



HPE NFV Director

Integrator guide for VIM Managed mode

Release 4.2

First Edition



Hewlett Packard
Enterprise

Notices

Legal notice

© Copyright 2017 Hewlett Packard Enterprise Development LP

Confidential computer software. Valid license from HPE 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.

The information contained herein is subject to change without notice. The only warranties for HPE 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. HPE shall not be liable for technical or editorial errors or omissions contained herein.

Printed in the US

Trademarks

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

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

Adobe®, Acrobat® and PostScript® are trademarks of Adobe Systems Incorporated.

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.

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.

Apache CouchDB, CouchDB, and the project logo are trademarks of The Apache Software Foundation.

Node.js project. Joyent® and Joyent's logo are registered trademarks of Joyent, Inc.

Neo4j is a trademark of Neo Technology.

VMware ESX, VMWare ESXi, VMWare vCenter and VMWare vSphere are either registered trademarks or trademarks of VMware, Inc. in the United States and other jurisdictions.

Contents

Notices	1
About this guide.....	8
Audience	8
Document history	8
Chapter 1 Operations tree.....	9
Chapter 2 Deploy of an Organization – Bottom-Up.....	10
2.1 Specific Elements of the TLD Deploy Organization.....	11
2.2 TLD ORGANIZATION STATUS CHANGE: Organization Status Change.....	11
Chapter 3 Deploy of a Tenant – Bottom-Up.....	12
3.1 Specific Elements of the TLD Deploy Tenant.....	13
3.2 TLD DEPLOY TENANT: Quota Assignment Task.....	13
3.3 . TLD STATUS CHANGE: Tenant_Status_Change task.....	14
Chapter 4 Deploy of a VNF Group – Bottom-Up.....	15
4.1 Specific Elements of the TLD Deploy VNF Group.....	16
4.2 TLD DEPLOY VNF_GROUP: Quota Assignment Task.....	16
4.3 . TLD CHANGE STATUS: VNF Group Status Change task.....	17
Chapter 5 Deploy of a VNF – Bottom-Up.....	18
5.1 Specific Elements of the TLD Deploy VNF.....	19
5.2 TLD DEPLOY VNF: Quota Assignment.....	19
5.3 TLD RESOURCE ASSIGNMENT: ALLOCATE RESOURCES VNF.....	20
5.4 TLD RESOURCE ASSIGNMENT: ALLOCATE RESOURCES VNFC.....	21
5.5 TLD RESOURCE ASSIGNMENT: ALLOCATE RESOURCES VNFC WITH AFFINITY.....	22
5.6 TLD RESOURCE ASSIGNMENT: ALLOCATE RESOURCES VNF WITH AFFINITY.....	23
5.7 TLD IMAGE PERMISSION: GET IMAGE.....	24
5.8 TLD VIM SPECS: CREATE_KEYPAIR.....	25
5.9 TLD VIM SPECS: GET FLAVOR.....	26
5.10 TLD VIM SPECS: CONNECT_SUBNET.....	27
5.11 TLD PREPROCESSING: PREPROCESS.....	28
5.12 TLD DEPLOY MONITOR VOLUME: MONITOR DEPLOY.....	29
5.13 TLD DEPLOY MONITOR VOLUME: ACTIVATE_FLAVOR.....	30
5.14 TLD DEPLOY MONITOR VOLUME: CREATE VOLUME.....	31
5.15 TLD DEPLOY MONITOR VOLUME: ACTIVATE_KEYPAIR.....	32
5.16 TLD ACTIVATE PORT: ACTIVATE PORT.....	33
5.17 TLD ACTIVATE: ACTIVATE_VM.....	34
5.18 TLD ATTACH: ATTACH VOLUME.....	35
5.19 TLD POSTPROCESSING: POSTPROCESS.....	36
5.20 . TLD START MONITORS: MONITOR START.....	37
5.21 . TLD VNF STATUS CHANGE: VNF_STATUS_CHANGE.....	38
Chapter 6 Scale In of a VNF – Bottom-Up.....	39
6.1 Specific Elements of the TLD Scale In VNF.....	41
6.2 TLD SCALE IN: SELECT COMPONENTS.....	41
6.3 SCALE IN PRE TLD: PREPROCESS.....	42
6.4 TLD STOP MONITOR: STOP MONITOR.....	43
6.5 TLD STOP MONITOR: DETACH VOLUME.....	44
6.6 TLD DEACTIVATE VM: DEACTIVATE VM.....	45
6.7 TLD UNDEPLOY POST: UNDEPLOY POST.....	46
6.8 TLD DELETE VPORT: DELETE PORT.....	47
6.9 TLD UNDEPLOY MONITOR VOLUME: DELETE VOLUME.....	48
6.10 TLD UNDEPLOY MONITOR VOLUME: UNDEPLOY_MONITOR.....	49
6.11 . Delete Inventory: DELETE_INVENTORY_COMPONENT_LEVEL.....	50
6.12 . TLD Delete Inventory VM: DELETE_INVENTORY_VM_LEVEL.....	51
Chapter 7 Scale Out of a VNF – Bottom-Up.....	52
7.1 Specific Elements of the TLD Scale Out of a VNF.....	54
7.2 TLD SCALE OUT: SCALE INVENTORY.....	54
7.3 TLD QUOTA ASSIGNMENT: QUOTA ASSIGNMENT.....	55
7.4 TLD RESOURCE ASSIGNMENT: ALLOCATE RESOURCES VNF WITH AFFINITY.....	56
7.5 TLD RESOURCE ASSIGNMENT: ALLOCATE RESOURCES VNF.....	57
7.6 TLD IMAGE: GET IMAGE.....	58
7.7 TLD VIM SPECS: GET_KEYPAIR.....	59

7.8 TLD VIM SPECS: GET FLAVOR	60
7.9 TLD VIM SPECS: CONNECT ENDPOINTS	61
7.10 TLD VIM SPECS: CONNECT_SUBNET	62
7.11 TLD PREPROCESSING: PREPROCESS	63
7.12 TLD DEPLOY MONITOR VOLUME: ACTIVATE KEYPAIR	64
7.13 TLD DEPLOY MONITOR VOLUME: ACTIVATE_FLAVOR	65
7.14 TLD DEPLOY MONITOR VOLUME: MONITOR DEPLOY	66
7.15 TLD DEPLOY MONITOR VOLUME: CREATE VOLUME	67
7.16 TLD ACTIVATE PORT: ACTIVATE PORT	68
7.17 TLD ACTIVATE: ACTIVATE VM	69
7.18 TLD ATTACH: ATTACH VOLUME	70
7.19 TLD POSTPROCESSING: POSTPROCESS	71
7.20 TLD START MONITOR: MONITOR START	72
Chapter 8 Start VM – Bottom-Up	73
8.1 Specific Elements of the TLD Start VM	74
8.2 TLD START VM: START_VM	74
8.3 TLD START MONITOR: START MONITOR	75
Chapter 9 Stop VM – Bottom-Up	76
9.1 Specific Elements of the TLD Stop VM	77
9.2 TLD STOP MONITOR: STOP MONITOR	77
9.3 TLD STOP VM: STOP_VM	78
Chapter 10 Undeploy of an Organization – Bottom-Up	79
10.1 Specific Elements of the TLD Undeploy Organization	80
10.2 TLD UNDEPLOY ORGANIZATION: Undeploy Check	80
10.3 TLD DELETE TREE: Delete Organization Tree	81
Chapter 11 Undeploy of a Tenant – Bottom-Up	82
11.1 Specific Elements of the TLD Undeploy Tenant	83
11.2 TLD UNDEPLOY TENANT: Undeploy Check	83
11.3 TLD DELETE IDS: DELETE VDC ID OF OS TENANTS	84
11.4 TLD DELETE TREE: Change Status Instantiated	85
11.5 TLD DELETE TREE: Delete Tenant Tree	86
Chapter 12 Undeploy of a Virtual Link – Bottom-Up	87
12.1 Specific Elements of the TLD Undeploy Virtual Link	88
12.2 TLD UNDEPLOY VIRTUAL LINK : UNDEPLOY CHECK	88
12.3 TLD CHECK MULTIPLE VDC : CHECK MULTIPLE VDC	89
12.4 TLD DEACTIVATE OPENSTACK SUBNET: CHECK SCOPE	90
12.5 TLD DEACTIVATE OPENSTACK SUBNET: DEACTIVATE_SUBNETWORK_OPENSTACK	91
12.6 TLD DEACTIVATE OPENSTACK NET: DEACTIVATE_NETWORK_OPENSTACK	92
12.7 TLD INVENTORY DELETE NETWORKS: DELETE NETWORK	93
12.8 TLD DELETE IDS: DELETE NETWORK SUBNETWORK IDS	94
12.9 TLD INVENTORY DELETE VIRTUAL LINK: VIRTUAL LINK INVENTORY DELETE	95
Chapter 13 Undeploy of a VNF Group – Bottom-Up	96
13.1 Specific Elements of the TLD Undeploy VNF Group	97
13.2 TLD UNDEPLOY VNF GROUP: Undeploy Check	97
13.3 TLD DELETE VNF_GROUP: VNF_GROUP Delete	98
Chapter 14 Undeploy of a VNF - Bottom-Up	99
14.1 Specific Elements of the TLD Undeploy VNF	100
14.2 TLD UNDEPLOY VNF: PREPROCESS	100
14.3 TLD STOP MONITOR: STOP MONITOR	101
14.4 TLD DEACTIVATE VM : DEACTIVATE VM	102
14.5 TLD UNDEPLOY POST: UNDEPLOY POST	103
14.6 TLD DELETE VPORT: DELETE PORT	104
14.7 LD Undeploy Monitor, Volume: UNDEPLOY_MONITOR	105
14.8 TLD Undeploy Monitor, Volume: DELETE VOLUME	106
14.9 TLD VNF Inventory Delete: DELETE_INVENTORY	107
Abbreviations	108

