not be deleted and plan for vnf will be marked with a failure message.";

}

what-is pafw-smart-license {

description "Currently not supported. Placeholder for future changes.";

}

[ok][2018-07-20 15:44:12]

[edit]

admin@ncs%

### Yang Model

Pictorial representation of the Model, reference to Model sources, High level description on the usage, boundaries, caveats, any future use or experimental elements etc. Capture events/notification models as well.

### Prerequisites/Dependencies

### Usage with Sample Payloads

### Payloads Validation

### Verification

### Troubleshooting and Diagnostic

### Packaging, Install, versioning

### Testcase

### Sources

### Webex Recording

# Appendix A. Yang Models and Payloads

## SAE Model

module cisco-sae-core-fp-cfs {

  namespace "[http://com/cisco/nso/sae-core-fp-cfs"](http://com/cisco/nso/sae-core-fp-cfs);

  prefix cisco-sae-core-fp-cfs;

  import ietf-inet-types {

    prefix inet;

  }

  import tailf-common {

    prefix tailf;

  }

  import tailf-ncs {

    prefix ncs;

  }

  import resource-allocator {

    prefix ralloc;

  }

  import ipaddress-allocator {

    prefix ipalloc;

  }

  import id-allocator {

    prefix idalloc;

  }

  import tailf-etsi-rel2-nfvo {

    prefix nfvo;

  }

  import core-fp-common {

    prefix core-fp-common;

  }

  import cisco-sae-core-fp-common {

    prefix cisco-sae-core-fp-common;

  }

  import tailf-etsi-rel2-nfvo {

    prefix nfvo-rel2;

  }

  import tailf-etsi-rel2-nfvo-esc {

    prefix nfvo-rel2-esc;

  }

  import tailf-etsi-rel2-nfvo-ro {

    prefix nfvo-rel2-ro;

  }

  import csp-vim {

    prefix csp-vim;

  }

  import cisco-extension-framework {

    prefix cisco-extension-framework;

  }

  import custom-template-hook {

    prefix ct-hook;

  }

  revision 2018-07-27 {

      description

       "Initial revision for 1.0.0 Release.";

  }

  typedef variableName {

    type string {

      pattern "[a-zA-Z0-9!#$%&=:/\_\\\\|\\.\\-\\^\\?\\\*\\+\\{\\}\\(\\)]+";

    }

  }

  typedef service-name {

    type string {

      pattern "[a-zA-Z0-9\\.\\-]+" {

        error-message "Invalid characters in site-name!";

      }

      length "1..25" {

        error-message "Name can not exceed 25 characters!";

      }

    }

  }

  typedef network-name {

    type string {

      length 1..1024;

    }

  }

  grouping vnf-data {

    leaf vdu {

      tailf:info "VDU for this VNF";

      description "VDU for this VNF";

      type leafref {

        path "/nfvo-rel2:nfvo/nfvo-rel2:vnfd[nfvo-rel2:id = current()/../vnfd]/"

        + "nfvo-rel2:vdu/nfvo-rel2:id";

      }

      mandatory true;

    }

    /\*    leaf type {

      tailf:info "Type of VNF (host/transit/hairpin)";

      type enumeration {

        enum "host" {

          tailf:info

            "A host type VNF has one or more interfaces connected to the network fabric to

            receive and consume traffic. The VNF doesn’t forward traffic to the next node

            in the service chain. Examples: Packet capture VMs, Web Server,

            Network Analysis Module, etc.";

          description

            "A host type VNF has one or more interfaces connected to the network fabric to

            receive and consume traffic. The VNF doesn’t forward traffic to the next node

            in the service chain. Examples: Packet capture VMs, Web Server,

            Network Analysis Module, etc.";

        }

        enum "transit" {

          tailf:info

              "A transit type VNF receives traffic from the fabric on one interface (inside)

              and forwards the traffic to the fabric through another interface (outside).

              Examples: Router, firewall, VPN concentrator, etc.

              The VNF may also use an additional interface for cluster/failover connectivity

              and synchronizing state information with a standby VNF";

          description

              "A transit type VNF receives traffic from the fabric on one interface (inside)

              and forwards the traffic to the fabric through another interface (outside).

              Examples: Router, firewall, VPN concentrator, etc.

              The VNF may also use an additional interface for cluster/failover connectivity

              and synchronizing state information with a standby VNF";

        }

        enum "hairpin" {

          tailf:info

              "A hair-pin type VNF receives traffic from one VNF in the service chain,

              performs a service and then sends the traffic back to the VNF.

              For example, the WAAS VM receives traffic from a router VM using a protocol like

              AppNav, optimizes the traffic and sends the traffic back to the router VM.";

          description

              "A hair-pin type VNF receives traffic from one VNF in the service chain,

              performs a service and then sends the traffic back to the VNF.

              For example, the WAAS VM receives traffic from a router VM using a protocol like

              AppNav, optimizes the traffic and sends the traffic back to the router VM.";

        }

      }

      default "transit";

    }

    leaf cluster-or-failover {

      when "../type ='transit'";

      tailf:info "Set to true if VNF natively supports some form of redundancy";

      description "true: VNF natively support some form of redundancy either Cluster

                          (active-active) or Fail-over (active-standby)

                  false: VNF doest not support redundancy natively";

      type boolean;

      default "false";

    }

\*/

    leaf mode {

      tailf:info "VNF mode";

      description "VNF mode";

      type enumeration {

        enum "transparent" {

          tailf:info

                    "VNF is in transparent mode and is not router hop in the network";

          description

                    "VNF is in transparent mode and is not router hop in the network";

        }

        enum "routed" {

          tailf:info

                    "Routed mode VNF, each interface that you want to route between is on a"

                  + " different subnet";

          description

                    "Routed mode VNF, each interface that you want to route between is on a"

                    + " different subnet";

        }

      }

      default "routed";

    }

  }

  grouping variables {

    list var {

      tailf:info "Variable as key/value pair";

      description "Variable as key/value pair";

      key name;

      leaf name {

        type variableName;

        tailf:info "Variable name";

        description "Variable name";

      }

      choice valtype {

        leaf val {

          tailf:info "static variable value";

          description "static variable value";

          type string;

          mandatory true;

        }

        leaf encrypted-val {

          tailf:info "Encrypted Variable value.";

          description "Encrypted Variable value.";

          type tailf:aes-cfb-128-encrypted-string;

          tailf:suppress-echo true;

          mandatory true;

        }

      }

    }

  }

  grouping day1-config {

    list day1-config {

      tailf:info "List of Day1 Config to be applied on the device";

      description "List of Day1 Config to be applied on the device";

      key name;

      leaf name {

        tailf:info "Config name";

        description "Config name";

        type string;

      }

      leaf cfg-template {

        tailf:info "Config template name";

        description "Config template name";

        type string;

        mandatory true;

      }

      uses variables;

    }

  }

  grouping deployment-params {

    leaf intangible {

      tailf:info "Set this leaf if VNF is intangible, no day1 config to be applied";

      description "Set this leaf if VNF is intangible, no day1 config to be applied";

      type empty;

    }

    leaf bootup-time {

      tailf:info "Maximum bootup time in seconds";

      description "Maximum bootup time in seconds";

      type int32;

      default 600;

    }

    leaf recovery-wait-time {

      tailf:info "Recovery wait time in seconds";

      description "Recovery wait time in seconds";

      type int32;

      default 90;

    }

    leaf vnf-authgroup {

      tailf:info "Device Authgroup for VNF";

      description "Device Authgroup for VNF";

      when "not(../intangible)";

      type leafref {

        path "/ncs:devices/ncs:authgroups/ncs:group/ncs:name";

      }

      default "default";

    }

    leaf ssh-authgroup {

      tailf:info "Points out the authentication group used to ssh into the device if a non-cli"

        +" device is used";

      description "Points out the authentication group used to ssh into the device if a non-cli"

      +" device is used";

      type leafref {

        path '/ncs:devices/ncs:authgroups/ncs:group/ncs:name';

      }

    }

    leaf vnc-password {

      tailf:info "Password to be set for VNC log-in";

      type string;

      default "ciscoAdmin";

    }

    list day0 {

      tailf:info "List of VNF's day0.";

      description "List of VNF's day0.";

      key destination;

      leaf destination {

        tailf:info "Destination filename for this day0 file.";

        description "Destination filename for this day0 file.";

        type leafref {

          path "/nfvo-rel2:nfvo/nfvo-rel2:vnfd[nfvo-rel2:id = "

          + "current()/../../cisco-sae-core-fp-cfs:vnfd]"

          + "/nfvo-rel2:vdu[nfvo-rel2:id = current()/../../cisco-sae-core-fp-cfs:vdu]"

          + "/nfvo-rel2-esc:day0/nfvo-rel2-esc:destination";

        }

      }

      choice source {

        description "Specifies where to get the day0 content";

        mandatory true;

        leaf url {

          tailf:info "Day0 file URL";

          description "Day0 file URL";

          type inet:uri;

        }

        leaf data {

          tailf:info "Inline day0 text";

          description "Inline day0 text";

          type tailf:aes-cfb-128-encrypted-string;

        }

      }

      uses variables;

    }

    uses day1-config {

      when "not(intangible)";

    }

  }

  grouping external-end-points {

    list external-end-point {

      tailf:info

            "Represents the Consumers (For example Employees, Customers, Partners) or the Providers"

              +" (For example AWS,Azure, Enterprise-Data Center) in SAE service-chain.";

      key name;

      ordered-by user;

      description

            "Represents the Consumers (For example Employees, Customers, Partners) or the Providers"

                +" (For example AWS,Azure, Enterprise-Data Center) in SAE service-chain.";

      leaf name {

        tailf:info "Unique name for external-end-point";

        description "Unique name for external-end-point";

        type string {

          pattern "[a-zA-Z0-9\\-]+";

          length "1..24" {

            error-message "external-end-point name can not exceed 24 characters!";

          }

        }

      }

      leaf traffic-hand-off-method {

        tailf:info "Traffic method from Consumer/Provider onto first VNF in service-chain";

        description "Represents how the traffic from the Consumer/Provider enters the first VNF in"

                +" the SAE service-chain";

        type cisco-sae-core-fp-common:traffic-hand-off-type-t;

        default cisco-sae-core-fp-common:vlan-only;

      }

      leaf vlan-number {

        tailf:info "VLAN number assigned to Consumer/Provider";

        description "VLAN number assigned to the Consumer/Provider. All traffic from"

                +" Consumer/Provider is in this VLAN";

        type uint16 {

          range "1..4096";

        }

        mandatory true;

      }

      leaf branch-bgp-as-number {

        tailf:info "external-end-point bgp-as towards the branch";

        type uint32 {

          range "0..4294967295";

        }

      }

      leaf branch-remote-as-number {

        tailf:info "external-end-point bgp-as towards the colo";

        type uint32 {

          range "0..4294967295";

        }

      }

      leaf endpoint-ipaddress {

        tailf:info "ASR peer ipaddress";

        type string;

      }

      leaf endpoint-netmask {

        tailf:info "Netmast to be used in VNF's interface connecting to endpoint";

        type inet:ip-address;

        mandatory true;

      }

      uses variables;

      uses day1-config;

    }

  }

  grouping resource-pools {

    container resource-pools {

      presence "sae";

      tailf:info "SAE Resource Pools";

      description "SAE Resource Pools";

      tailf:validate "resource-validation" {

        tailf:dependency ".";

      }

      leaf as-pool {

        tailf:info "Resource Pool for autonomous system number Allocation";

        description "Resource Pool for autonomous system number Allocation";

        type leafref {

          path "/ralloc:resource-pools/idalloc:id-pool/idalloc:name";

        }

        default default-as-pool;

      }

      leaf mgmt-ip-pool {

        tailf:info "Resource Pool for Management IP Address Allocation";

        description "Resource Pool for Management IP Address Allocation";

        type leafref {

          path "/ralloc:resource-pools/ipalloc:ip-address-pool/ipalloc:name";

        }

        mandatory true;

      }

      leaf internal-ip-pool {

        tailf:info "Resource Pool for internal link IP Address Allocation";

        description "Resource Pool for internal link IP Address Allocation";

        type leafref {

          path "/ralloc:resource-pools/ipalloc:ip-address-pool/ipalloc:name";

        }

        mandatory true;

      }

    }

  }

  grouping custom-template-grouping {

    uses ct-hook:template-hook {

      refine "custom-template/iteration/variable/name" {

        tailf:cli-completion-actionpoint "cli-completion" {

          tailf:cli-completion-id "get-variables";

        }

      }

      refine "custom-template/name" {

        tailf:cli-completion-actionpoint "cli-completion" {

          tailf:cli-completion-id "get-templates";

        }

      }

    }

  }

  grouping location-details {

    container location {

      tailf:info "Used to store information [optional] for northbound layer";

      leaf address {

        type string;

      }

      leaf latitude {

        type string;

      }

      leaf longitude {

        type string;

      }

      leaf note {

        type string;

      }

    }

  }

  container sae-validation {

    tailf:info "This shows SAE validation for interface supported-types";

    description "This shows SAE validation for interface supported-types";

    container interface {

      leaf-list supported-type {

        type string;

      }

    }

  }

  list sae-provider {

    tailf:info "List of SAE service providers";

    description "List of SAE service providers";

    key name;

    leaf name {

      tailf:info "SAE service provider name";

      description "SAE service provider name";

      type string;

    }

    leaf sae-provider-catalog {

      tailf:info "Catalog belonging to this SAE service provider";

      description "Catalog belonging to this SAE service provider";

      type leafref {

        path "/cisco-sae-core-fp-cfs:sae-catalog/cisco-sae-core-fp-cfs:name";

      }

      mandatory true;

    }

    list sae-tenant {

      tailf:info "List of SAE service provider's tenants";

      description "List of SAE service provider's tenants";

      key name;

      leaf name {

        tailf:info "SAE tenant name";

        description "SAE tenant name";

        type string;

      }

      leaf sae-tenant-catalog {

        tailf:info "Catalog belonging to this SAE tenant";

        description "Catalog belonging to this SAE tenant";

        type leafref {

          path "/cisco-sae-core-fp-cfs:sae-catalog/cisco-sae-core-fp-cfs:name";

        }

        tailf:default-ref "../../sae-provider-catalog";

      }

    }

    leaf timestamp {

      tailf:hidden sae;

      type string;

      description "This timestamp should be updated internally by redeploy action";

    }

    tailf:action custom-redeploy {

      tailf:info "Action to trigger occupancy copy";

      tailf:actionpoint "cfs-redep-occupancy-action-point";

      output {

        leaf status {

          type string;

        }

        leaf error {

          type string;

        }

      }

    }

  }

  list sae-catalog {

    tailf:info "The catalog which contains details about the types of VNFs and the deployments"

        +" supported in SAE function pack.";

    description "The catalog which contains details about the types of VNFs and the deployments"

        +" supported in SAE function pack.";

    key name;

    leaf name {

      tailf:info "Name of Catalog";

      description "Name of Catalog";

      type string;

    }

    uses variables;

    list csp {

      tailf:info "List of CSP types";

      description "List of CSP types";

      key name;

      leaf name {

        tailf:info "Type of CSP";

        description "Type of CSP";

        type string;

      }

      uses day1-config;

      list virtio-network {

        tailf:info "List of non-sriov networks on CSP";

        description "List of non-sriov networks on CSP";

        key name;

        ordered-by user;

        leaf name {

          tailf:info "Non-SRIOV network name";

          description "Non-SRIOV network name";

          type network-name;

        }

      }

      list sriov-networks {

        tailf:info "List of SRIOV Networks on CSP";

        description "List of SRIOV Networks on CSP";

        key name;

        ordered-by user;

        leaf name {

          tailf:info "SRIOV network name";

          description "SRIOV network name";

          type network-name;

        }

      }

      leaf mgmt-network {

        tailf:info "Management network name on CSP";

        description "Management network name on CSP";

        type network-name;

      }

    }

    list nsd-deployment {

      tailf:info "List of NSD Deployment Details";

      description "List of NSD Deployment Details";

      tailf:validate "nsd-validation" {

        tailf:dependency ".";

      }

      key name;

      leaf name {

        tailf:info "Unique NSD Deployment name";

        description "Unique NSD Deployment name";

        type string;

      }

      leaf nsd {

        tailf:info "NSD for which deployment details are to be specified";

        description "NSD for which deployment details are to be specified";

        type leafref {

          path "/nfvo:nfvo/nfvo:nsd/nfvo:id";

        }

      }

      leaf nsd-deployment-flavor {

        tailf:info "Pick NSD's deployment-flavor defining the NSD";

        description "Pick NSD's deployment-flavor defining the NSD";

        mandatory true;

        type leafref {

          path "deref(../cisco-sae-core-fp-cfs:nsd)/../nfvo-rel2:deployment-flavor/nfvo-rel2:id";

        }

      }

      list vnf-deployment {

        tailf:info "List of vnfd-deployment for each vnf-profile in NSD";

        description "List of vnfd-deployment for each vnf-profile in NSD";

        key "vnf-profile";

        leaf vnf-profile {

          tailf:info "vnf-profile from NSD's deployment-flavor";

          description "vnf-profile from NSD's deployment-flavor";

          type leafref {

            path "deref(../../cisco-sae-core-fp-cfs:nsd-deployment-flavor)/../" +

                        "nfvo-rel2:vnf-profile/nfvo-rel2:id";

          }

        }

        leaf vnfd-deployment {

          tailf:info "vnfd-deployment for the vnf-profile";

          description "vnfd-deployment for the vnf-profile";

          type leafref {

            path "../../../vnfd-deployment/name";

          }

        }

      }

    }

    list vnfd-deployment {

      tailf:info "List of VNFD Deployments";

      description "List of VNFD Deployments";

      tailf:validate "vnfd-validation" {

        tailf:dependency ".";

      }

      key name;

      leaf name {

        tailf:info "Unique VNFD Deployment name";

        description "Unique VNFD Deployment name";

        type string;

