# 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**

# 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



## Trademark Notices

Adobe®, Acrobat® and PostScript® 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™ is a trademark of Oracle and/or its affiliates.

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

Firefox® is a registered trademark of the Mozilla Foundation.

Google Chrome® is a trademark of Google Inc.

Oracle® is a registered U.S. trademark of Oracle Corporation, Redwood City, California.

EnterpriseDB® is a registered trademark of EnterpriseDB.

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

UNIX® is a registered trademark of The Open Group.

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

Red Hat® is a registered trademark of the Red Hat Company.

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

Neo4j is a trademark of Neo Technology.

# Contents

# Figures

# 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.

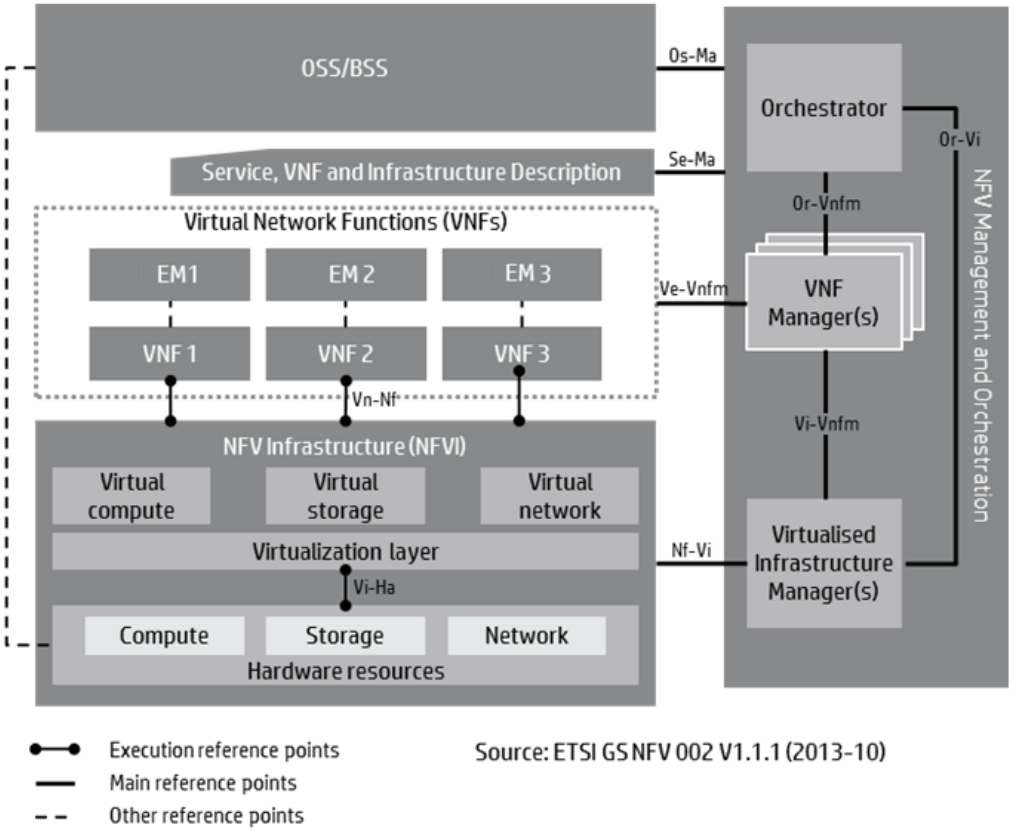
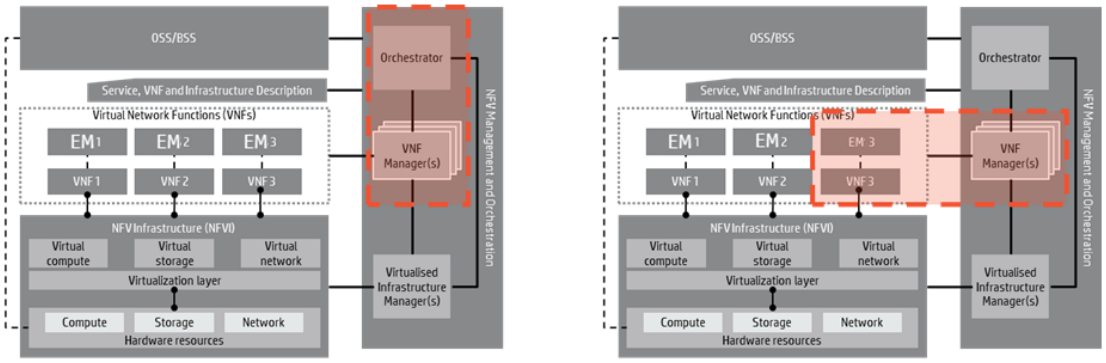



**Figure 1 ETSI Reference Architecture**

## 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.




Generic VNF manager coupled with NFV orchestrator

VNF manager coupled with EM

- Responsibility of VNF provider

**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

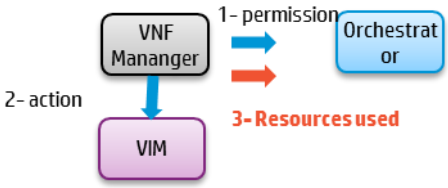



**Figure 4 Direct VNFM**

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

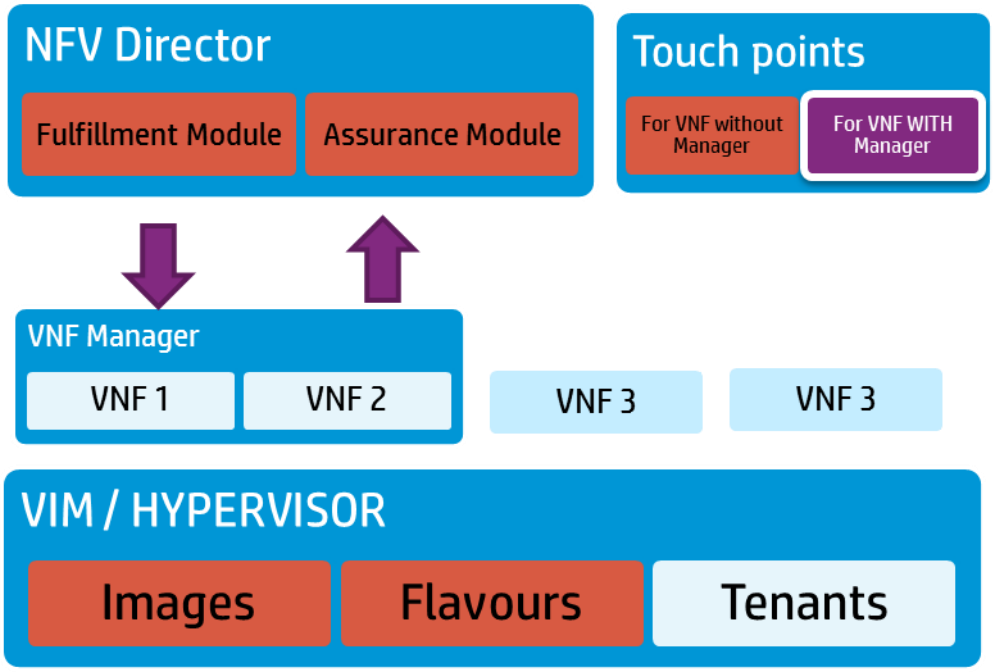



**Figure 5 VNF Integration Touch Points**

## 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?
    - 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
    - All the VMs MUST run in different server
    - 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?
    - 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?
    - 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)
    - 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)
    - 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
    - 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