List of Figures

<i>Figure: 1 Deployment tree.....</i>	<i>9</i>
<i>Figure: 2 Organization Status Change TLD.....</i>	<i>11</i>
<i>Figure: 3 Quota Assignment task.....</i>	<i>13</i>
<i>Figure: 4 Change status of the Tenant.....</i>	<i>14</i>
<i>Figure: 5 Quota Assignment task.....</i>	<i>16</i>
<i>Figure: 6 Change status of the VNF Group.....</i>	<i>17</i>
<i>Figure: 7 Quota Assignment task.....</i>	<i>19</i>
<i>Figure: 8 Allocate Resources for VNF.....</i>	<i>20</i>
<i>Figure: 9 Allocate Resources for VNF.....</i>	<i>21</i>
<i>Figure: 10 Allocate Resources for VNF with Affinities.....</i>	<i>22</i>
<i>Figure: 11 Allocate Resources for VNF with Affinities.....</i>	<i>23</i>
<i>Figure: 12 Get Image.....</i>	<i>24</i>
<i>Figure: 13 Get Key Pair.....</i>	<i>25</i>
<i>Figure: 14 Creation of a standard flavor.....</i>	<i>26</i>
<i>Figure: 15 Connection of the Subnetwork to the Virtual Port.....</i>	<i>27</i>
<i>Figure: 16 Deploying Pre-Process policies.....</i>	<i>28</i>
<i>Figure: 17 Deployment of a Monitor.....</i>	<i>29</i>
<i>Figure: 18 Deployment of a Flavor.....</i>	<i>30</i>
<i>Figure: 19 Activation of a Volume.....</i>	<i>31</i>
<i>Figure: 20 Activation of the KeyPair.....</i>	<i>32</i>
<i>Figure: 21 Activation of a Virtual Port.....</i>	<i>33</i>
<i>Figure: 22 Activation of Virtual Machine related to the VNF.....</i>	<i>34</i>
<i>Figure: 23 Attach Volume.....</i>	<i>35</i>
<i>Figure: 24 Deploying Post-Processing policies.....</i>	<i>36</i>
<i>Figure: 25 Starting Monitor.....</i>	<i>37</i>
<i>Figure: 26 Changing the status of the VNF.....</i>	<i>38</i>

<i>Figure: 27 Selection of the components for the Scale In.</i>	41
<i>Figure: 28 Undeployment of Pre-Processing policies</i>	42
<i>Figure: 29 Stopping monitor.</i>	43
<i>Figure: 30 Detach Volume.</i>	44
<i>Figure: 31 Deactivation of a Virtual Machine.</i>	45
<i>Figure: 32 Undeploying Post-Processing policies.</i>	46
<i>Figure: 33 Deletion of a Virtual Port.</i>	47
<i>Figure: 34 Deletion of the Volume</i>	48
<i>Figure: 35 Undeploy Monitor</i>	49
<i>Figure: 36 Deletion at component level.</i>	50
<i>Figure: 37 Delete Virtual Machine from scaled in VNF.</i>	51
<i>Figure: 38 Scale Out Operations MODE&GROUP.</i>	53
<i>Figure: 39 Scale Inventory Task.</i>	54
<i>Figure: 40 Quota Assignment.</i>	55
<i>Figure: 41 Assignment of the resources with affinity.</i>	56
<i>Figure: 42 Assignment of the resources without affinity.</i>	57
<i>Figure: 43 Checking the image permissions.</i>	58
<i>Figure: 44 Creation the Keypair element.</i>	59
<i>Figure: 45 Get Flavor.</i>	60
<i>Figure: 46 Connection of the Endpoints.</i>	61
<i>Figure: 47 Connection of the Subnetwork to the VPort.</i>	62
<i>Figure: 48 Deploying Pre-Process policies.</i>	63
<i>Figure: 49 Activation of the Keypair.</i>	64
<i>Figure: 50 Activation of the Flavor.</i>	65
<i>Figure: 51 Deployment of a Monitor.</i>	66
<i>Figure: 52 Activation of the Volume.</i>	67
<i>Figure: 53 Activation of a Virtual Port.</i>	68
<i>Figure: 54 Activation of Virtual Machine related to the VNF.</i>	69
<i>Figure: 55 Attaching of the Volume used.</i>	70

<i>Figure: 56 Deploying Post-Processing policies.</i>	71
<i>Figure: 57 Starting Monitor.</i>	72
<i>Figure: 58 Start Virtual Machine</i>	74
<i>Figure: 59 Start Monitor.</i>	75
<i>Figure: 60 Stop Monitor</i>	77
<i>Figure: 61 Stop Virtual Machine.</i>	78
<i>Figure: 62 Undeploy check.</i>	80
<i>Figure: 63 Organization Delete.</i>	81
<i>Figure: 64 Undeploy Check.</i>	83
<i>Figure: 65 Delete ID of TENANT OS</i>	84
<i>Figure: 66 Deactivating Egress policies for the Tenant.</i>	85
<i>Figure: 67 Delete Tenant.</i>	86
<i>Figure: 68 Undeploy Check.</i>	88
<i>Figure: 69 Check Multiple VDC.</i>	89
<i>Figure: 70 Check Scope.</i>	90
<i>Figure: 71 Deactivate Subnetwork Openstack.</i>	91
<i>Figure: 72 Deactivate Network Openstack.</i>	92
<i>Figure: 73 Delete Network</i>	93
<i>Figure: 74 Delete Network.</i>	94
<i>Figure: 75 Delete Virtual Link.</i>	95
<i>Figure: 76 Undeploy check.</i>	97
<i>Figure: 77 VNF Group Delete.</i>	98
<i>Figure: 78 Undeploy Preprocessing policies.</i>	100
<i>Figure: 79 Stop Monitor.</i>	101
<i>Figure: 80 Deactivation of a stopped VM.</i>	102
<i>Figure: 81 Undeploy Post-processing policies.</i>	103
<i>Figure: 82 Delete vPort.</i>	104
<i>Figure: 83 Undeploy Monitor.</i>	105
<i>Figure: 84 Delete Volume.</i>	106

Figure: 85 Delete Inventory..... 107

Preface

About this guide

This guide is intended to support the NFV Director integrator to understand and customize if needed “VIM Managed” mode.

It is NEVER recommended to modify the VIM manages mode directly but to create a full new copy with NEW ids for all artifacts and trees in order to avoid any conflict with the out of the box mode.

Audience

This document is targeting integrators specially HPE delivery and NFVD global practice so they can understand and customize if needed the behavior and limitations of current NFVD version.

For On boarding VNFs please refer to the *HPE NFV Director On boarding Guide*.

Document history

Table 1: Document history

Edition	Date	Description
1.0	14 March, 2017	First Edition.

Chapter 1

Operations tree.

These are the different operations represented in a tree shape.

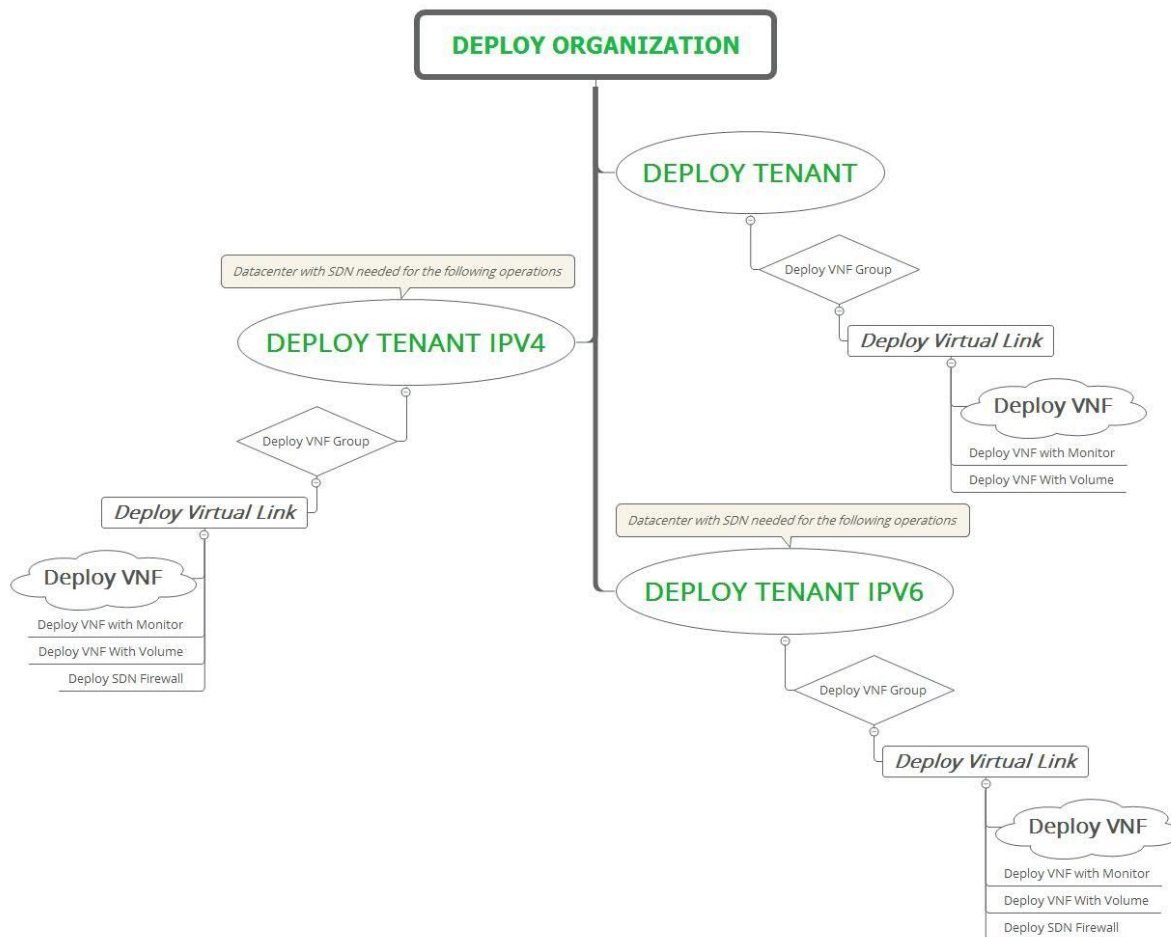
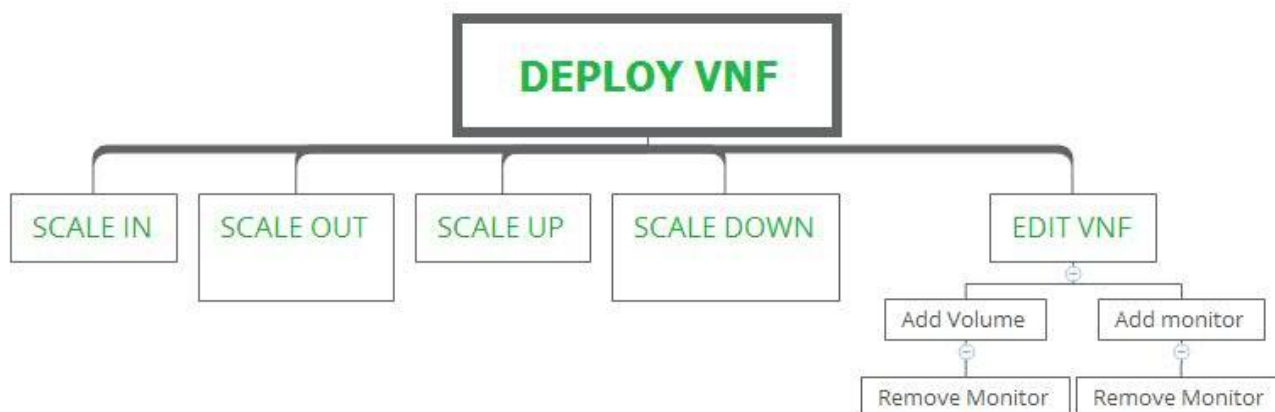