      }

      leaf vnfd {

        tailf:info "VNF descriptor name";

        description "VNF descriptor name";

        type leafref {

          path "/nfvo-rel2:nfvo/nfvo-rel2:vnfd/nfvo-rel2:id";

        }

        mandatory true;

      }

      uses vnf-data;

      uses variables;

      uses deployment-params;

      leaf vnf-type {

        type cisco-extension-framework:extension-group-t;

        tailf:info "Represents the type of the vnf, defaults to generic.";

        default cisco-extension-framework:generic;

      }

      leaf controller {

        tailf:info "For a SDN based VNF, captures the controller device name.";

        type leafref {

          path "/ncs:devices/ncs:device/ncs:name";

        }

      }

      container ftdv-settings {

        when "../vnf-type='cisco-sae-ftdv-extension:ftdv'";

        leaf-list license\_caps {

          tailf:info "licenses to be applied on the FTDv. ex : BASE, MALWARE, THREAT, URLFilter";

          type string {

            tailf:info "ex : BASE, MALWARE, THREAT, URLFilter";

          }

        }

        leaf accessPolicy {

          tailf:info "Policy to be applied";

          type string;

        }

        must "accessPolicy" {

          error-message "access policy for FTDv devices is mandatory";

        }

      }

      container asav-ha-settings {

        when "../vnf-type='cisco-sae-asav-extension:asav'";

        leaf ha-role {

          tailf:info "ASAv HA ROLE active/standby";

          type enumeration {

            enum active;

            enum standby;

          }

        }

        leaf-list failover-connection-point {

          tailf:info "Failover CP name which connect ASAv HA pair";

          type leafref {

            path "deref(../../vnfd)/../"

                  +"nfvo-rel2:vdu/nfvo-rel2:internal-connection-point-descriptor/"

                  +"nfvo-rel2:external-connection-point-descriptor";

          }

          max-elements 2;

        }

        must "ha-role and failover-connection-point" {

          error-message "HA role & connection-point name is mandatory for ASAv devices";

        }

      }

    }

    leaf timestamp {

      tailf:hidden sae;

      type string;

      description "This timestamp should be updated internally by redeploy action";

    }

    tailf:action custom-redeploy {

      tailf:info "Action to trigger catalog copy";

      tailf:actionpoint "cfs-redep-catalog-action-point";

      output {

        leaf status {

          type string;

        }

        leaf error {

          type string;

        }

      }

    }

  }

  list sae-site {

    tailf:info "List of SAE sites";

    description

        "North bound yang model for Secure Agile Exchange (SAE) Core Function

                Pack";

    key site;

    leaf site {

      tailf:info "Unique name to represent a SAE site";

      description "Unique name to represent a SAE site";

      type service-name;

    }

    leaf sae-provider {

      tailf:info "SAE service provider";

      description "SAE service provider";

      type leafref {

        path "/cisco-sae-core-fp-cfs:sae-provider/cisco-sae-core-fp-cfs:name";

      }

      mandatory true;

    }

    leaf sae-tenant {

      tailf:info "SAE service provider's Tenant";

      description "SAE service provider's Tenant";

      type leafref {

        path "/cisco-sae-core-fp-cfs:sae-provider" +

                "[cisco-sae-core-fp-cfs:name = current()/../sae-provider]"

        + "/cisco-sae-core-fp-cfs:sae-tenant/cisco-sae-core-fp-cfs:name";

      }

    }

    container vnf-mgmt-resources {

      presence "sae";

      tailf:info "Specify VNF's management interface resources";

      tailf:validate "vnf-resource-validation" {

        tailf:dependency ".";

      }

      leaf vnf-mgmt-netmask {

        tailf:info "Netmast to be used in VNF's Mgmt interface";

        type inet:ip-address;

        mandatory true;

      }

      leaf vnf-mgmt-vlan {

        tailf:info "VLAN to be configured for Mgmt vnic in CSP";

        type uint16 {

          range "1..4096";

        }

        mandatory true;

      }

      leaf vnf-mgmt-gateway {

        tailf:info "VNF's Mgmt interface gateway ip-address";

        type inet:ip-address;

        mandatory true;

      }

    }

    must "vnf-mgmt-resources" {

      error-message "Please specify VNF's management interface resources to be used";

    }

    uses variables;

    uses location-details;

    uses external-end-points;

    uses resource-pools;

    container infrastructure {

      tailf:info "Represents SAE Infrastructure";

      description "Represents SAE Infrastructure";

      container switching {

        leaf type {

          tailf:info "Type of Leaf/Spine Switches";

          description "Type of Leaf/Spine Switches";

          type cisco-sae-core-fp-common:switching-type-t;

          default cisco-sae-core-fp-common:n9k-switch-pair;

        }

        leaf vnid-pool {

          when "../type = 'cisco-sae-core-fp-common:n9k-spine-leaf'";

          tailf:info "Resource Pool for VNID Allocation";

          description "Resource Pool for VNID Allocation";

          type leafref {

            path "/ralloc:resource-pools/idalloc:id-pool/idalloc:name";

          }

          mandatory true;

        }

        leaf bgp-asn {

          tailf:info "Global bgp asn value. Default: 100";

          description "Global bgp asn value. Default: 100";

          type string;

          default "100";

        }

      }

      //Note:: leaf-switches should be unique, not allowed to share between clusters

      tailf:unique-selector "compute-clusters/leaf-switches" {

        tailf:unique-leaf "name";

      }

      //Note:: CSP servers should be unique, not allowed to share between clusters

      tailf:unique-selector "compute-clusters/servers" {

        tailf:unique-leaf "name";

      }

      list compute-clusters {

        tailf:info "List of Compute Clusters";

        description "List of Compute Clusters";

        key name;

        leaf name {

          tailf:info "Compute cluster name";

          description "Compute cluster name";

          type string;

        }

        leaf vlan-pool {

          tailf:info "Resource Pool for VLAN Allocation";

          description "Resource Pool for VLAN Allocation";

          type leafref {

            path "/ralloc:resource-pools/idalloc:id-pool/idalloc:name";

          }

          must "/ralloc:resource-pools/idalloc:id-pool[idalloc:name=current()]"

          + "/idalloc:range/idalloc:start>=2"

          + " and /ralloc:resource-pools/idalloc:id-pool[idalloc:name=current()]"

          + "/idalloc:range/idalloc:end<=3967" {

            error-message "VLAN ID pool range should be between 2-3967";

          }

          mandatory true;

        }

        list leaf-switches {

          tailf:info "List of leaf Switches";

          max-elements 2;

          key name;

          ordered-by user;

          leaf name {

            tailf:info "Name of switch device in NSO device tree";

            description "Name of switch device in NSO device tree";

            tailf:cli-completion-actionpoint "device-cli-completion" {

              tailf:cli-completion-id "n9k-switch-device";

            }

            type string;

            must "(/ncs:devices/ncs:device[ncs:name=current()])" {

                error-message "Switch device not oboarded on NSO device tree";

            }

          }

          uses custom-template-grouping;

          leaf timestamp {

            tailf:hidden sae;

            type string;

            description "This timestamp should be updated internally by redeploy action";

          }

          tailf:action custom-redeploy {

            tailf:info "Action to redeploy custom-template on leaf switches";

            tailf:actionpoint "cfs-redep-switch-action-point";

            output {

              leaf status {

                type string;

              }

              leaf error {

                type string;

              }

            }

          }

        }

        list servers {

          uses ncs:service-data;

          ncs:servicepoint cfs-sae-compute-servicepoint;

          tailf:info "List of CSPs in a given Compute Cluster";

          description "List of CSPs in a given Compute Cluster";

          key name;

          leaf name {

            tailf:info "Name of CSP device";

            description "Name of CSP device";

            tailf:cli-completion-actionpoint "device-cli-completion" {

              tailf:cli-completion-id "csp-device";

            }

            type string;

            must "(/ncs:devices/ncs:device[ncs:name=current()])" {

              error-message "CSP device not oboarded on NSO device tree";

            }

          }

          leaf type {

            tailf:info "Type of CSP device as described in sae-catalog->csp";

            description "Type of CSP device as described in sae-catalog->csp";

            type leafref {

              path "/cisco-sae-core-fp-cfs:sae-catalog/cisco-sae-core-fp-cfs:csp/"

              + "cisco-sae-core-fp-cfs:name";

            }

            //Allow selecting a CSP type from list of sae-catalog owned by chosen provider/tenant

            must "/cisco-sae-core-fp-cfs:sae-catalog[cisco-sae-core-fp-cfs:name="

            + "deref(current()/../../../../sae-tenant)/../"

            + "cisco-sae-core-fp-cfs:sae-tenant-catalog]/cisco-sae-core-fp-cfs:csp"

            + "[cisco-sae-core-fp-cfs:name=current()]"

            + " or (not(../../../../sae-tenant) and /cisco-sae-core-fp-cfs:sae-catalog"

            + "[cisco-sae-core-fp-cfs:name=deref(current()/../../../../sae-provider)/../"

            + "cisco-sae-core-fp-cfs:sae-provider-catalog]"

            + "/cisco-sae-core-fp-cfs:csp[cisco-sae-core-fp-cfs:name=current()])" {

              error-message "Incorrect CSP type for the chosen sae-provider/sae-tenant";

              tailf:dependency "../../../../sae-provider";

              tailf:dependency "../../../../sae-tenant";

              tailf:dependency "/cisco-sae-core-fp-cfs:sae-catalog/cisco-sae-core-fp-cfs:csp";

            }

            mandatory true;

          }

          list virtio-network {

            tailf:info "Switch connection information for non-sriov network";

            description "Switch connection information for non-sriov network";

            key pc-name;

            leaf pc-name {

              tailf:info "Non-SRIOV port channel name";

              description "Non-SRIOV port channel name";

              type leafref {

                path "/cisco-sae-core-fp-cfs:sae-catalog/cisco-sae-core-fp-cfs:csp/"

                + "cisco-sae-core-fp-cfs:virtio-network/cisco-sae-core-fp-cfs:name";

              }

              //Allow selecting virtio-network based on CSP type selected

              must "/cisco-sae-core-fp-cfs:sae-catalog[cisco-sae-core-fp-cfs:name="

              + "deref(current()/../../../../../sae-tenant)/../"

              + "cisco-sae-core-fp-cfs:sae-tenant-catalog]/cisco-sae-core-fp-cfs:csp"

              + "[cisco-sae-core-fp-cfs:name=deref(current()/../../type)]/"

              + "cisco-sae-core-fp-cfs:virtio-network[cisco-sae-core-fp-cfs:name=current()]"

              + " or (not(../../../../../sae-tenant) and /cisco-sae-core-fp-cfs:sae-catalog"

              + "[cisco-sae-core-fp-cfs:name="

              + "deref(current()/../../../../../sae-provider)/../"

              + "cisco-sae-core-fp-cfs:sae-provider-catalog]"

              + "/cisco-sae-core-fp-cfs:csp[cisco-sae-core-fp-cfs:name="

              + "deref(current()/../../type)]/cisco-sae-core-fp-cfs:virtio-network"

              + "[cisco-sae-core-fp-cfs:name=current()])" {

                error-message "Incorrect virtio-network name for selected CSP type";

                tailf:dependency "../../../../../sae-provider";

                tailf:dependency "../../../../../sae-tenant";

                tailf:dependency "../../type";

              }

            }

            list pnics {

              max-elements 2;

              tailf:info "List of PNICs forming non-sriov port-channel";

              description "List of PNICs forming non-sriov port-channel";

              key name;

              leaf name {

                tailf:info "PNIC name";

                description "PNIC name";

                type string;

              }

              leaf connected-switch {

                tailf:info "Switch name connected to this pnic";

                description "Switch name connected to this pnic";

                type leafref {

                  path "../../../../leaf-switches/name";

                }

              }

              leaf switch-interface-name {

                tailf:validate "interface-type-validation" {

                  tailf:dependency ".";

                }

                tailf:info "Interface Type and port";

                type string;

              }

              leaf bandwidth {

                tailf:info "PNIC bandwidth";

                description "PNIC bandwidth";

                type int32;

                default 10000;

              }

            }

          }

          must "((count(virtio-network) = count(/cisco-sae-core-fp-cfs:sae-catalog"

          +"[cisco-sae-core-fp-cfs:name= deref(current()/../../../sae-tenant)/../"

          + "cisco-sae-core-fp-cfs:sae-tenant-catalog]/cisco-sae-core-fp-cfs:csp"

          + "[cisco-sae-core-fp-cfs:name=deref(current()/type)]/"

          + "cisco-sae-core-fp-cfs:virtio-network))"

          + " or (not(../../../sae-tenant) and (count(virtio-network) = count(/" +

                    "cisco-sae-core-fp-cfs:sae-catalog[cisco-sae-core-fp-cfs:name="

          + "deref(current()/../../../sae-provider)/../"

          + "cisco-sae-core-fp-cfs:sae-provider-catalog]"

          + "/cisco-sae-core-fp-cfs:csp[cisco-sae-core-fp-cfs:name="

          + "deref(current()/type)]/cisco-sae-core-fp-cfs:virtio-network))))" {

            error-message "no. of virtio-networks definition should match to no. of virtio-networks"

                +" defined under catalog CSP type";

            tailf:dependency "../../../sae-provider";

            tailf:dependency "../../../sae-tenant";

            tailf:dependency "virtio-network";

            tailf:dependency "/cisco-sae-core-fp-cfs:sae-catalog/cisco-sae-core-fp-cfs:csp";

          }

          list sriov-networks {

            tailf:info "Switch connection information for sriov network";

            description "Switch connection information for sriov network";

            key name;

            ordered-by user;

            leaf name {

              tailf:info "SRIOV network name";

              description "SRIOV network name";

              type leafref {

                path "/cisco-sae-core-fp-cfs:sae-catalog/cisco-sae-core-fp-cfs:csp/"

                + "cisco-sae-core-fp-cfs:sriov-networks/cisco-sae-core-fp-cfs:name";

              }

              //Allow selecting sriov-networks based on CSP type selected

              must "/cisco-sae-core-fp-cfs:sae-catalog[cisco-sae-core-fp-cfs:name="

              + "deref(current()/../../../../../sae-tenant)/../"

              + "cisco-sae-core-fp-cfs:sae-tenant-catalog]/cisco-sae-core-fp-cfs:csp"

              + "[cisco-sae-core-fp-cfs:name=deref(current()/../../type)]/"

              + "cisco-sae-core-fp-cfs:sriov-networks"

              + "[cisco-sae-core-fp-cfs:name=current()]"

              + " or (not(../../../../../sae-tenant) and /cisco-sae-core-fp-cfs:sae-catalog"

              + "[cisco-sae-core-fp-cfs:name="

              + "deref(current()/../../../../../sae-provider)/../"

              + "cisco-sae-core-fp-cfs:sae-provider-catalog]"

              + "/cisco-sae-core-fp-cfs:csp[cisco-sae-core-fp-cfs:name="

              + "deref(current()/../../type)]/cisco-sae-core-fp-cfs:sriov-networks"

              + "[cisco-sae-core-fp-cfs:name=current()])" {

                error-message "Incorrect sriov-network name for selected CSP type";

                tailf:dependency "../../../../../sae-provider";

                tailf:dependency "../../../../../sae-tenant";

                tailf:dependency "../../type";

              }

            }

            leaf connected-switch {

              tailf:info "Switch name connected to this pnic";

              description "Switch name connected to this pnic";

              type leafref {

                path "../../../leaf-switches/name";

              }

            }

            leaf switch-interface-name {

              tailf:validate "interface-type-validation" {

                tailf:dependency ".";

              }

              tailf:info "Interface Type and port";

              type string;

            }

            leaf sriov-type {

              tailf:info "CSP sriov pnic type";

              description "CSP sriov pnic type";

              type cisco-sae-core-fp-common:csp-sriov-type-t;

              default cisco-sae-core-fp-common:X520;

            }

            leaf bandwidth {

              tailf:info "PNIC bandwidth";

              description "PNIC bandwidth";

              type int32;

              default 10000;

            }

          }

          must "((count(sriov-networks) = count(/cisco-sae-core-fp-cfs:sae-catalog"

          +"[cisco-sae-core-fp-cfs:name=deref(current()/../../../sae-tenant)/../"

          + "cisco-sae-core-fp-cfs:sae-tenant-catalog]/cisco-sae-core-fp-cfs:csp"

          + "[cisco-sae-core-fp-cfs:name=deref(current()/type)]/"

          + "cisco-sae-core-fp-cfs:sriov-networks))"

          + " or (not(../../../sae-tenant) and (count(sriov-networks) = count(/" +

                    "cisco-sae-core-fp-cfs:sae-catalog[cisco-sae-core-fp-cfs:name="

          + "deref(current()/../../../sae-provider)/../"

          + "cisco-sae-core-fp-cfs:sae-provider-catalog]"

          + "/cisco-sae-core-fp-cfs:csp[cisco-sae-core-fp-cfs:name="

          + "deref(current()/type)]/cisco-sae-core-fp-cfs:sriov-networks))))" {

            error-message "no. of sriov-networks definition should match to no. of sriov-networks"

              +" defined under catalog CSP type";

            tailf:dependency "../../../sae-provider";

            tailf:dependency "../../../sae-tenant";

            tailf:dependency "sriov-networks";

            tailf:dependency "/cisco-sae-core-fp-cfs:sae-catalog/cisco-sae-core-fp-cfs:csp";

          }

          uses variables;

          uses custom-template-grouping;

          leaf timestamp {

            tailf:hidden sae;

            type string;

            description "This timestamp should be updated internally by redeploy action";

          }

          tailf:action custom-redeploy {

            tailf:info "Action to redeploy day1-config on CSP device";

            tailf:actionpoint "cfs-redep-csp-day1-action-point";

            output {

              leaf status {

                type string;

              }

              leaf error {

                type string;

              }

            }

          }

        }

      }

      list vnf-manager {

        tailf:info "List of VNFMs. Eg: ESC";

        description "List of VNFMs. Eg: ESC";

        uses ncs:service-data;

        ncs:servicepoint cfs-sae-vnfm-servicepoint;

        max-elements 1;

        key name;

        leaf name {

          tailf:info "Name of the VNFM service";

          description "Name of the VNFM service";

          type string;