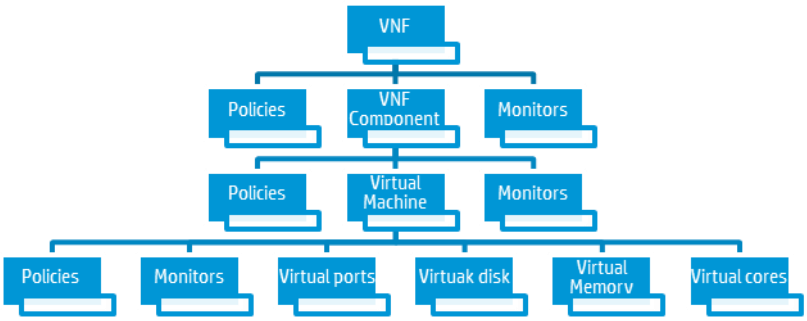



**Figure 6 VNF Descriptor**

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

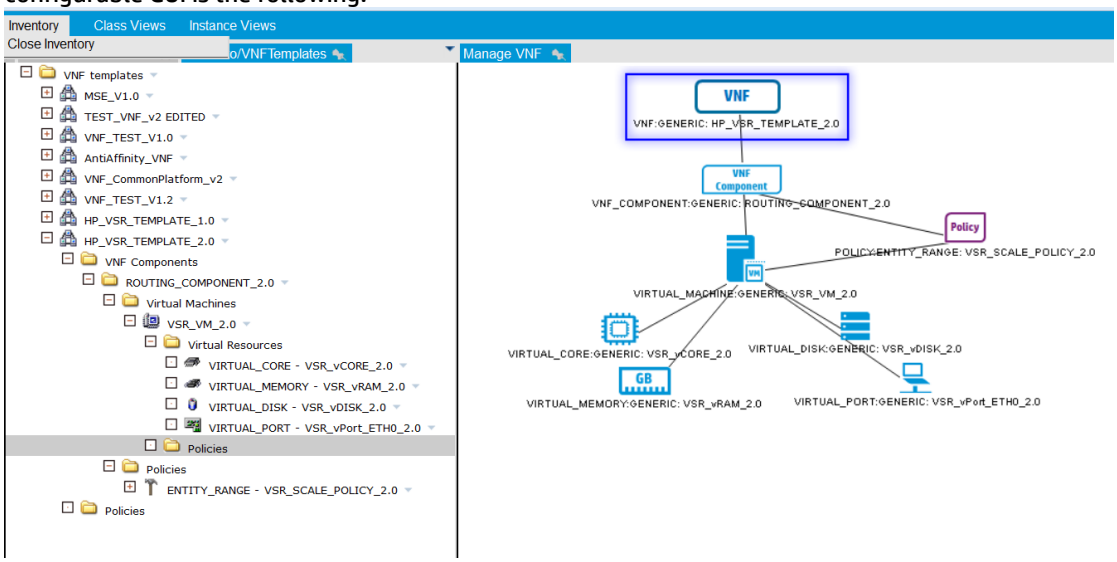


**Figure 7 VNF Director Configurable GUI**

## 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

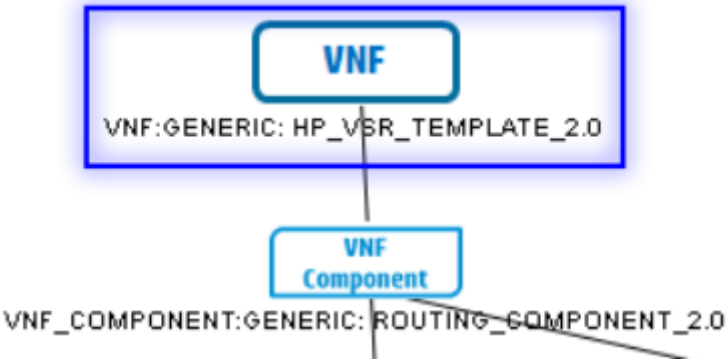



**Figure 8 Example XML template of VNF Level**

```
<templates xmlns="http://www.model.bll.nfv.activator.ov.hp.com">
<languageCode xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">en EN</languageCode>
<artifactTemplates>

        <artifactTemplate>
                <artifactFamily>VNF</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>HP VSR TEMPLATE 2.0</value>
                                        </attribute>
                                </attributes>
```

```
                                <categories/>
                        </category>
                </categories>
                <id>HP VSR TEMPLATE 2.0</id>
                <identifier>HP_VSR_TEMPLATE_2.0</identifier>
                <status>
                        <label>ENABLED</label>
                        <visibleStatusLabel>ENABLED</visibleStatusLabel>
                        <enabled>true</enabled>
                </status>
        </artifactTemplate>

</artifactTemplates>
<relationshipTemplates>

        <relationshipTemplate>
                <type>INCLUDE</type>
                <parentId>HP_VSR_TEMPLATE_2.0</parentId>
                <parentArtifactFamily>VNF</parentArtifactFamily>
                <parentArtifactCategory>GENERIC</parentArtifactCategory>
                <childId>ROUTING_COMPONENT_2.0</childId>
                <childArtifactFamily>VNF COMPONENT</childArtifactFamily>
                <childArtifactCategory>GENERIC</childArtifactCategory>
                <enabled>true</enabled>
                <status>
                        <label>ENABLED</label>
                        <visibleStatusLabel>ENABLED</visibleStatusLabel>
                        <enabled>true</enabled>
                </status>
        </relationshipTemplate>

</relationshipTemplates>
</templates>
```

### 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
  - Example 1: A policy to set up some attributes of the VNF Component
  - 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

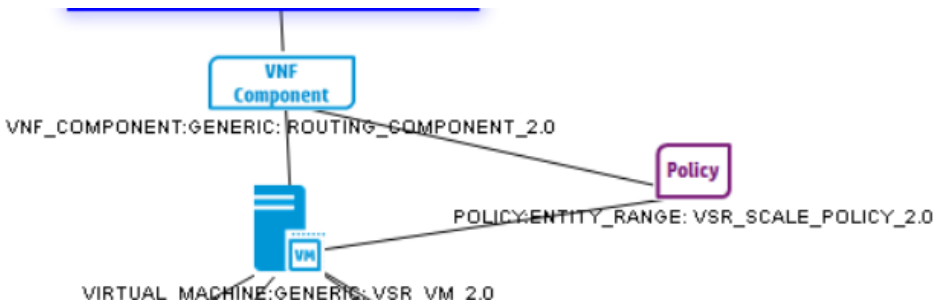



**Figure 9 Example XML Template of VNF Component Level**

```
<templates xmlns="http://www.model.bll.nfv.activator.ov.hp.com">
<languageCode xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema"
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:
  - Example 1; A policy to set up some attributes of the VM
  - 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.

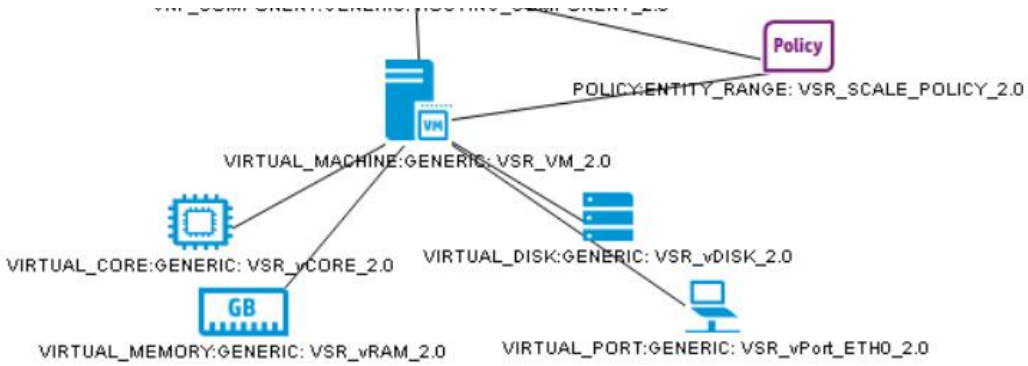



**Figure 10 Example XML Template of Virtual Machine Level**

```
<templates xmlns="http://www.model.bll.nfv.activator.ov.hp.com">
<languageCode xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">en_EN</languageCode>
<artifactTemplates>


        <artifactTemplate>
                <artifactFamily>VIRTUAL MACHINE</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>VSR_VM_2.0</value>
                                        </attribute>
                                </attributes>
                                <categories/>
                        </category>
                </categories>
                <id>VSR_VM_2.0</id>
                <identifier>VSR VM 2.0</identifier>
                <status>
                        <label>ENABLED</label>
                        <visibleStatusLabel>ENABLED</visibleStatusLabel>
                        <enabled>true</enabled>
                </status>
        </artifactTemplate>

        <artifactTemplate>
                <artifactFamily>POLICY</artifactFamily>
                <artifactCategory>ENTITY_RANGE</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>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</parentId>
                <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