Figure: 1 Deployment tree.

Once the VNF has been activated, the operations available change.



Chapter 2 Deploy of an Organization – Bottom-Up.

From now on, and to make easier the understanding of the TLDs, we are going to explain the functionality of each set of `TASK_LIST_DEFINITION:GENERIC`, and the number of `TASK_DEFINITION:GENERIC` children of the previously mentioned `TASK_LIST_DEFINITION:GENERIC`.

Basically, the `TASK_LIST_DEFINITION:GENERIC` connect what we can consider “units of execution”, those are the `TASK_DEFINITION:GENERIC`, that have a `WORKFLOW` assigned to be executed when the execution of the TLD reach them.

If you like to have a more deep knowledge about the workflows mentioned through this document please refer to the specific document.

If in the category `FIND`, the attribute `Path` is present, the attribute `FIND.ArtifactType` will be the starting artifact for the `Path`, but the `FIND.Status` attribute refers to the last artifact on the `Path`.

```
FIND.ArtifactType == VIRTUAL_MACHINE
FIND.Status==      INSTANTIATED
FIND.Path ==
    VIRTUAL_MACHINE>VIRTUAL_CORE<CORE<CPU<SERVER<AVAILABILITY_ZONE<REGION>
    COMPUTE>FLAVOR
```

In this example, we are looking for a `FLAVOR` in status `INSTANTIATED`, we do not expect to get a `VIRTUAL_MACHINE`, in status `INSTANTIATED`.

If during the use of the TLDs, the “Regenerate UUIDs” option is used, the user should check the `Id` of the tree that brings all the elements of the TLD, this “`id`” is specific and it will be the same for all the tree groups in all the TLDs.

The two modes available are “Default” and “Simulated”, the second one is only available if it is configured previously, by defect, and the mode that will be used is “Default”.

2.1 Specific Elements of the TLD Deploy Organization

In this chapter the different elements of the specific TLD will be explained conscientiously.

2.2 TLD ORGANIZATION STATUS CHANGE: Organization Status Change.

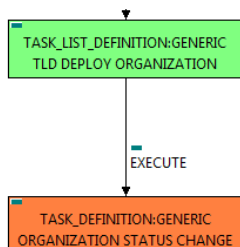


Figure: 2 Organization Status Change TLD.

The TDs that have present in their names “Status Change”, are Task Definitions responsible of the change in the status of the entity associated, in this case an ORGANIZATION. When the WF has finished we will have an ORGANIZATION with status ACTIVE in case of successful execution, or status ERROR in case of error, or simply not any change in the status because a ROLLBACK during the execution.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

GENERAL.Name ==	Activate Enterprise
FIND.Condition ==	status==constant:INSTANTIATED
SET.Running_Status ==	INSTANTIATED.
SET.Status ==	ACTIVE.
EXECUTE.Inactive==	false
ROLLBACK.Behaviour_on_error ==	STOP
ROLLBACK.Number_of_retries ==	0
DATA.Lock ==	true

The TASK_DEFINITION do not execute any workflow, with the attributes present in the categories it is enough to change the status of the entity.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a “STOP” set as behavior, so no Rollback it is going to be initiated, due to that the value of the attribute DATA.Lock is true, when the Task Definition has finished the artifact that was used in the workflow executed will be set as “Locked”.

Chapter 3 Deploy of a Tenant – Bottom-Up.

From now on, and to make easier the understanding of the TLDs, we are going to explain the functionality of each set of TASK_LIST_DEFINITION:GENERIC, and the number of TASK_DEFINITION:GENERIC children of the previously mentioned TASK_LIST_DEFINITION:GENERIC.

Basically, the TASK_LIST_DEFINITION:GENERIC connect what we can consider “units of execution”, those are the TASK_DEFINITION:GENERIC, that have a WORKFLOW assigned to be executed when the execution of the TLD reach them.

If you like to have a more deep knowledge about the workflows mentioned through this document please refer to the specific document.

If in the category FIND, the attribute Path is present, the attribute FIND.ArtifactType will be the starting artifact for the Path, but the FIND.Status attribute refers to the last artifact on the Path.

```
FIND.ArtifactType == VIRTUAL_MACHINE
FIND.Status==      INSTANTIATED
FIND.Path ==
    VIRTUAL_MACHINE>VIRTUAL_CORE<CORE<CPU<SERVER<AVAILABILITY_ZONE<REGION>
    COMPUTE>FLAVOR
```

In this example, we are looking for a FLAVOR in status INSTANTIATED, we do not expect to get a VIRTUAL_MACHINE, in status INSTANTIATED.

If during the use of the TLDs, the “Regenerate UUIDs” option is used, the user should check the Id of the tree that brings all the elements of the TLD, this “id” is specific and it will be the same for all the tree groups in all the TLDs.

The two modes available are “Default” and “Simulated”, the second one is only available if it is configured previously, by defect, and the mode that will be used is “Default”.

3.1 Specific Elements of the TLD Deploy Tenant.

In this chapter the different elements of the specific TLD will be explained conscientiously.

3.2 TLD DEPLOY TENANT: Quota Assignment Task.

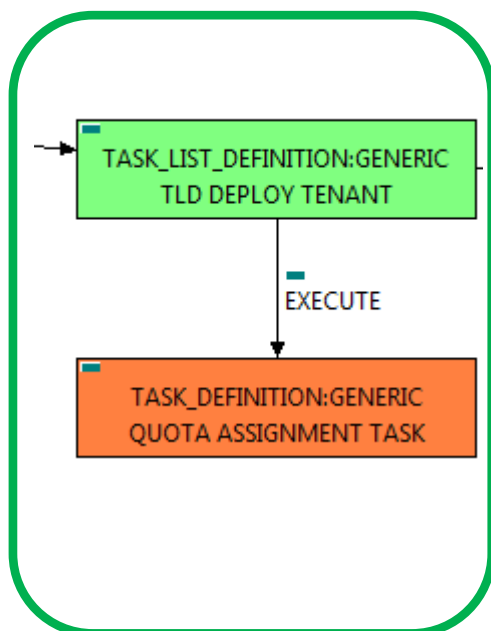


Figure: 3 Quota Assignment task.

The TDs that have present in the their names “Assignment”, are Task Definitions responsible of the assignation of resources for an specific artifact, in the case of the quotas, the TLD it is going to assign an amount of each resource needed for the correct execution of the deployment.

Once finished, our VNF should have every quota needed for a successful deployment assigned, having taken in consideration all the rules for the assignment. This is crucial, because our component consume quotas during the execution of the TLD.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

```

GENERAL.Name ==                Quota_Assignment Task
EXECUTE.Workflow ==            "WF_NFVD_CREATE_CONSUMED_RELATIONSHIP"
EXECUTE.Inactive ==            false
ROLLBACK.Behaviour_on_error == ROLLBACK
ROLLBACK.Number_of_retries ==  0
DATA.Lock ==                    false
INPUT_MAPPING.MAPPING_LIST ==  resourceTreeID=nfvd#quotaResourceID
  
```

The Workflow present in EXECUTE.Workflow it is going to seek the artifact identified by the Id given, this id should belong to an artifact TENANT:GENERIC in Status INSTANTIATED in the DDBB, when the WF find it, it will start. This workflow will create the relationships with the parent quotas needed by the TENANT:GENERIC to get a successful Deploy.