        }

        leaf device-on-boarding {

          tailf:info "Defines if the VNFM is on-boarded by user or by SAE CFP";

          description "Defines if the VNFM is on-boarded by user or by SAE CFP";

          type enumeration {

            enum nso-managed {

              tailf:info "VNFM will be onboarded by SAE Core Function Pack";

              description "VNFM will be onboarded by SAE Core Function Pack";

            }

            enum nso-unmanaged {

              tailf:info "VNFM is onboarded by user on to NSO device tree";

              description "VNFM is onboarded by user on to NSO device tree";

            }

          }

          default nso-unmanaged;

        }

        leaf vnfm-device-name {

          tailf:info "User specified VNFM device name";

          description "User specified VNFM device name";

          when "../device-on-boarding = 'nso-unmanaged'";

          tailf:cli-completion-actionpoint "device-cli-completion" {

            tailf:cli-completion-id "esc-device";

          }

          type string;

          mandatory true;

          must "( ../device-on-boarding='nso-unmanaged' and current() and "

          + "(/ncs:devices/ncs:device[ncs:name=current()]) )"

          + " or (../device-on-boarding='nso-managed')" {

            error-message "For nso-unmanaged devices, vnfm-device-name must match existing device"

              +" in /ncs:devices/ncs:device tree";

          }

        }

        leaf deployment-name {

          tailf:info "VNFD Deployment from catalog for the VNFM";

          description "VNFD Deployment from catalog for the VNFM";

          when "../device-on-boarding = 'nso-managed'";

          type leafref {

            path "/cisco-sae-core-fp-cfs:sae-catalog/cisco-sae-core-fp-cfs:vnfd-deployment/"

            + "cisco-sae-core-fp-cfs:name";

          }

          // Allow selecting a vnfd-deployment from list of sae-catalog

          // owned by chosen provider/tenant

          must "/cisco-sae-core-fp-cfs:sae-catalog[cisco-sae-core-fp-cfs:name="

          + "deref(current()/../../../sae-tenant)/../"

          + "cisco-sae-core-fp-cfs:sae-tenant-catalog]/cisco-sae-core-fp-cfs:vnfd-deployment"

          + "[cisco-sae-core-fp-cfs:name=current()]"

          + " or (not(../../../sae-tenant) and /cisco-sae-core-fp-cfs:sae-catalog"

          + "[cisco-sae-core-fp-cfs:name=deref(current()/../../../sae-provider)/../"

          + "cisco-sae-core-fp-cfs:sae-provider-catalog]"

          + "/cisco-sae-core-fp-cfs:vnfd-deployment[cisco-sae-core-fp-cfs:name=current()])" {

            error-message "Incorrect vnfd-deployment for the chosen sae-provider/sae-tenant";

            tailf:dependency "../../../sae-provider";

            tailf:dependency "../../../sae-tenant";

            tailf:dependency "/cisco-sae-core-fp-cfs:sae-catalog/"

            + "cisco-sae-core-fp-cfs:vnfd-deployment";

          }

          mandatory true;

        }

        leaf-list resource-zone {

          description "List of resource zones from which to select computes for this vnf-manager";

          tailf:info "List of resource zones from which to select computes for this vnf-manager";

          when "../device-on-boarding='nso-managed'";

          type leafref {

            path "/nfvo-rel2:nfvo/nfvo-rel2-ro:resource-orchestration/"

            + "nfvo-rel2-ro:vim-type/nfvo-rel2-ro:vim/csp-vim:csp-resource-zone/csp-vim:id";

          }

        }

        leaf vnfm-mgmt-address {

          when "../device-on-boarding='nso-managed'";

          tailf:info "vnf-manager Mgmt Address";

          description "vnf-manager Mgmt Address";

          type inet:ip-address;

          mandatory true;

        }

        leaf vnfm-mgmt-netmask {

          when "../device-on-boarding='nso-managed'";

          tailf:info "vnf-manager Mgmt Netmask";

          description "vnf-manager Mgmt Netmask";

          type inet:ip-address;

          mandatory true;

        }

        leaf vnfm-mgmt-gateway {

          when "../device-on-boarding='nso-managed'";

          tailf:info "vnf-manager Mgmt gateway";

          description "vnf-manager Mgmt gateway";

          type inet:ip-address;

          mandatory true;

        }

        leaf vnfm-mgmt-vlan {

          when "../device-on-boarding='nso-managed'";

          tailf:info "vnf-manager Mgmt VLAN ID";

          description "vnf-manager Mgmt VLAN ID";

          type uint16 {

            range "1..4096";

          }

          mandatory true;

        }

        uses variables;

        leaf timestamp {

          tailf:hidden sae;

          type string;

          description "This timestamp should be updated internally by redeploy action";

        }

        tailf:action custom-redeploy {

          tailf:info "Action to redeploy VNF Manager service";

          tailf:actionpoint "cfs-redep-vnf-manager-action-point";

          output {

            leaf status {

              type string;

            }

            leaf error {

              type string;

            }

          }

        }

      }

      must "((switching[type!='cisco-sae-core-fp-common:n9k-switch-pair']) or"

          +" ((switching[type='cisco-sae-core-fp-common:n9k-switch-pair'])"

          +" and (count(compute-clusters) < 2)))" {

        error-message "For stand-alone topology, list of compute cluster should be one";

      }

    }

    list endpoint-gateway-vnf {

      uses ncs:service-data;

      ncs:servicepoint cfs-sae-endpoint-gateway-vnf-servicepoint;

      tailf:info "Endpoint Gateway VNF for Provider or Consumer End";

      ordered-by user;

      key name;

      leaf name {

        tailf:info "Endpoint Gateway VNF service name";

        type string;

      }

      leaf bandwidth-requirement-mbps {

        tailf:info "bandwidth requirement for endpoint-gateway-vnf";

        type int32;

        default 512;

      }

      leaf deployment-name {

        tailf:info

                "Network Service Descriptor (NSD)'s deployment from sae-catalog -> nsd-deployment";

        type leafref {

          path "/cisco-sae-core-fp-cfs:sae-catalog/cisco-sae-core-fp-cfs:nsd-deployment/"

          + "cisco-sae-core-fp-cfs:name";

        }

        //Allow selecting a nsd-deployment from list of sae-catalog owned by chosen provider/tenant

        must "/cisco-sae-core-fp-cfs:sae-catalog[cisco-sae-core-fp-cfs:name="

        + "deref(current()/../../sae-tenant)/../"

        + "cisco-sae-core-fp-cfs:sae-tenant-catalog]/cisco-sae-core-fp-cfs:nsd-deployment"

        + "[cisco-sae-core-fp-cfs:name=current()]"

        + " or (not(../../sae-tenant) and /cisco-sae-core-fp-cfs:sae-catalog"

        + "[cisco-sae-core-fp-cfs:name=deref(current()/../../sae-provider)/../"

        + "cisco-sae-core-fp-cfs:sae-provider-catalog]"

        + "/cisco-sae-core-fp-cfs:nsd-deployment[cisco-sae-core-fp-cfs:name=current()])" {

          error-message "nsd-deployment (deployment-name) is invalid";

          tailf:dependency "../../sae-provider";

          tailf:dependency "../../sae-tenant";

          tailf:dependency "/cisco-sae-core-fp-cfs:sae-catalog/"

          + "cisco-sae-core-fp-cfs:nsd-deployment";

        }

        mandatory true;

      }

      leaf gateway-vnf-device-name {

        tailf:info

                "User specified endpoint-gateway-vnf device"

              +" name when onboarded onto NSO device tree";

        type string;

      }

      tailf:unique-selector "external-end-point/connectivity/external-access-point" {

        tailf:unique-leaf "gateway-access-point";

      }

      list external-end-point {

        when "string-length(../deployment-name) > 0";

        tailf:info "External endpoints connecting to endpoint-gateway-vnf in user specified order";

        key name;

        min-elements 1;

        leaf name {

          tailf:info "Consumer/Provider endpoint";

          type leafref {

            path "../../../external-end-point/name";

          }

        }

        container connectivity {

          presence true;

          tailf:info "Gateway VNF's interface association with the external end point";

          list external-access-point {

            max-elements 1;

            tailf:info "Defines connectivity towards external-endpoint";

            key gateway-access-point;

            leaf gateway-access-point {

              tailf:info "Service access point of the network service deployment (NSD)";

              mandatory true;

              tailf:cli-incomplete-command;

              type string;

              tailf:cli-completion-actionpoint "gateway-sapd-cli-completion" {

                tailf:cli-completion-id "get-external-sapds";

              }

            }

            list ip {

              key address;

              leaf address {

                type inet:ip-address;

              }

              leaf virtual-ip {

                type empty;

                tailf:info "address for virtual interface";

              }

              leaf vnf-profile {

                tailf:info "vnf-profile for which this address applies."

                      +" Optional when NSD has only one profile";

                type string;

                tailf:cli-completion-actionpoint "vnf-profile-completion" {

                  tailf:cli-completion-id "get-vnf-profile";

                }

              }

            }

          }

          list next-hop-access-point {

            tailf:info "Defines connectivity towards next-hop in the service chain";

            key gateway-access-point;

            leaf gateway-access-point {

              tailf:info "Service access point of the network service deployment (NSD)";

              type string;

              tailf:cli-completion-actionpoint "gateway-sapd-cli-completion" {

                tailf:cli-completion-id "get-next-hop-sapds";

              }

            }

            leaf subnet-size {

              tailf:info "Size of the subnet to be reserved for the access-point";

              type uint8 {

                range "1..32";

              }

              default 29;

            }

          }

        }

      }

      uses variables;

      must " (../resource-pools) " {

        error-message "resource pools should be present before defining endpoint-gateway,"

          +" it is not allowed to modify/delete them once endpoint-gateway present";

      }

      leaf-list resource-zone {

        type leafref {

          path "/nfvo-rel2:nfvo/nfvo-rel2-ro:resource-orchestration/"

          + "nfvo-rel2-ro:vim-type/nfvo-rel2-ro:vim/csp-vim:csp-resource-zone/csp-vim:id";

        }

        tailf:info "List of resource zones from which to select computes for this

                                VNF";

      }

      //uses custom-template-grouping;

      container customization {

        tailf:info "Customization of deployed VNFs in this service";

        list vnf-deployment {

          tailf:info "vnf-deployment profile to customize corresponding VNF";

          key profile;

          leaf profile {

            type leafref {

              path "deref(../../../cisco-sae-core-fp-cfs:deployment-name)/"

                +"../cisco-sae-core-fp-cfs:vnf-deployment/cisco-sae-core-fp-cfs:vnf-profile";

            }

          }

          uses custom-template-grouping;

        }

      }

      leaf timestamp {

        tailf:hidden sae;

        type string;

        description "This timestamp should be updated internally by redeploy action";

      }

      tailf:action custom-redeploy {

        tailf:info "Action to redeploy endpoint gateway vnf service";

        tailf:actionpoint "cfs-redep-ep-gw-vnf-action-point";

        output {

          leaf status {

            type string;

          }

          leaf error {

            type string;

          }

        }

      }

      tailf:action custom-undeploy-redeploy {

        tailf:info "Will undeploy & redeploy service vnfs ";

        description "Will undeploy & redeploy service vnfs";

        tailf:actionpoint "undeploy-redeploy";

        output {

          leaf status {

            type string;

            mandatory true;

          }

        }

      }

    }

    list service-chain {

      uses ncs:service-data;

      ncs:servicepoint cfs-sae-servicepoint;

      tailf:info "Instance of a Service Chain for a given Consumer and/or Provider";

      description

            "Instance of a Service Chain for a given Consumer and/or Provider";

      key name;

      leaf name {

        tailf:info "Unique name for service-chain";

        description "Unique name for service-chain";

        type string;

      }

      leaf bandwidth-requirement-mbps {

        tailf:info "bandwidth requirement for service-chain";

        description "bandwidth requirement for service-chain";

        type int32;

        default 512;

      }

      leaf deployment-name {

        tailf:info

                "Network Service Descriptor (NSD)'s deployment from sae-catalog -> nsd-deployment";

        description

                "Network Service Descriptor (NSD)'s deployment from sae-catalog -> nsd-deployment";

        type leafref {

          path "/cisco-sae-core-fp-cfs:sae-catalog/cisco-sae-core-fp-cfs:nsd-deployment/"

          + "cisco-sae-core-fp-cfs:name";

        }

        //Allow selecting a nsd-deployment from list of sae-catalog owned by chosen provider/tenant

        must "/cisco-sae-core-fp-cfs:sae-catalog[cisco-sae-core-fp-cfs:name="

        + "deref(current()/../../sae-tenant)/../"

        + "cisco-sae-core-fp-cfs:sae-tenant-catalog]/cisco-sae-core-fp-cfs:nsd-deployment"

        + "[cisco-sae-core-fp-cfs:name=current()]"

        + " or (not(../../sae-tenant) and /cisco-sae-core-fp-cfs:sae-catalog"

        + "[cisco-sae-core-fp-cfs:name=deref(current()/../../sae-provider)/../"

        + "cisco-sae-core-fp-cfs:sae-provider-catalog]"

        + "/cisco-sae-core-fp-cfs:nsd-deployment[cisco-sae-core-fp-cfs:name=current()])" {

          error-message "nsd-deployment (deployment-name) is invalid";

          tailf:dependency "../../sae-provider";

          tailf:dependency "../../sae-tenant";

          tailf:dependency "/cisco-sae-core-fp-cfs:sae-catalog/"

          + "cisco-sae-core-fp-cfs:nsd-deployment";

        }

        mandatory true;

      }

      leaf type {

        tailf:info "Type of SAE service-chain";

        description "Type of SAE service-chain";

        type enumeration {

          enum "consumer" {

            tailf:info "This is a half-chain representing the chain of virtualized network"

                +" funtions between the Consumer <-->SAE";

            description "This is a half-chain representing the chain of virtualized network"

                +" funtions between the Consumer <-->SAE";

          }

          enum "provider" {

            tailf:info "This is a half-chain representing the chain of virtualized network"

              +" funtions between SAE <--> Provider";

            description "This is a half-chain representing the chain of virtualized network"

              +" funtions between SAE <--> Provider";

          }

          enum "end-to-end" {

            tailf:info "This is a Full chain representing the chain of virtualized network"

              +" functions between Consumer <---SAE-->Provider";

            description "This is a Full chain representing the chain of virtualized network"

              +" functions between Consumer <---SAE-->Provider";

          }

        }

        mandatory true;

      }

      list consumer-end-point {

        tailf:info "Consumer for this service chain";

        when "(../type='consumer' or ../type='end-to-end') "

             + "and (string-length(../deployment-name) > 0)";

        key external-end-point;

        min-elements 1;

        leaf external-end-point {

          tailf:info "External endpoint for this service chain";

          description "External endpoint for this service chain";

          type leafref {

            path "../../../external-end-point/name";

          }

        }

        choice gateway-choice {

          case external-gw-case {

            container shared-gateway {

              presence true;

              leaf endpoint-gateway-vnf {

                tailf:info "External endpoint gateway for this service chain";

                description "External endpoint gateway for this service chain";

                type leafref {

                  path "../../../../endpoint-gateway-vnf/name";

                }

                must "current()/../../../../endpoint-gateway-vnf/cisco-sae-core-fp-cfs:"

                +"external-end-point[cisco-sae-core-fp-cfs:"

                +"name=current()/../../external-end-point]/../name=current()" {

                error-message "Invalid endpoint-gateway-vnf value. Selected endpoint-gateway-vnf"

                  +" is not connected to the external-end-point.";

                tailf:dependency "../../external-end-point";

                }

                mandatory true;

              }

              leaf gateway-access-point {

                when "string-length(../endpoint-gateway-vnf) > 0";

                tailf:info "Service access point of the network service deployment (NSD)";

                type string;

                tailf:cli-completion-actionpoint "chain-sapd-gw-cli-completion" {

                  tailf:cli-completion-id "get-endpoint-next-hop-sapd";

                }

              }

              leaf service-access-point {

                when "string-length(../endpoint-gateway-vnf) > 0";

                tailf:info "Service access point of the network service deployment (NSD)";

                type string;

                tailf:cli-completion-actionpoint "chain-sapd-gw-cli-completion" {

                  tailf:cli-completion-id "get-consumer-endpoint-sapd";

                }

              }

            }

          }

          case inline-gw-case {

            list connectivity {

              tailf:info "Gateway VNF's interface association with the external end point";

              key service-access-point;

              max-elements 1;

              leaf service-access-point {

                tailf:info "Service access point of the network service deployment (NSD)";

                type string;

                tailf:cli-completion-actionpoint "chain-sapd-cli-completion" {

                  tailf:cli-completion-id "get-consumer-endpoint-sapd";

                }

              }

              list ip {

                key address;

                leaf address {

                  type inet:ip-address;

                }

                leaf virtual-ip {

                  type empty;

                  tailf:info "address for virtual interface";

                }

                leaf vnf-profile {

                  tailf:info "vnf-profile for which this address applies."