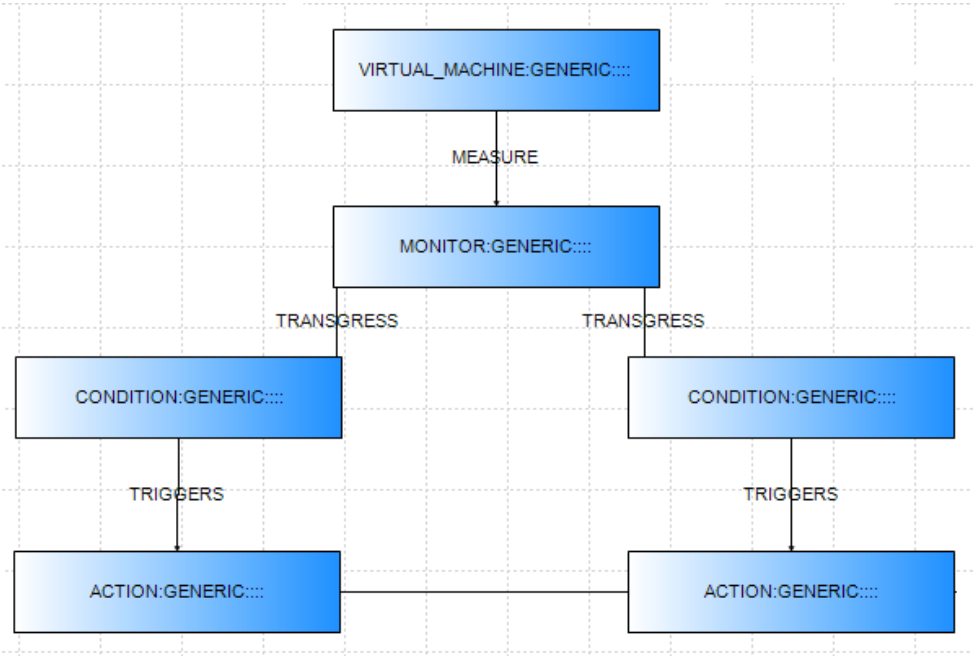



**Figure 11 Configure Monitors**

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.

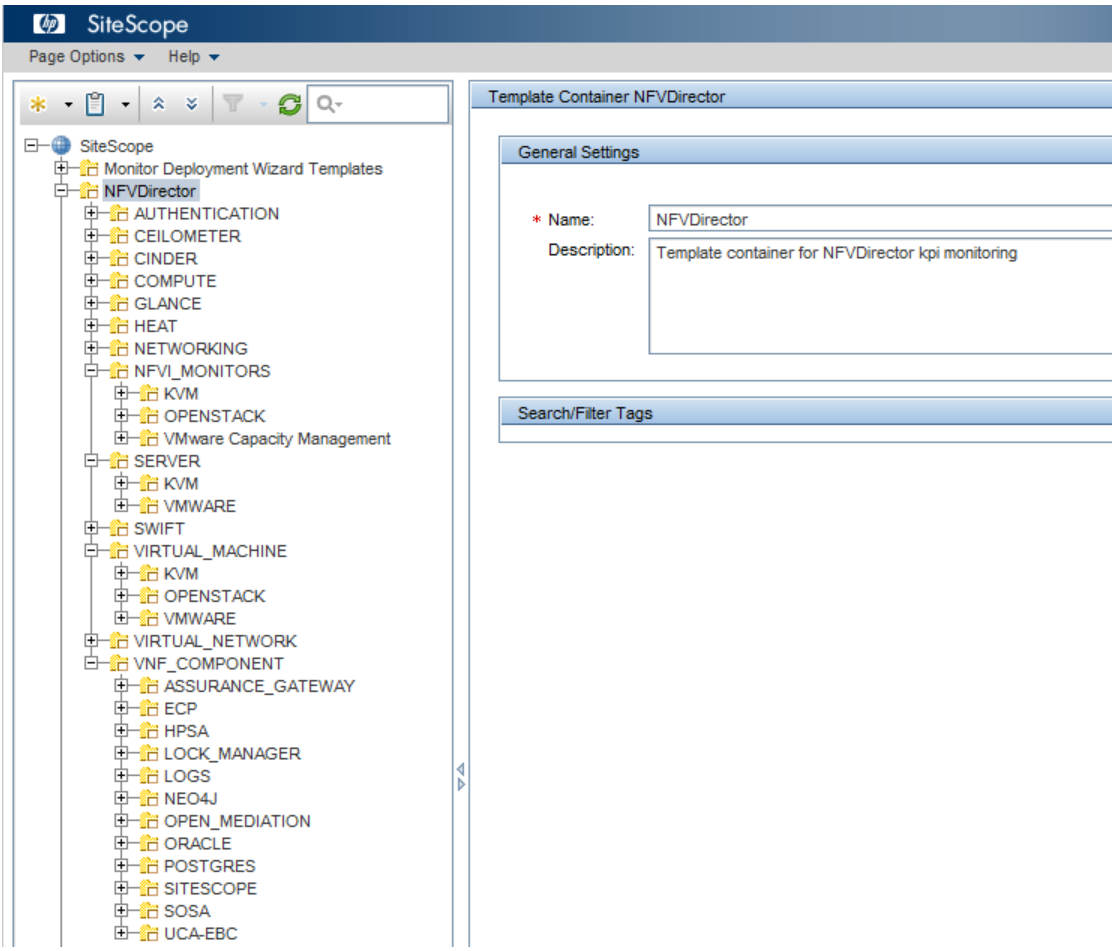



**Figure 12 SiteScope NFVD Templates**

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.

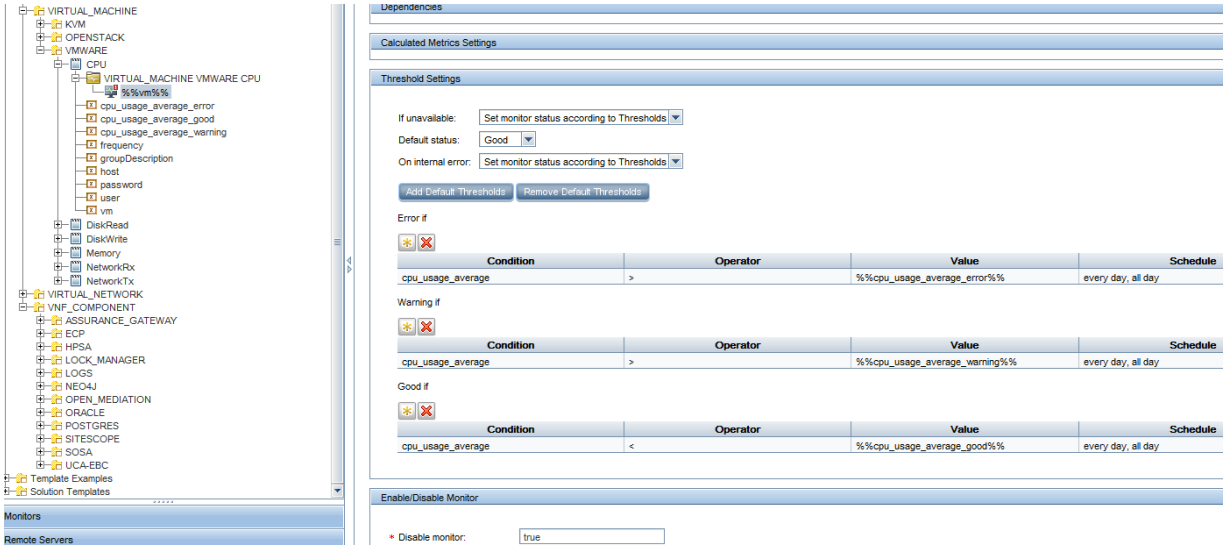



**Figure 13 SiteScope NFVD Monitor Template Parameters**

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):

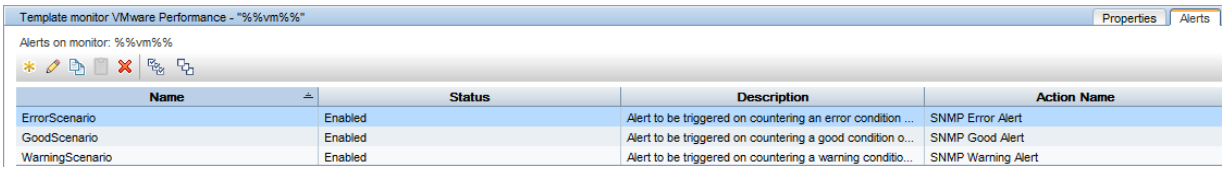



**Figure 14 SiteScope Alerts Definition 1**

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).