In case of error during the execution, the workflow jump to the ROLLBACK category, if the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, the attribute “number_of_retries” set the number of rollback attempts. In this case, the TLD has not assigned a rollback workflow, so in this case the TD will only change the status of the artifact which is being used.

Due to that the value of the attribute DATA.Lock is false, when the Task Definition has finished the artifact that was used in the workflow executed will remain unlocked.

3.3 • TLD STATUS CHANGE: Tenant_Status_Change task.

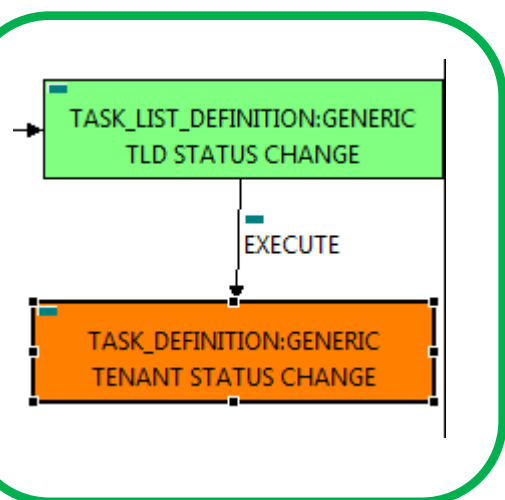


Figure: 4 Change status of the Tenant .

The TDs that have present in their names “Status Change”, are Task Definitions responsible of the change in the status of the entity associated, in this case a “VIRTUAL_LINK:MANAGEMENT”. When the WF has finished we will have an “VIRTUAL_LINK:MANAGEMENT” with status ACTIVE in case of successful execution, or status ERROR in case of error, or simply not any change in the status because a ROLLBACK during the execution.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

GENERAL.Name ==	Tenant Status Change
SET.Status ==	ACTIVE.
EXECUTE.Inactive==	false
ROLLBACK.Behaviour_on_error ==	STOP
ROLLBACK.Number_of_retries ==	0
DATA.Lock ==	true

The TASK_DEFINITION do not execute any workflow, with the attributes present in the categories it is enough to change the status of the entity.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a “STOP” set as behavior, so no Rollback it is going to be initiated, so the execution it is going to end here in case of error.

Chapter 4 Deploy of a VNF Group – Bottom-Up.

From now on, and to make easier the understanding of the TLDs, we are going to explain the functionality of each set of TASK_LIST_DEFINITION:GENERIC, and the number of TASK_DEFINITION:GENERIC children of the previously mentioned TASK_LIST_DEFINITION:GENERIC.

Basically, the TASK_LIST_DEFINITION:GENERIC connect what we can consider “units of execution”, those are the TASK_DEFINITION:GENERIC, that have a WORKFLOW assigned to be executed when the execution of the TLD reach them.

If you like to have a more deep knowledge about the workflows mentioned through this document please refer to the specific document.

If in the category FIND, the attribute Path is present, the attribute FIND.ArtifactType will be the starting artifact for the Path, but the FIND.Status attribute refers to the last artifact on the Path.

```
FIND.ArtifactType == VIRTUAL_MACHINE
FIND.Status==      INSTANTIATED
FIND.Path ==
    VIRTUAL_MACHINE>VIRTUAL_CORE<CORE<CPU<SERVER<AVAILABILITY_ZONE<REGION>
    COMPUTE>FLAVOR
```

In this example, we are looking for a FLAVOR in status INSTANTIATED, we do not expect to get a VIRTUAL_MACHINE, in status INSTANTIATED.

If during the use of the TLDs, the “Regenerate UUIDs” option is used, the user should check the Id of the tree that brings all the elements of the TLD, this “id” is specific and it will be the same for all the tree groups in all the TLDs.

The two modes available are “Default” and “Simulated”, the second one is only available if it is configured previously, by defect, and the mode that will be used is “Default”.

4.1 Specific Elements of the TLD Deploy VNF Group.

In this chapter the different elements of the specific TLD will be explained conscientiously.

4.2 TLD DEPLOY VNF_GROUP: Quota Assignment Task.

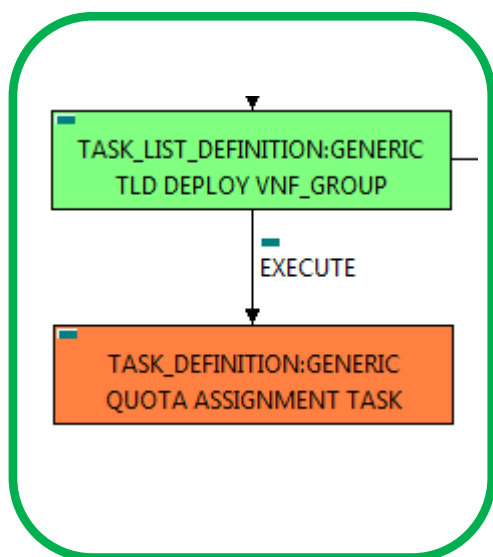


Figure: 5 Quota Assignment task.

The TDs that have present in their names “Assignment”, are Task Definitions responsible of the assignment of resources for an specific artifact, in the case of the quotas, the TLD it is going to assign an amount of each resource needed for the correct execution of the deployment.

Once finished, our VNF should have every quota needed for a successful deployment assigned, having taken in consideration all the rules for the assignment. This is crucial, because our component consume quotas during the execution of the TLD.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

```

GENERAL.Name ==                Quota_Assignment Task
EXECUTE.Workflow ==
“WF_NFVD_CREATE_CONSUMED_RELATIONSHIP”
EXECUTE.Inactive==                false
ROLLBACK.Behaviour_on_error ==    ROLLBACK
ROLLBACK.Number_of_retries ==    0
DATA.Lock ==                      false
INPUT_MAPPING.MAPPING_LIST ==
resourceTreeID=nfvd#quotaResourceID
  
```

The Workflow present in EXECUTE.Workflow it is going to seek the artifact identified by the Id given, this id should belong to an artifact VNF_GROUP:GENERIC in Status INSTANTIATED in the DDBB, when the WF find it, it will start. This workflow will create the relationships with the parent quotas needed by the VNF_GROUP:GENERIC to get a successful Deploy.

In case of error during the execution, the workflow jump to the ROLLBACK category, if the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, the attribute “number_of_retries” set the number of rollback attempts. In this case, the TLD has not assigned a rollback workflow, so in this case the TD will only change the status of the artifact which is being used.

Due to that the value of the attribute DATA.Lock is false, when the Task Definition has finished the artifact that was used in the workflow executed will remain unlocked.

4.3 • TLD CHANGE STATUS: VNF Group Status Change task.

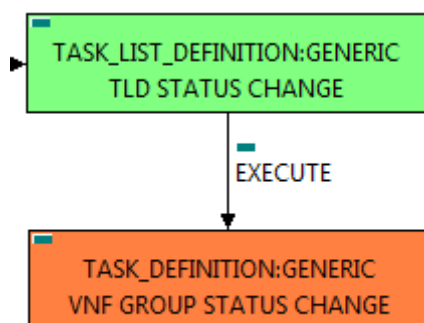


Figure: 6 Change status of the VNF Group .

The TDs that have present in their names “Status Change”, are Task Definitions responsible of the change in the status of the entity associated, in this case a “VNF_GROUP:GENERIC”. When the WF has finished we will have an “VNF_GROUP:GENERIC” with status ACTIVE in case of successful execution, or status ERROR in case of error, or simply not any change in the status because a ROLLBACK during the execution.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

GENERAL.Name ==	VNF GROUP Status Change
SET.Status ==	ACTIVE.
EXECUTE.Inactive==	false
ROLLBACK.Behaviour_on_error ==	STOP
ROLLBACK.Number_of_retries ==	0
DATA.Lock ==	true

The TASK_DEFINITION do not execute any workflow, with the attributes present in the categories it is enough to change the status of the entity.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a “STOP” set as behavior, so no Rollback it is going to be initiated, so the execution it is going to end here in case of error.

Chapter 5 Deploy of a VNF – Bottom-Up.

From now on, and to make easier the understanding of the TLDs, we are going to explain the functionality of each set of TASK_LIST_DEFINITION:GENERIC, and the number of TASK_DEFINITION:GENERIC children of the previously mentioned TASK_LIST_DEFINITION:GENERIC.

Basically, the TASK_LIST_DEFINITION:GENERIC connect what we can consider “units of execution”, those are the TASK_DEFINITION:GENERIC, that have a WORKFLOW assigned to be executed when the execution of the TLD reach them.

If you like to have a more deep knowledge about the workflows mentioned through this document please refer to the specific document.

If in the category FIND, the attribute Path is present, the attribute FIND.ArtifactType will be the starting artifact for the Path, but the FIND.Status attribute refers to the last artifact on the Path.

```
FIND.ArtifactType == VIRTUAL_MACHINE
FIND.Status==      INSTANTIATED
FIND.Path ==
    VIRTUAL_MACHINE>VIRTUAL_CORE<CORE<CPU<SERVER<AVAILABILITY_ZONE<REGION>
    COMPUTE>FLAVOR
```

In this example, we are looking for a FLAVOR in status INSTANTIATED, we do not expect to get a VIRTUAL_MACHINE, in status INSTANTIATED.

If during the use of the TLDs, the “Regenerate UUIDs” option is used, the user should check the Id of the tree that brings all the elements of the TLD, this “id” is specific and it will be the same for all the tree groups in all the TLDs.

The two modes available are “Default” and “Simulated”, the second one is only available if it is configured previously, by defect, and the mode that will be used is “Default”.

5.1 Specific Elements of the TLD Deploy VNF.

In this chapter the different elements of the specific TLD will be explained conscientiously.

