# **HP NFV Director**



## **HP NFV Director**

# Version 3.0

# **On-boarding Guide**

Edition: 1.0

For Red Hat Enterprise Linux Server 6.6

June 20155

© Copyright 2015 Hewlett-Packard Development Company, L.P.

# **Legal Notices**

#### Warranty

The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

#### License Requirement and U.S. Government Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

#### **Copyright Notices**

© Copyright 2015 Hewlett-Packard Development Company, L.P.

#### **Trademark Notices**

Adobe<sup>®</sup>, Acrobat<sup>®</sup> and PostScript<sup>®</sup> are trademarks of Adobe Systems Incorporated.

HP-UX Release 10.20 and later and HP-UX Release 11.00 and later (in both 32 and 64-bit configurations) on all HP 9000 computers are Open Group UNIX 95 branded products.

Java<sup>™</sup> is a trademark of Oracle and/or its affiliates.

Microsoft<sup>®</sup>, Internet Explorer, Windows<sup>®</sup>, Windows Server 2007<sup>®</sup>, Windows XP<sup>®</sup>, and Windows 7<sup>®</sup> are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Firefox<sup>®</sup> is a registered trademark of the Mozilla Foundation.

Google Chrome<sup>®</sup> is a trademark of Google Inc.

Oracle<sup>®</sup> is a registered U.S. trademark of Oracle Corporation, Redwood City, California.

EnterpriseDB<sup>®</sup> is a registered trademark of EnterpriseDB.

Postgres Plus® Advanced Server is a registered U.S. trademark of EnterpriseDB.

UNIX<sup>®</sup> is a registered trademark of The Open Group.

X/Open<sup>®</sup> is a registered trademark, and the X device is a trademark of X/Open Company Ltd. in the UK and other countries.

Red Hat<sup>®</sup> is a registered trademark of the Red Hat Company.

Linux<sup>®</sup> is a registered trademark of Linus Torvalds in the U.S. and other countries.

Neo4j is a trademark of Neo Technology.

# Contents

Legal Notice	es2
Contents	3
Figures	4
Preface	5
In this	s Guide5
Audie	nce5
Τγρο	graphical Conventions5
Chapter 1	6
	nee Architecture
E 1 51 Relere	
1.1	Managing VNFs
1.1.1	Indirect VNFM
1.1.2	Direct VNFM7
Chapter 2	8
Intograting (	a VNE into NEV Director
	Embedded VNE Manager
2.1	Ellibedded VINF Mallagel
2.2	Step and VNF must run on desired VIM or Humaniaar
2.3	Step they Clarify what peode to be medaled
2.4	Step two. Clarify what needs to be modeled
2.4.1	Madeling questions
2.4.2	Monitoring questions
2.4.3	Monitoring questions
2.4.4	Ad noc processing questions
2.4.5	License questions
2.5	Step two: writing a VNF descriptor
2.5.1	
2.5.2	VINF Component Level
2.5.3	Virtual machine level
2.6	Step three: configuring monitors
2.6.1	Example Configuring Monitors
2.7	Step iour: Creating custom monitors
2.8	Step five: Integrating custom alarm feeds
2.9	Step six: Creating Custom Correlation rules
2.9.1	Specific Definitions for NEVD
2.9.2	Inventory data populated in NEVD value pack inventory
2.9.3	NFVD/Parameters

# **Figures**

Figure 1 ETSI Reference Architecture	6
Figure 2 Managing VNFs	7
Figure 3 Indirect VNFM	7
Figure 4 Direct VNFM	7
Figure 5 VNF Integration Touch Points	8
Figure 6 VNF Descriptor	11
Figure 7 VNF Director Configurable GUI	12
Figure 8 Example XML template of VNF Level	12
Figure 9 Example XML Template of VNF Component Level	14
Figure 10 Example XML Template of Virtual Machine Level	16
Figure 11 Configure Monitors	19
Figure 12 SiteScope NFVD Templates	23
Figure 13 SiteScope NFVD Monitor Template Parameters	24
Figure 14 SiteScope Alerts Definition 1	24
Figure 15 SiteScope Alerts Definition 2	24
Figure 16 SiteScope Alerts configuration	25
Figure 17: SiteScope: Create custom templates	27
Figure 18: SiteScope: new template	27
Figure 19: SiteScope: new group	28
Figure 20: SiteScope: new Monitor	28
Figure 21: SiteScope: new Variable	29
Figure 22: SiteScope: new Variable details	29
Figure 23: SiteScope: enter variable to associate with template	30
Figure 24: SiteScope: hierarchy of custom template	30
Figure 25 UCA's HPSA Framework	33
Figure 26 HPSA Inventory – UCA/Parameters – Global Parameters	33
Figure 27 HPSA Inventory - UCA/Parameters Workflow Templates	33
Figure 28 UCA Automation Foundation Inventory – UCA/Services	34
Figure 29 UCA Automation Foundation Inventory – UCA/Services	34
Figure 30 UCA Automation Foundation Inventory – UCA/ActionFramework (Problems)	35
Figure 31 UCA Automation Foundation Inventory – UCA/Parameters	35
Figure 32 UCA Automation Monitoring Screen	36
Figure 33 HPSA UCA NFVD Workflows	36
Figure 34 NFVD Inventory – NFVD/Parameters	36
Figure 35 NFVD/Parameters	37
Figure 36 Creating new Workflow Templates	37
Figure 37 View Workflow Templates	37

# Preface

## In this Guide

This guide describes the NFV Director on boarding process.

## **Audience**

Here are some recommendations based on the possible reader profile:

- Solution Developers
- System Integrators

## **Typographical Conventions**

Courier Font:

- Source code and examples of file contents.
- Commands that you enter on the screen.
- Pathnames
- Keyboard key names

Italic Text:

- Filenames, programs and parameters.
- The names of other documents referenced in this manual.

#### Bold Text:

• To introduce new terms and to emphasize important words.

# **Chapter 1**

# **ETSI Reference Architecture**

ETSI reference is well known in the in the NFV industry where it describes the functional decomposition of the NFV ecosystem.

ETSI architecture is NOT an implementation diagram and so it happens on the market that the same set of functions are still addressed in different ways and with different set of HW and SW assets depending on the approach taken.



- Execution reference points Source: ETSI GS NFV 002 V1.1.1 (2013-10)
- Main reference points

Other reference points

**Figure 1 ETSI Reference Architecture** 

## **1.1 Managing VNFs**

The VNF Manager (VNFM) is responsible to manage the lifecycle of a VNF. The VNFM can be generic, often bundled as part of the NFV Orchestrator (NFVO), or specific to the VNF, often bundled with the element manager (EM).

The interactions between the VNFM, NFVO, and Virtaulised Infrastructure Manager (VIM) can take one of two forms: indirect interactions from VNFM to VIM and direct interactions from VNFM to VIM.

-	OSS/BSS		Orchestrator —
Servi Virtual EM1 VNF1	te, VNF and Infrast. Network Functions EM; 2 VNF 2	ructure Description (VNF5) EM 3 VNF 3	VNF Manager(s)
Virtual compute	/Infrastructure (NF Virtual storage	Vi) Virtual network	estration
	Virtualization layer	r .	Virtualised Infrastructure Manager(s)
Compute	Storage lardware resource	Network	

Generic VNF manager coupled with NFV orchestrator

OSS/BSS Orchestrator Virtual Network Functions (WHS) EM 1 EM 2 WHF 3 VNF 1 VNF 2 VHF 3 VNF 3 VHF 3 VNF 3 VHF 3 VNF 4 VHF 3 VIrtual Sector (NFVI) Sector (NFVI) Virtual Sector (NFVI) VII Secto



#### **Figure 2 Managing VNFs**

An NFV Orchestrator has two main modes of operation

#### 1.1.1 Indirect VNFM

- 1. The VNF manager asks NFV Orchestrator to perform every action (create VM, scale, VNF, monitor, etc.)
- 2. NFV Orchestrator informs the VNF manager when each operation finishes
- 3. NFV Orchestrator has the capacity consumption detail of each component



#### **Figure 3 Indirect VNFM**

#### 1.1.2 Direct VNFM

- 1. The VNF manager asks NFV Orchestrator only for permission to consume resources; the response may include a tenant a potentially a target set of resources
- 2. The VNF managers performs every action (create VM, scale, VNF, monitor, etc.)
- 3. VNF manager informs NFV Orchestrator when each operation finishes
- 4. NFV Orchestrator would have the capacity consumption detail of each component ONLY if the VNF manager informs NFV Orchestrator



**Figure 4 Direct VNFM** 

# Chapter 2

# **Integrating a VNF into NFV Director**

## 2.1 Embedded VNF Manager

HP NFVD Director has a generic VNF Manager embedded that allows it to manage the lifecycle of any VNF

## 2.2 Touch Points for VNF integration

When integrating with NFVD director the main touch points are:

- Fulfillment Module of NFV Director
  - Where the VNF Descriptor is modeled
  - Where fulfillment integration with VNFM Managers is configured
- Assurance Module of NFV Director
  - Where VNF monitors are defined
  - Where Monitors Correlation rules are defines
  - Where assurance integration with VNFM Managers is configured
  - VIM/HYPERVISOR
    - Where images are deployed and must rum



#### **Figure 5 VNF Integration Touch Points**

## 2.3 Step one: VNF must run on desired VIM or Hypervisor

The Images provided by the provider of the VNF must run on the target server and virtual infrastructure (either a VIM or an HYPERVISOR)

The images must be loaded in the VIM or HYPERVISOR and proper flavors created on the VIM as part of this process

Note

OpenStack is the most common VIM today so most of the VNFs must be tested to run in an OpenStack environment and not just on a Hypervisor (like VMware or KVM)

As part of the HP OpenNFV initiative HP provides access to a certification program where any company can certify their Virtual Network functions to be compatible with HP HW, HP VIM and HP Orchestrator

## 2.4 Step two: Clarify what needs to be modeled

In order to write a proper VNF descriptor and model the desired behavior in NFV Director, a set of questions must be clarified with the VNF provider.