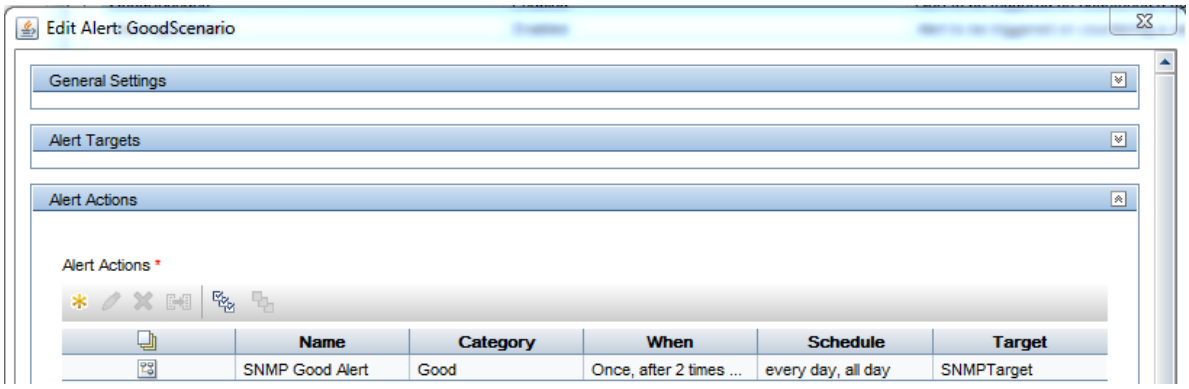



**Figure 15 SiteScope Alerts Definition 2**

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

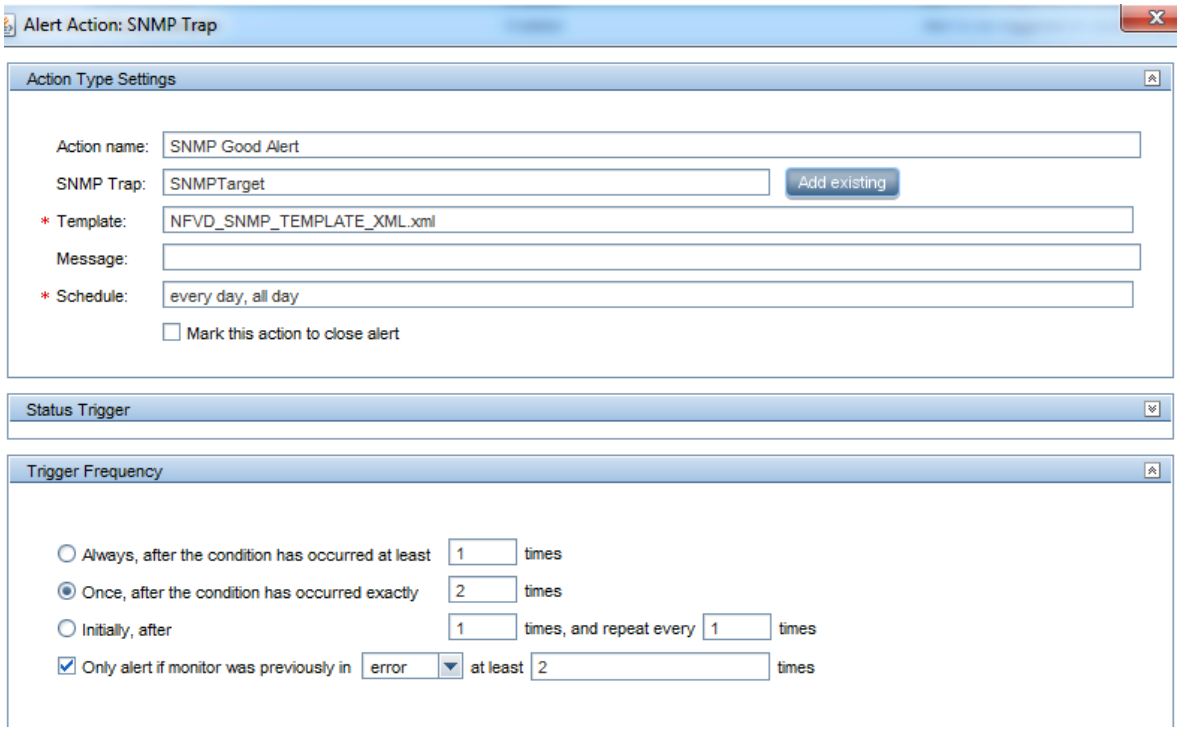



**Figure 16 SiteScope Alerts configuration**

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-
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=".1.3.6.1.4.1.11.2.53.2.2.3.1.2.1.11" 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=<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>