5.2 TLD DEPLOY VNF: Quota Assignment

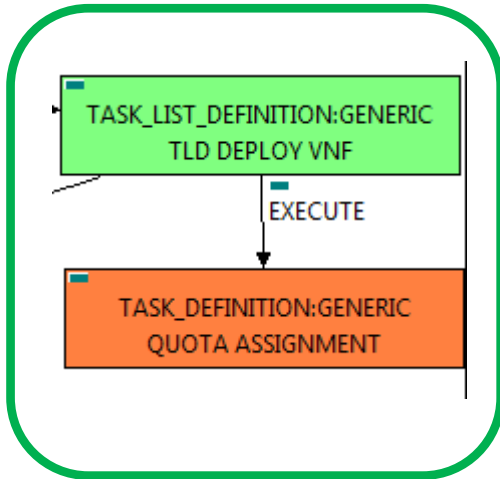


Figure: 7 Quota Assignment task.

The TDs that have present in the their names “Assignment”, are Task Definitions responsible of the assignation of resources for an specific artifact, in the case of the quotas, the TLD it is going to assign an amount of each resource needed for the correct execution of the deployment.

Once finished, our VNF should have every quota needed for a successful deployment assigned. This is crucial, because our component consume quotas during the execution of the TLD.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

```

GENERAL.Name ==                               Quota_Assignment
EXECUTE.Workflow ==
    "WF_NFVD_ASSIGNMENT_QUOTA"
EXECUTE.Inactive==                             false
ROLLBACK.Behaviour_on_error ==                 ROLLBACK
ROLLBACK.Number_of_retries ==                  0
DATA.Lock ==                                   true
INPUT_MAPPING.MAPPING_LIST ==
resourceTreeID=nfvd#quotaResourceID
  
```

The Workflow present in EXECUTE.Workflow it is going to seek a VNF in the DDBB, when the WF is found, it will start. This workflow assign all the quotas needed by the VNF to get a successful Deploy, it will check the available quotas and make the relationships without taking in mind the amounts.

In case of error during the execution, the workflow jump to the ROLLBACK category, if the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, the attribute “number_of_retries” set the number of rollback attempts. In this case, the TLD has not assigned a rollback workflow, so in this case the TD will only change the status of the artifact which is being used.

Due to that the value of the attribute DATA.Lock is true, once the TD has finished, the Quota recently assigned, will be blocked.

5.3 TLD RESOURCE ASSIGNMENT: ALLOCATE RESOURCES VNF.

The TDs that have present in their names “Assignment” or “Allocate”, are Task Definitions responsible of the assignation of resources for a specific artifact or deploy, in this case, we are looking for a VNF to assign the resources needed for the future deployment. In order to have a successful assignation we must have in our TLD Deploy VNF an artifact POLICY: ASSIGMENT_RELATIONSHIP, with a GENERAL.Name==”Resource_Assignment”, also this artifact must be related with the OPERATION_GROUP: GENERIC of our TLD with a relationship of type PROVIDES and status ENABLED.

Once finished, our VNF should have every resources needed for a successful deployment assigned, having taken in consideration all the rules for the assignment.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

```

GENERAL.Name ==                               Allocate Resources
FIND.MainArtifact ==
VNF#ORIGIN_CREATION.ResourceTreeID!=null
FIND.Condition ==
!{[VNF>POLICY:AFFINITY][VNF>POLICY:ANTI_AFFINITY]
[GENERAL.Name!=""]}
EXECUTE.Workflow ==
    “WF_NFVD_RESOURCE_ALLOCATION”
EXECUTE.Inactive==                             false
ROLLBACK.Behaviour_on_error ==                 ROLLBACK
ROLLBACK.Number_of_retries ==                  0
DATA.Lock ==                                   true
INPUT_MAPPING.MAPPING_LIST ==
resourceTreeID=resourceArtifactID;
def_exclusion_list=TENANT:OPENSTACK,NETWORKING,
COMPUTE,IMAGE_STORAGE

```

The Workflow present in EXECUTE.Workflow it is going to seek a VNF with its attribute ORIGIN_CREATION.ResourceTreeID not null in the DDBB and without AFFINITY or ANTI_AFFINITY artifacts, when the WF find it, it will start. This workflow assign all the resources needed by the VNF to get a successful Deploy, it will check the available resources and assign them to the artifact id contained in the ORIGIN_CREATION.ResourceTreeID attribute.

In case of error during the execution, the workflow jump to the ROLLBACK category, if the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, the attribute “number_of_retries” set the number of rollback attempts. In this case, the TLD has not assigned a rollback workflow, so in this case the TD will only change the status of the artifact which is being used.

Due to that the value of the attribute DATA.Lock is true, once the TD has finished, the artifact recently assigned, will be blocked.

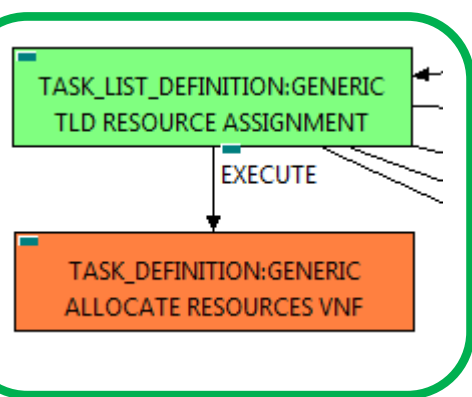


Figure: 8 Allocate Resources for VNF.

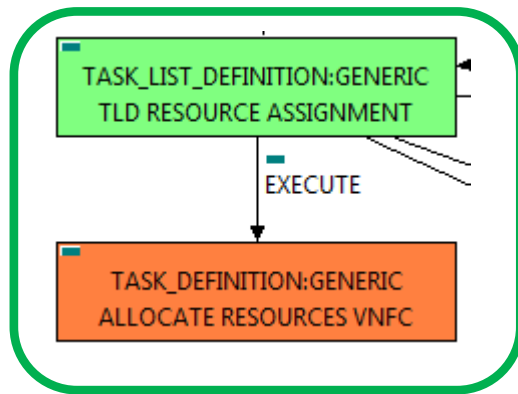


Figure: 9 Allocate Resources for VNF.

The TDs that have present in their names “Assignment” or “Allocate”, are Task Definitions responsible of the assignation of resources for a specific artifact or deploy, in this case, we are looking for a VNF_COMPONENT to assign the resources needed for the future deployment. In order to have a successful assignation we must have in our TLD Deploy VNF an artifact POLICY: ASSIGMENT_RELATIONSHIP, with a GENERAL.Name==”Resource_Assignment”, also this artifact must be related with the OPERATION_GROUP: GENERIC of our TLD with a relationship of type PROVIDES and status ENABLED.

Once finished, our VNF should have every resources needed for a successful deployment assigned, having taken in consideration all the rules for the assignment.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

```

GENERAL.Name == Allocate Resources
FIND.MainArtifact ==
VNF>VNF_COMPONENT#ORIGIN_CREATION.ResourceTreeID!=null
FIND.Condition == ![VNF_COMPONENT<VNF>POLICY:AFFINITY]
[VNF_COMPONENT<VNF>POLICY:ANTI_AFFINITY]
[GENERAL.Name!=""]
EXECUTE.Workflow ==
    “WF_NFVD_RESOURCE_ALLOCATION”
EXECUTE.Inactive== false
ROLLBACK.Behaviour_on_error == ROLLBACK
ROLLBACK.Number_of_retries == 0
DATA.Lock == true
INPUT_MAPPING.MAPPING_LIST ==
resourceTreeID=resourceArtifactID;
def_exclusion_list=TENANT:OPENSTACK,NETWORKING
,COMPUTE,IMAGE_STORAGE
  
```

The Workflow present in EXECUTE.Workflow it is going to seek a VNF_COMPONENT with its attribute ORIGIN_CREATION.ResourceTreeID not null in the DDBB and without AFFINITY or ANTI_AFFINITY artifacts, when the WF find it, it will start. This workflow assign all the resources needed by the VNF to get a successful Deploy, it will check the available resources and assign them to the artifact id contained in the ORIGIN_CREATION.ResourceTreeID attribute.

In case of error during the execution, the workflow jump to the ROLLBACK category, if the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, the attribute “number_of_retries” set the number of rollback attempts. In this case, the TLD has not assigned a rollback workflow, so in this case the TD will only change the status of the artifact which is being used.

Due to that the value of the attribute DATA.Lock is true, once the TD has finished, the artifact recently assigned, will be blocked.

5.5 TLD RESOURCE ASSIGNMENT: ALLOCATE RESOURCES VNF WITH AFFINITY.

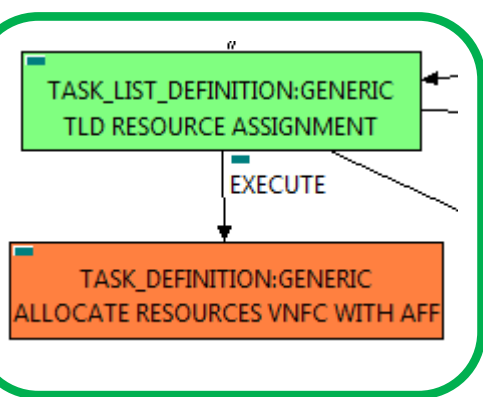


Figure: 10 Allocate Resources for VNF with Affinities.

The TDs that have present in their names “Assignment” or “Allocate”, are Task Definitions responsible of the assignation of resources for a specific artifact or deploy, in this case, we are looking for a VNF_COMPONENT to assign the resources needed for the future deployment. In order to have a successful assignation we must have in our TLD Deploy VNF an artifact POLICY: ASSIGMENT_RELATIONSHIP, with a GENERAL.Name==”Resource_Assignment”, also this artifact must be related with the OPERATION_GROUP: GENERIC of our TLD with a relationship of type PROVIDES and status ENABLED.