#### Note

The same VNF can have multiple ways of being deploy (with HA, without HA, in cluster, everything on a single VM, etc...) in order to address all the VNF flavors, NFV Director uses the concept of templates where each VNF flavor can be modeled as a separate template.

### 2.4.1 Infrastructure questions

- Are images of the VNF able to run in an OpenStack environment or do those need VMware or KVM directly?
- How many images (vmdk or qcow2) are needed?
- Which is the guest Operating System?
- Are there any specific network requirements?
  - o For example, dedicated bandwidth
- Does it matter in which server each VMs runs? Or there are dependencies like:
  - All VMs must be running in the same server
  - o All the VMs MUST run in different server
  - $\circ~$  All the VMs need to be connected to the same network
- Are there any limitations if the VMs are in servers in different datacenters?
- To how many networks does each VM need to connect?
- Are there any specific requirements for any of the VMs?
  - For example, DPDK or 10GB of dedicated bandwidth

#### 2.4.2 Modeling questions

- How many components does the VNF have?
- How many Virtual machines does each VNF component have?

- What resources (vcores, vRAM, vdisk and vport) does each virtual machine require?
- Does the VNF have a manager?
- Is the VNF using any physical components?
  - Like a VNFM or shared DB
- How the VMs connect between them?
- What are the external network requirements?
- How does the VNF scale
  - In/out, up/down or both
- How many VNFs are planned?
- What is the simplest deployment? (minimum number of VMs and simplest configuration)
- What is one of the more complex deployments? (maximum number of VMs and more complex configuration)

### 2.4.3 Monitoring questions

- What things need be monitored at the guest OS level?
  - Running processes, output traffic, specific measures, ...
- What things need be monitor at the VM level?
  - o vCPU, vRAM, vdisk, virtual port, ...
- What things need be monitor at the application level?
  - Number of calls, number of I/O, physical port, jitter, ...
- What are the thresholds for each monitor?
  - $\circ~$  For example, 80% of CPU or RAM at 50% and at the same time vCPU 50%
- What are the monitors that are required apart from out-of-the box monitors provided by NFVD (refer to section 3.4 of NFVD integrators guide).
- Are there any action that require derived metrics (combining multiple monitored values)
  - $\circ$  For example, trigger scale up only if 80% CPU at 2 VMs in a VNF
- What are the actions to trigger for each threshold violation
  - For example, trigger scale out at 80% of CPU
- What are the actions that needs to be automated (scenarios)
  - o For example, perform some tests after a new VNF is created
- Are there any alarms that needs to be listened which contributes to any automated action
  - $\circ$  For example, alarms from VIM, Hypervisor, VNFM

### 2.4.4 Ad hoc processing questions

- What needs to be done before or after deploying a new VNF?
- What needs to be done before or after deploying a new VNF component?
- What needs to be done before or after starting a virtual machine?
  - For example, start the services, request EMS to configure, or get licenses

- Can the VMs belonging to the VNF scale out or it can they only scale up
  - Does the VNF can allow restart of a VM when scaling up/down (default OpenStack behaviors is to restart for each VM flavor change)

### 2.4.5 License questions

- Do the VNF need any license keys to run or it is the license just a commercial agreement?
- If license keys are required how are they acquired and deployed?
- Are demo licenses available (even if functional restrictions apply)?
- Is there any license agreement with HP to demo to HP customers that VNF or do we need to request a license each time we want to demo to customers
- Do the virtual machines need Internet access to fully configure?

## 2.5 Step two: Writing a VNF descriptor

NFV Director allows to define a VNF with the following structure (following ETSI standards) using XML format



#### **Figure 6 VNF Descriptor**

One of the multiple potential representations of that structure using NFV Director configurable GUI is the following:



#### Figure 7 VNF Director Configurable GUI

### 2.5.1 VNF Level

A VNF is a container that will define

- How many components the VNF has
  - Example 1: a Routing component and a Security component
  - Example 2: a frontend component and a backend component
- If there is any policy that apply to the entire VNF:
  - Example 1; A policy to set up some attributes of the VNF
  - Example 2: A policy to execute certain actions

#### 2.5.1.1 Example XML template of VNF Level

If we follow NFV Director XML notation we can express a VNF Containing just one Component as following



#### Figure 8 Example XML template of VNF Level





### 2.5.2 VNF Component Level

A VNF Component is a container that will define

- How many and what type of virtual machines are needed for that component
  - Example 1 (two component shares a VM): a Routing component need 1 VM and a Security component runs on the same machine of the routing component
  - Example 2 (each component has different set of VMs): a frontend component need 1 VM and a backend component need 2 VMs
- If there is any policy that apply to the VNF component
  - o Example 1: A policy to set up some attributes of the VNF Component
  - o Example 2: A policy to define the scaling factors for the VNF Component

#### 2.5.2.1 Example XML template of VNF Component Level

If we follow NFV Director XML notation, we can express a VNF Component containing just one VM with a policy that defines the number of those and the scaling in/out policy for that VM



#### Figure 9 Example XML Template of VNF Component Level

```
<templates xmlns="http://www.model.bll.nfv.activator.ov.hp.com">
<languageCode xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema"</pre>
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">en EN</languageCode>
<artifactTemplates>
       <artifactTemplate>
               <artifactFamily>VNF COMPONENT</artifactFamily>
               <artifactCategory>GENERIC</artifactCategory>
               <categories>
                       <category>
                              <label>GENERAL</label>
                              <version>1</version>
                              <order>1</order>
                              <attributes>
                                      <attribute>
                                              <label>Name</label>
                                              <type>
                                                     <label>TEXT</label>
                                              </type>
                                              <unit>TEXT</unit>
                                              <mandatory>false</mandatory>
                                              <order>1</order>
<value>ROUTING COMPONENT 2.0</value>
                                      </attribute>
                              </attributes>
                              <categories/>
                       </category>
               </categories>
               <id>ROUTING COMPONENT 2.0</id>
               <identifier>ROUTING COMPONENT 2.0</identifier>
               <status>
                       <label>ENABLED</label>
                       <visibleStatusLabel>ENABLED</visibleStatusLabel>
                       <enabled>true</enabled>
               </status>
       </artifactTemplate>
</artifactTemplates>
<relationshipTemplates>
       <relationshipTemplate>
               <type>INCLUDE</type>
               <parentId>ROUTING_COMPONENT_2.0/parentId>
               <parentArtifactFamily>VNF_COMPONENT</parentArtifactFamily>
               <parentArtifactCategory>GENERIC</parentArtifactCategory>
               <childId>VSR VM 2.0</childId>
```

```
<childArtifactFamily>VIRTUAL MACHINE</childArtifactFamily>
               <childArtifactCategory>GENERIC</childArtifactCategory>
               <enabled>true</enabled>
               <status>
                      <label>ENABLED</label>
                      <visibleStatusLabel>ENABLED</visibleStatusLabel>
                       <enabled>true</enabled>
               </status>
       </relationshipTemplate>
       <relationshipTemplate>
               <type>APPLY</type>
               <parentId>ROUTING COMPONENT 2.0/parentId>
               <parentArtifactFamily>VNF COMPONENT</parentArtifactFamily>
               <parentArtifactCategory>GENERIC</parentArtifactCategory>
               <childId>VSR SCALE POLICY 2.0</childId>
               <childArtifactFamily>POLICY</childArtifactFamily>
               <childArtifactCategory>ENTITY RANGE</childArtifactCategory>
               <enabled>true</enabled>
               <status>
                      <label>ENABLED</label>
                      <visibleStatusLabel>ENABLED</visibleStatusLabel>
                      <enabled>true</enabled>
               </status>
       </relationshipTemplate>
</relationshipTemplates>
</templates>
```

## 2.5.3 Virtual machine level

A Virtual machine indicates the attributes of the virtual machine:

- How many virtual cores, virtual RAM and virtual disk are needed for that component
  - Example 1: One VM need 2 cores and 5 GB and another 10 cores and 500Gb of RAM
- If there is any policy that apply to the VM:
  - o Example 1; A policy to set up some attributes of the VM
  - $\circ~$  Example 2: A policy to execute certain actions before or after the creation of the VM, such as
    - Get licenses
    - Start service
    - Configure the VM before starting services

#### 2.5.3.1 Example XML template of virtual machine level

If we follow NFV Director XML notation we can express a virtual machine, using a specified number of virtual cores, RAM, and disk, defining a policy that will drive the cardinality of the VM at creation and at scaling in/out time.