<SourceIdentifier>NFVD_Source</SourceIdentifier>
<OriginatingManagedEntity></OriginatingManagedEntity>
<Problem>SCALE_IN_CPU</Problem> <ActionPreset>true</ActionPreset>
</ActionRequest> </header> <body> <Parameters> <Parameter>
<attribute>ArtifactInstanceId</attribute> <value>14066367111091</value>
</Parameter> </Parameters> <Service> <serviceTypeID>NFVD</serviceTypeID>
<serviceInstanceID></serviceInstanceID> </Service> <Resource>
<resourceTypeID>INVENTORY</resourceTypeID>
<resourceInstanceID>14066367111091</resourceInstanceID> </Resource> </body><
/msg>
```

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**.

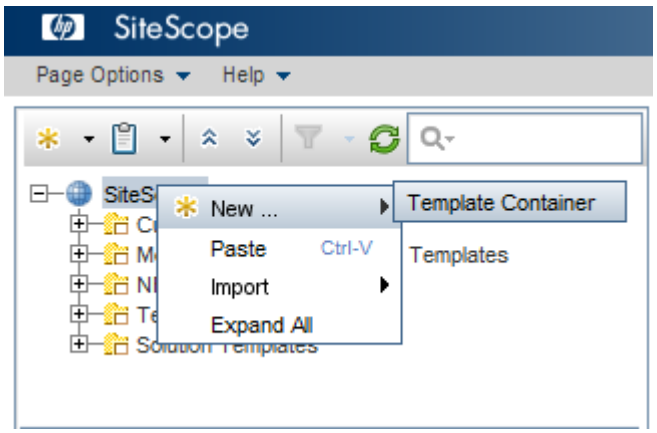



**Figure 17: SiteScope: Create custom templates**

3. Enter the name of the template container and click the **OK** button.
4. Right-click this new template container node and select **New** > **Template**.

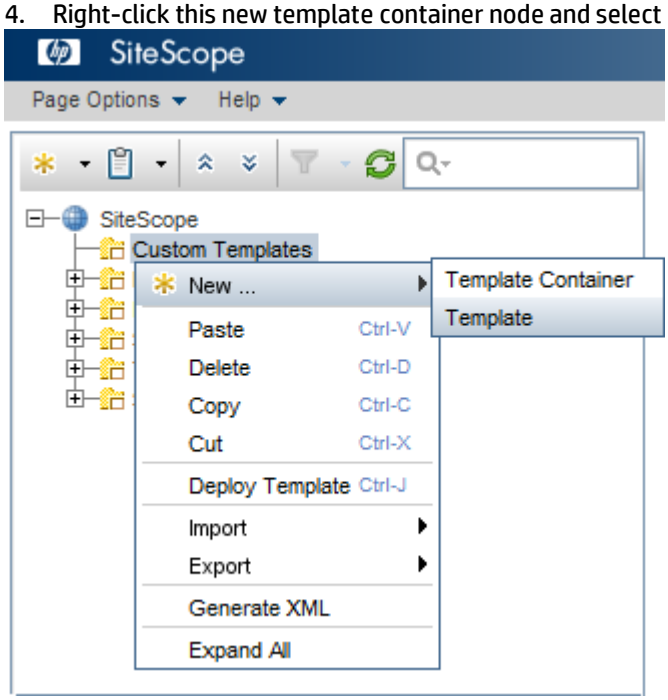



**Figure 18: SiteScope: new template**

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**.

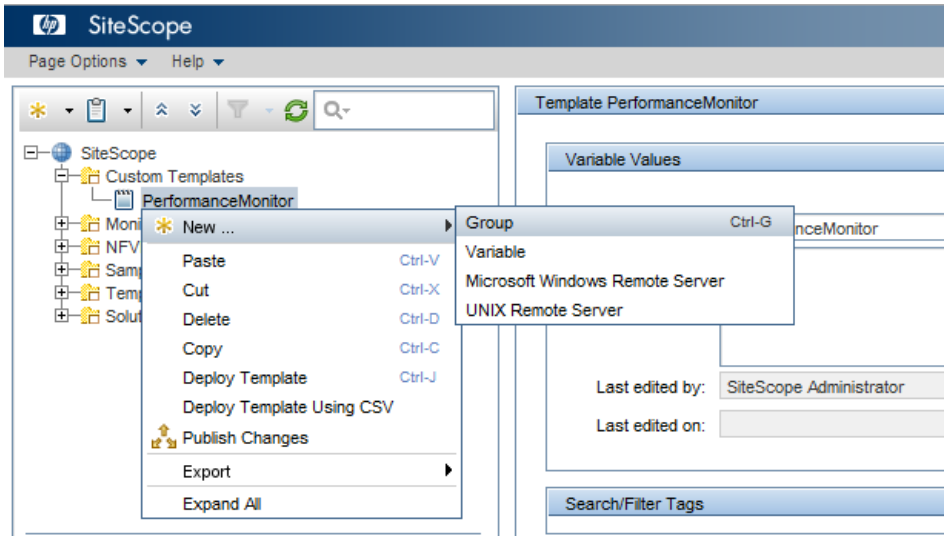



**Figure 19: SiteScope: new group**

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**.

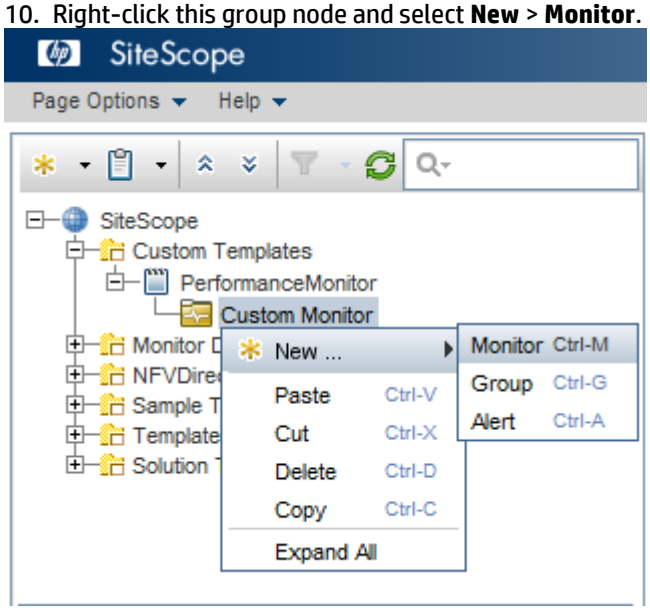



**Figure 20: SiteScope: new Monitor**

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**.

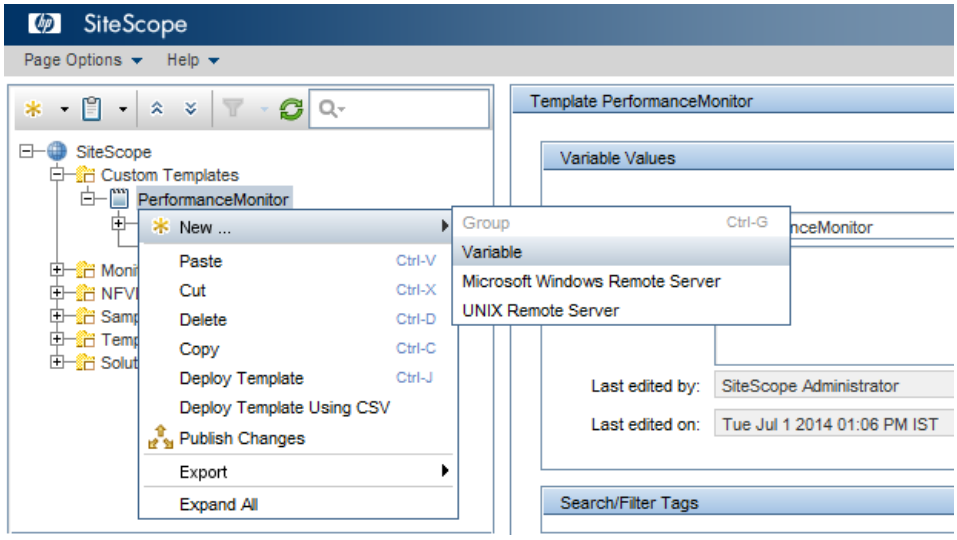



**Figure 21: SiteScope: new Variable**

The following window opens.

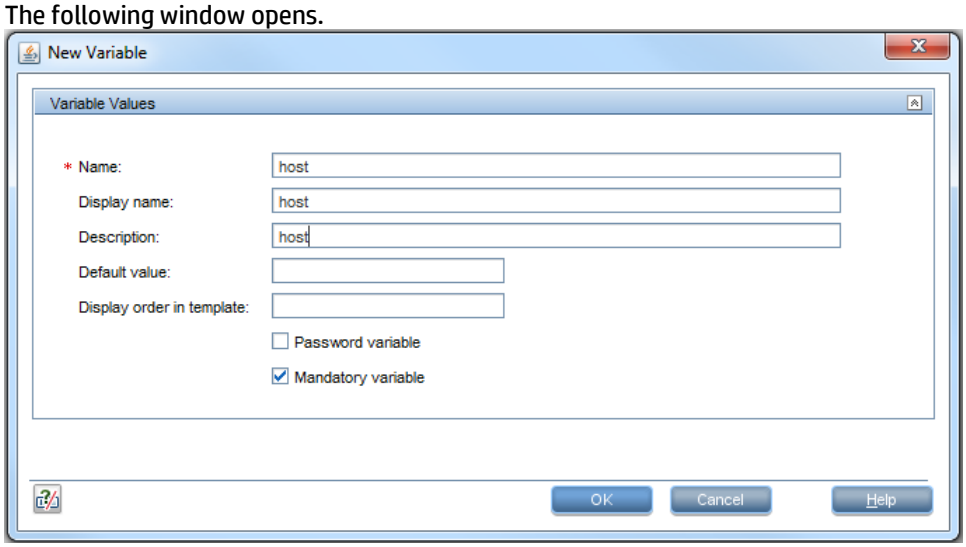



**Figure 22: SiteScope: new Variable details**

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.

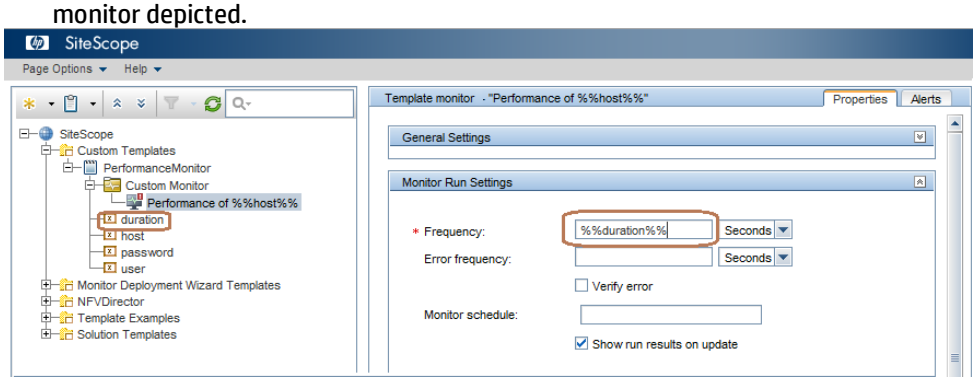


**Figure 23: SiteScope: enter variable to associate with template**

Following is the complete hierarchy of the Custom Template created.

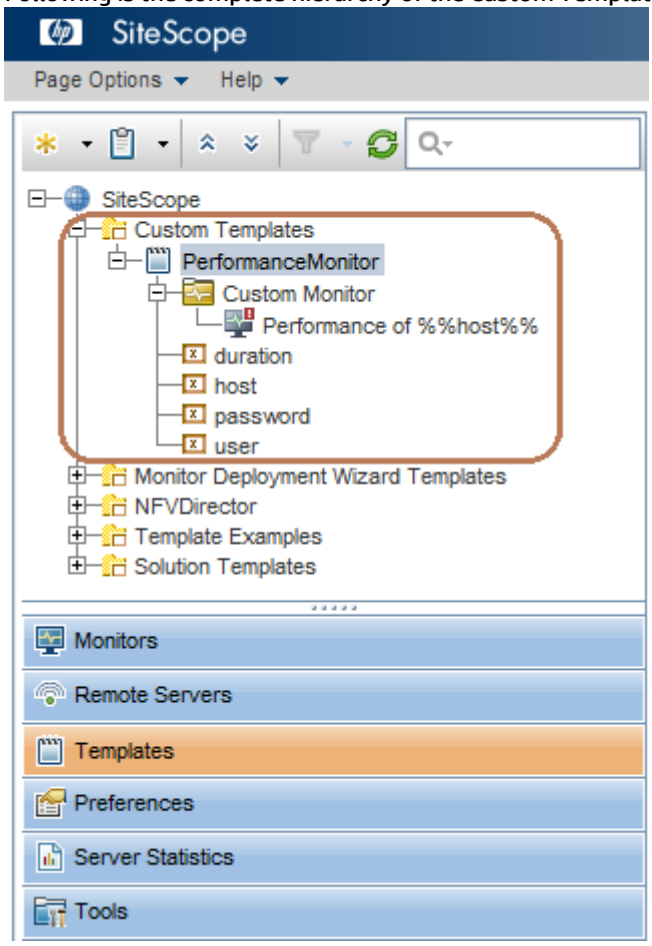



**Figure 24: SiteScope: hierarchy of custom template**

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">
    <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>
        <classInstance instance="ossvm1.ind.hp.com" class="Host"/>
    </originatingManagedEntityStructure>
    <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-
001|CONDITION=ERROR|group=KVM VM CPU
Monitor|groupdescription=CPU|groupID=201070542|id=1</additionalText>
    <notificationIdentifier>0</notificationIdentifier>
    <correlationNotificationIdentifiers>Attribute not
available</correlationNotificationIdentifiers>
</AlarmCreationInterface>
```