                        +" Optional when NSD has only one profile";

                  type string;

                  tailf:cli-completion-actionpoint "vnf-profile-completion" {

                    tailf:cli-completion-id "get-sc-vnf-profile";

                  }

                }

              }

            }

          }

        }

        must "(shared-gateway/endpoint-gateway-vnf or connectivity)" {

          error-message "consumer-end-point must define either shared-gateway or in-line"

            +" connectivity";

          tailf:dependency "shared-gateway/endpoint-gateway-vnf";

          tailf:dependency "external-end-point";

        }

      }

      list provider-end-point {

        when "(../type='provider' or ../type='end-to-end') "

             + "and (string-length(../deployment-name) > 0)";

        tailf:info "Provider for this service chain";

        description "Provider for this service chain";

        key external-end-point;

        min-elements 1;

        leaf external-end-point {

          tailf:info "External endpoint for this service chain";

          description "External endpoint for this service chain";

          type leafref {

            path "../../../external-end-point/name";

          }

        }

        choice gateway-choice {

          case external-gw-case {

            container shared-gateway {

              presence true;

              leaf endpoint-gateway-vnf {

                tailf:info "External endpoint gateway for this service chain";

                description "External endpoint gateway for this service chain";

                type leafref {

                  path "../../../../endpoint-gateway-vnf/name";

                }

                must "current()/../../../../endpoint-gateway-vnf/cisco-sae-core-fp-cfs:"

                +"external-end-point[cisco-sae-core-fp-cfs:"

                +"name=current()/../../external-end-point]/../name=current()" {

                error-message "Invalid endpoint-gateway-vnf value. Selected endpoint-gateway-vnf"

                  +" is not connected to the external-end-point.";

                tailf:dependency "../../external-end-point";

                }

                mandatory true;

              }

              leaf gateway-access-point {

                when "string-length(../endpoint-gateway-vnf) > 0";

                tailf:info "Service access point of the network service deployment (NSD)";

                type string;

                tailf:cli-completion-actionpoint "chain-sapd-gw-cli-completion" {

                  tailf:cli-completion-id "get-endpoint-next-hop-sapd";

                }

              }

              leaf service-access-point {

                when "string-length(../endpoint-gateway-vnf) > 0";

                tailf:info "Service access point of the network service deployment (NSD)";

                type string;

                tailf:cli-completion-actionpoint "chain-sapd-gw-cli-completion" {

                  tailf:cli-completion-id "get-provider-endpoint-sapd";

                }

              }

            }

          }

          case inline-gw-case {

            list connectivity {

              tailf:info "Gateway VNF's interface association with the external end point";

              key service-access-point;

              max-elements 1;

              leaf service-access-point {

                tailf:info "Service access point of the network service deployment (NSD)";

                type string;

                tailf:cli-completion-actionpoint "chain-sapd-cli-completion" {

                  // Displays only provider endpoints of the service chain.

                  tailf:cli-completion-id "get-provider-endpoint-sapd";

                }

              }

              list ip {

                key address;

                leaf address {

                  type inet:ip-address;

                }

                leaf virtual-ip {

                  type empty;

                  tailf:info "address for virtual interface";

                }

                leaf vnf-profile {

                  tailf:info "vnf-profile for which this address applies."

                    +" Optional when NSD has only one profile";

                  type string;

                  tailf:cli-completion-actionpoint "vnf-profile-completion" {

                    tailf:cli-completion-id "get-sc-vnf-profile";

                  }

                }

              }

            }

          }

        }

        must "(shared-gateway/endpoint-gateway-vnf or connectivity)" {

          error-message "provider-end-point  must define either shared-gateway or in-line"

            +" connectivity.";

          tailf:dependency "shared-gateway/endpoint-gateway-vnf";

          tailf:dependency "external-end-point";

        }

      }

      uses variables;

      leaf-list resource-zone {

        type leafref {

          path "/nfvo-rel2:nfvo/nfvo-rel2-ro:resource-orchestration/"

          + "nfvo-rel2-ro:vim-type/nfvo-rel2-ro:vim/csp-vim:csp-resource-zone/csp-vim:id";

        }

        description "List of resource zones from which to select computes for this service chain";

        tailf:info "List of resource zones from which to select computes for this service chain";

      }

      container customization {

        tailf:info "Customization of deployed VNFs in this service chain";

        description "Customization of deployed VNFs in this service chain";

        list vnf-deployment {

          tailf:info "vnf-deployment profile to customize corresponding VNF";

          description "vnf-deployment profile to customize corresponding VNF";

          key profile;

          leaf profile {

            type leafref {

              path "deref(../../../cisco-sae-core-fp-cfs:deployment-name)/.."

                +"/cisco-sae-core-fp-cfs:vnf-deployment/cisco-sae-core-fp-cfs:vnf-profile";

            }

          }

          uses custom-template-grouping;

        }

      }

      leaf timestamp {

        tailf:hidden sae;

        type string;

        description "This timestamp should be updated internally by redeploy action";

      }

      tailf:action custom-redeploy {

        tailf:info "Action to redeploy service chain";

        tailf:actionpoint "cfs-redep-service-chain-action-point";

        output {

          leaf status {

            type string;

          }

          leaf error {

            type string;

          }

        }

      }

      tailf:action custom-undeploy-redeploy {

        tailf:info "Will undeploy & redeploy service vnfs ";

        description "Will undeploy & redeploy service vnfs";

        tailf:actionpoint "undeploy-redeploy";

        output {

          leaf status {

            type string;

            mandatory true;

          }

        }

      }

    }

    list stitching-service {

      uses ncs:service-data;

      ncs:servicepoint cfs-sae-stitching-servicepoint;

      tailf:info "Stitching service chain created from two already instantiated half-chains or"

          +" connect 2 endpoint-gateways";

      description "Stitching service chain created from two already instantiated half-chains or"

          +" connect 2 endpoint-gateways";

      key name;

      leaf name {

        tailf:info "Unique name for stitching-service";

        description "Unique name for stitching-service";

        type string;

      }

      container consumer-end {

        tailf:info "Consumer half-service/endpoint-gateway-vnf participating in the stitching";

        choice half-chain-or-epgw {

          case half-chain {

            leaf service-chain {

              tailf:info "Consumer half-service chain participating in the stitching";

              mandatory true;

              type leafref {

                path "../../../service-chain/name";

              }

              must "(deref(current())/../cisco-sae-core-fp-cfs:type = 'consumer'"

              + " or deref(current())/../cisco-sae-core-fp-cfs:type = 'provider') and "

              + "(not(../../provider-end/service-chain) or "

              + "../../provider-end/service-chain != current())" {

                error-message "Selected consumer service-chain should be a half-chain and " +

                "should not be selected as provider-end service-chain in this stitching service";

                tailf:dependency "../../../service-chain";

              }

            }

          }

          case epgw {

            container shared-gateway {

              tailf:info "Shared consumer endpoint-gateway participating in the stitching";

              presence true;

              leaf endpoint-gateway-vnf {

                tailf:info "Name of the shared consumer endpoint-gateway-vnf service";

                mandatory true;

                type leafref {

                  path "../../../../endpoint-gateway-vnf/name";

                }

              }

              leaf external-end-point {

                when "string-length(../endpoint-gateway-vnf) > 0";

                tailf:info "External-endpoint";

                type leafref {

                  path "deref(../cisco-sae-core-fp-cfs:endpoint-gateway-vnf)/../"

                  + "cisco-sae-core-fp-cfs:external-end-point/cisco-sae-core-fp-cfs:name";

                }

                must "(not(../../../provider-end/shared-gateway/external-end-point) or "

                + "../../../provider-end/shared-gateway/external-end-point != current())" {

                  error-message "external-end-point can not be same as provider-end";

                  tailf:dependency "../../../../endpoint-gateway-vnf/external-end-point";

                }

              }

              leaf gateway-access-point {

                when "string-length(../external-end-point) > 0";

                tailf:info "Endpoint-gateway-vnf's chain-end access point to be used for stitching";

                type string;

                tailf:cli-completion-actionpoint "chain-sapd-gw-cli-completion" {

                  tailf:cli-completion-id "get-endpoint-next-hop-sapd";

                }

              }

              must "(count(deref(endpoint-gateway-vnf)/../external-end-point) > 1" +

              " and external-end-point) or " +

              "(count(deref(endpoint-gateway-vnf)/../external-end-point) = 1)" {

                error-message "Please select external-end-point, because selected" +

                " endpoint-gateway-vnf has multiple external-end-points";

                tailf:dependency "endpoint-gateway-vnf";

              }

            }

          }

        }

      }

      container provider-end {

        tailf:info "Provider half-service/endpoint-gateway-vnf participating in the stitching";

        choice half-chain-or-epgw {

          case half-chain {

            leaf service-chain {

              tailf:info "Provider half-service chain participating in the stitching";

              mandatory true;

              type leafref {

                path "../../../service-chain/name";

              }

              must "(deref(current())/../cisco-sae-core-fp-cfs:type = 'consumer'"

              + " or deref(current())/../cisco-sae-core-fp-cfs:type = 'provider') and "

              + "(not(../../consumer-end/service-chain) or "

              + "../../consumer-end/service-chain != current())"{

                error-message "Selected provider service-chain should be a half-chain and " +

                "should not be selected as consumer-end service-chain in this stitching service";

                tailf:dependency "../../../service-chain";

              }

            }

          }

          case epgw {

            container shared-gateway {

              tailf:info "Shared provider endpoint-gateway participating in the stitching";

              presence true;

              leaf endpoint-gateway-vnf {

                tailf:info "Name of the shared provider endpoint-gateway-vnf service";

                mandatory true;

                type leafref {

                  path "../../../../endpoint-gateway-vnf/name";

                }

              }

              leaf external-end-point {

                when "string-length(../endpoint-gateway-vnf) > 0";

                tailf:info "External-endpoint";

                type leafref {

                  path "deref(../cisco-sae-core-fp-cfs:endpoint-gateway-vnf)/../"

                  + "cisco-sae-core-fp-cfs:external-end-point/cisco-sae-core-fp-cfs:name";

                }

                must "(not(../../../consumer-end/shared-gateway/external-end-point) or "

                + "../../../consumer-end/shared-gateway/external-end-point != current())" {

                  error-message "external-end-point can not be same as consumer-end";

                  tailf:dependency "../../../../endpoint-gateway-vnf/external-end-point";

                }

              }

              leaf gateway-access-point {

                when "string-length(../external-end-point) > 0";

                tailf:info "endpoint-gateway-vnf's chain-end access point to be used for stitching";

                type string;

                tailf:cli-completion-actionpoint "chain-sapd-gw-cli-completion" {

                  tailf:cli-completion-id "get-endpoint-next-hop-sapd";

                }

              }

              must "(count(deref(endpoint-gateway-vnf)/../external-end-point) > 1" +

              " and external-end-point) or " +

              "(count(deref(endpoint-gateway-vnf)/../external-end-point) = 1)" {

                error-message "Please select external-end-point, because selected" +

                " endpoint-gateway-vnf has multiple external-end-points";

                tailf:dependency "endpoint-gateway-vnf";

              }

            }

          }

        }

      }

      must "((provider-end/service-chain or provider-end/shared-gateway) and"+

      " (consumer-end/service-chain or consumer-end/shared-gateway))" {

        error-message "consumer-end and provider-end details are a must for stitching."

          +" Please provide complete details";

      }

      uses variables;

      leaf timestamp {

        tailf:hidden sae;

        type string;

        description "This timestamp should be updated internally by redeploy action";

      }

      tailf:action custom-redeploy {

        tailf:info "Action to redeploy stitching service";

        tailf:actionpoint "cfs-redep-stitch-action-point";

        output {

          leaf status {

            type string;

          }

          leaf error {

            type string;

          }

        }

      }

    }

    leaf timestamp {

      tailf:hidden sae;

      type string;

      description "This timestamp should be updated internally by redeploy action";

    }

    tailf:action custom-redeploy {

      tailf:info "Action to redeploy a SAE site";

      tailf:actionpoint "cfs-redep-site-action-point";

      output {

        leaf status {

          type string;

        }

        leaf error {

          type string;

        }

      }

    }

  }

  container sae-site-actions {

    tailf:info "sae-site actions";

    description "sae-site actions";

    tailf:action services-on-csp {

      tailf:info "List services on CSP device";

      description "List services on CSP device";

      tailf:actionpoint "services-on-csp";

      input {

        leaf csp-device {

          tailf:info "Name of CSP device";

          description "Name of CSP device";

          tailf:cli-completion-actionpoint "device-cli-completion" {

              tailf:cli-completion-id "csp-device";

            }

          type string;

          mandatory true;

        }

      }

      output {

        leaf status {

          type string;

          mandatory true;

        }

      }

    }

    tailf:action undep-redep-services-on-csp {

      tailf:info "undep-redep services on CSP device";

      description "undep-redep services on CSP device";

      tailf:actionpoint "undep-redep-services-on-csp";

      input {

        leaf csp-device {

          tailf:info "Name of CSP device";

          description "Name of CSP device";

          tailf:cli-completion-actionpoint "device-cli-completion" {

              tailf:cli-completion-id "csp-device";

            }

          type string;

          mandatory true;

        }

      }

      output {

        leaf status {

          type string;

          mandatory true;

        }

      }

    }

    tailf:action recover-vnf-on-csp {

      tailf:info "redeploy vnf from one csp to another csp";

      description "redeploy vnf from one csp to another csp";

      tailf:actionpoint "recover-vnf-on-csp";

      input {

        leaf sae-site {

          type leafref {

            path "/cisco-sae-core-fp-cfs:sae-site/site";

          }

          tailf:info "Name of SAE Site";

          description "Name of SAE Site";

          mandatory true;

        }

        choice service-name {

          case half-chain {

            leaf service-chain {

              tailf:info "Name of service chain";

              description "Name of service chain";

              type leafref {

                path "deref(../sae-site)/../cisco-sae-core-fp-cfs:service-chain/"

                  +"cisco-sae-core-fp-cfs:name";

                }

            }

          }

          case epgw-name {

            leaf endpoint-gateway-vnf-service {

              tailf:info "Name of Endpoint Gateway VNF service";

              description "Name of Endpoint Gateway VNF service";

              type leafref {

                path "/cisco-sae-core-fp-cfs:sae-site[cisco-sae-core-fp-cfs:site=current()/"

                  + "../sae-site]/cisco-sae-core-fp-cfs:endpoint-gateway-vnf/name";

                }

            }

          }

        }

        leaf vnf-profile {

          tailf:info "Name of vnf-profile to be redeployed";

          description "Name of vnf-profile to be redeployed";

          type leafref {

            path "/nfvo-rel2:nfvo/nfvo-rel2:nsd/nfvo-rel2:deployment-flavor/"

              + "nfvo-rel2:vnf-profile/nfvo-rel2:id";

            }

        }

      }

      output {

        leaf status {

          type string;

          mandatory true;

        }

      }

    }

  }

  container sae-cleanup {

    tailf:info "Cleanup actions for SAE site";

    description "Cleanup actions for SAE site";

    tailf:action csp\_cleanup {

      tailf:info "Will cleanup leftover data for csp-day1 service";

      description "Will cleanup leftover data for csp-day1 service";

      tailf:actionpoint "csp-cleanup";

      input {

        leaf sae-site {

          tailf:info "Name of SAE Site";

          description "Name of SAE Site";

          type string;

          mandatory true;

        }

        leaf compute-clusters {

          tailf:info "Name of Compute Cluster";

          description "Name of Compute Cluster";

          type string;

          mandatory true;

        }

        leaf csp-device {

          tailf:info "Name of CSP Device for Day1 cleanup";

          description "Name of CSP Device for Day1 cleanup";

          tailf:cli-completion-actionpoint "device-cli-completion" {

              tailf:cli-completion-id "csp-device";

            }

          type string;

          mandatory true;

        }

      }

      output {

        leaf status {

          type string;

          mandatory true;

        }

      }

      tailf:confirm-text "########################\n" +

            "#        Warning       #\n" +

            "########################\n" +

            "You are about to forcefully clean up a compute intent\n" +

            "This could southbound lock CSP device momentarily, affecting

                        any actively" +

            " deploying service(s) on this site\n" +

            "Also, this might leave CSP and NSO out-of-sync\n" +

            "Are you sure, you want to proceed?" {

        tailf:confirm-default false;

      }

    }

    tailf:action vnf-manager-cleanup {

      tailf:info "Will cleanup leftover data for vnf-manager service";

      description "Will cleanup leftover data for vnf-manager service";

      tailf:actionpoint "vnf-manager-cleanup";

      input {

        leaf sae-site {

          tailf:info "Name of SAE Site";

          description "Name of SAE Site";

          type string;

          mandatory true;

        }

        leaf vnf-manager-service {

          tailf:info "Name of VNF Manager service to be cleaned";

          description "Name of VNF Manager service to be cleaned";

          type string;

          mandatory true;

        }

        leaf more {

          tailf:info "To provide more parameters about the service that helps clean-up every" +

                    " possible trace of detail from NSO";

          type boolean;

          default false;

          must "current() = 'false' or ../isManaged" {

            error-message "Please provide if the vnf-manager is a managed device";

            tailf:dependency "../isManaged";

          }

        }

        leaf isManaged {

          when "../more='true'";

          tailf:info "Is the VNF Manager managed by NSO";

          type boolean;

          must "current() = 'false' or ../sae-site-provider" {

            error-message "Please provide provider name of the sae-site";

            tailf:dependency "../sae-site-provider";

          }

          must "current() = 'false' or ../vnf-manager-device" {

            error-message "Please provide name of the vnf-manager device";

            tailf:dependency "../vnf-manager-device";

          }

        }

        leaf vnf-manager-device {

          when "../isManaged='true'";

          tailf:info "Name of the vnf-manager device";

          type string;

        }

        leaf sae-site-provider {

          when "../isManaged='true'";

          tailf:info "Provider name of the sae-site";

          type string;

        }

      }

      output {

        leaf status {

          type string;

          mandatory true;

        }

      }

      tailf:confirm-text "########################\n" +

            "#        Warning       #\n" +

            "########################\n" +

            "You are about to forcefully clean up a vnf-manager intent\n" +

            "This could southbound lock CSP device momentarily, affecting

                        any actively" +

            " deploying service(s) on this site\n" +

            "Also, this might leave CSP and NSO out-of-sync\n" +

            "Are you sure, you want to proceed?" {

        tailf:confirm-default false;

      }

    }

    tailf:action endpoint-gateway-vnf-cleanup {

      tailf:info "Will cleanup leftover data for endpoint-gateway-vnf service";

      description "Will cleanup leftover data for endpoint-gateway-vnf service";

      tailf:actionpoint "endpoint-gateway-vnf-cleanup";

      input {

        leaf sae-site {

          tailf:info "Name of SAE Site";

          description "Name of SAE Site";

          type string;

          mandatory true;