#### Figure 10 Example XML Template of Virtual Machine Level



```
<attributes>
                                      <attribute>
                                             <label>Name</label>
                                             <tvpe>
                                                     <label>TEXT</label>
                                              </type>
                                              <unit>TEXT</unit>
                                              <mandatory>false</mandatory>
                                             <order>1</order>
<value>VSR SCALE POLICY 2.0</value>
                                      </attribute>
                              </attributes>
                              <categories/>
                      </category>
               </categories>
               <id>VSR SCALE POLICY 2.0</id>
               <identifier>VSR SCALE POLICY 2.0</identifier>
               <status>
                       <label>ENABLED</label>
                      <visibleStatusLabel>ENABLED</visibleStatusLabel>
                       <enabled>true</enabled>
               </status>
       </artifactTemplate>
</artifactTemplates>
<relationshipTemplates>
       <relationshipTemplate>
               <type>USES</type>
               <parentId>VSR VM 2.0</parentId>
               <parentArtifactFamily>VIRTUAL MACHINE</parentArtifactFamily>
               <parentArtifactCategory>GENERIC</parentArtifactCategory>
               <childId>VSR vCORE 2.0</childId>
               <childArtifactFamily>VIRTUAL CORE</childArtifactFamily>
               <childArtifactCategory>GENERIC</childArtifactCategory>
               <enabled>true</enabled>
               <status>
                      <label>ENABLED</label>
                      <visibleStatusLabel>ENABLED</visibleStatusLabel>
                      <enabled>true</enabled>
               </status>
       </relationshipTemplate>
       <relationshipTemplate>
               <type>USES</type>
               <parentId>VSR VM 2.0</parentId>
               <parentArtifactFamily>VIRTUAL MACHINE</parentArtifactFamily>
               <parentArtifactCategory>GENERIC</parentArtifactCategory>
               <childId>VSR vRAM 2.0</childId>
               <childArtifactFamily>VIRTUAL MEMORY</childArtifactFamily>
               <childArtifactCategory>GENERIC</childArtifactCategory>
               <enabled>true</enabled>
               <status>
                      <label>ENABLED</label>
                       <visibleStatusLabel>ENABLED</visibleStatusLabel>
                       <enabled>true</enabled>
               </status>
       </relationshipTemplate>
       <relationshipTemplate>
               <type>USES</type>
               <parentId>VSR VM 2.0</parentId>
               <parentArtifactFamily>VIRTUAL MACHINE</parentArtifactFamily>
               <parentArtifactCategory>GENERIC</parentArtifactCategory>
```

```
<childId>VSR vDISK 2.0</childId>
               <childArtifactFamily>VIRTUAL DISK</childArtifactFamily>
               <childArtifactCategory>GENERIC</childArtifactCategory>
               <enabled>true</enabled>
               <status>
                      <label>ENABLED</label>
                      <visibleStatusLabel>ENABLED</visibleStatusLabel>
                      <enabled>true</enabled>
               </status>
       </relationshipTemplate>
       <relationshipTemplate>
               <type>USES</type>
               <parentId>VSR VM 2.0</parentId>
               <parentArtifactFamily>VIRTUAL_MACHINE</parentArtifactFamily>
               <parentArtifactCategory>GENERIC</parentArtifactCategory>
               <childId>VSR vPort ETH0 2.0</childId>
               <childArtifactFamily>VIRTUAL PORT</childArtifactFamily>
               <childArtifactCategory>GENERIC</childArtifactCategory>
               <enabled>true</enabled>
               <status>
                      <label>ENABLED</label>
                      <visibleStatusLabel>ENABLED</visibleStatusLabel>
                      <enabled>true</enabled>
               </status>
       </relationshipTemplate>
       <relationshipTemplate>
               <type>OVER</type>
               <parentId>VSR SCALE POLICY 2.0</prentId>
               <parentArtifactFamily>POLICY</parentArtifactFamily>
               <parentArtifactCategory>ENTITY RANGE</parentArtifactCategory>
               <childId>VSR VM 2.0</childId>
               <childArtifactFamily>VIRTUAL MACHINE</childArtifactFamily>
               <childArtifactCategory>GENERIC</childArtifactCategory>
               <enabled>true</enabled>
               <status>
                      <label>ENABLED</label>
                      <visibleStatusLabel>ENABLED</visibleStatusLabel>
                      <enabled>true</enabled>
               </status>
       </relationshipTemplate>
</relationshipTemplates>
</templates>
```



# 2.6 Step three: configuring monitors

#### **Figure 11 Configure Monitors**

If we follow NFV Director XML notation we can express a monitor using some conditions, and actions. Here, Monitor corresponds to a template in SiteScope.

Below is a snippet of MONITOR, CONDITION and ACTION.

```
<template internal-id="" local-uri="" xmlns="http://www.hp.com/nfvd">
 <artifact-templates>
   <artifact-template>
     <artifact-definition>
       <family>MONITOR</family>
       <category>GENERIC</category>
     </artifact-definition>
      <status>
       <label>DEPLOYED</label>
       <visible-label>DEPLOYED</visible-label>
       <enabled>true</enabled>
      </status>
      <categories>
       <category>
          <attributes>
            <attribute>
              <label>Name</label>
              <description />
              <mandatory>false</mandatory>
              <order>1</order>
              <type>TEXT</type>
              <unit>TEXT</unit>
              <value />
            </attribute>
            <attribute>
              <label>Description</label>
              <description />
              <mandatory>false</mandatory>
              <order>2</order>
              <type>TEXT</type>
```

```
<unit>TEXT</unit>
          <value />
        </attribute>
        <attribute>
          <label>Type</label>
          <description />
          <mandatory>true</mandatory>
          <order>3</order>
          <type>TEXT</type>
          <unit>TEXT</unit>
          <value>CPU</value>
        </attribute>
        <attribute>
          <label>Frequency</label>
          <description />
          <mandatory>true</mandatory>
          <order>4</order>
          <type>Number</type>
          <unit>seconds</unit>
          <value>600</value>
        </attribute>
      </attributes>
      <label>GENERAL</label>
      <order>1</order>
    </category>
    <category>
      <attributes>
        <attribute>
          <label>Path</label>
          <description />
          <mandatory>false</mandatory>
          <order>1</order>
          <type>TEXT</type>
          <unit>TEXT</unit>
          <value />
        </attribute>
        <attribute>
          <label>Type</label>
          <description />
          <mandatory>false</mandatory>
          <order>2</order>
         <type>TEXT</type>
          <unit>TEXT</unit>
         <value>AUTO</value>
        </attribute>
      </attributes>
      <label>DEPLOYMENT</label>
      <order>2</order>
    </category>
 <creation-timestamp>02015-04-30T10:59:01.776+0200</creation-timestamp>
 <id>f8bbd4dc-098a-4899-b884-c8377f85c750</id>
 <identifier>MONITOR:f8bbd4dc-098a-4899-b884-c8377f85c750</identifier>
 <physical>false</physical>
  <update-timestamp>02015-04-30T10:59:02.507+0200</update-timestamp>
</artifact-template>
<artifact-template>
 <artifact-definition>
    <family>CONDITION</family>
    <category>GENERIC</category>
 </artifact-definition>
  <status>
    <label>ENABLED</label>
    <visible-label>ENABLED</visible-label>
```

```
<enabled>true</enabled>
  </status>
 <categories>
    <category>
     <attributes>
        <attribute>
          <label>Name</label>
          <description />
          <mandatory>false</mandatory>
          <order>1</order>
          <type>TEXT</type>
          <unit>TEXT</unit>
          <value>GOOD Condition</value>
        </attribute>
        <attribute>
          <label>Type</label>
          <description />
          <mandatory>true</mandatory>
          <order>2</order>
          <type>TEXT</type>
          <unit>TEXT</unit>
          <value>GOOD</value>
        </attribute>
        <attribute>
          <label>Expression</label>
          <description />
          <mandatory>true</mandatory>
          <order>3</order>
         <type>TEXT</type>
          <unit>TEXT</unit>
          <value>cpu usage average &lt; 30</value>
        </attribute>
      </attributes>
      <label>GENERAL</label>
      <order>1</order>
    </category>
  </categories>
 <creation-timestamp>2015-06-11T11:31:09.769+0000</creation-timestamp>
 <id>61547f6c-adbe-4d54-8436-80d551ee0867</id>
 <identifier>CONDITION_GENERIC____31</identifier>
 <physical>false</physical>
 <update-timestamp>2015-06-11T11:32:52.034+0000</update-timestamp>
</artifact-template>
<artifact-template>
 <artifact-definition>
   <family>ACTION</family>
   <category>GENERIC</category>
 </artifact-definition>
 <status>
   <label>ENABLED</label>
    <visible-label>ENABLED</visible-label>
    <enabled>true</enabled>
 </status>
 <categories>
    <category>
      <attributes>
        <attribute>
          <label>Name</label>
          <description />
          <mandatory>true</mandatory>
          <order>1</order>
          <type>TEXT</type>
          <unit>TEXT</unit>
```

```
<value>SCALE IN</value>
        </attribute>
        <attribute>
          <label>Description</label>
          <description />
          <mandatory>false</mandatory>
          <order>2</order>
          <type>TEXT</type>
          <unit>TEXT</unit>
          <value>Description</value>
        </attribute>
        <attribute>
          <label>Type</label>
          <description />
          <mandatory>true</mandatory>
          <order>3</order>
         <type>TEXT</type>
         <unit>TEXT</unit>
          <value>ScaleIn</value>
        </attribute>
        <attribute>
          <label>ScaleValue</label>
          <description />
         <mandatory>false</mandatory>
         <order>4</order>
         <type>Number</type>
         <unit>TEXT</unit>
          <value>10</value>
        </attribute>
        <attribute>
          <label>Path</label>
          <description />
          <mandatory>false</mandatory>
          <order>5</order>
          <type>TEXT</type>
          <unit>TEXT</unit>
          <value>Script Path</value>
        </attribute>
        <attribute>
         <label>Operation_Mode</label>
         <description />
         <mandatory>false</mandatory>
          <order>6</order>
         <type>TEXT</type>
         <unit>TEXT</unit>
         <value>CLOSED LOOP</value>
       </attribute>
     </attributes>
     <label>GENERAL</label>
     <order>1</order>
   </category>
 </categories>
 <creation-timestamp>2015-06-11T11:34:48.634+0000</creation-timestamp>
 <id>4ebbd09b-8e45-4139-a8a9-334296a844c6</id>
 <identifier>ACTION_GENERIC____33</identifier>
 <physical>false</physical>
  <update-timestamp>2015-06-11T11:35:42.576+0000</update-timestamp>
</artifact-template>
```