Once finished, our VNF should have every resources needed for a successful deployment assigned, having taken in consideration all the rules for the assignment.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

```

GENERAL.Name == Allocate Resources
FIND.MainArtifact ==
VNF>VNF_COMPONENT#ORIGIN_CREATION.ResourceTreeID!=null
FIND.Condition == {[VNF_COMPONENT<VNF>POLICY:AFFINITY]
[VNF_COMPONENT<VNF>POLICY:ANTI_AFFINITY]
[GENERAL.Name!=""]}
EXECUTE.Workflow ==
  “WF_NFVD_RESOURCE_ALLOCATION_WITH_AFFINITIES”
EXECUTE.Inactive== false
ROLLBACK.Behaviour_on_error == ROLLBACK
ROLLBACK.Number_of_retries == 0
DATA.Lock == true
INPUT_MAPPING.MAPPING_LIST ==
assignmentRelationshipID=Resource_Assignment;
resourceTreeID=resourceArtifactID;
def_exclusion_list=TENANT:OPENSTACK,NETWORKING,
COMPUTE_IMAGE_STORAGE

```

The Workflow present in EXECUTE.Workflow it is going to seek a VNF_COMPONENT with its attribute ORIGIN_CREATION.ResourceTreeID not null in the DDBB and with AFFINITY or ANTI_AFFINITY artifacts, when the WF find it, it will start. This workflow assign all the resources needed by the VNF to get a successful Deploy, it will check the available resources and assign them to the artifact id contained in the ORIGIN_CREATION.ResourceTreeID attribute.

In case of error during the execution, the workflow jump to the ROLLBACK category, if the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, the attribute “number_of_retries” set the number of rollback attempts. In this case, the TLD has not assigned a rollback workflow, so in this case the TD will only change the status of the artifact which is being used.

Due to that the value of the attribute DATA.Lock is true, once the TD has finished, the artifact recently assigned, will be blocked.

5.6 TLD RESOURCE ASSIGNMENT: ALLOCATE RESOURCES VNF WITH AFFINITY.

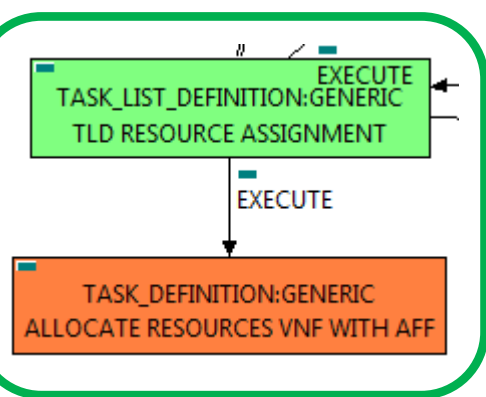


Figure: 11 Allocate Resources for VNF with Affinities.

The TDs that have present in their names “Assignment” or “Allocate”, are Task Definitions responsible of the assignation of resources for a specific artifact or deploy, in this case, we are looking for a VNF to assign the resources needed for the future deployment. In order to have a successful assignation we must have in our TLD Deploy VNF an artifact POLICY: ASSIGMENT_RELATIONSHIP, with a GENERAL.Name==”Resource_Assignment”, also this artifact must be related with the OPERATION_GROUP: GENERIC of our TLD with a relationship of type PROVIDES and status ENABLED.

Once finished, our VNF should have every resources needed for a successful deployment assigned, having taken in consideration all the rules for the assignment.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

```

GENERAL.Name ==                               Allocate Resources
FIND.MainArtifact ==
                                                VNF#ORIGIN_CREATION.ResourceTreeID!=null
FIND.Condition ==
{[VNF>POLICY:AFFINITY][VNF>POLICY:ANTI_AFFINITY]
[GENERAL.Name!=""]}
EXECUTE.Workflow ==
“WF_NFVD_RESOURCE_ALLOCATION_WITH_AFFINITIES”
EXECUTE.Inactive==                               false
ROLLBACK.Behaviour_on_error ==                   ROLLBACK
ROLLBACK.Number_of_retries ==                     0
DATA.Lock ==
true
INPUT_MAPPING.MAPPING_LIST ==
assignmentRelationshipID=Resource_Assignment;
resourceTreeID=resourceArtifactID;
def_exclusion_list=TENANT:OPENSTACK,NETWORKING,
COMPUTE,IMAGE_STORAGECOMPUTE,IMAGE_STORAGE
  
```

The Workflow present in EXECUTE.Workflow it is going to seek a VNF_COMPONENT with its attribute ORIGIN_CREATION.ResourceTreeID not null in the DDBB and with AFFINITY or ANTI_AFFINITY artifacts, when the WF find it, it will start. This workflow assign all the resources needed by the VNF to get a successful Deploy, it will check the available resources and assign them to the artifact id contained in the ORIGIN_CREATION.ResourceTreeID attribute.

In case of error during the execution, the workflow jump to the ROLLBACK category, if the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, the attribute “number_of_retries” set the number of rollback attempts. In this case, the TLD has not assigned a rollback workflow, so in this case the TD will only change the status of the artifact which is being used.

Due to that the value of the attribute DATA.Lock is true, once the TD has finished, the artifact recently assigned, will be blocked.

5.7 TLD IMAGE PERMISSION: GET IMAGE.

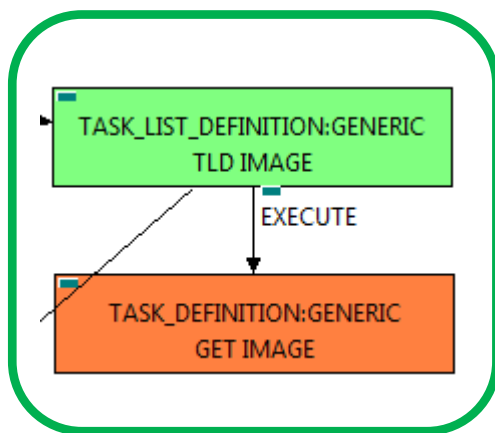


Figure: 12 Get Image.

The TDs that have present in the their names “Provision”, are Task Definitions responsible of the creation and store of an artifact in DDBB, in this case, the artifact that it is going to be provisioned it is an “IMAGE:OPENSTACK”, if it does not exist. This means, when this workflow finish, we will have a new artifact “IMAGE:OPENSTACK” in our DDBB, as well, due to the nature of the artifact, the artifact will be prepared to be activated in the platform Openstack when will be required.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

```

GENERAL.Name ==                               Get Image
FIND.MainArtifact ==
VNF>VNF_COMPONENT>
VIRTUAL_MACHINE@status=INSTANTIATED
EXECUTE.Workflow ==
    “WF_TS_CHECK_VM_IMAGE”
EXECUTE.Inactive==                             false
ROLLBACK.Behaviour_on_error ==                 ROLLBACK
ROLLBACK.Number_of_retries ==                  0
DATA.Lock ==                                   true
  
```

The Workflow present in EXECUTE.Workflow it is going to seek a VIRTUAL_MACHINE in Status INSTANTIATED in the DDBB, when the WF find it, it will start. This workflow will start another two more, the one that check the permissions of the IMAGE, “WF_TS_CHECK_IMAGE_PERMISSIONS”, and the one that will deploy our IMAGE if it is not deployed, “WF_TS_DEPLOY_IMAGE”.

In case of IMAGE managed by VIM, “WF_TS_CHECK_IMAGE_PERMISSIONS will work the same way, however in that case “WF_TS_DEPLOY_IMAGE” couldn’t work the same way, the image can be reused it previously has been deployed in the VIM, but can not generate a new deployed IMAGE.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, the attribute “number_of_retries” set the number of rollback attempts. If an error take place in this TD , no action will be taken, the execution of the TLD will try to start a rollback workflow but there is not a workflow to be executed in the attribute ROLLBACK.Workflow, so the execution will continue.

Due to that the value of the attribute DATA.Lock is true, once the TD has finished, the artifact recently checked, will be blocked.

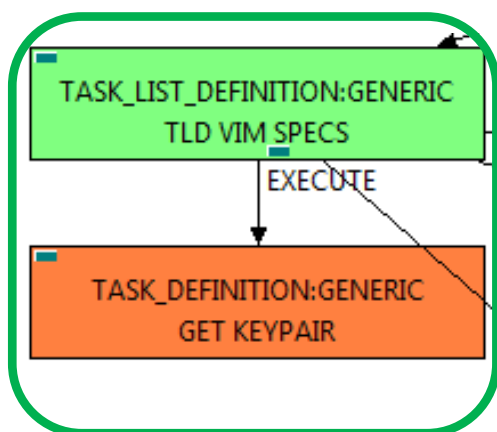


Figure: 13 Get Key Pair.

The TDs that have present in the their names “Create” or “Inventory”, are Task Definitions responsible of the provision of an element, in the platform targeted or in the DDBB, or both, in this case, the artifact that is going to be provisioned is a “KEYPAIR”, this means, when this workflow finish, we will have a KEYPAIR with status INSTANTIATED.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

```

GENERAL.Name ==                               Get KeyPair
FIND.MainArtifact ==
VNF>VNF_COMPONENT>
VIRTUAL_MACHINE@status=INSTANTIATED
FIND.Condition==
KEYPAIR.Pubkey_Data != null || KEYPAIR.Pubkey_Path != null
EXECUTE.Workflow ==
    “WF_TS_NFVD_CREATE_KEY_PAIR_INVENTORY”
EXECUTE.Inactive==                             false
ROLLBACK.Behaviour_on_error ==                 ROLLBACK
ROLLBACK.Number_of_retries ==                  0
DATA.Lock ==                                   true
  
```

The Workflow present in EXECUTE.Workflow it is going to seek for a KEYPAIR element in Status INSTANTIATED in the DDBB, when the WF find it, it will start.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, the attribute “number_of_retries” set the number of rollback attempts. If an error take place in this TD , no action will be taken, the execution of the TLD will try to start a rollback workflow but there is not a workflow to be executed in the attribute ROLLBACK.Workflow, so the execution will continue.