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

| | |
|---|---|
| <additionalText> | The value should contain groupdescription=<MonitorName> and <MonitorName> should be the value of property GENERAL.Name 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:

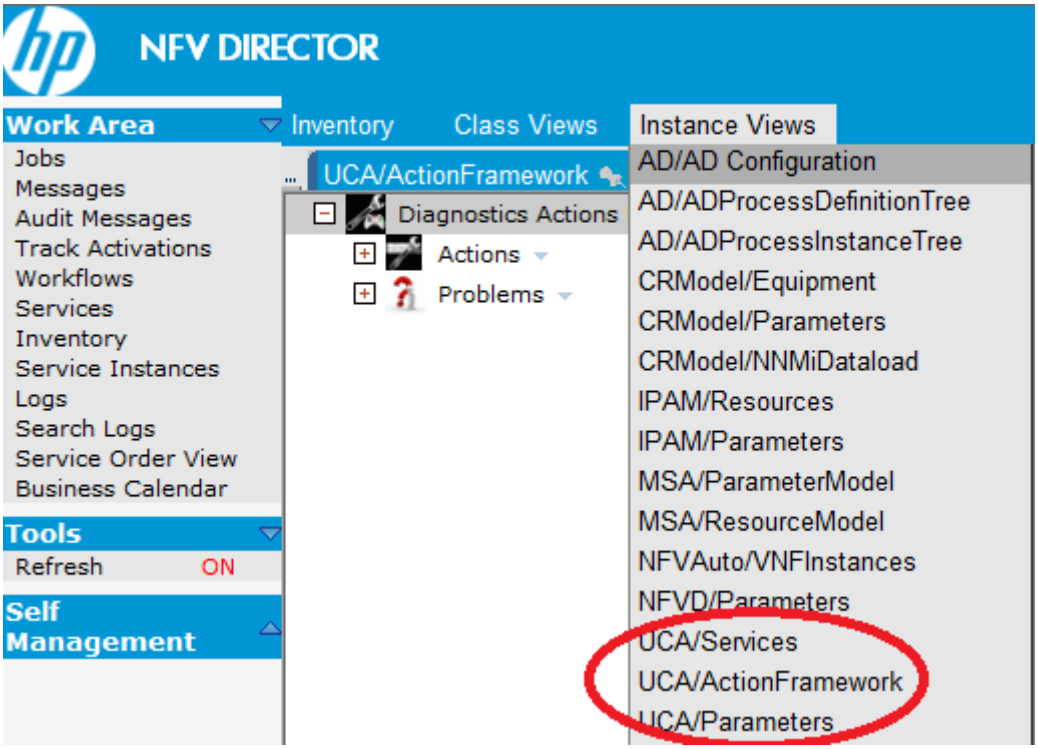


**Figure 25 UCA's HPSA Framework**

**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)

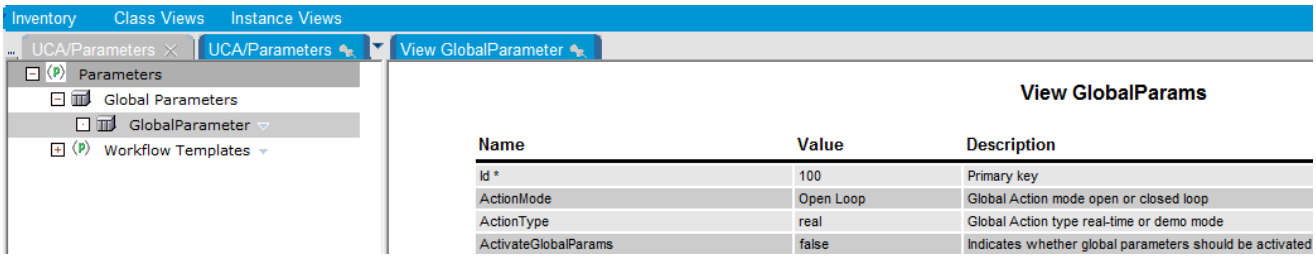



**Figure 26 HPSA Inventory – UCA/Parameters – Global Parameters**

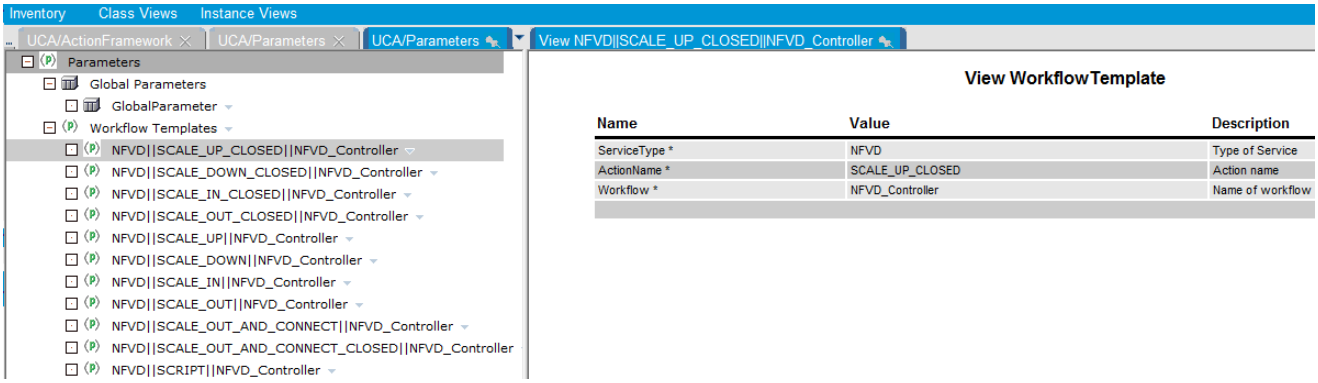



**Figure 27 HPSA Inventory - UCA/Parameters Workflow Templates**

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.