### 2.6.1 Example Configuring Monitors

SiteScope is the component of NFVD that gets the real-time status of the network and infrastructure to be monitored.

In SiteScope you define the monitors and metrics that you want to take care applied. NFVD has a comprehensive set of predefined templates for SiteScope monitors.



Figure 12 SiteScope NFVD Templates

In the NFVD inventory the artifact "Monitor" and the possible "Actions" (i.e. scale out/scale in) that could be associated to a value get by a monitor, have to be declared.

A SiteScope monitor can be instantiated by assigning concrete values to the parameters of the measurement of the SiteScope monitors templates. This assignment is done by the NFVD Fulfillment Monitor artifact definition.

#### Let's see an example:

To achieve a case where : for checking every **120 seconds** the value of the **CPU usage average** of a Virtual Machine and, for if this CPU usage average value is lower than 5, the **scale in action** will be invoked. In this case, these steps have to be done in SiteScope:

In SiteScope the product template can be used (or create a new one if there are no templates for doing the monitoring which we are looking for):

In the SiteScope Monitor template left panel are the output parameters that are needed: "frequency", "good scenario" (not error or warning scenario because we are with the scale in scenario so it means good scenario), host of the VM, user, pw and name of the VM.

	Dependencies					
THE OPENSTACK						
E WWARE	Calculated Metrics Sett	ngs				
É-M CPU						
E-ER VIRTUAL MACHINE VMWARE CRU	Threshold Settings					
96.96.000 96.96	Thround Coungo					
	If unavailable:	Set monitor status according to 1	hresholds 💌			
Cou usage average warning						
Trequency	Default status:	Good 💌				
aroupDescription	On internal error	Sat monitor status according to 1	brooholdo			
- I host	On internal error.	Set monitor status according to	filesholds -			
- I password		1				
-I user	Add Default Three	holds Remove Default Thresh	iolds			
T vm						
E- DiskRead	Error					
E DiskWrite	V V					
E Memory						
E NetworkRx		Condition	Operator	Value		Schedule
NetworkTx	cpu_usage_avera	je	>	%%cpu_usage_average_error%%	every day, all day	
E CIRTUAL_NETWORK						
E VNF_COMPONENT	Warning if					
🕀 📩 ECP	· · ·					
🕀 📅 HPSA		Condition	Operator	Value		Schedule
E-CK_MANAGER	cpu_usage_avera	je	>	%%cpu_usage_average_warning%%	every day, all day	
🕀 🚼 LOGS						
🕀 🔐 NEO4J	Good if					
OPEN_MEDIATION						
	× ×					
H POSTGRES		Condition	Operator	Value		Schedule
U SITESCOPE	cou usage avera	26	<	%%cpu usage average good%%	every day, all day	
UT SOSA						
Cabilities Templates						
	Enable/Disable Monitor					
Monitors						
	* Disable monitor:	true				

Figure 13 SiteScope NFVD Monitor Template Parameters

In the Assurance part, for the monitor calling an action (in this case scale in) when the measurement reaches the good scenario condition (in our case the value of the CPU usage average < 5) an alert that sends a trap when the system gets in the good scenario must be defined (in SiteScope Monitor Template):

Template monitor VMware Performance - "%%vm%%"			Properties Alerts
Alerts on monitor: %%vm%%			
* 🖉 🐚 📕 🗞 🗞			
Name 📥	Status	Description	Action Name
ErrorScenario	Enabled	Alert to be triggered on countering an error condition	SNMP Error Alert
GoodScenario	Enabled	Alert to be triggered on countering a good condition o	SNMP Good Alert
WarningScenario	Enabled	Alert to be triggered on countering a warning conditio	SNMP Warning Alert

#### Figure 14 SiteScope Alerts Definition 1

We can select the special conditions for sending the trap when the good scenario is reached (in this case we have configured the trap to be sent only when the good condition has been reached 3 period of time of 120 seconds).

Edit Alert: GoodScenario		(hanna)				23
General Settings					¥	
Alert Targets					۷	
Alert Actions					8	
Alert Actions *						
* / 🗙 🖂 🗞	Ъ.					
- Li	Name	Category	When	Schedule	Target	
명	SNMP Good Alert	Good	Once, after 2 times	every day, all day	SNMPTarget	

#### Figure 15 SiteScope Alerts Definition 2

The target of the trap is UCA and is configured previously in the NFVD assurance part configuration.

Alert Action: SNN	ИР Тгар			X
Action Type Settin	gs			*
Antina annai	SNMD Cood Alast			1
SNMP Trap:	SNMP Good Alen		Add existing	
* Template:	NFVD_SNMP_TEMPLATE_XML.xml	]		
Message:				]
* Schedule:	every day, all day			
	Mark this action to close alert			
Status Trigger				8
Trigger Frequency	,			
- mggor r roquono,				
<ul> <li>Always, af</li> <li>Once, afte</li> </ul>	ter the condition has occurred at least 1 times the condition has occurred exactly 2 times	;		
Initially, aft	er 1 times	and repeat every 1	times	
Only alert	f monitor was previously in error at least 2		times	