        }

        leaf endpoint-gateway-vnf-service {

          tailf:info "Name of Endpoint Gateway VNF service to be cleaned";

          description "Name of Endpoint Gateway VNF service to be cleaned";

          type string;

          mandatory true;

        }

        leaf more {

          tailf:info "To provide more parameters about the service that helps clean-up every" +

                    " possible trace of detail from NSO";

          type boolean;

          default false;

          must "current() = 'false' or ../sae-site-provider" {

            error-message "Please provide provider name of the sae-site";

            tailf:dependency "../sae-site-provider";

          }

          must "current() = 'false' or ../catalog" {

            error-message "Please provide catalog used for sae-site";

            tailf:dependency "../catalog";

          }

          must "current() = 'false' or ../nsd-deployment" {

            error-message "Please provide nsd-deployment used for endpoint-gateway intent";

            tailf:dependency "../nsd-deployment";

          }

        }

        leaf sae-site-provider {

          when "../more='true'";

          tailf:info "Provider name of the sae-site";

          type string;

        }

        leaf catalog {

          when "../more='true'";

          tailf:info "Catalog used for sae-site";

          type string;

        }

        leaf nsd-deployment {

          when "../more='true'";

          tailf:info "NSD-deployment used for endpoint-gateway intent";

          type string;

        }

      }

      output {

        leaf status {

          type string;

          mandatory true;

        }

      }

      tailf:confirm-text "########################\n" +

            "#        Warning       #\n" +

            "########################\n" +

            "You are about to forcefully clean up a endpoint-gateway-vnf intent\n" +

            "This could southbound lock ESC and/or N9K devices momentarily, affecting any actively"

            + " deploying service(s) on this site\n" +

            "Also, this might leave ESC and/or N9K devices and NSO out-of-sync\n" +

            "Are you sure, you want to proceed?" {

        tailf:confirm-default false;

      }

    }

    tailf:action service-chain-cleanup {

      tailf:info "Will cleanup leftover data for service-chain service";

      description "Will cleanup leftover data for service-chain service";

      tailf:actionpoint "service-chain-cleanup";

      input {

        leaf sae-site {

          tailf:info "Name of SAE Site";

          description "Name of SAE Site";

          type string;

          mandatory true;

        }

        leaf service-chain {

          tailf:info "Name of service chain to be cleaned";

          description "Name of service chain to be cleaned";

          type string;

          mandatory true;

        }

        leaf more {

          tailf:info "To provide more parameters about the service that helps clean-up every" +

                    " possible trace of detail from NSO";

          type boolean;

          default false;

          must "current() = 'false' or ../sae-site-provider" {

            error-message "Please provide provider name of the sae-site";

            tailf:dependency "../sae-site-provider";

          }

          must "current() = 'false' or ../catalog" {

            error-message "Please provide catalog used for sae-site";

            tailf:dependency "../catalog";

          }

          must "current() = 'false' or ../nsd-deployment" {

            error-message "Please provide nsd-deployment used for service-chain intent";

            tailf:dependency "../nsd-deployment";

          }

        }

        leaf sae-site-provider {

          when "../more='true'";

          tailf:info "Provider name of the sae-site";

          type string;

        }

        leaf catalog {

          when "../more='true'";

          tailf:info "Catalog used for sae-site";

          type string;

        }

        leaf nsd-deployment {

          when "../more='true'";

          tailf:info "NSD-deployment used for service-chain intent";

          type string;

        }

      }

      output {

        leaf status {

          type string;

          mandatory true;

        }

      }

      tailf:confirm-text "########################\n" +

            "#        Warning       #\n" +

            "########################\n" +

            "You are about to forcefully clean up a service chain intent\n" +

            "This could southbound lock ESC and/or N9K devices momentarily, affecting any actively"

            + " deploying service(s) on this site\n" +

            "Also, this might leave ESC and/or N9K devices and NSO out-of-sync\n" +

            "Are you sure, you want to proceed?" {

        tailf:confirm-default false;

      }

    }

    tailf:action stitching-service-cleanup {

      tailf:info "Will cleanup leftover data for stitching service";

      description "Will cleanup leftover data for stitching service";

      tailf:actionpoint "stitching-service-cleanup";

      input {

        leaf sae-site {

          tailf:info "Name of SAE Site";

          description "Name of SAE Site";

          type string;

          mandatory true;

        }

        leaf stitching-service {

          tailf:info "Name of stitching service to be cleaned";

          description "Name of stitching service to be cleaned";

          type string;

          mandatory true;

        }

      }

      output {

        leaf status {

          type string;

          mandatory true;

        }

      }

      tailf:confirm-text "########################\n" +

            "#        Warning       #\n" +

            "########################\n" +

            "You are about to forcefully clean up a stitching-service chain

                        intent\n" +

            "This could southbound lock N9K devices momentarily, affecting any actively" +

            " deploying service(s) on this site\n" +

            "Also, this might leave N9K devices and NSO out-of-sync\n" +

            "Are you sure, you want to proceed?" {

        tailf:confirm-default false;

      }

    }

  }

}

## VNF Model

submodule tailf-etsi-rel2-nfvo-vnf {

  belongs-to tailf-etsi-rel2-nfvo {

    prefix nfvo-rel2;

  }

  import tailf-common {

    prefix tailf;

  }

  import ietf-inet-types {

    prefix inet;

  }

  include tailf-etsi-rel2-nfvo-common {

    revision-date 2017-01-20;

  }

  description

    "Models for VNFD according to GS NFV-IFA 011.";

  revision 2017-01-20 {

    description

      "Initial revision

       Common data structure to support VNFD

       according to:

       ETSI GS NFV-IFA 011 V2.1.1 (2016-10)";

    reference "ETSI GS NFV-IFA 011 V2.1.1 (2016-10)";

  }

  grouping cpd-information-element {

    reference "7.1.6.2 Cpd Information Element";

    leaf id {

      type string;

    }

    leaf layer-protocol {

      mandatory true;

      type identityref {

        base layer-protocol;

      }

    }

    leaf role {

      type cp-role;

    }

    leaf description {

      tailf:info "Human readable description of the connection point";

      type string;

    }

    list address-data {

      key "address-type";

      leaf address-type {

        type enumeration {

          enum mac-address;

          enum ip-address;

        }

      }

      container l2-address-data {

        when "../nfvo-rel2:address-type='mac-address'";

      }

      container l3-address-data {

        when "../nfvo-rel2:address-type='ip-address'";

        leaf ip-address-assignment {

          description

            "Specify if the address assignment is the

             responsibility of management and

             orchestration function or not.";

          type empty;

        }

        leaf floating-ip-activated {

          description

            "Specify if the floating IP scheme is

             activated on the CP or not.";

          type empty;

        }

        leaf ip-address-type {

          type enumeration {

            enum "IPv4";

            enum "IPv6";

          }

        }

        leaf number-of-ip-addresses {

          type uint32;

        }

      }

    }

  }

  grouping virtual-network-interface-requirement-list {

    list virtual-network-interface-requirement {

      key "name";

      leaf name {

        description

          "Provides a human readable name for the

           requirement.";

        type string;

      }

      leaf description {

        type string;

      }

      leaf support-mandatory {

        description

          "Indicates whether fulfilling the constraint is

          mandatory for successful operation or desirable.";

        type empty;

      }

      leaf requirement {

        mandatory true;

        type enumeration {

          enum "SR-IOV";

          enum "DPDK";

        }

      }

    }

  }

  grouping vnfd {

    list vnfd {

      key "id";

      description

        "A VNF Descriptor (VNFD) is a deployment template which describes a VNF

         in terms of deployment and operational behaviour requirements.

         It also contains connectivity, interface and virtualised resource

         requirements";

      reference "GS NFV IFA011: VNFD information element";

      leaf id {

        tailf:info "Unique ID of the VNFD";

        type string;

      }

      leaf provider {

        // should be mandatory

        tailf:info "Provider of the VNF and of the VNFD";

        type string;

      }

      leaf product-name {

        // should be mandatory

        tailf:info "Human readable name for the VNF Product.";

        type string;

      }

      leaf software-version {

        // should be mandatory

        tailf:info "Software version of the VNF.";

        type string;

      }

      leaf version {

        // should be mandatory

        tailf:info "Identifies the version of the VNFD";

        type string;

      }

      leaf product-info-name {

        tailf:info "Human readable name of the VNFD";

        type string;

      }

      leaf product-info-description {

        tailf:info "Human readable description of the VNFD";

        type string;

      }

      leaf-list vnfm-info {

        // Should be min-elements 1 according to specification.

        type string;

      }

      leaf localization-language {

        description "Information about the language of the VNF.";

        type string;

      }

      leaf default-localization-language {

        type string;

      }

      list vdu {

        key "id";

        min-elements 1;

        description

          "The Virtualisation Deployment Unit (VDU) is a construct supporting

           the description of the deployment and operational behaviour of a

           VNF component, or the entire VNF if it was not componentized in

           components";

        reference "GS NFV IFA011: VDU information element";

        leaf id {

          tailf:info "Unique ID of the VDU";

          type string;

        }

        leaf name {

          description "Human readable name of the VDU";

          type string;

        }

        leaf description {

          tailf:info "Human readable description of the VDU";

          type string;

        }

        list internal-connection-point-descriptor {

          key "id";

          min-elements 1;

          description

            "A internal-connection-point element is a type of connection point and describes

             network connectivity between a VDU instance and an internal Virtual Link or an

             external connection point.";

          reference "GS NFV IFA011: VduCpd information element";

          choice cp-connection {

            mandatory true;

            description

              "A connection point must either connect to an internal virtual

               link or to an external connection points.";

            leaf virtual-link-descriptor {

              tailf:info "Connect to an internal VLD";

              type leafref {

                path "../../../nfvo-rel2:virtual-link-descriptor/nfvo-rel2:id";

              }

            }

            leaf external-connection-point-descriptor {

              tailf:info "Connect to an external connection point";

              type leafref {

                path "../../../nfvo-rel2:external-connection-point-descriptor/nfvo-rel2:id";

              }

            }

          }

          leaf bitrate-requirement {

            type uint64;

            units "Mbps";

          }

          uses virtual-network-interface-requirement-list;

          uses cpd-information-element;

        }

        leaf virtual-compute-descriptor {

          type leafref {

            path "../../nfvo-rel2:virtual-compute-descriptor/nfvo-rel2:id";

          }

        }

        leaf-list virtual-storage-descriptor {

          type leafref {

            path "../../nfvo-rel2:virtual-storage-descriptor/nfvo-rel2:id";

          }

        }

        list boot-order {

          ordered-by user;

          key "virtual-storage-descriptor";

          leaf virtual-storage-descriptor {

            type leafref {

              path "../../nfvo-rel2:virtual-compute-descriptor";

            }

          }

        }

        container software-image-descriptor {

          presence "Can be used to specify image data if the VNF should be onboarded to the VIM by the NFVO";

          description

            "VM image data for the image used to boot the VDU.";

          reference "GS NFV IFA011: SwImageDescriptor information element";

          leaf name {

            type string;

          }

          leaf version {

            type string;

          }

          leaf checksum {

            type string;

          }

          leaf container-format {

            mandatory true;

            type enumeration {

              enum "bare";

              enum "ovf";

            }

            description

              "VIM specific information for the container format";

          }

          leaf disk-format {

            mandatory true;

            type enumeration {

              enum "qcow2";

              enum "raw";

              enum "vmdk";

            }

            description

              "VIM specific information for format of the image";

          }

          leaf min-disk {

            type uint64;

            units "Gb";

          }

          leaf min-ram {

            type decimal64 {

              fraction-digits 1;

              range "0..max";

            }

            units "Gb";

          }

          leaf size {

            type uint64;

          }

          leaf image {

            mandatory true;

            type inet:uri;

            description

              "URI to the image";

          }

          leaf operating-system {

            type string;

          }

          leaf-list supported-virtualization-environment {

            type string;

          }

        }

        leaf-list nfvi-constraint {

          type string;

        }

        list monitoring-parameter {

          key "id";

          leaf id {

            type string;

          }

          uses monitoring-parameter;

        }

        container configurable-properties {

        }

      }

      list virtual-compute-descriptor {

        key "id";

        description

          "Specifies the sizing of the virtual machine.";

        leaf id {

          type string;

        }

        list request-additional-capability {

          key "name";

          leaf name {

            description

              "Identifies a requested additional capability

               for the VDU. ETSI GS NFV-IFA 002 [i.1]

               describes acceleration capabilities.";

            type string;

          }

          leaf support-mandatory {

            type empty;

          }

          leaf min-version {

            type string;

          }

          leaf preferred-version {

            type string;

          }

          list target-performance-parameters {

            key "key";

            leaf key {

              type string;

            }

            leaf value {

              type string;

            }

          }

        }

        container virtual-memory {

          leaf virtual-memory-size {

            tailf:info "Size in Gb";

            type decimal64 {

              fraction-digits 1;

              range "0..max";

            }

            units "GB";

            mandatory true;

          }

          leaf virtual-memory-subscription-policy {

            type string;

          }

          leaf numa-enabled {

            type empty;

          }

        }

        container virtual-cpu {

          leaf cpu-architecture {

            type string;

          }

          leaf number-of-virtual-cpus {

            type uint16 {

              range "1..max";

            }

            mandatory true;

          }

          leaf virtual-cpu-clock {

            type uint32;

            units "MHz";

          }

          leaf virtual-cpu-oversubscription-policy {

            type string;

          }

          container virtual-cpu-pinning {

            presence "Set to specify CPU pinning.";

            leaf cpu-pinning-policy {

              mandatory true;

              type enumeration {

                enum "static";

                enum "dynamic";

              }

            }

            container cpu-pinning-map {

              when "../nfvo-rel2:cpu-pinning-policy='static'";

            }

          }

        }

      }

      list virtual-storage-descriptor {

        key "id";

        description

          "Storage requirements for a Virtual Storage instance

           attached to the VNFC created from this VDU";

        leaf id {

          type string;

        }

        leaf type-of-storage {

          // Needed to be able to onboard images

          type enumeration {

            enum "root";

            enum "swap";

            enum "ephemeral";

          }

          mandatory true;

        }

        leaf size-of-storage {

          tailf:info "Size in Gib";

          type uint64;

          units "GiB";

          mandatory true;

        }

        leaf rdma-enabled {

          type empty;

        }

        /\*

          swImageDesc

          The specification prescribes a reference to a VDU's software image

          the VDU already has information of which image it uses, so this would

          unnecessarily complicate things.

        \*/

      }

      list virtual-link-descriptor {

        key "id";

        description

          "Represents the type of network connectivity mandated by the VNF

           vendor between two or more Connection";

        reference "GS NFV IFA011: Information elements related to the Vld";

        leaf id {

          type string;

        }

        list descriptor-flavor {

          key "id";

          leaf id {

            type string;

          }

          uses link-bitrate-requirements;

          container qos {

            leaf latency {

              mandatory true;

              description

                "Specifies the maximum latency in ms.";

              type uint32;

              units "ms";

            }

            leaf packet-delay-variation {

              mandatory true;

              description

                "Specifies the maximum jitter in ms.";

              type uint32;

              units "ms";

            }

            leaf packet-loss-ratio {

              description

                "Specifies the maximum packet loss ratio.";

              type decimal64 {

                fraction-digits "2";

                range "0..1.00";

              }

            }

          }

        }

        uses connectivity-type;

        leaf-list test-access {

          type string;

        }

        leaf description {

          tailf:info "Human readable description of the VLD";

          type string;

        }

        list monitoring-parameter {

          key "id";

          leaf id {

            type string;

          }

          uses monitoring-parameter;

        }

      }

      list external-connection-point-descriptor {

        key "id";

        min-elements 1;

        description

          "Describes an external interface exposed by this VNF enabling

           connection with a Virual Link";

        leaf virtual-link-descriptor {

          tailf:info "Optional connection to an internal VLD";

          type leafref {

            path "../../nfvo-rel2:virtual-link-descriptor/nfvo-rel2:id";

          }

        }

        /\*

          intCpd

          we reference from the VDU's CP to the ext-CP instead of the

          ext-CP to the VDU's CP.

        \*/

        uses virtual-network-interface-requirement-list;

        uses cpd-information-element;

      }

      list deployment-flavor {

        must "nfvo-rel2:default-instantiation-level or count(nfvo-rel2:instantiation-level) = 1";

        key "id";

        min-elements 1;

        description

          "Describes a specific deployment version of a VNF with specific

           requirements for capacity and performance.";

        reference "GS NFV IFA011: Information elements related to the DeploymentFlavour";

        leaf id {

          type string;

        }

        leaf description {

          tailf:info "Human readable description of the deployment flavor";

          type string;

        }

        list vdu-profile {

          key "vdu";

          min-elements 1;

          description

            "The Vduprofile describes additional instantiation data for

             a given VDU used in a deployment flavour.";

          reference "GS NFV IFA011: VduProfile information element";

          leaf vdu {

            type leafref {

              path "../../../nfvo-rel2:vdu/nfvo-rel2:id";

            }

          }

          leaf min-number-of-instances {

            type uint16;

            mandatory true;

          }

          leaf max-number-of-instances {

            type uint16;

            mandatory true;

            must ". >= ../nfvo-rel2:min-number-of-instances";

          }

          uses local-affinity-or-anti-affinity-rule-list;

          list affinity-or-anti-affinity-group {

            key "id";

            leaf id {

              type leafref {

                path "../../../nfvo-rel2:affinity-or-anti-affinity-group/nfvo-rel2:id";

              }

            }

          }

        }

        list virtual-link-profile {

          key "virtual-link-descriptor flavor";

          leaf virtual-link-descriptor {

            type leafref {

              path "../../../nfvo-rel2:virtual-link-descriptor/nfvo-rel2:id";

            }

          }

          leaf flavor {

            type leafref {

              path "deref(../nfvo-rel2:virtual-link-descriptor)/../nfvo-rel2:descriptor-flavor/nfvo-rel2:id";

            }

          }

          uses local-affinity-or-anti-affinity-rule-list;

          list affinity-or-anti-affinity-group {

            key "id";

            leaf id {

              type leafref {

                path "../../../nfvo-rel2:affinity-or-anti-affinity-group/nfvo-rel2:id";

              }

            }

          }

        }

        list instantiation-level {

          key "id";

          min-elements 1;

          description

            "The InstantiationLevel information element describes a

             given level of resources to be instantiated within a

             deployment flavour in term of the number of VNFC instances

             to be created from each VDU.