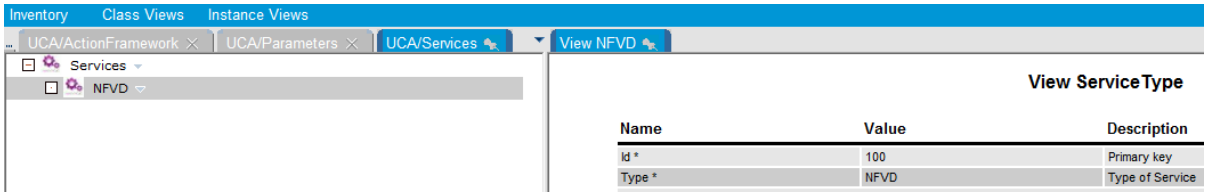



**Figure 28 UCA Automation Foundation Inventory – UCA/Services**

Action definitions for NFVD

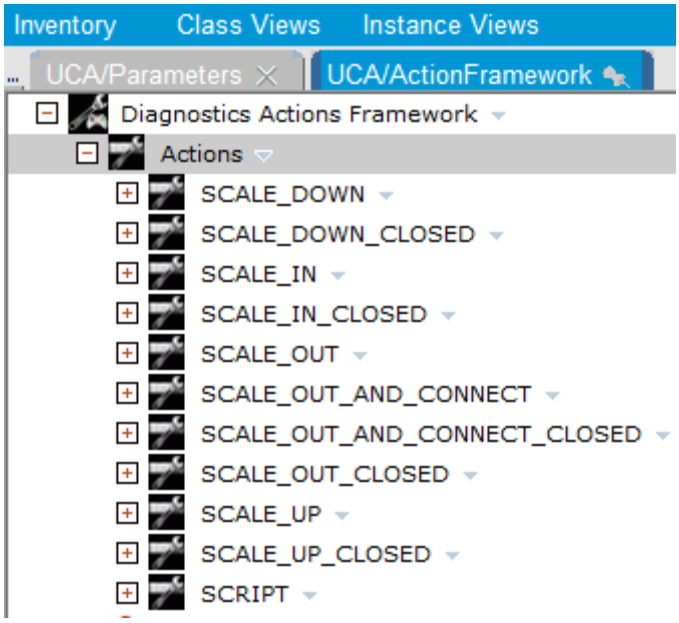



**Figure 29 UCA Automation Foundation Inventory – UCA/Services**

Problem definitions for NFVD:

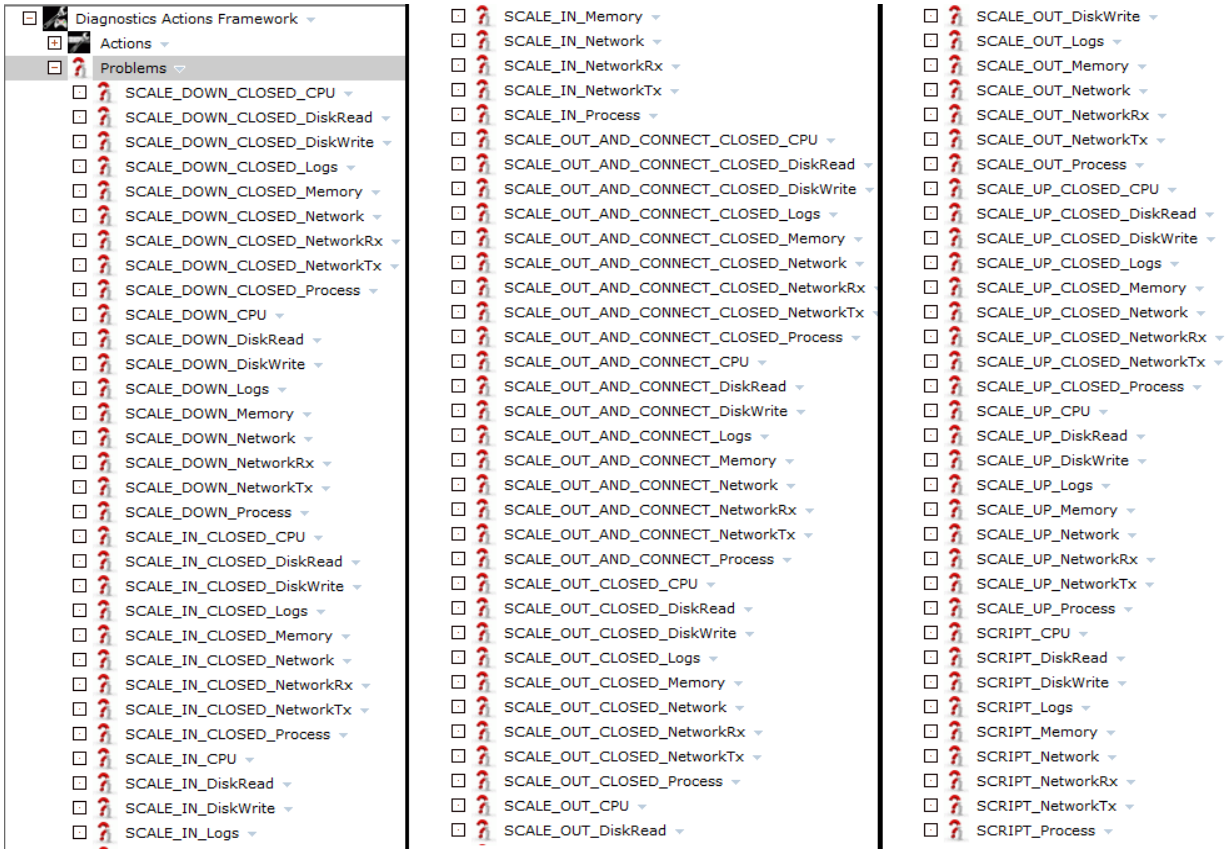



**Figure 30 UCA Automation Foundation Inventory – UCA/ActionFramework (Problems)**

NFVD Controller workflow mapping definition:

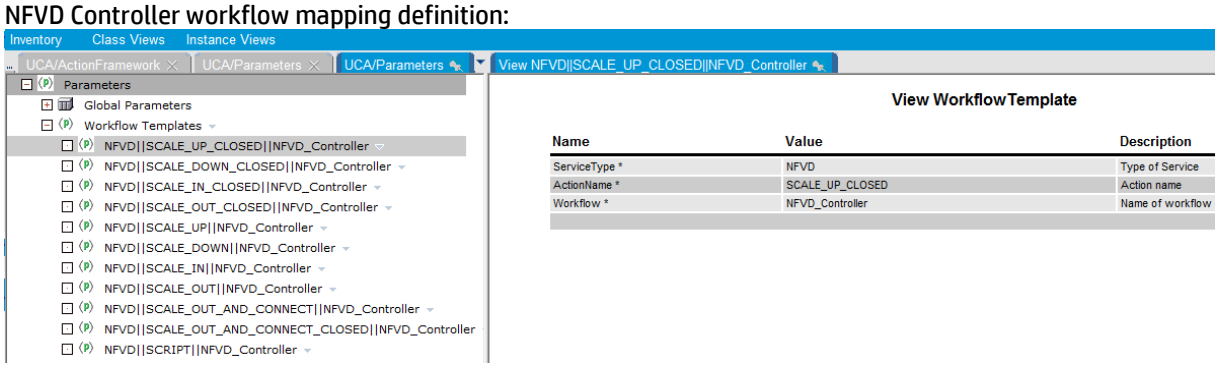



**Figure 31 UCA Automation Foundation Inventory – UCA/Parameters**

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:

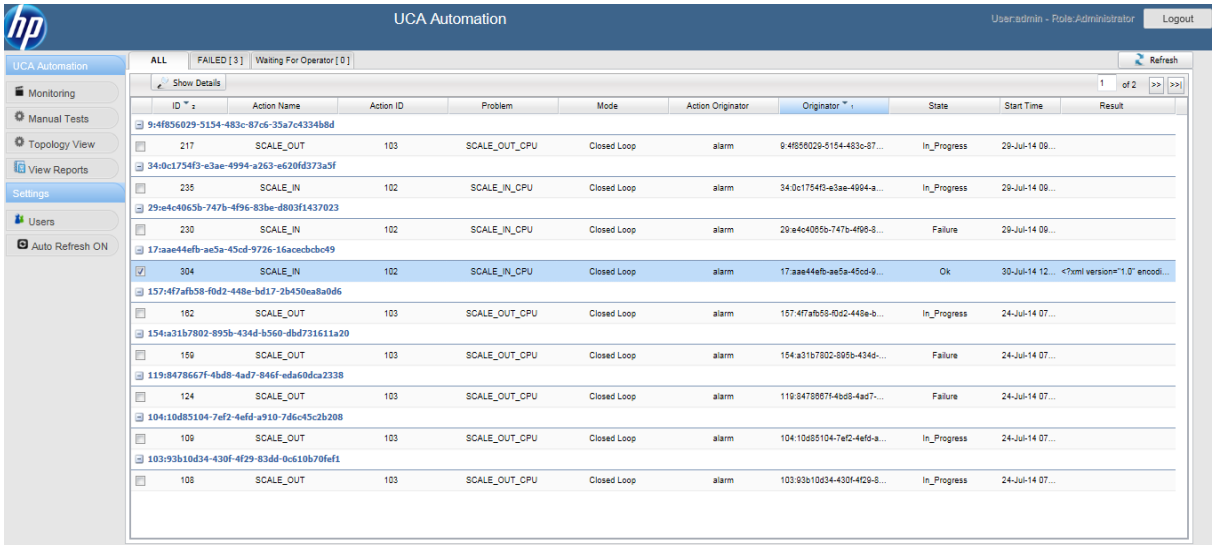



**Figure 32 UCA Automation Monitoring Screen**

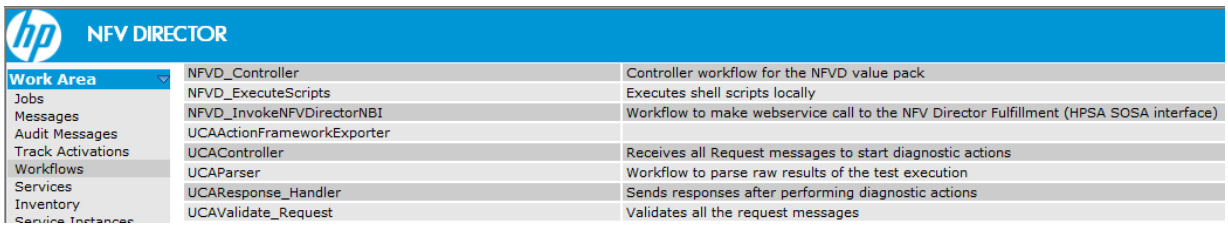



**Figure 33 HPSA UCA NFVD Workflows**

## 2.9.2 Inventory data populated in NFVD value pack inventory

Action mapping to NFVD child workflows

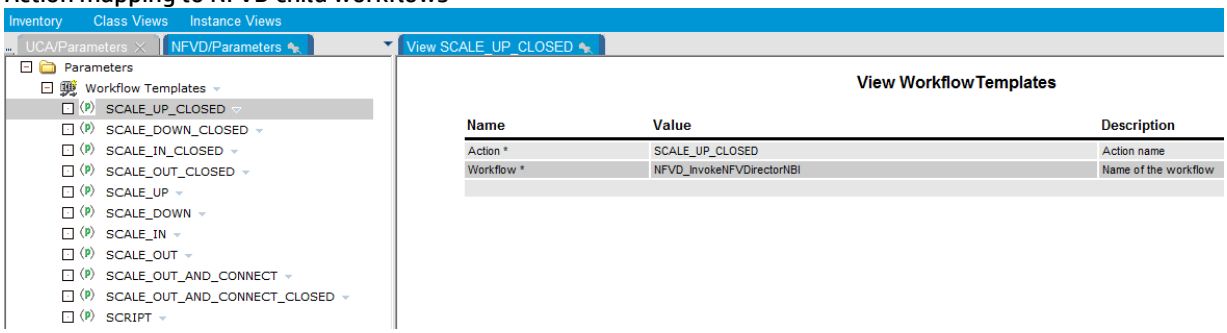



**Figure 34 NFVD Inventory – NFVD/Parameters**

## 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 → Parameters → Create Workflow Template.

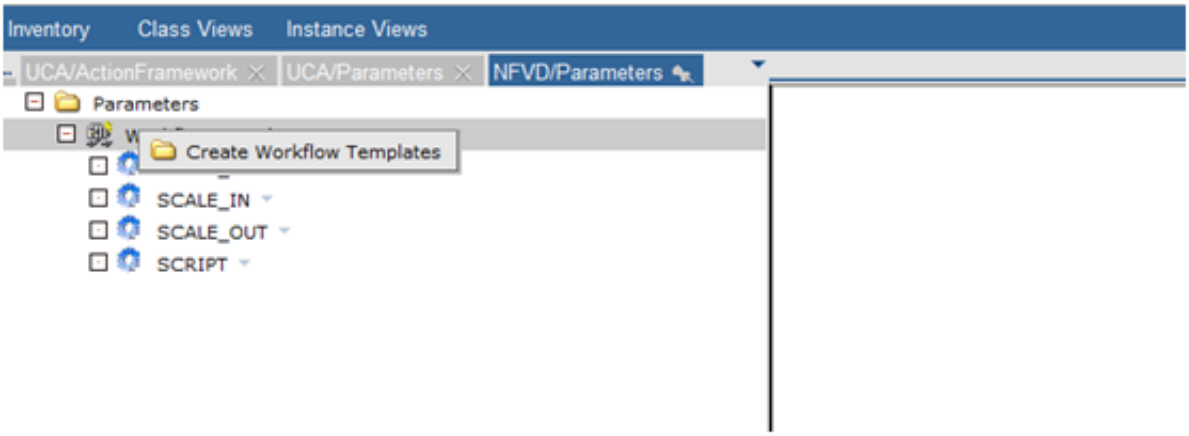



**Figure 35 NFVD/Parameters**

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

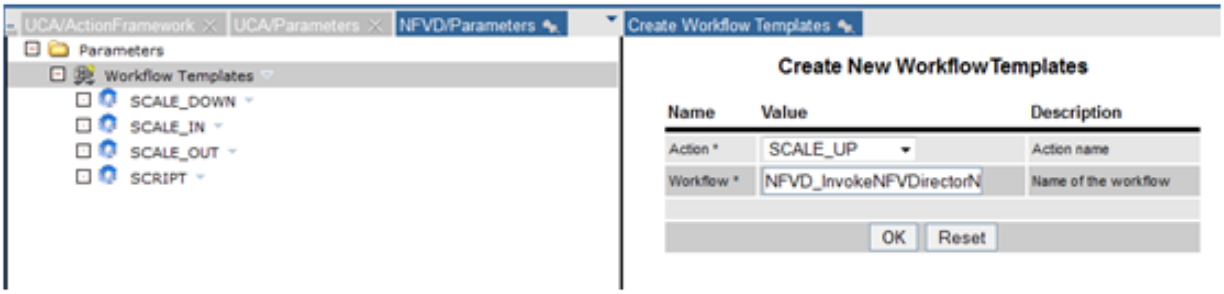



**Figure 36 Creating new Workflow Templates**

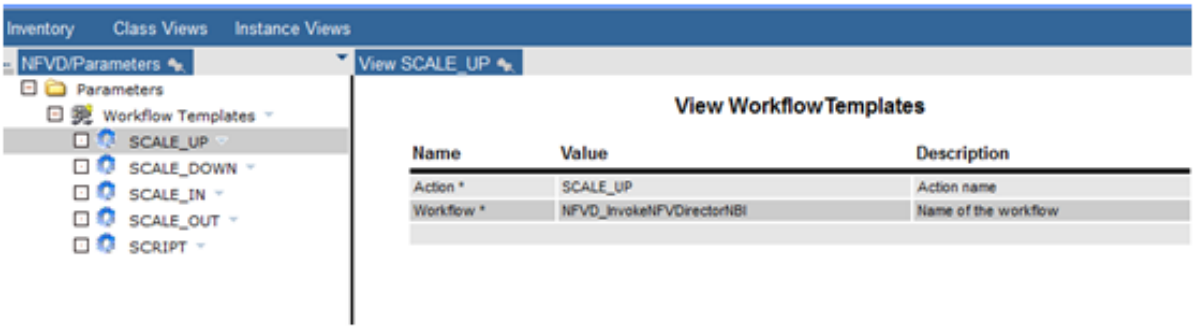



**Figure 37 View Workflow Templates**

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.