#### Figure 16 SiteScope Alerts configuration

In the template of the alert in SiteScope, you configure the trap to be sent. In this xml we can see the good condition trap, the error condition trap and the normal condition trap that is sent to UCA when the conditions are reached:

```
<?xml version="1.0" encoding="UTF-8"?>
<SNMPAction xmlns:xsi="http://www.w3.org/2001/XMLSchema-</pre>
instance"><FieldList><Field type="OID" refvalue="<<snmpVersion>>"
description="SNMP Trap OID " OID="1.3.6.1.4.1.11.15.1.0" name="Trap
OID"><option name="V1" value=".1.3.6.1.4.1.11.2.53.2.2.3.1.2.0.1"/><option
name="V2" value=".1.3.6.1.4.1.11.2.53.2.2.3.1.2.1"/></Field><Field
type="STRING" refvalue="<name>" description="Remote Machine Name"
OID=".1.3.6.1.4.1.11.2.53.2.2.3.1.2.1.1" name="RemoteMachineName"/><Field
type="STRING"
refvalue="<<monitorType>>" description=" SiteScope monitorType"
OID=".1.3.6.1.4.1.11.2.53.2.2.3.1.2.1.2" name="SiteScope monitorType"/><Field
type="STRING" refvalue="<fullMonitorName>" description="SiteScope
fullMonitorName" OID=".1.3.6.1.4.1.11.2.53.2.2.3.1.2.1.3" name="SiteScope
fullMonitorName"/><Field type="STRING" refvalue="QUALITY_OF_SERVICE_ALARM"
description="QUALITY OF SERVICE ALARM to be updated by NOM"
OID=".1.3.6.1.4.1.11.2.53.2.2.3.1.2.1.4" name="QUALITY OF SERVICE
ALARM"/><Field type="STRING" refvalue="<category>" description="Category of
SiteScope alert" OID=".1.3.6.1.4.1.11.2.53.2.2.3.1.2.1.5" name="SiteScope
category"/><Field type="STRING" refvalue="< classifier>" description="SiteScope
alert classifier" OID=".1.3.6.1.4.1.11.2.53.2.2.3.1.2.1.6" name="SiteScope
classifier"/><Field type="STRING" refvalue="<currentTime>"
description="SiteScope server alert generated time"
OID=".1.3.6.1.4.1.11.2.53.2.2.3.1.2.1.7" name="SiteScope currentTime"/><Field
type="STRING" refvalue="<errorOnly>" description="SiteScope error counter"
OID=".1.3.6.1.4.1.11.2.53.2.2.3.1.2.1.8" name="SiteScope errorOnly"/><Field
type="STRING" refvalue="<warningOnly>" description="SiteScope warning counter"
OID=".1.3.6.1.4.1.11.2.53.2.2.3.1.2.1.9" name="SiteScope warningOnly"/><Field
type="STRING" refvalue="<goodOnly>" description="SiteScope goodOnly counter"
OID=".1.3.6.1.4.1.11.2.53.2.2.3.1.2.1.10" name="SiteScope goodOnly"/><Field
```

```
type="STRING" refvalue="<sample>" description="SiteScope alert Id"
OID=".1.3.6.1.4.1.111.2.53.2.2.3.1.2.1.11" name="SiteScope alertId"/><Field
type="STRING"
refvalue="_customPropertiesValues=<_customPropertiesValues>|_httpPort=<_httpPor
t>|_webserverAddress=<_webserverAddress>|alertHelpURL=<alertHelpURL>|diagnostic
TraceRoute=<diagnosticTraceRoute>|errorOnly=<errorOnly>|goodOnly=<goodOnly>|Ful
lGroupId=<FullGroupId>|group>|groupdescription=<groupdescription>|groupI
D=<groupID>|id=<id>|mainStateProperties=<mainStateProperties>|monitorDrilldownU
rl=<<monitorDrilldownUrl>>|monitorServiceId=<<monitorServiceId>>|monitorTypeDis
playName=<<monitorTypeDisplayName>>|monitorUUID=<<monitorUUID>>|mountName=<moun
tName>|multiViewUrl=<multiViewUrl>|fullMonitorName=<fullMonitorName>|newSiteSco
peURL=<newSiteScopeURL>|sample=<sample>|siteScopeBaseUrl=<siteScopeBaseUrl>|sit
eScopeHost=<siteScopeHost>|sitescopeURL=<sitescopeURL>|sitescopeuserurl=<sitesc
opeuserurl>|state=<state>|tag=<tag>|targetHost=<<targetHost>>|targetIP=<<target
IP>>|targetIPVersion=<<targetIPVersion>>|templateDeployPath=<<templateDeployPat
h>>|time=<time>|warningOnly=<warningOnly>|customerId=<customerId>"
description="SiteScope alert additionalText"
OID=".1.3.6.1.4.1.11.2.53.2.2.3.1.2.1.12" name="SiteScope
additionalText"/><Field type="STRING" refvalue="NFVD Source"
description="SiteScope server IP address"
OID=".1.3.6.1.4.1.11.2.53.2.2.3.1.2.1.13" name="SiteScope
webserverAddress"/></FieldList></SNMPAction>
```

This is the generic template. In the case that a Good Scenario is reached, the traps that is sent from SiteScope alert to UCA is (enriched by EBC CA):

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>< msg
xmlns="http://types.ws.ucaautomation.hp.com/"> <header> <ActionRequest
Originator="alarm" OpenLoop="false" Mode="real"> <ActionId></ActionId>
<Operation>SCALE_IN</Operation> <OrignatorId>29:e4c4065b-747b-4f96-83be-
d803f1437023</OrignatorId>