             All the VDUs referenced in the level shall be part of the

             corresponding deployment flavour and their number shall

             be within the range (min/max) for this deployment flavour.";

          reference "GS NFV IFA011: InstantiationLevel information element";

          leaf id {

            type string;

          }

          leaf description {

            tailf:info "Human readable description of the instantiation level";

            type string;

          }

          list vdu-level {

            key "vdu";

            min-elements 1;

            description

              "Sets the number of instances for the VDU in this

               instantiation level.";

            leaf vdu {

              type leafref {

                path "../../../../nfvo-rel2:vdu/nfvo-rel2:id";

              }

            }

            leaf number-of-instances {

              type uint16;

              must ". <= ../../../nfvo-rel2:vdu-profile[vdu=current()/../nfvo-rel2:vdu]/nfvo-rel2:max-number-of-instances";

              must ". >= ../../../nfvo-rel2:vdu-profile[vdu=current()/../nfvo-rel2:vdu]/nfvo-rel2:min-number-of-instances";

              mandatory true;

            }

          }

          list scaling-info {

            key "aspect";

            leaf aspect {

              type leafref {

                path "../../../nfvo-rel2:scaling-aspect/nfvo-rel2:id";

              }

            }

            leaf scale-level {

              type uint32;

            }

          }

        }

        leaf default-instantiation-level {

          type leafref {

            path "../nfvo-rel2:instantiation-level/nfvo-rel2:id";

          }

        }

        leaf-list supported-operation {

          type string;

        }

        container lcm-operations-configuration {

          leaf instantiate-vnf-op-config {

            type string;

          }

          leaf scale-vnf-op-config {

            type string;

          }

          leaf scale-vnf-to-level-op-config {

            type string;

          }

          leaf heal-vnf-op-config {

            type string;

          }

          leaf terminate-vnf-op-config {

            type string;

          }

          leaf operate-vnf-op-config {

            type string;

          }

        }

        list affinity-or-anti-affinity-group {

          key "id";

          leaf id {

            type string;

          }

          leaf affinity-type {

            mandatory true;

            type affinity-type;

          }

          leaf affinity-scope {

            mandatory true;

            type affinity-scope;

          }

        }

        list monitoring-parameter {

          key "id";

          leaf id {

            type string;

          }

          uses monitoring-parameter;

        }

        list scaling-aspect {

          key "id";

          leaf id {

            type string;

          }

          leaf name {

            type string;

          }

          leaf description {

            type string;

          }

          leaf associated-group {

            type leafref {

              path "../../../nfvo-rel2:element-group/nfvo-rel2:id";

            }

          }

          leaf max-scale-level {

            type uint32 {

              range "1..max";

            }

          }

        }

      }

      container configurable-properties {

        leaf auto-scalable {

          type empty;

        }

        leaf auto-healable {

          type empty;

        }

        leaf-list additional-configurable-property {

          type string;

        }

      }

      container modifiable-attributes {

        leaf-list extension {

          type string;

        }

        leaf-list metadata {

          type string;

        }

      }

      list lifecycle-management-script {

        key "event";

        leaf event {

          type string;

        }

        leaf script {

          type string;

        }

      }

      list element-group {

        key "id";

        leaf id {

          type string;

        }

        leaf description {

          type string;

        }

        leaf-list vdu {

          type leafref {

            path "../../nfvo-rel2:vdu/nfvo-rel2:id";

          }

        }

        leaf-list virtual-link-descriptor {

          type leafref {

            path "../../nfvo-rel2:virtual-link-descriptor/nfvo-rel2:id";

          }

        }

      }

      list indicator {

        key "id";

        leaf id {

          type string;

        }

        leaf name {

          type string;

        }

        leaf-list indicator-value {

          type string;

        }

        leaf source {

          type enumeration {

            enum "VNF";

            enum "EM";

            enum "both";

          }

        }

      }

      leaf-list auto-scale {

        type string;

      }

    }

  }

}

## NSD Model

submodule tailf-etsi-rel2-nfvo-ns {

  belongs-to tailf-etsi-rel2-nfvo {

    prefix nfvo-rel2;

  }

  include tailf-etsi-rel2-nfvo-common {

    revision-date 2017-01-20;

  }

  import ietf-inet-types { prefix inet; }

  description

    "Models for NS according to ETSI GS NFV-IFA 014.";

  revision 2017-01-20 {

    description

      "Initial revision

       Common data structure to support NSD

       according to:

       ETSI GS NFV-IFA 014 V2.1.1 (2016-10)";

    reference "ETSI GS NFV-IFA 014 V2.1.1 (2016-10)";

  }

  grouping security-parameters {

    container security {

      leaf signature {

        type string;

      }

      leaf algorithm {

        type string;

      }

      leaf certificate {

        type string;

      }

    }

  }

  grouping resource-handle {

    leaf vim-id {

      type string;

    }

    leaf resrouce-provider-id {

      type string;

    }

    leaf resource-id {

      type string;

    }

  }

  grouping nsd {

    list nsd {

      key "id";

      leaf id {

        type string;

        description

          "Uniquely identifies an network service descriptor";

      }

      leaf designer {

        type string;

        description

          "Identifies the designer of the network service descriptor";

      }

      leaf version {

        type string;

        description

          "Identifies the version of the network service descriptor";

      }

      leaf name {

        type string;

        description

          "Provides the human readable name of the network service. descriptor";

      }

      leaf invariant-id {

        description

          "Identifies an network service descriptor in a version

          independent manner. This attribute is invariant across

          versions of the network service descriptor.";

        type string;

      }

      leaf-list nested-nsd {

        type leafref {

          path "../../nfvo-rel2:nsd/nfvo-rel2:id";

        }

        must ". != ../nfvo-rel2:id";

      }

      list vnfd {

        key "vnfd";

        min-elements 1;

        description

          "This lists the VNFDs this network service

           descriptor consists of.";

        leaf vnfd {

          type leafref {

            path "/nfvo-rel2:nfvo/nfvo-rel2:vnfd/nfvo-rel2:id";

          }

        }

      }

      leaf-list pnfd {

        type string;

      }

      list service-access-point-descriptor {

        key "id";

        min-elements 1;

        description

          "A service access point serves as the network

           service's external connection point";

        leaf id {

          type string;

        }

        leaf address-assignment {

          description

            "Specify whether the SAP address

             assignment is under the responsibility of

             management and orchestration functions

             or not. If it is set to True, management and

             orchestration functions are responsible for

             assigning addresses to the access points

             instantiated from this SAPD";

          type empty;

        }

        leaf description {

          type string;

        }

        leaf layer-protocol {

          type identityref {

            base layer-protocol;

          }

        }

        leaf role {

          type cp-role;

        }

        leaf virtual-link-descriptor {

          type leafref {

            path "../../nfvo-rel2:virtual-link-descriptor/nfvo-rel2:id";

          }

          description

            "Can optionally connect this service-access-point to an virtual-link. If not set,

             it's assumed a VNFD's connection-points references this

             service-access-point.";

        }

        /\*

          associatedCpdId:

          Deliberatly not according to standard, see

          vnf-profile -> sapd-connectivity

        \*/

      }

      list virtual-link-descriptor {

        key "id";

        description

          "The virtual-link-descriptor information element provides general information

           enabling the instantiation of virtual links.";

        leaf id {

          type string;

        }

        leaf provider {

          description

            "Defines the organization generating the VLD.  ";

          type string;

        }

        leaf version {

          mandatory true;

          type string;

        }

        uses connectivity-type;

        list deployment-flavor {

          key "id";

          leaf id {

            type string;

          }

          container qos {

            leaf latency {

              mandatory true;

              description

                "Specifies the maximum latency in ms.";

              type uint32;

              units "ms";

            }

            leaf packet-delay-variation {

              mandatory true;

              description

                "Specifies the maximum jitter in ms.";

              type uint32;

              units "ms";

            }

            leaf packet-loss-ratio {

              description

                "Specifies the maximum packet loss ratio.";

              type decimal64 {

                fraction-digits "2";

                range "0..1.00";

              }

            }

            leaf priority {

              description

                "Specifies the priority level in case of

                 congestion on the underlying physical links.";

              type uint32;

            }

          }

          leaf service-availability-level {

            reference "ETSI GS NFV-REL 001 [i.5]";

            type enumeration {

              enum "level-1";

              enum "level-2";

              enum "level-3";

            }

          }

        }

        leaf test-access {

          type enumeration {

            enum none;

            enum passive-monitoring;

            enum active;

          }

        }

        leaf description {

          type string;

        }

        uses security-parameters;

      }

      // vnffgd skipped since not fully specified by the spec

      list monitored-info {

        key "id";

        leaf id {

          type string;

        }

        choice source {

          // We have no way of refererencing a VNF MP today

          container ns-monitoring-parameter {

            uses monitoring-parameter;

          }

          container vnf-indicator {

            leaf vnfd {

              type leafref {

                path "/nfvo-rel2:nfvo/nfvo-rel2:vnfd/nfvo-rel2:id";

              }

            }

            leaf indicator {

              type leafref {

                path "deref(../nfvo-rel2:vnfd)/../nfvo-rel2:indicator/nfvo-rel2:id";

              }

            }

          }

        }

      }

      leaf-list autoscale-rule {

        type string;

      }

      list lifecycle-management-script {

        key "event";

        leaf event {

          type string;

        }

        leaf script {

          type string;

        }

      }

      list deployment-flavor {

        min-elements 1;

        must "nfvo-rel2:default-instantiation-level or count(nfvo-rel2:instantiation-level) = 1";

        key "id";

        description

          "The NsDf information element specifies the properties of a variant of an NS.";

        leaf id {

          type string;

        }

        leaf flavor-key {

          type leafref {

            path "../../nfvo-rel2:monitored-info/nfvo-rel2:id";

          }

        }

        list vnf-profile {

          must "nfvo-rel2:min-number-of-instances<=nfvo-rel2:max-number-of-instances";

          must "count(../../nfvo-rel2:vnfd[vnfd=current()/nfvo-rel2:vnfd])=1" {

            error-message "VNFDs in the profile has to be listed as a dependency of the network service descriptor.";

          }

          key "id";

          leaf id {

            type string;

          }

          leaf vnfd {

            mandatory true;

            type leafref {

              path "/nfvo-rel2:nfvo/nfvo-rel2:vnfd/nfvo-rel2:id";

            }

          }

          leaf flavor {

            mandatory true;

            type leafref {

              path "deref(../nfvo-rel2:vnfd)/../nfvo-rel2:deployment-flavor/nfvo-rel2:id";

            }

          }

          leaf instantiation-level {

            mandatory true;

            type leafref {

              path "deref(../nfvo-rel2:flavor)/../nfvo-rel2:instantiation-level/nfvo-rel2:id";

            }

          }

          leaf min-number-of-instances {

            type uint16;

            mandatory true;

          }

          leaf max-number-of-instances {

            type uint16;

            mandatory true;

          }

          uses local-affinity-or-anti-affinity-rule-list;

          list affinity-or-anti-affinity-group {

            key "id";

            leaf id {

              type leafref {

                path "../../../nfvo-rel2:affinity-or-anti-affinity-group/nfvo-rel2:id";

              }

            }

          }

          list virtual-link-connectivity {

            key "virtual-link-profile cp";

            leaf virtual-link-profile {

              type leafref {

                path "../../../nfvo-rel2:virtual-link-profile/nfvo-rel2:virtual-link-descriptor";

              }

            }

            leaf cp {

              type leafref {

                path "deref(../../nfvo-rel2:vnfd)/../nfvo-rel2:external-connection-point-descriptor/nfvo-rel2:id";

              }

            }

          }

          // Not according to standard

          list sapd-connectivity {

            key "sapd cp";

            leaf sapd {

              type leafref {

                path "../../../../nfvo-rel2:service-access-point-descriptor/nfvo-rel2:id";

              }

            }

            leaf cp {

              type leafref {

                path "deref(../../nfvo-rel2:vnfd)/../nfvo-rel2:external-connection-point-descriptor/nfvo-rel2:id";

              }

            }

          }

        }

        list pnf-profile {

          key "pnfd";

          leaf pnfd {

            type leafref {

              path "../../../nfvo-rel2:pnfd";

            }

          }

          // no virtual-link-connectivity as we don't have PNFD modelled.

        }

        list virtual-link-profile {

          key "virtual-link-descriptor";

          leaf virtual-link-descriptor {

            mandatory true;

            type leafref {

              path "../../../nfvo-rel2:virtual-link-descriptor/nfvo-rel2:id";

            }

          }

          leaf flavor {

            mandatory true;

            type leafref {

              path "deref(../nfvo-rel2:virtual-link-descriptor)/../nfvo-rel2:deployment-flavor/nfvo-rel2:id";

            }

          }

          uses local-affinity-or-anti-affinity-rule-list;

          list affinity-or-anti-affinity-group {

            key "id";

            leaf id {

              type leafref {

                path "../../../nfvo-rel2:affinity-or-anti-affinity-group/nfvo-rel2:id";

              }

            }

          }

          container max-bitrate-requirements {

            uses link-bitrate-requirements;

          }

          container min-bitrate-requirements {

            uses link-bitrate-requirements;

          }

        }

        list scaling-aspect {

          key "aspect";

          leaf aspect {

            type string;

          }

          leaf name {

            type string;

          }

          leaf description {

            type string;

          }

          leaf scaling-level {

            type leafref {

              path "../../nfvo-rel2:instantiation-level/nfvo-rel2:id";

            }

          }

        }

        list affinity-or-anti-affinity-group {

          key "id";

          leaf id {

            type string;

          }

          leaf affinity-type {

            mandatory true;

            type affinity-type;

          }

          leaf affinity-scope {

            mandatory true;

            type affinity-scope;

          }

        }

        list instantiation-level {

          min-elements 1;

          key "id";

          leaf id {

            type string;

          }

          leaf description {

            mandatory true;

            type string;

          }

          list vnf-to-level-mapping {

            key "vnf-profile";

            leaf vnf-profile {

              type leafref {

                path "../../../nfvo-rel2:vnf-profile/nfvo-rel2:id";

              }

            }

            leaf number-of-instances {

              mandatory true;

              type uint32;

            }

          }

          list vl-to-level-mapping {

            key "virtual-link-profile";

            leaf virtual-link-profile {

              type leafref {

                path "../../../nfvo-rel2:virtual-link-profile/nfvo-rel2:virtual-link-descriptor";

              }

            }

            uses link-bitrate-requirements;

          }

          must "(count(nfvo-rel2:vnf-to-level-mapping) + count(nfvo-rel2:vl-to-level-mapping)) > 0";

          list ns-to-level-mapping {

            key "ns-profile";

            leaf ns-profile {

              type leafref {

                path "../../../nfvo-rel2:ns-profile/nfvo-rel2:nsd";

              }

            }

            leaf number-of-instances {

              mandatory true;

              type uint32;

            }

          }

        }

        leaf default-instantiation-level {

          type leafref {

            path "../nfvo-rel2:instantiation-level/nfvo-rel2:id";

          }

        }

        list ns-profile {

          must "nfvo-rel2:min-number-of-instances<=nfvo-rel2:max-number-of-instances";

          key "nsd";

          leaf nsd {

            type leafref {

              path "../../../nfvo-rel2:nested-nsd";

            }

          }

          leaf deployment-flavor {

            type leafref {

              path "/nfvo-rel2:nfvo/nfvo-rel2:nsd[id=current()/../nfvo-rel2:nsd]/nfvo-rel2:deployment-flavor/nfvo-rel2:id";

            }

          }

          leaf instantiation-level {

            type leafref {

              path "deref(../nfvo-rel2:deployment-flavor)/../nfvo-rel2:instantiation-level/nfvo-rel2:id";

            }

          }

          leaf min-number-of-instances {

            type uint16;

            mandatory true;

          }

          leaf max-number-of-instances {

            type uint16;

            mandatory true;

          }

          uses local-affinity-or-anti-affinity-rule-list;

          list affinity-or-anti-affinity-group {

            key "id";

            leaf id {

              type leafref {

                path "../../../nfvo-rel2:affinity-or-anti-affinity-group/nfvo-rel2:id";

              }

            }

          }

        }

        list dependency {

          description

            "NFV Management and Orchestration functions shall instantiate

             VNFs from the VnfProfile and/or nested NSs from the NsProfile

             referenced in the primary attribute before instantiating VNFs

             from the VnfProfile and/or nested NSs from the NsProfile

             referenced in the secondary attribute.";

          key "id";

          leaf id {

            type string;

          }

          choice primary {

            leaf primary-vnf-profile {

              type leafref {

                path "../../nfvo-rel2:vnf-profile/nfvo-rel2:vnfd";

              }

            }

            leaf primary-ns-profile {

              type leafref {

                path "../../nfvo-rel2:ns-profile/nfvo-rel2:nsd";

              }

            }

          }

          choice secondary {

            leaf secondary-vnf-profile {

              type leafref {

                path "../../nfvo-rel2:vnf-profile/nfvo-rel2:vnfd";

              }

            }

            leaf secondary-ns-profile {

              type leafref {

                path "../../nfvo-rel2:ns-profile/nfvo-rel2:nsd";

              }

            }

          }

        }

      }

      uses security-parameters;