Due to that the value of the attribute DATA.Lock is true, once the TD has finished, the artifact recently checked, will be blocked.

5.9 TLD VIM SPECS: GET FLAVOR

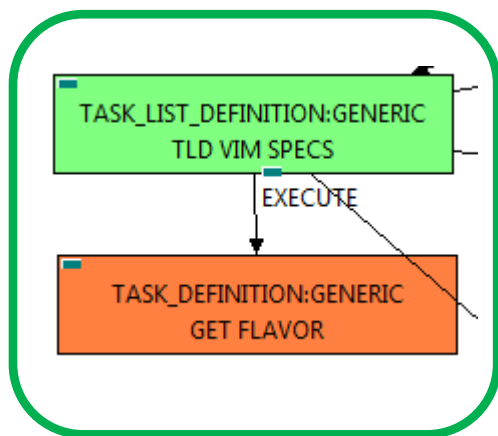


Figure: 14 Creation of a standard flavor.

This TD it is going to create or get the FLAVORS needed for each VMs to be activated later on, **this means, the WFs implied in this TLD are going to check each element of our VMs to gather all the information needed to create a specific FLAVOR artifact, during the execution of the TD, the ENTITY_SCALE Policies are going to be consulted, the situation of these policies are required for the correct creation of the FLAVOR.**

Once finished, we will have a number of FLAVORS bonded to a VM or VMs, prepare to be activated with these FLAVORS.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

GENERAL.Name ==	Get Flavor
EXECUTE.Workflow ==	
“WF_NFVD_CREATE_FLAVOR_INSTANCES”	
EXECUTE.Inactive==	false
ROLLBACK.Behaviour_on_error ==	ROLLBACK
ROLLBACK.Number_of_retries ==	0
DATA.Lock ==	true

The Workflow present in EXECUTE.Workflow it is going to seek a VNF with Running_Status INSTANTIATED in the DDBB, if the WF find some artifact that fill all the conditions, it will start.

This workflow will start another two more, the one that check if the FLAVOR needs Extra_Specs, a special set of configurations for the FLAVOR, “WF_NFVD_CREATE_FLAVOR_EXTRA_INVENTORY”, and the one that will create the Flavor Instance in OPENSTACK platform, “WF_NFVD_CREATE_FLAVOR_OS_INSTANCE”.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, the attribute “number_of_retries” set the number of rollback attempts. If an error take place in this TD, no action will be taken, the execution of the TLD will try to start a rollback workflow but there is not a workflow to be executed in the attribute ROLLBACK.Workflow, so the execution will continue without error.

Due to that the value of the attribute DATA.Lock is true, once the TD has finished, the artifact recently created, will be blocked.

5.10 TLD VIM SPECS: CONNECT_SUBNET

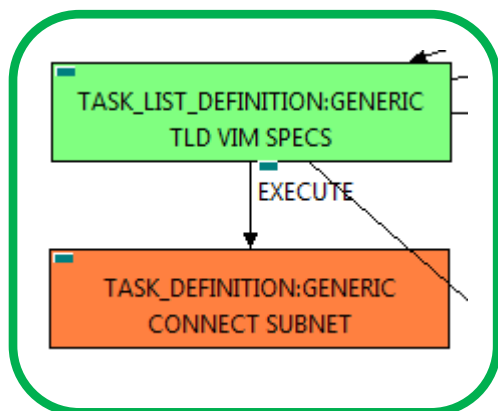


Figure: 15 Connection of the Subnetwork to the Virtual Port.

The TDs that have present in their names “Connect”, are Task Definitions responsible of the connection between artifacts, this means, this TDs will create relationship of specific kind between concrete artifacts, in this case, the WF it is going to query the DDBB looking for all the VIRTUAL_LINK:END_POINTS with Status “TO_BE_CONNECTED”, once the WF has the VL:EP, it will query for all the SUBNETWORKS, NETWORKS and IPADDRESS of the VL, when the WF reach this point, it will query for the VPORTS related to these artifacts, after that, it is going to evaluate the relationships between the previously mentioned artifacts and the VPORTS, creating VPORTs and new relationships of type ALLOCATED and USES depending on the artifacts which are going to be related, mainly, SUBNETWORKs and VPORTs. The last thing this WF will do is change the status of the relationship between VL:EPs and the VNF:EP to CONNECTED.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

GENERAL.Name ==	Connect Subnet
FIND.MainArtifact ==	VNF>VNF_ENDPOINT
EXECUTE.Workflow ==	“WF_TS_CONNECT_VM_SUBNET”
EXECUTE.Inactive==	false
ROLLBACK.Behaviour_on_error ==	ROLLBACK
ROLLBACK.Number_of_retries ==	0
DATA.Lock ==	true

The Workflow present in EXECUTE.Workflow it is going to seek a VNF_ENDPOINT with Running_Status INSTANTIATED in the DDBB, if the WF find some artifact that fill all the conditions, it will start.

In case of error during the execution, the workflow jump to the ROLLBACK category, if the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, the attribute “number_of_retries” set the number of rollback attempts. If an error take place in this TD, no action will be taken, the execution of the TLD will try to start a rollback workflow but there is not a workflow to be executed in the attribute ROLLBACK.Workflow, so the execution will continue without error.

Due to that the value of the attribute DATA.Lock is true, once the TD has finished, the artifact recently created, will be blocked.

5.11 TLD PREPROCESSING: PREPROCESS.

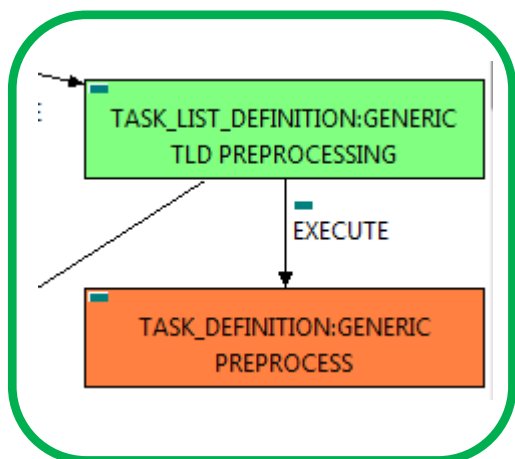


Figure: 16 Deploying Pre-Process policies.

This TD is responsible of the provision in the right order of the artifacts referenced by the PreProcessing policies, these policies allow the user to set a number of elements that should be taken in consideration in a certain order, in other case, the execution will fail depending on the event occurred.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

FIND.MainArtifact ==	POLICY:POSTPRE_PROCESSING
FIND.Condition ==	
PROCESSING_JOB.Job_type==constant:PRE&&	
PROCESSING_JOB.Operation==constant:DEPLOY	
EXECUTE.OrderBy ==	PROCESSING_JOB.OrderBy
ROLLBACK.Behaviour_on_error ==	STOP
ROLLBACK.Number_of_retries ==	0
DATA.Lock ==	false

In this TD there is not a workflow to be executed, the target of this TD is process in the correct order the PreProcessing policies present in the VNF, these policies should be executed in a specific order to make the changes or configurations properly, in other case an error will be launched

If the TD ends successfully, the Pre-Processing policies will have been processed adequately.

In case of error during the execution, the TD will jump to the ROLLBACK category, If the "Behaviour_on_error" attribute its set on "ROLLBACK" the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a "STOP" as value set for behavior, so no Rollback it is going to be initiated, the execution will stop.

Due to that the value of the attribute DATA.Lock is false, when the Task Definition has finished the artifact that was used in the workflow executed will remain unlocked.

5.12 TLD DEPLOY MONITOR VOLUME: MONITOR DEPLOY.

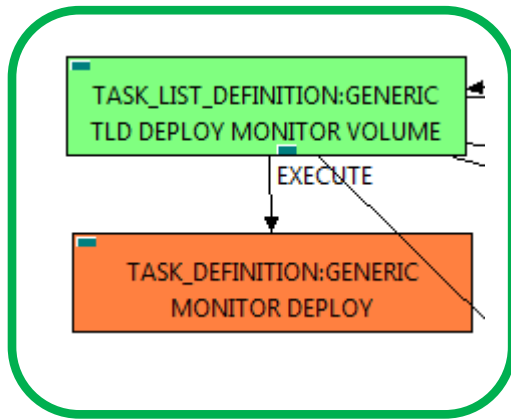


Figure: 17 Deployment of a Monitor.

The TDs that have present in the their names “Deploy” are Task Definitions responsible of the deployment in the platform targeted and the updating of the status in the platform and the DDBB , these deployments are slightly different to the ones we launch for our entities, as a rule, they are small components as the MONITORS. In this case, the artifact that is going to be deployed is a “MONITOR”, this means, when this workflow finish, we will have a MONITOR deployed with status DEPLOYED.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

```

GENERAL.Name == CREATE_INGRESS_ENTRY
FIND.MainArtifact== MONITOR
SET.Running_Status == INSTANTIATED.
Set.Status == DEPLOYED.
EXECUTE.Workflow ==
    "WF_TS_MONITOR_DEPLOY"
EXECUTE.Inactive== false
EXECUTE.OrderBy == GENERAL.Order
ROLLBACK.Behaviour_on_error == STOP
ROLLBACK.Number_of_retries == 0
DATA.Lock == true
  
```

The Workflow present in EXECUTE.Workflow attribute it is going to seek a MONITOR in Status INSTANTIATED in the DDBB . Once found , the WF will start the deployment, if the deployment is successful we set the status of the artifact as the SET.Status attribute dictates. The attribute SET.Running_Status concern about the temporal status that the artifact it is going to maintain until the final change of status that comes from SET.Status.