</pre
```

So SiteScope, with this trap, is notifying UCA that there is a "**problem**", in this case **SCALE\_IN** \_**CPU** that is associated to an **action** to be invoked that is **SCALE IN**.

When UCA receives this notification, if there is no another stronger order relating to this (i.e. the minimum number of VW is 1), the SCALE IN action is sent to the NFVD fulfillment part to be done in the host and the NFVD fulfillment part will delete the VM as result of the scale in operation.

Apart of the NFVD SiteScope templates, a lot of monitors could be customized in SiteScope to adapt the customer monitoring requirements (see 2.7).

## 2.7 Step four: Creating custom monitors

This is an optional step. It is required only if he comprehensive set of predefined SiteScope monitors is insufficient and you want to take advantage of NFVD's ability to support custom/VNF specific monitors.

Custom templates broaden the capabilities of the regular SiteScope monitors beyond what is provided by the NFVD out-of-the-box monitors. Custom monitors can help in tracking availability and performance of monitored environments. The custom templates enable you to create your own monitor by using any existing monitors provided by SiteScope.

- 1. Select the templates context.
- 2. Right-click the SiteScope root node from the tree and select **New > Template Container**.



Figure 17: SiteScope: Create custom templates

3. Enter the name of the template container and click the **OK** button.

. Right-ci	ick this new ter	mplate conta	amer node and select
Ø Sit	eScope		
Page Option	ns 👻 Help 👻		
* • 🖺	• × × 1	7 - <b>G</b> Q	t.
E- Site	Scope	_	
	Custom Template	s	Templete Capteiner
	* New	•	remplate Container
	Paste	Ctrl-V	Template
	Delete	Ctrl-D	
Ū- <u>6</u> ;	Сору	Ctrl-C	
	Cut	Ctrl-X	
	Deploy Terr	nplate Ctrl-J	
	Import	•	
	Export	•	
	Generate X	ML	
	Expand All		
L	-		-

Right-click this new template container node and select New > Template.

#### Figure 18: SiteScope: new template

- 5. Enter the name of the template and click the **OK** button.
- 6. In the example provided in this section, a template container is created with the name Custom Templates.

7. Right-click this new template node and select **New > Group**.

Ø SiteS	соре					
Page Options	▪ Help ▼					
* • 🖺 •	* * 🝸 - 🗗 Q-			Template PerformanceM	onitor	
⊟–⊕ SiteScop	pe om Templates PerformanceMonitor			Variable Values		
E Moni	* New	•	Group		Ctrl-G	nceMonitor
	Paste	Ctrl-V	Variable			
t tem	Cut	Ctrl-X	Microsoft	Windows Remote Serve	r	
🗄 🛗 Solut	Delete	Ctrl-D	UNIX Ren	note Server		
	Сору	Ctrl-C				
	Deploy Template	Ctrl-J		Last edited by:	SiteSco	e Administrator
	Deploy Template Using CS	SV		Last edited on:		
	Publish Changes			Lust cultur on.		
	Export	•				
	Expand All			Search/Filter Tags		

Figure 19: SiteScope: new group

- 8. Enter the name of the group and click the **OK** button.
- 9. In the example provided in this section, a template is created with name PerformanceMonitor.
- 10. Right-click this group node and select **New > Monitor**.

Page Options ▼ Help ▼ * ▼ 🖺 ▼   ネ ×   🗑 ∽ 🗗 Q ⊡-⊕ SiteScope
* • 🖺 • 🗍 ∗ 🗧 🖓 - 🞜 Q. ⊡-⊕ SiteScope
⊡-@ SiteScope
Custom Templates
Copy Ctrl-C Expand All

Figure 20: SiteScope: new Monitor

- 11. Select any one of the monitors of your choice from the list.
- 12. In the example provided in this section, a group is created with the name Custom Monitor.
- 13. Right-click the PerformanceMonitor template node and select **New > Variable**.

Ø SiteSco	ре						
Page Options 🔻	Help 🔻						
* • 🖺 • 🙁	* 🔻 - 🞜 Q-			Т	emplate PerformanceM	onitor	
E- DiteScope	Templates				Variable Values		
	New	•	Group	)		Ctrl-G	nceMonitor
te - <u>C</u> ii Moni te - <u>C</u> ii NFVI te - <u>C</u> ii Samo	Paste Cut Delete	Ctrl-V Ctrl-X Ctrl-D	Variab Micros UNIX	ole soft \ Rem	Windows Remote Serve ote Server	er	
⊕–∰ Temp ⊕–∰ Solut	Copy Deploy Template Deploy Template Using CS	Ctrl-C Ctrl-J			Last edited by:	SiteSco	De Administrator
	Publish Changes Export Expand All	•			Last edited on: Search/Filter Tags	Tue Jul	1 2014 01:06 PM IST

Figure 21: SiteScope: new Variable

The following	window	opens
---------------	--------	-------

🛓 New Variable		x
Variable Values		*
* Name:	host	
Display name:	host	
Description:	host	
Default value:		
Display order in template:		
	Password variable	
	Mandatory variable	
2/	_	11-1-
		Help

Figure 22: SiteScope: new Variable details

14. Enter the details for configuring a variable to associate it with the template.

In the following example, the host variable is configured. After the variable is configured, it appears in the tree under the Template node. You can configure any number of variables. The following example shows how to use these variables in the monitor depicted.

Ø SiteScope		
Page Options 👻 Help 👻		
* • 🖞 • 🙁 × 🝸 - 💋 Q-	Template monitor · "Performance of %%host%%"	Properties Alerts
□-@ SiteScope □-@ Custom Templates	General Settings	
Custom Monitor	Monitor Run Settings	A
	Frequency:     Seconds     Seconds     Seconds	
Honitor Deployment Wizard Templates     Honitor Deployment Wizard Templates     Honitor Deployment Wizard Templates	Venify error Monitor schedule:	
	Show run results on update	=

#### Figure 23: SiteScope: enter variable to associate with template



Following is the complete hierarchy of the Custom Template created.

#### Figure 24: SiteScope: hierarchy of custom template

- 15. Create variables for conditions for custom monitors to achieve threshold in below format:
  - a. For Error Condition: <variable>\_error
  - b. For Warning Condition: <variable>\_warning
  - c. For Good Condition: <variable>\_good

For eg: If variable name is 'network\_bytes\_in\_received' then we will have below template variables for threshold:

network\_bytes\_in\_received\_error

network\_bytes\_in\_received\_warning

network\_bytes\_in\_received\_good

16. To create the KPI alert, use the <variable> name used in the previous step.

Example: network\_bytes\_in\_received.

For more details please refer to section 16 of the "SiteScope Monitor Reference" guide.

# 2.8 Step five: Integrating custom alarm feeds

This is an optional step. It is required only if you want to take advantage of NFVD's Mediation Framework to integrate other alarm sources into NFVD.

Built-in action (SCALE IN/OUT, SCALE UP/DOWN or SCRIPT) can be invoked automatically by specifying a suitable CONDITION.

In cases where VNF specific correlation and actions needs to be performed, new correlation rules can be created which rely on alarm feeds from SiteScope or new alarm sources can be integrated into NFVD. Following tables provide a sample SiteScope alarm as received by NFVD's correlation engine i.e. UCA EBC, and description of important fields in the alarm.

<alarmcreationinterface xmlns="http://hp.com/uca/expert/x733Alarm"></alarmcreationinterface>
<identifier>201019787:1654434a-731b-4547-974d-0bb9e3e0709f</identifier>
<sourceidentifier>NFVD_Source</sourceidentifier>
<alarmraisedtime>2014-05-05T20:41:00.848+05:30</alarmraisedtime>
<originatingmanagedentity>KVM TestVM</originatingmanagedentity>
<originatingmanagedentitystructure></originatingmanagedentitystructure>
<classinstance class="Host" instance="ossvml.ind.hp.com"></classinstance>
<alarmtype>QUALITY_OF_SERVICE_ALARM</alarmtype>
<probablecause>UtilizationPercentage</probablecause>
<perceivedseverity>CRITICAL</perceivedseverity>
<networkstate>NOT_CLEARED</networkstate>
<operatorstate>NOT ACKNOWLEDGED</operatorstate>
<problemstate>NOT_HANDLED</problemstate>
<probleminformation>Attribute not available</probleminformation>
<specificproblem>ERROR</specificproblem>
<additionaltext>Sitescope alarm MONITOR.cpuMonitor-</additionaltext>
001 CONDITION=ERROR group=KVM VM CPU
Monitor groupdescription=CPU groupID=201070542 id=1
<notificationidentifier>0</notificationidentifier>