    }

  }

}

## EPGW Payloads

**NSD CSR NO HA**

<config xmlns="[http://tail-f.com/ns/config/1.0"](http://tail-f.com/ns/config/1.0)>  
  <nfvo xmlns="[http://tail-f.com/pkg/tailf-etsi-rel2-nfvo"](http://tail-f.com/pkg/tailf-etsi-rel2-nfvo)>  
  <nsd>  
    <id>SAE-NSD-EPGW-CSR-NO-HA</id>  
    <vnfd>  
      <vnfd>SAE-CSR</vnfd>  
    </vnfd>  
     <service-access-point-descriptor>  
      <id>sapd-inside</id>  
      <chain-endpoint xmlns="[http://com/cisco/nso/sae-core-fp-common"](http://com/cisco/nso/sae-core-fp-common)/>  
     </service-access-point-descriptor>  
     <service-access-point-descriptor>  
      <id>sapd-outside</id>  
      <provider-endpoint xmlns="[http://com/cisco/nso/sae-core-fp-common"](http://com/cisco/nso/sae-core-fp-common)/>  
     </service-access-point-descriptor>  
     <service-access-point-descriptor>  
      <id>sapd-extra3</id>  
      <consumer-endpoint xmlns="[http://com/cisco/nso/sae-core-fp-common"](http://com/cisco/nso/sae-core-fp-common)/>  
     </service-access-point-descriptor>  
     <service-access-point-descriptor>  
      <id>sapd-extra4</id>  
      <chain-endpoint xmlns="[http://com/cisco/nso/sae-core-fp-common"](http://com/cisco/nso/sae-core-fp-common)/>  
      </service-access-point-descriptor>  
     <service-access-point-descriptor>  
      <id>sapd-mgmt</id>  
     </service-access-point-descriptor>  
    <deployment-flavor>  
      <id>basic</id>  
      <vnf-profile>  
        <id>csr-NO-HA-profile</id>  
        <vnfd>SAE-CSR</vnfd>  
        <flavor>CSR</flavor>  
        <instantiation-level>SAE-CSR</instantiation-level>  
        <min-number-of-instances>1</min-number-of-instances>  
        <max-number-of-instances>1</max-number-of-instances>  
        <affinity-or-anti-affinity-group>  
          <id>anti-group1</id>  
        </affinity-or-anti-affinity-group>  
        <sapd-connectivity>  
          <sapd>sapd-inside</sapd>  
          <cp>inside</cp>  
        </sapd-connectivity>  
         <sapd-connectivity>  
          <sapd>sapd-outside</sapd>  
          <cp>outside</cp>  
      </sapd-connectivity>  
       <sapd-connectivity>  
          <sapd>sapd-extra3</sapd>  
          <cp>extra3</cp>  
        </sapd-connectivity>  
         <sapd-connectivity>  
          <sapd>sapd-extra4</sapd>  
          <cp>extra4</cp>  
        </sapd-connectivity>  
        <sapd-connectivity>  
         <sapd>sapd-mgmt</sapd>  
         <cp>mgmt</cp>  
        </sapd-connectivity>  
      </vnf-profile>  
      <affinity-or-anti-affinity-group>  
        <id>anti-group1</id>  
        <affinity-type>anti-affinity</affinity-type>  
        <affinity-scope>nfvi-node</affinity-scope>  
      </affinity-or-anti-affinity-group>  
      <instantiation-level>  
        <id>basic</id>  
        <description>basic</description>  
        <vnf-to-level-mapping>  
          <vnf-profile>csr-NO-HA-profile</vnf-profile>  
          <number-of-instances>1</number-of-instances>  
        </vnf-to-level-mapping>  
      </instantiation-level>  
    </deployment-flavor>  
  </nsd>  
  </nfvo>  
</config>

**VNFD CSR NO HA**

<config xmlns="[http://tail-f.com/ns/config/1.0"](http://tail-f.com/ns/config/1.0)>  
  <nfvo xmlns="[http://tail-f.com/pkg/tailf-etsi-rel2-nfvo"](http://tail-f.com/pkg/tailf-etsi-rel2-nfvo)>  
  <vnfd>  
    <id>SAE-EP-GW-CSR</id>  
    <product-name>CSR 1000v</product-name>  
    <provider>Cisco</provider>  
    <version>1.0</version>  
    <product-info-description>Cloud router</product-info-description>  
    <vdu>  
      <id>CSR</id>  
      <internal-connection-point-descriptor>  
        <id>mgmt</id>  
        <external-connection-point-descriptor>mgmt</external-connection-point-descriptor>  
        <layer-protocol>Ethernet</layer-protocol>  
        <virtual-network-interface-requirement>  
          <name>mgmt</name>  
          <support-mandatory/>  
          <requirement>SR-IOV</requirement>  
          <sriov-type xmlns="[http://com/cisco/nso/sae-core-fp-common"](http://com/cisco/nso/sae-core-fp-common)>X520</sriov-type>  
         </virtual-network-interface-requirement>  
        <interface-id xmlns="[http://tail-f.com/pkg/tailf-etsi-rel2-nfvo-esc"](http://tail-f.com/pkg/tailf-etsi-rel2-nfvo-esc)>0</interface-id>  
      </internal-connection-point-descriptor>  
      <internal-connection-point-descriptor>  
        <id>cp1</id>  
        <external-connection-point-descriptor>cp1</external-connection-point-descriptor>  
        <layer-protocol>Ethernet</layer-protocol>  
        <virtual-network-interface-requirement>  
          <name>cp1</name>  
          <support-mandatory/>  
          <requirement>SR-IOV</requirement>  
          <sriov-type xmlns="[http://com/cisco/nso/sae-core-fp-common"](http://com/cisco/nso/sae-core-fp-common)>X520</sriov-type>  
         </virtual-network-interface-requirement>  
        <interface-id xmlns="[http://tail-f.com/pkg/tailf-etsi-rel2-nfvo-esc"](http://tail-f.com/pkg/tailf-etsi-rel2-nfvo-esc)>1</interface-id>  
      </internal-connection-point-descriptor>  
      <internal-connection-point-descriptor>  
        <id>cp2</id>  
        <external-connection-point-descriptor>cp2</external-connection-point-descriptor>  
        <layer-protocol>Ethernet</layer-protocol>  
        <virtual-network-interface-requirement>  
          <name>cp2</name>  
          <support-mandatory/>  
          <requirement>SR-IOV</requirement>  
          <sriov-type xmlns="[http://com/cisco/nso/sae-core-fp-common"](http://com/cisco/nso/sae-core-fp-common)>X520</sriov-type>  
         </virtual-network-interface-requirement>  
        <interface-id xmlns="[http://tail-f.com/pkg/tailf-etsi-rel2-nfvo-esc"](http://tail-f.com/pkg/tailf-etsi-rel2-nfvo-esc)>2</interface-id>  
      </internal-connection-point-descriptor>  
      <internal-connection-point-descriptor>  
        <id>cp3</id>  
        <external-connection-point-descriptor>cp3</external-connection-point-descriptor>  
        <layer-protocol>Ethernet</layer-protocol>  
        <virtual-network-interface-requirement>  
          <name>cp3</name>  
          <support-mandatory/>  
          <requirement>SR-IOV</requirement>  
          <sriov-type xmlns="[http://com/cisco/nso/sae-core-fp-common"](http://com/cisco/nso/sae-core-fp-common)>X520</sriov-type>  
         </virtual-network-interface-requirement>  
        <interface-id xmlns="[http://tail-f.com/pkg/tailf-etsi-rel2-nfvo-esc"](http://tail-f.com/pkg/tailf-etsi-rel2-nfvo-esc)>3</interface-id>  
      </internal-connection-point-descriptor>  
      <internal-connection-point-descriptor>  
        <id>cp4</id>  
        <external-connection-point-descriptor>cp4</external-connection-point-descriptor>  
        <layer-protocol>Ethernet</layer-protocol>  
        <virtual-network-interface-requirement>  
          <name>cp4</name>  
          <support-mandatory/>  
          <requirement>SR-IOV</requirement>  
          <sriov-type xmlns="[http://com/cisco/nso/sae-core-fp-common"](http://com/cisco/nso/sae-core-fp-common)>X520</sriov-type>  
         </virtual-network-interface-requirement>  
        <interface-id xmlns="[http://tail-f.com/pkg/tailf-etsi-rel2-nfvo-esc"](http://tail-f.com/pkg/tailf-etsi-rel2-nfvo-esc)>4</interface-id>  
      </internal-connection-point-descriptor>  
      <internal-connection-point-descriptor>  
        <id>cp5</id>  
        <external-connection-point-descriptor>cp5</external-connection-point-descriptor>  
        <layer-protocol>Ethernet</layer-protocol>  
        <virtual-network-interface-requirement>  
          <name>cp5</name>  
          <support-mandatory/>  
          <requirement>SR-IOV</requirement>  
          <sriov-type xmlns="[http://com/cisco/nso/sae-core-fp-common"](http://com/cisco/nso/sae-core-fp-common)>X520</sriov-type>  
         </virtual-network-interface-requirement>  
        <interface-id xmlns="[http://tail-f.com/pkg/tailf-etsi-rel2-nfvo-esc"](http://tail-f.com/pkg/tailf-etsi-rel2-nfvo-esc)>5</interface-id>  
      </internal-connection-point-descriptor>  
      <internal-connection-point-descriptor>  
        <id>cp6</id>  
        <external-connection-point-descriptor>cp6</external-connection-point-descriptor>  
        <layer-protocol>Ethernet</layer-protocol>  
        <virtual-network-interface-requirement>  
          <name>cp6</name>  
          <support-mandatory/>  
          <requirement>SR-IOV</requirement>  
          <sriov-type xmlns="[http://com/cisco/nso/sae-core-fp-common"](http://com/cisco/nso/sae-core-fp-common)>X520</sriov-type>  
         </virtual-network-interface-requirement>  
        <interface-id xmlns="[http://tail-f.com/pkg/tailf-etsi-rel2-nfvo-esc"](http://tail-f.com/pkg/tailf-etsi-rel2-nfvo-esc)>6</interface-id>  
      </internal-connection-point-descriptor>  
      <internal-connection-point-descriptor>  
        <id>cp7</id>  
        <external-connection-point-descriptor>cp7</external-connection-point-descriptor>  
        <layer-protocol>Ethernet</layer-protocol>  
        <virtual-network-interface-requirement>  
          <name>cp7</name>  
          <support-mandatory/>  
          <requirement>SR-IOV</requirement>  
          <sriov-type xmlns="[http://com/cisco/nso/sae-core-fp-common"](http://com/cisco/nso/sae-core-fp-common)>X520</sriov-type>  
         </virtual-network-interface-requirement>  
        <interface-id xmlns="[http://tail-f.com/pkg/tailf-etsi-rel2-nfvo-esc"](http://tail-f.com/pkg/tailf-etsi-rel2-nfvo-esc)>7</interface-id>  
      </internal-connection-point-descriptor>  
      <virtual-compute-descriptor>vcd</virtual-compute-descriptor>  
      <virtual-storage-descriptor>roottest1</virtual-storage-descriptor>  
      <virtual-storage-descriptor>roottest2</virtual-storage-descriptor>  
      <software-image-descriptor>  
        <name>csr1000v-universalk9.16.06.01.qcow2</name>  
        <container-format>bare</container-format>  
        <disk-format>qcow2</disk-format>  
        <image>http://127.0.0.1:8082/images/csr1000v-universalk9.16.06.01.qcow2</image>  
          <additional-setting xmlns="[http://tail-f.com/pkg/tailf-etsi-rel2-nfvo-esc"](http://tail-f.com/pkg/tailf-etsi-rel2-nfvo-esc)>  
              <id>disk\_type</id>  
              <value>virtio</value>  
          </additional-setting>  
      </software-image-descriptor>  
      <device-type xmlns="[http://tail-f.com/pkg/tailf-etsi-rel2-nfvo-esc"](http://tail-f.com/pkg/tailf-etsi-rel2-nfvo-esc)>  
        <cli>  
          <ned-id>ios-id:cisco-ios</ned-id>  
        </cli>  
      </device-type>  
      <vnf-wait-time xmlns="[http://tail-f.com/pkg/tailf-etsi-rel2-nfvo-esc"](http://tail-f.com/pkg/tailf-etsi-rel2-nfvo-esc)>3</vnf-wait-time>  
      <day0 xmlns="[http://tail-f.com/pkg/tailf-etsi-rel2-nfvo-esc"](http://tail-f.com/pkg/tailf-etsi-rel2-nfvo-esc)>  
        <destination>iosxe\_config.txt</destination>  
        <mandatory/>  
      </day0>  
    </vdu>  
    <virtual-compute-descriptor>  
      <id>vcd</id>  
      <virtual-memory>  
        <virtual-memory-size>4.0</virtual-memory-size>  
      </virtual-memory>  
      <virtual-cpu>  
        <number-of-virtual-cpus>2</number-of-virtual-cpus>  
      </virtual-cpu>  
    </virtual-compute-descriptor>  
    <virtual-storage-descriptor>  
      <id>roottest1</id>  
      <type-of-storage>root</type-of-storage>  
      <size-of-storage>10</size-of-storage>  
    </virtual-storage-descriptor>  
    <virtual-storage-descriptor>  
      <id>roottest2</id>  
      <type-of-storage>root</type-of-storage>  
      <size-of-storage>100</size-of-storage>  
    </virtual-storage-descriptor>  
    <virtual-storage-descriptor>  
      <id>swaptest</id>  
      <type-of-storage>swap</type-of-storage>  
      <size-of-storage>10</size-of-storage>  
    </virtual-storage-descriptor>  
    <external-connection-point-descriptor>  
      <id>mgmt</id>  
      <layer-protocol>Ethernet</layer-protocol>  
      <management xmlns="[http://tail-f.com/pkg/tailf-etsi-rel2-nfvo-esc"](http://tail-f.com/pkg/tailf-etsi-rel2-nfvo-esc)/>  
    </external-connection-point-descriptor>  
    <external-connection-point-descriptor>  
      <id>cp1</id>  
      <layer-protocol>Ethernet</layer-protocol>  
    </external-connection-point-descriptor>  
    <external-connection-point-descriptor>  
      <id>cp2</id>  
      <layer-protocol>Ethernet</layer-protocol>  
    </external-connection-point-descriptor>  
    <external-connection-point-descriptor>  
      <id>cp3</id>  
      <layer-protocol>Ethernet</layer-protocol>  
    </external-connection-point-descriptor>  
    <external-connection-point-descriptor>  
      <id>cp4</id>  
      <layer-protocol>Ethernet</layer-protocol>  
    </external-connection-point-descriptor>  
    <external-connection-point-descriptor>  
      <id>cp5</id>  
      <layer-protocol>Ethernet</layer-protocol>  
    </external-connection-point-descriptor>  
    <external-connection-point-descriptor>  
      <id>cp6</id>  
      <layer-protocol>Ethernet</layer-protocol>  
    </external-connection-point-descriptor>  
    <external-connection-point-descriptor>  
      <id>cp7</id>  
      <layer-protocol>Ethernet</layer-protocol>  
    </external-connection-point-descriptor>  
    <deployment-flavor>  
      <id>CSR</id>  
      <vdu-profile>  
        <vdu>CSR</vdu>  
        <min-number-of-instances>1</min-number-of-instances>  
        <max-number-of-instances>1</max-number-of-instances>  
      </vdu-profile>  
      <instantiation-level>  
        <id>SAE-EP-GW-CSR</id>  
        <vdu-level>  
          <vdu>CSR</vdu>  
          <number-of-instances>1</number-of-instances>  
        </vdu-level>  
      </instantiation-level>  
      <default-instantiation-level>SAE-EP-GW-CSR</default-instantiation-level>  
    </deployment-flavor>  
  </vnfd>  
  </nfvo>  
</config>

**CATALOG CSR NO HA**

<config xmlns="[http://tail-f.com/ns/config/1.0"](http://tail-f.com/ns/config/1.0)>  
  <sae-catalog xmlns="[http://com/cisco/nso/sae-core-fp-cfs"](http://com/cisco/nso/sae-core-fp-cfs)>  
    <name>catalogEPGWNOHA</name>  
    <var>  
      <name>CATALOG\_VAR\_TRIAL</name>  
      <val>cat\_var\_value</val>  
    </var>  
    <var>  
      <name>CATALOG\_VAR\_TRIAL\_2</name>  
      <val>cat\_var\_value\_2</val>  
    </var>  
    <csp>  
      <name>csp-2100-2.2.3.32</name>  
      <day1-config>  
        <name>ntp</name>  
        <cfg-template>csp-ntp-server</cfg-template>  
        <var>  
          <name>NTP\_SERVER</name>  
          <val>127.0.0.1</val>  
        </var>  
      </day1-config>  
      <virtio-network>  
        <name>virt-A</name>  
      </virtio-network>  
      <sriov-networks>  
        <name>sriov-1</name>  
      </sriov-networks>  
      <sriov-networks>  
        <name>sriov-2</name>  
      </sriov-networks>  
      <mgmt-network>mgmt-A</mgmt-network>  
    </csp>  
    <nsd-deployment>  
      <name>NSD-EPGW-NO-HA-CSR-DEPLOYMENT</name>  
      <nsd>SAE-NSD-EPGW-CSR-NO-HA</nsd>  
      <nsd-deployment-flavor>basic</nsd-deployment-flavor>  
      <vnf-deployment>  
        <vnf-profile>csr-NO-HA-profile</vnf-profile>  
        <vnfd-deployment>CSR-DEP</vnfd-deployment>  
      </vnf-deployment>  
   </nsd-deployment>  
   <nsd-deployment>  
      <name>NSD-EPGW-NO-HA-POCH-CSR-DEPLOYMENT</name>  
      <nsd>SAE-NSD-EPGW-CSR-PORTCH-NO-HA</nsd>  
      <nsd-deployment-flavor>basic</nsd-deployment-flavor>  
      <vnf-deployment>  
        <vnf-profile>csr-PORTCH-NO-HA-profile</vnf-profile>  
        <vnfd-deployment>EP-GW-CSR-POCH-DEP</vnfd-deployment>  
      </vnf-deployment>  
   </nsd-deployment>  
      