<correlationnotificationidentifiers>Attribute not</correlationnotificationidentifiers>
available

Alarm Attributed Name	Description
<identifier></identifier>	The value should be unique, so UUID is generated by Channel adapter and is appended with alert.
<sourceldentifier></sourceldentifier>	The value should be configured at channel adapter to constant NFVD_Source configure channel adapter.
<perceivedseverity></perceivedseverity>	CRITICAL for ERROR condition of monitor. WARNING for WARNING condition of monitor. CLEAR for GOOD condition of the monitor.
<alarmtype></alarmtype>	The value should be configured at channel adapter to constant QUALITY_OF_SERVICE_ALARM.

<additionaltext></additionaltext>	The value should contain groupdescription= <monitorname> and <monitorname> should be the value of property GENERAL.Name</monitorname></monitorname>
	of Monitor artifact.

#### **Table 1 NFVD Alarm attributes**

The general format of the alarm is as per the standards specified by Open Mediation in section "4.5.2 Alarm message format" of functional specification guide.

Any new alarm sources like VIM or VNFM can be integrated into NFVD using Open Mediation. If the alarm source is SNMP based then the NFVD's "Generic SNMP Channel Adapter" can be customized to handle the traps from this new source also. Otherwise new custom channel adapters can be developed which receives alarms from respective source and transforms it into Open Mediation compliant format using Open Mediation toolkit. For more details please refer to the OSS Open Mediation Development Guide.

## 2.9 Step six: Creating Custom Correlation rules

This is an optional step. It is required only if you want to create custom/VNF specific correlation and/or automated actions using NFVD's Automation framework (powered by UCA Automation & UCA EBC)

In cases where VNF specific correlation and actions needs to be performed, new correlation rules can be created using UCA Automation framework.

When UCA Automation service pack is deployed, we get in the UCA's HPSA the UCA/NetworkResources, UCA/Services, UCA/Actionframework and UCA/Parameters inventory model:

	ECTOR			
Work Area	Inventory	Class Views	Instance Views	
Jobs	UCA/Ac	tionFramework 🔦	AD/AD Configura	tion
Messages Audit Messages	ΕÅρ	iagnostics Actions	AD/ADProcessD	efinitionTree
Track Activations	+	Actions -	AD/ADProcessIn	stanceTree
Workflows	F 2	Problems 👻	CRModel/Equipm	nent
Services			CRModel/Parame	eters
Service Instances			CRModel/NNMiD	ataload
Logs			IPAM/Resources	
Search Logs Service Order View			IPAM/Parameters	S
Business Calendar			MSA/Parameter	Nodel
Tools	-		MSA/ResourceM	odel
Refresh ON			NFVAuto/VNFIns	tances
Self			NFVD/Parameter	S
Management <sup>4</sup>		/	UCA/Services	
		(	UCA/ActionFram	ework
			UCA/Parameters	

#### Figure 25 UCA's HPSA Framework

**UCA/Services** provide option to create the domain name (service name) to be supported.

**UCA/Action Framework**, an alarm from a Resource Instance (device) can be identified as a Problem in the UCA Automation solution. Each of these problems can be associated with an action to be taken as a part of resolution. Action involves invocations to HP Service Activator workflows to perform specific task on the resource instance. In UCA/Action Framework the different problems, actions and associations between actions and resources are defined and created.

**UCA/Parameters** are for defining the global parameters of the new actions that needs to be used by the application (Global Parameters figure 19), and for assigning a workflow for each group of values: Service Type, Problem and Action (workflow template, figure 20)

Inventory Class Views Instance Views			
📕 UCA/Parameters 🗙 🚺 UCA/Parameters 🔩 🚺	View GlobalParameter 🔩		
Parameters			
🖃 🗊 Global Parameters			View GlobalParams
🖸 🔟 GlobalParameter 🤝			
🛨 (🌶) Workflow Templates 👻	Name	Value	Description
	ld *	100	Primary key
	ActionMode	Open Loop	Global Action mode open or closed loop
	ActionType	real	Global Action type real-time or demo mode
	ActivateGlobalParams	false	Indicates whether global parameters should be activated

#### Figure 26 HPSA Inventory – UCA/Parameters – Global Parameters

Inventory Class Views Instance Views				
UCA/ActionFramework $ imes  ilde{I}$ UCA/Parameters	s 🗙 🛛 UCA/Parameters 🔩 🝸 Vi	ew NFVD  SCALE_UP_CLOSED	NFVD_Controller	
Parameters				
🖃 🗊 Global Parameters			View Worktio	wiemplate
🖸 🗊 GlobalParameter 🤟				
🖃 (🎙) Workflow Templates 🤟		Name	Value	Description
(P) NFVD  SCALE_UP_CLOSED  NFVI	D_Controller -	ServiceType *	NFVD	Type of Service
(P) NFVD[ SCALE_DOWN_CLOSED ]	NFVD_Controller -	ActionName *	SCALE_UP_CLOSED	Action name
NFVD  SCALE_IN_CLOSED  NFVE	D_Controller -	Workflow *	NFVD_Controller	Name of workflow
(P) NFVD  SCALE_OUT_CLOSED  NF	VD_Controller -			
(P) NFVD  SCALE_UP  NFVD_Control	ler 🗸			
(P) NFVD  SCALE_DOWN  NFVD_Cor	ntroller 👻			
(P) NFVD  SCALE_IN  NFVD_Control	ler 👻			
(P) NFVD  SCALE_OUT  NFVD_Contr	oller 👻			
(P) NFVD  SCALE_OUT_AND_CONNE	CT  NFVD_Controller -			
○ (₱) NFVD  SCALE_OUT_AND_CONNE	CT_CLOSED  NFVD_Controller			
(P) NFVD  SCRIPT  NFVD_Controller	-			

#### Figure 27 HPSA Inventory - UCA/Parameters Workflow Templates

Summarizing, for a generic case we have to create on the UCA HPSA Inventory, the new resources, the new services, the new problems, the new actions and the new workflows. So when a new event arrives to UCA, the UCA generic workflow "UCAController" is triggered for validating the parameters "action", "origin" and "task", and then other generic UCA workflows can be invoked (i.e. "UCAValidate\_Request"l or "UCAResponse\_Handler"), and/or the new workflows defined for the new action and new problems will be invoked for finish the new action and getting the result.

For more details on problem-action framework, refer to HP UCA Automation Administrator and User Interface Guide.

### 2.9.1 Specific Definitions for NFVD

The HPSA NFVD solution value pack already has a set of predefined data for the NFVD solution.

Inventory Class Views	Instance Views				
UCA/ActionFramework $ imes$	UCA/Parameters 🗙 UCA/Services 🔩	-	View NFVD 🔩		
🖃 🂁 Services 🗸					
🖸 🂁 NFVD 🗟					View ServiceType
			Name	Value	Description
			ld *	100	Primary key
			Type *	NFVD	Type of Service
1					

Figure 28 UCA Automation Foundation Inventory – UCA/Services

Action definitions for NFVD Inventory Class Views Instance Views UCA/Parameters X UCA/ActionFramework 🖸 🌠 Diagnostics Actions Framework 👻 - Actions F SCALE\_DOWN -E SCALE\_DOWN\_CLOSED -🛨 📝 SCALE\_IN 🗸 🛨 🚰 SCALE\_IN\_CLOSED 🗸 🛨 📝 SCALE\_OUT 👻 SCALE\_OUT\_AND\_CONNECT -E SCALE\_OUT\_AND\_CONNECT\_CLOSED -🕀 📷 SCALE\_OUT\_CLOSED 🗸 + 📝 SCALE\_UP 🗸 + F SCALE\_UP\_CLOSED -🛨 🐋 SCRIPT 👻

#### Figure 29 UCA Automation Foundation Inventory – UCA/Services

Problem definitions for NFVD:

_ =		
Diagnostics Actions Framework 👻	SCALE_IN_Memory -	SCALE_OUT_DiskWrite -
🛨 📝 Actions 👻	SCALE_IN_Network -	SCALE_OUT_Logs -
🖻 🚹 Problems 🤝	SCALE_IN_NetworkRx	SCALE_OUT_Memory -
SCALE_DOWN_CLOSED_CPU -	SCALE_IN_NetworkTx	SCALE_OUT_Network
🖸 👔 SCALE_DOWN_CLOSED_DiskRead 👻	SCALE_IN_Process	🖸 👔 SCALE_OUT_NetworkRx 👻
🖸 👔 SCALE_DOWN_CLOSED_DiskWrite 👻	SCALE_OUT_AND_CONNECT_CLOSED_CPU -	🖸 👔 SCALE_OUT_NetworkTx 👻
SCALE_DOWN_CLOSED_Logs -	🖸 🐔 SCALE_OUT_AND_CONNECT_CLOSED_DiskRead 👻	SCALE_OUT_Process -
SCALE_DOWN_CLOSED_Memory ~	🖸 🐔 SCALE_OUT_AND_CONNECT_CLOSED_DiskWrite 👻	🖸 👔 SCALE_UP_CLOSED_CPU 🗸
SCALE_DOWN_CLOSED_Network ~	SCALE_OUT_AND_CONNECT_CLOSED_Logs -	🖸 👔 SCALE_UP_CLOSED_DiskRead 🗸
SCALE_DOWN_CLOSED_NetworkRx -	SCALE_OUT_AND_CONNECT_CLOSED_Memory -	🖸 👔 SCALE_UP_CLOSED_DiskWrite 🗸
SCALE_DOWN_CLOSED_NetworkTx -	SCALE_OUT_AND_CONNECT_CLOSED_Network	SCALE_UP_CLOSED_Logs ~
SCALE_DOWN_CLOSED_Process -	SCALE_OUT_AND_CONNECT_CLOSED_NetworkRx	SCALE_UP_CLOSED_Memory ~
SCALE_DOWN_CPU -	SCALE_OUT_AND_CONNECT_CLOSED_NetworkTx	SCALE_UP_CLOSED_Network -
SCALE_DOWN_DiskRead -	SCALE_OUT_AND_CONNECT_CLOSED_Process	🖸 👔 SCALE_UP_CLOSED_NetworkRx 👻
SCALE_DOWN_DiskWrite -	SCALE_OUT_AND_CONNECT_CPU -	🖸 👔 SCALE_UP_CLOSED_NetworkTx 👻
SCALE_DOWN_Logs -	SCALE_OUT_AND_CONNECT_DiskRead -	SCALE_UP_CLOSED_Process
SCALE_DOWN_Memory -	SCALE_OUT_AND_CONNECT_DiskWrite	SCALE_UP_CPU -
SCALE_DOWN_Network -	SCALE_OUT_AND_CONNECT_Logs -	🖸 👔 SCALE_UP_DiskRead 🗸
SCALE_DOWN_NetworkRx -	SCALE_OUT_AND_CONNECT_Memory -	🖸 👔 SCALE_UP_DiskWrite 👻
SCALE_DOWN_NetworkTx -	SCALE_OUT_AND_CONNECT_Network ~	SCALE_UP_Logs -
SCALE_DOWN_Process -	SCALE_OUT_AND_CONNECT_NetworkRx	🖸 👔 SCALE_UP_Memory 👻
SCALE_IN_CLOSED_CPU -	SCALE_OUT_AND_CONNECT_NetworkTx -	SCALE_UP_Network -
SCALE_IN_CLOSED_DiskRead -	SCALE_OUT_AND_CONNECT_Process	SCALE_UP_NetworkRx -
SCALE_IN_CLOSED_DiskWrite -	SCALE_OUT_CLOSED_CPU -	SCALE_UP_NetworkTx -
SCALE_IN_CLOSED_Logs -	SCALE_OUT_CLOSED_DiskRead -	SCALE_UP_Process -
SCALE_IN_CLOSED_Memory -	SCALE_OUT_CLOSED_DiskWrite ~	🖸 👔 SCRIPT_CPU 👻
SCALE_IN_CLOSED_Network -	SCALE_OUT_CLOSED_Logs ~	🖸 👔 SCRIPT_DiskRead 🗸
SCALE_IN_CLOSED_NetworkRx -	SCALE_OUT_CLOSED_Memory -	🖸 👔 SCRIPT_DiskWrite 🗸
SCALE_IN_CLOSED_NetworkTx -	SCALE_OUT_CLOSED_Network ~	🖸 👔 SCRIPT_Logs 👻
SCALE_IN_CLOSED_Process -	SCALE_OUT_CLOSED_NetworkRx ~	🖸 👔 SCRIPT_Memory 👻
SCALE_IN_CPU -	SCALE_OUT_CLOSED_NetworkTx -	SCRIPT_Network -
SCALE_IN_DiskRead -	SCALE_OUT_CLOSED_Process ~	SCRIPT_NetworkRx -
SCALE_IN_DiskWrite -	SCALE_OUT_CPU -	SCRIPT_NetworkTx -
SCALE_IN_Logs -	SCALE_OUT_DiskRead -	SCRIPT_Process -

#### Figure 30 UCA Automation Foundation Inventory – UCA/ActionFramework (Problems)

Inventory Class Views Instance Views			
📕 UCA/ActionFramework 🗙 🗍 UCA/Parameters 🗙 🚺 UCA/Parameters 🔩 💌 🗸	ew NFVD  SCALE_UP_CLOSED	INFVD_Controller 🔩	
Parameters			
🕂 🔟 Global Parameters	View Workflow Template		
(p) Workflow Templates			
NFVD  SCALE_UP_CLOSED  NFVD_Controller	Name	Value	Description
(P) NFVD  SCALE_DOWN_CLOSED  NFVD_Controller	ServiceType *	NFVD	Type of Service
(P) NFVD  SCALE_IN_CLOSED  NFVD_Controller	ActionName *	SCALE_UP_CLOSED	Action name
(P) NFVD  SCALE_OUT_CLOSED  NFVD_Controller	Workflow *	NFVD_Controller	Name of workflow
(P) NFVD  SCALE_UP  NFVD_Controller			
(P) NFVD  SCALE_DOWN  NFVD_Controller			
(P) NFVD  SCALE_IN  NFVD_Controller			
(P) NFVD  SCALE_OUT  NFVD_Controller			
(P) NFVD  SCALE_OUT_AND_CONNECT  NFVD_Controller			
(P) NFVD  SCALE_OUT_AND_CONNECT_CLOSED  NFVD_Controller			
(P) NFVD  SCRIPT  NFVD_Controller			

#### Figure 31 UCA Automation Foundation Inventory – UCA/Parameters

With this specific NFVD definitions and mappings, when an NFVD event arrives to UCA whit the specific NFVD parameters predefined, when the UCA\_controler workflow invokes the UCA Generic workflows (UCAValidate\_Request.xml and UCAResponse\_Handler), if the Service is NFVD, the new specific workflow "NFVD\_Controller" is invoked, and it invokes the other new NFVD specific workflow" NFVD\_InvokeNFVDirectorNBI" whit the specific NFVD problem and the specific NFVD action. "InvokeNFVDirectorNBI" invokes NFVD fulfillment side (web service) for executing the action in the CPD infrastructure and sends the result to UCA:

hp i			UCA	Automation						tola:Administrator
UCA Automation	ALL FAILE	D[3] Waiting For Operator [0	1							2 Refr
Monitoring	2 Shon Details									1 of 2 >>
0	ID 🔭 2	Action Name	Action ID	Problem	Mode	Action Originator	Originator T 1	State	Start Time	Result
Manual Tests	9:4f856029-5154	-483c-87c6-35a7c4334b8d								
C Topology View	217	SCALE_OUT	103	SCALE_OUT_CPU	Closed Loop	alarm	9:4f856029-5154-483c-87	In_Progress	29-Jul-14 09	
View Reports	∃ 34:0c1754f3-e3a	e-4994-a263-e620fd373a5f								
	235	SCALE_IN	102	SCALE_IN_CPU	Closed Loop	alarm	34:Dc1754f3-e3ae-4994-a	In_Progress	29-Jul-14 09	
	⊒ 29:e4c4065b-747	b-4f96-83be-d803f1437023								
Users	230	SCALE_IN	102	SCALE_IN_CPU	Closed Loop	alarm	29:e4c4065b-747b-4f96-8	Failure	29-Jul-14 09	
Auto Refresh ON	□ 17:aae44efb-ae5a-45cd-9726-16acechchct9									
	304	SCALE_IN	102	SCALE_IN_CPU	Closed Loop	alarm	17:aae44efb-ae5a-45cd-9	Ok	30-Jul-14 12	xml version="1.0" encodi.</td
	3157:4f7afb38-fbd2-448e-bd17-2b430ea8a0d6									
	182	SCALE_OUT	103	SCALE_OUT_CPU	Closed Loop	alarm	157:4f7afb58-f0d2-448e-b	In_Progress	24-Jul-14 07	
	■ 154:a31b7802-89	5b-434d-b560-dbd731611a2	D							
	159	SCALE_OUT	103	SCALE_OUT_CPU	Closed Loop	alarm	154:a31b7802-895b-434d	Failure	24-Jul-14 07	
	□ 119:8478667f-4bd8-4ad7-846f-eda60dca2338									
	124	SCALE_OUT	103	SCALE_OUT_CPU	Closed Loop	alarm	119:8478667f-4bd8-4ad7	Failure	24-Jul-14 07	
	□ 104:10d85104-7ef2-4ef4-a910-7d6c45c2b208									
	109	SCALE_OUT	103	SCALE_OUT_CPU	Closed Loop	alarm	104:10d85104-7ef2-4efd-a	In_Progress	24-Jul-14 07	
	□ 103:93b10d34-430f-4f29-83dd-0c610b70fef1									
		SCALE OUT	102	SCALE OUT OPU	Closed Loop	alarm	102-02510424 4201 4520 9	In Progress	24- Jul-14 07	

#### Figure 32 UCA Automation Monitoring Screen

	CTOR				
Work Area 🛛 🔻	NFVD_Controller	Controller workflow for the NFVD value pack Executes shell scripts locally			
lobs	NFVD_ExecuteScripts				
Messages	NFVD_InvokeNFVDirectorNBI	Workflow to make webservice call to the NFV Director Fulfillment (HPSA SOSA interface)			
Audit Messages	UCAActionFrameworkExporter				
Track Activations	UCAController	Receives all Request messages to start diagnostic actions			
Workflows	UCAParser	Workflow to parse raw results of the test execution			
Services	UCAResponse_Handler	Sends responses after performing diagnostic actions			
Inventory	UCAValidate_Request	Validates all the request messages			
Service Instances	UCAValluare_Request	validates all the request messages			

#### Figure 33 HPSA UCA NFVD Workflows

## 2.9.2 Inventory data populated in NFVD value pack inventory

Action mapping to NFVD child workflows

inventory class views instance views			
UCA/Parameters 🗙 NFVD/Parameters 🔩	View SCALE_UP_CLOSED 🔩		
🖃 🚞 Parameters			
🖃 🏨 Workflow Templates 🤟		View Workflow Templates	
SCALE_UP_CLOSED -			
SCALE_DOWN_CLOSED ~	Name	Value	Description
SCALE_IN_CLOSED -	Action *	SCALE_UP_CLOSED	Action name
SCALE_OUT_CLOSED ~	Workflow *	NFVD_InvokeNFVDirectorNBI	Name of the workflow
· (₱) SCALE_UP ▼			
· (♥) SCALE_DOWN ↓			
· ⟨₱⟩ SCALE_IN ↓			
. (₱) SCALE_OUT			
SCALE_OUT_AND_CONNECT ~			
SCALE_OUT_AND_CONNECT_CLOSED			
· (P) SCRIPT -			

Figure 34 NFVD Inventory – NFVD/Parameters

#### 2.9.3 NFVD/Parameters

NFVD/Parameters provides mapping of the actions to the child workflows of NFVD.

Create a new workflow template by performing a right click on NFVD/Parameters  $\rightarrow$  Parameters  $\rightarrow$  Create Workflow Template.

Inventory	Class Views	Instance Views		
UCA/Actio	onFramework $ imes$	UCA/Parameters ×	NFVD/Parameters 🔩	- ×,
🗆 🗀 Pa	rameters			
- 99 	Create W	orkflow Templates		
	SCALE_OUT	*		
_				

#### Figure 35 NFVD/Parameters

Select the required action and enter the name of the child workflow. Click OK.

UCA/ActionFramework × UCA/Parameters × NFVD/Parameters •	Create Workflow	Templates 👟 Create New Workflow Te	emplates
	Name	Value	Description
Scale_out -	Action *	SCALE_UP ·	Action name
🖸 🔍 SCRIPT -	Workflow *	NFVD_InvokeNFVDirectorN	Name of the workflow
		OK Reset	

#### Figure 36 Creating new Workflow Templates

Inventory Class Views Instance Views				
• NFVD/Parameters 🔩	View SCALE_UP 🔩			
Parameters     Workflow Templates ~		View Workflow Templates		
SCALE_UP	Name	Value	Description	
SCALE_DOWN *	Action *	SCALE_UP	Action name	
SCALE_OUT -	Workflow *	NFVD_InvokeNFVDirectorNBI	Name of the workflow	
SCRIPT -				

#### Figure 37 View Workflow Templates

These mappings can be edited or deleted by performing a right click on them.

Refer to the HP UCA Automation Integrators Guide for details on overall procedure for creating automated correlation/actions using UCA Automation.

Refer to the HP UCA for EBC Value Pack Development Guide for details on development of a value pack.