    <vnfd-deployment>  
      <name>CSR-DEP</name>  
      <vnfd>SAE-CSR</vnfd>  
      <vdu>CSR</vdu>  
      <bootup-time>600</bootup-time>  
      <recovery-wait-time>90</recovery-wait-time>  
      <vnf-authgroup>csr\_authgroup</vnf-authgroup>  
      <day0>  
        <destination>iosxe\_config.txt</destination>  
        <url>http://127.0.0.1:8082/day0/csr.txt</url>  
        <var>  
          <name>HOSTNAME</name>  
          <val>csr\_host</val>  
        </var>  
        <var>  
          <name>PASSWORD</name>  
          <val>cisco123</val>  
        </var>  
      </day0>  
      <day1-config>  
        <name>csr-day1</name>  
        <cfg-template>day1\_csr\_banner</cfg-template>  
        <var>  
          <name>BANNER\_VALUE</name>  
          <val>Hello</val>  
        </var>  
      </day1-config>  
    </vnfd-deployment>  
    <vnfd-deployment>  
      <name>EP-GW-CSR-POCH-DEP</name>  
      <vnfd>SAE-EP-GW-CSR-POCH</vnfd>  
      <vdu>CSR</vdu>  
      <vnf-authgroup>csr\_authgroup</vnf-authgroup>  
      <bootup-time>600</bootup-time>  
      <recovery-wait-time>90</recovery-wait-time>  
      <day0>  
        <destination>iosxe\_config.txt</destination>  
        <url>http://127.0.0.1:8082/day0/ep-gw-csr.txt</url>  
        <var>  
          <name>HOSTNAME</name>  
          <val>csr\_host</val>  
        </var>  
        <var>  
          <name>PASSWORD</name>  
          <val>cisco123</val>  
        </var>  
      </day0>  
      <day1-config>  
        <name>csr-day1-banner</name>  
        <cfg-template>day1\_csr\_banner</cfg-template>  
        <var>  
          <name>BANNER\_VALUE</name>  
          <val>Login Successful</val>  
        </var>  
      </day1-config>  
    </vnfd-deployment>  
    <vnfd-deployment>  
      <name>ESC-DEP</name>  
      <vnfd>SAE-ESC</vnfd>  
      <vdu>ESC</vdu>  
      <vnf-authgroup>esc\_authgroup</vnf-authgroup>  
      <ssh-authgroup>default</ssh-authgroup>  
      <day0>  
        <destination>user-data</destination>  
        <url>http://127.0.0.1:8082/day0/esc-day0.txt</url>  
        <var>  
          <name>ESC\_VIP\_ADDRESS</name>  
          <val>10.245.188.4</val>  
        </var>  
        <var>  
          <name>ESC\_VIP\_NODE\_2\_IP</name>  
          <val>10.245.188.3</val>  
        </var>  
        <var>  
          <name>PASSWORD</name>  
          <val>$6$rounds=4096$adWA0K3AZ30Pe$EHUa2Z7o/Rvhz8Pk2eET5Rhto3G7MmKjUef.AyOJNtBfRo0fUapoxmFd7LUn2PEUPWtWPl5tCD7pO9bae672T1</val>  
        </var>  
        <var>  
          <name>SSH\_PASSWORD</name>  
          <val>xy7gqo67gn3nM</val>  
        </var>  
        <var>  
          <name>VRI\_ID</name>  
          <val>138</val>  
        </var>  
      </day0>  
  </vnfd-deployment>  
  </sae-catalog>  
</config>

**SAE SITE PAYLOAD CSR NO HA**

<config xmlns="[http://tail-f.com/ns/config/1.0"](http://tail-f.com/ns/config/1.0)>  
  <sae-site xmlns="[http://com/cisco/nso/sae-core-fp-cfs"](http://com/cisco/nso/sae-core-fp-cfs)>  
    <site>sae-site-1</site>  
    <sae-provider>sae-provider-admin</sae-provider>  
    <var>  
      <name>SAE\_VAR\_1</name>  
      <val>sae\_var\_1\_val</val>  
    </var>  
    <var>  
      <name>SAE\_VAR\_2</name>  
      <val>sae\_var\_2\_val</val>  
    </var>  
    <external-end-point>  
      <name>external-ep1</name>  
      <traffic-hand-off-method xmlns:cisco-sae-core-fp-common="[http://com/cisco/nso/sae-core-fp-common"](http://com/cisco/nso/sae-core-fp-common)>cisco-sae-core-fp-common:vlan-only</traffic-hand-off-method>  
      <vlan-number>1001</vlan-number>  
      <branch-bgp-as-number>420000013</branch-bgp-as-number>  
      <branch-remote-as-number>420000014</branch-remote-as-number>  
      <day1-config>  
        <name>epgw-vlan</name>  
        <cfg-template>epgw-vlan</cfg-template>  
        <var>  
          <name>VLAN\_NUMBER</name>  
          <val>56</val>  
        </var>  
        </day1-config>  
      <!--  No HA day1 variable application in day1 template  
      <day1-config>  
       <name>day1-interface-vrf-mgmt</name>  
        <cfg-template>day1-interface-vrf-mgmt</cfg-template>  
        <var>  
          <name>INTERFACE\_ID</name>  
          <val>0</val>  
        </var>  
      </day1-config> -->  
    </external-end-point>  
    <external-end-point>  
      <name>external-ep2</name>  
      <traffic-hand-off-method xmlns:cisco-sae-core-fp-common="[http://com/cisco/nso/sae-core-fp-common"](http://com/cisco/nso/sae-core-fp-common)>cisco-sae-core-fp-common:vlan-only</traffic-hand-off-method>  
      <vlan-number>2001</vlan-number>  
      <branch-bgp-as-number>420000015</branch-bgp-as-number>  
      <branch-remote-as-number>420000016</branch-remote-as-number>  
      <day1-config>  
        <name>epgw-vlan</name>  
        <cfg-template>epgw-vlan</cfg-template>  
        <var>  
          <name>VLAN\_NUMBER</name>  
          <val>56</val>  
        </var>  
      </day1-config>  
    </external-end-point>  
    <external-end-point>  
      <name>external-ep3</name>  
      <traffic-hand-off-method xmlns:cisco-sae-core-fp-common="[http://com/cisco/nso/sae-core-fp-common"](http://com/cisco/nso/sae-core-fp-common)>cisco-sae-core-fp-common:vlan-only</traffic-hand-off-method>  
      <vlan-number>3001</vlan-number>  
      <day1-config>  
        <name>epgw-vlan</name>  
        <cfg-template>epgw-vlan</cfg-template>  
        <var>  
          <name>VLAN\_NUMBER</name>  
          <val>56</val>  
        </var>  
      </day1-config>  
    </external-end-point>  
    <external-end-point>  
      <name>external-ep4</name>  
      <traffic-hand-off-method xmlns:cisco-sae-core-fp-common="[http://com/cisco/nso/sae-core-fp-common"](http://com/cisco/nso/sae-core-fp-common)>cisco-sae-core-fp-common:vlan-only</traffic-hand-off-method>  
      <vlan-number>4001</vlan-number>  
      <branch-bgp-as-number>420000034</branch-bgp-as-number>  
      <branch-remote-as-number>420000035</branch-remote-as-number>  
      <day1-config>  
        <name>epgw-vlan</name>  
        <cfg-template>epgw-vlan</cfg-template>  
        <var>  
          <name>VLAN\_NUMBER</name>  
          <val>56</val>  
        </var>  
      </day1-config>  
    </external-end-point>  
    <external-end-point>  
      <name>external-ep5</name>  
      <traffic-hand-off-method xmlns:cisco-sae-core-fp-common="[http://com/cisco/nso/sae-core-fp-common"](http://com/cisco/nso/sae-core-fp-common)>cisco-sae-core-fp-common:vlan-with-poch</traffic-hand-off-method>  
      <vlan-number>3100</vlan-number>  
      <branch-bgp-as-number>420000036</branch-bgp-as-number>  
      <branch-remote-as-number>420000037</branch-remote-as-number>  
    </external-end-point>  
    <external-end-point>  
      <name>external-ep6</name>  
      <traffic-hand-off-method xmlns:cisco-sae-core-fp-common="[http://com/cisco/nso/sae-core-fp-common"](http://com/cisco/nso/sae-core-fp-common)>cisco-sae-core-fp-common:vlan-with-poch</traffic-hand-off-method>  
      <vlan-number>3200</vlan-number>  
      <branch-bgp-as-number>420000038</branch-bgp-as-number>  
      <branch-remote-as-number>420000039</branch-remote-as-number>  
    </external-end-point>  
    <resource-pools>  
      <mgmt-ip-pool>mgmt-network-pool</mgmt-ip-pool>  
      <internal-ip-pool>local-network-pool</internal-ip-pool>  
      <as-pool>default-as-pool</as-pool>  
    </resource-pools>  
    <infrastructure>  
      <switching>  
        <type xmlns:cisco-sae-core-fp-common="[http://com/cisco/nso/sae-core-fp-common"](http://com/cisco/nso/sae-core-fp-common)>cisco-sae-core-fp-common:n9k-switch-pair</type>  
      </switching>  
      <compute-clusters>  
        <name>cluster-1</name>  
        <vlan-pool>sae-vlan-pool</vlan-pool>  
        <leaf-switches>  
          <name>nexus0</name>  
        </leaf-switches>  
        <leaf-switches>  
          <name>nexus1</name>  
        </leaf-switches>  
        <servers>  
          <name>csp0</name>  
          <type>csp-2100-2.2.3.32</type>  
          <virtio-network>  
            <pc-name>virt-A</pc-name>  
            <pnics>  
              <name>enp10s0f0</name>  
              <connected-switch>nexus0</connected-switch>  
              <switch-interface-name>port-channel0</switch-interface-name>  
              <bandwidth>4000</bandwidth>  
            </pnics>  
            <pnics>  
              <name>enp10s0f1</name>  
              <connected-switch>nexus1</connected-switch>  
              <switch-interface-name>port-channel1</switch-interface-name>  
              <bandwidth>4000</bandwidth>  
            </pnics>  
          </virtio-network>  
          <sriov-networks>  
            <name>sriov-1</name>  
            <connected-switch>nexus0</connected-switch>  
            <switch-interface-name>Ethernet1/1</switch-interface-name>  
            <bandwidth>4000</bandwidth>  
            <sriov-type xmlns:cisco-sae-core-fp-common="[http://com/cisco/nso/sae-core-fp-common"](http://com/cisco/nso/sae-core-fp-common)>cisco-sae-core-fp-common:X520</sriov-type>  
          </sriov-networks>  
          <sriov-networks>  
            <name>sriov-2</name>  
            <connected-switch>nexus1</connected-switch>  
            <switch-interface-name>Ethernet1/2</switch-interface-name>  
            <bandwidth>4000</bandwidth>  
            <sriov-type xmlns:cisco-sae-core-fp-common="[http://com/cisco/nso/sae-core-fp-common"](http://com/cisco/nso/sae-core-fp-common)>cisco-sae-core-fp-common:X520</sriov-type>  
          </sriov-networks>  
          <var>  
            <name>NTP\_SERVER</name>  
            <val>10.10.0.0</val>  
          </var>  
        </servers>  
        <servers>  
          <name>csp1</name>  
          <type>csp-2100-2.2.3.32</type>  
          <virtio-network>  
            <pc-name>virt-A</pc-name>  
            <pnics>  
              <name>enp10s0f0</name>  
              <connected-switch>nexus0</connected-switch>  
              <switch-interface-name>port-channel2</switch-interface-name>  
              <bandwidth>4000</bandwidth>  
            </pnics>  
            <pnics>  
              <name>enp10s0f1</name>  
              <connected-switch>nexus1</connected-switch>  
              <switch-interface-name>port-channel3</switch-interface-name>  
              <bandwidth>4000</bandwidth>  
            </pnics>  
          </virtio-network>  
          <sriov-networks>  
            <name>sriov-1</name>  
            <connected-switch>nexus0</connected-switch>  
            <switch-interface-name>Ethernet1/3</switch-interface-name>  
            <bandwidth>4000</bandwidth>  
            <sriov-type xmlns:cisco-sae-core-fp-common="[http://com/cisco/nso/sae-core-fp-common"](http://com/cisco/nso/sae-core-fp-common)>cisco-sae-core-fp-common:X520</sriov-type>  
          </sriov-networks>  
          <sriov-networks>  
            <name>sriov-2</name>  
            <connected-switch>nexus1</connected-switch>  
            <switch-interface-name>Ethernet1/4</switch-interface-name>  
            <bandwidth>4000</bandwidth>  
            <sriov-type xmlns:cisco-sae-core-fp-common="[http://com/cisco/nso/sae-core-fp-common"](http://com/cisco/nso/sae-core-fp-common)>cisco-sae-core-fp-common:X520</sriov-type>  
          </sriov-networks>  
        </servers>  
        <servers>  
          <name>csp2</name>  
          <type>csp-2100-2.2.3.32</type>  
          <virtio-network>  
            <pc-name>virt-A</pc-name>  
            <pnics>  
              <name>enp10s0f0</name>  
              <connected-switch>nexus0</connected-switch>  
              <switch-interface-name>port-channel4</switch-interface-name>  
              <bandwidth>4000</bandwidth>  
            </pnics>  
            <pnics>  
              <name>enp10s0f1</name>  
              <connected-switch>nexus1</connected-switch>  
              <switch-interface-name>port-channel5</switch-interface-name>  
              <bandwidth>4000</bandwidth>  
            </pnics>  
          </virtio-network>  
          <sriov-networks>  
            <name>sriov-1</name>  
            <connected-switch>nexus0</connected-switch>  
            <switch-interface-name>Ethernet1/5</switch-interface-name>  
            <bandwidth>4000</bandwidth>  
            <sriov-type xmlns:cisco-sae-core-fp-common="[http://com/cisco/nso/sae-core-fp-common"](http://com/cisco/nso/sae-core-fp-common)>cisco-sae-core-fp-common:X520</sriov-type>  
          </sriov-networks>  
          <sriov-networks>  
            <name>sriov-2</name>  
            <connected-switch>nexus1</connected-switch>  
            <switch-interface-name>Ethernet1/6</switch-interface-name>  
            <bandwidth>4000</bandwidth>  
            <sriov-type xmlns:cisco-sae-core-fp-common="[http://com/cisco/nso/sae-core-fp-common"](http://com/cisco/nso/sae-core-fp-common)>cisco-sae-core-fp-common:X520</sriov-type>  
          </sriov-networks>  
          <var>  
            <name>NTP\_SERVER</name>  
            <val>10.10.0.0</val>  
          </var>  
        </servers>  
        <servers>  
          <name>csp3</name>  
          <type>csp-2100-2.2.3.32</type>  
          <virtio-network>  
            <pc-name>virt-A</pc-name>  
            <pnics>  
              <name>enp10s0f0</name>  
              <connected-switch>nexus0</connected-switch>  
              <switch-interface-name>port-channel6</switch-interface-name>  
              <bandwidth>4000</bandwidth>  
            </pnics>  
            <pnics>  
              <name>enp10s0f1</name>  
              <connected-switch>nexus1</connected-switch>  
              <switch-interface-name>port-channel7</switch-interface-name>  
              <bandwidth>4000</bandwidth>  
            </pnics>  
          </virtio-network>  
          <sriov-networks>  
            <name>sriov-1</name>  
            <connected-switch>nexus0</connected-switch>  
            <switch-interface-name>Ethernet1/1</switch-interface-name>  
            <bandwidth>4000</bandwidth>  
            <sriov-type xmlns:cisco-sae-core-fp-common="[http://com/cisco/nso/sae-core-fp-common"](http://com/cisco/nso/sae-core-fp-common)>cisco-sae-core-fp-common:X520</sriov-type>  
          </sriov-networks>  
          <sriov-networks>  
            <name>sriov-2</name>  
            <connected-switch>nexus1</connected-switch>  
            <switch-interface-name>Ethernet1/2</switch-interface-name>  
            <bandwidth>4000</bandwidth>  
            <sriov-type xmlns:cisco-sae-core-fp-common="[http://com/cisco/nso/sae-core-fp-common"](http://com/cisco/nso/sae-core-fp-common)>cisco-sae-core-fp-common:X520</sriov-type>  
          </sriov-networks>  
          <var>  
            <name>NTP\_SERVER</name>  
            <val>10.10.0.0</val>  
          </var>  
        </servers>  
        <servers>  
          <name>csp4</name>  
          <type>csp-2100-2.2.3.32</type>  
          <virtio-network>  
            <pc-name>virt-A</pc-name>  
            <pnics>  
              <name>enp10s0f0</name>  
              <connected-switch>nexus0</connected-switch>  
              <switch-interface-name>port-channel8</switch-interface-name>  
              <bandwidth>4000</bandwidth>  
            </pnics>  
            <pnics>  
              <name>enp10s0f1</name>  
              <connected-switch>nexus1</connected-switch>  
              <switch-interface-name>port-channel9</switch-interface-name>  
              <bandwidth>4000</bandwidth>  
            </pnics>  
          </virtio-network>  
          <sriov-networks>  
            <name>sriov-1</name>  
            <connected-switch>nexus0</connected-switch>  
            <switch-interface-name>Ethernet1/3</switch-interface-name>  
            <bandwidth>4000</bandwidth>  
            <sriov-type xmlns:cisco-sae-core-fp-common="[http://com/cisco/nso/sae-core-fp-common"](http://com/cisco/nso/sae-core-fp-common)>cisco-sae-core-fp-common:X520</sriov-type>  
          </sriov-networks>  
          <sriov-networks>  
            <name>sriov-2</name>  
            <connected-switch>nexus1</connected-switch>  
            <switch-interface-name>Ethernet1/1</switch-interface-name>  
            <bandwidth>4000</bandwidth>  
            <sriov-type xmlns:cisco-sae-core-fp-common="[http://com/cisco/nso/sae-core-fp-common"](http://com/cisco/nso/sae-core-fp-common)>cisco-sae-core-fp-common:X520</sriov-type>  
          </sriov-networks>  
        </servers>  
      </compute-clusters>  
      <vnf-manager>  
        <name>unmanaged-trial</name>  
        <device-on-boarding>nso-unmanaged</device-on-boarding>  
        <vnfm-device-name>esc0</vnfm-device-name>  
      </vnf-manager>  
    </infrastructure>  
  </sae-site>  
</config>