In case of error during the execution, the TD will jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, the TD has not a rollback workflow set, so no Rollback it is going to be initiated, the execution will stop.

Due to that the value of the attribute DATA.Lock is true, when the Task Definition has finished the artifact that was used in the workflow executed will be set as “Locked”.

5.13 TLD DEPLOY MONITOR VOLUME: ACTIVATE_FLAVOR.

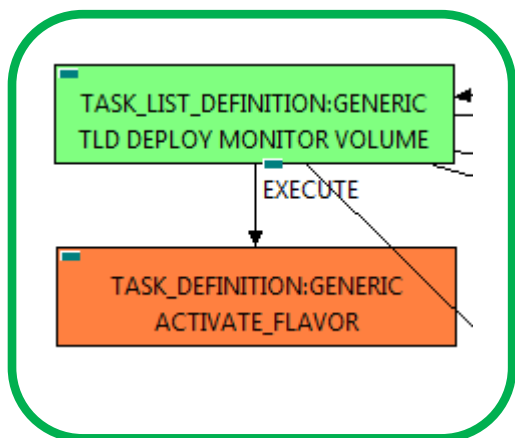


Figure: 18 Deployment of a Flavor.

The TDs that have present in the their names “Activate”, are Task Definitions responsible of the activation in the platform targeted and the updating of the status in the platform and the DDBB, in this case, the artifact that is going to be activated is a “FLAVOR”, this means, when this workflow finish, we will have a FLAVOR with status ACTIVE associated to the VIRTUAL_MACHINE that it is going to use it in the activation.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

```

GENERAL.Name ==                                ACTIVATE_FLAVOR
FIND.MainArtifact==
VNF>VNF_COMPONENT>VIRTUAL_MACHINE
FIND.Path ==
VIRTUAL_MACHINE>VIRTUAL_CORE<CORE<CPU
<SERVER<AVAILABILITY_ZONE<REGION>COMPUTE>
FLAVOR@status=INSTANTIATED
FIND.Condition ==                                Id==% VIM.FlavorID%
SET.Running_Status ==                            INSTANTIATED.
Set.Status ==                                    ACTIVE.
EXECUTE.Workflow ==
                                “WF_TS_ACTIVATE_FLAVOR”
EXECUTE.Inactive ==                                false
ROLLBACK.Behaviour_on_error ==                    STOP
ROLLBACK.Number_of_retries ==                      0
DATA.Lock ==                                       true
  
```

The Workflow present in EXECUTE.Workflow attribute it is going to seek a FLAVOR in Status INSTANTIATED in the DDBB . Notice that we are not trying to get a VIRTUAL_MACHINE in status INSTANTIATED. The query it is going to use the Path present in the category FIND. Once found, the WF will start the activation, if the activation it is successful we set the status of the artifact as the SET.Status attribute dictates. The attribute SET.Running_Status concern about the temporal status that the artifact it is going to maintain until the final change of status that comes from SET.Status.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a “STOP” set as behavior, so no Rollback it is going to be initiated, so the execution it is going to end here in case of error.

Due to that the value of the attribute DATA.Lock is true, when the Task Definition has finished the artifact that was used in the workflow executed will be set as “Locked”.

5.14 TLD DEPLOY MONITOR VOLUME: CREATE VOLUME

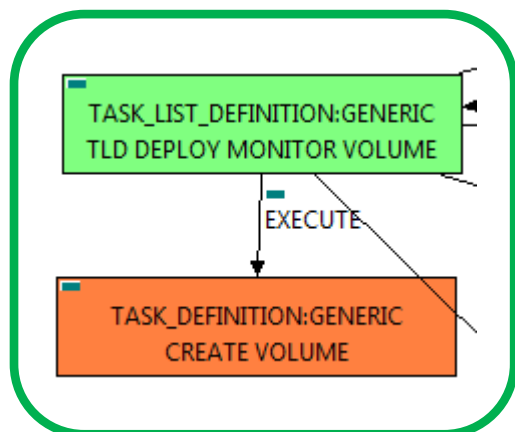


Figure: 19 Activation of a Volume.

The TDs that have present in the their names “Activate”, are Task Definitions responsible of the activation in the platform targeted and the updating of the status in the platform and the DDBB, in this case, the artifact that is going to be activated is a “VOLUME”, this means, when this workflow finish, we will have a VOLUME with status CREATED.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

GENERAL.Name ==	Create Volume
FIND.MainArtifact==	
VNF>VNF_COMPONENT>VIRTUAL_MACHINE>	
VIRTUAL_LUN@status=INSTANTIATED	
SET.Running_Status ==	INSTANTIATED.
Set.Status ==	CREATED.
EXECUTE.Workflow ==	
“WF_TS_CREATE_VOLUME”	
EXECUTE.Inactive ==	false
ROLLBACK.Behaviour_on_error ==	STOP
ROLLBACK.Number_of_retries ==	0
DATA.Lock ==	true

The Workflow present in EXECUTE.Workflow attribute it is going to seek a VIRTUAL_LUN in Status INSTANTIATED in the DDBB . Once found , the WF will start the activation, if the activation it is successful we set the status of the artifact as the SET.Status attribute dictates. The attribute SET.Running_Status concern about the temporal status that the artifact it is going to maintain until the final change of status that comes from SET.Status.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a “STOP” set as behavior, so no Rollback it is going to be initiated, so the execution it is going to end here in case of error.

Due to that the value of the attribute DATA.Lock is true, when the Task Definition has finished the artifact that was used in the workflow executed will be set as “Locked”.

5.15 TLD DEPLOY MONITOR VOLUME: ACTIVATE KEYPAIR.

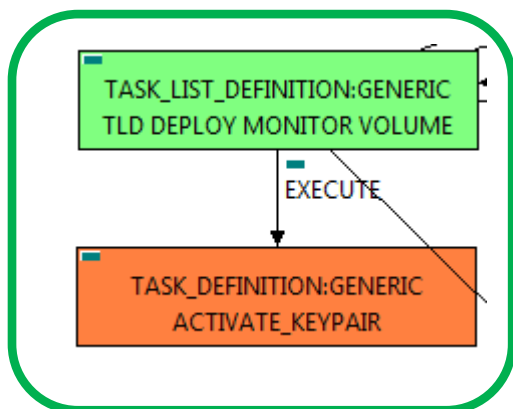


Figure: 20 Activation of the KeyPair.

The TDs that have present in the their names “Activate”, are Task Definitions responsible of the activation in the platform targeted and the updating of the status in the platform and the DDBB, in this case, the artifact that is going to be activated is a “KEYPAIR”, this means, when this workflow finish, we will have a KEYPAIR with status ACTIVE.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

```

GENERAL.Name ==                                ACTIVATE_KEYPAIR
FIND.MainArtifact==
VNF>VNF_COMPONENT>VIRTUAL_MACHINE>
VIRTUAL_CORE<CORE<CPU<SERVER
<AVAILABILITY_ZONE<REGION>COMPUTE>KEY_PAIR@status=INSTANTIATED
SET.Running_Status ==                          INSTANTIATED.
Set.Status ==                                  ACTIVE.
EXECUTE.Workflow ==
        “WF_TS_CREATE_KEY_PAIR”
EXECUTE.Inactive ==                             false
ROLLBACK.Behaviour_on_error ==                  STOP
ROLLBACK.Number_of_retries ==                   0
DATA.Lock ==                                    true
  
```

The Workflow present in EXECUTE.Workflow attribute it is going to seek a KEYPAIR that match the FIND.Condition attribute with value KEYPAIR.Pubkey_Data==%GENERAL.Pubkey_Data% in Status INSTANTIATED in the DDBB, notice that we are not trying to get a VIRTUAL_MACHINE in status INSTANTIATED. The query it is going to use the Path present in the category FIND. Once found , the WF will start the activation, if the activation it is successful we set the status of the artifact as the SET.Status attribute dictates. The attribute SET.Running_Status concern about the temporal status that the artifact it is going to maintain until the final change of status that comes from SET.Status.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a “STOP” set as behavior, so no Rollback it is going to be initiated, so the execution it is going to end here in case of error.

Due to that the value of the attribute DATA.Lock is true, when the Task Definition has finished the artifact that was used in the workflow executed will be set as “Locked”.

5.16 TLD ACTIVATE PORT: ACTIVATE PORT.

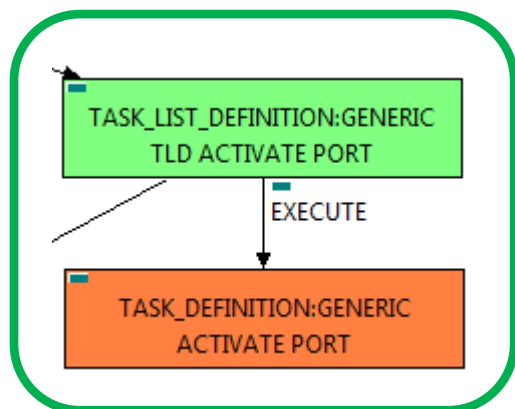


Figure: 21 Activation of a Virtual Port

The TDs that have present in the their names “Activate”, are Task Definitions responsible of the activation in the platform targeted and the updating of the status in the platform and the DDBB, in this case, the artifact that is going to be activated is a “VIRTUAL PORT”, this means, when this workflow finish, we will have a VIRTUAL PORT with status ACTIVE.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

```

GENERAL.Name ==                               Activate Port
FIND.MainArtifact==
VNF>VNF_COMPONENT>VIRTUAL_MACHINE>VIRTUAL_PORT@status=INSTANTIATED
FIND.Condition == {[VIRTUAL_PORT<SUBNETWORK<NETWORK
<TENANT:OPENSTACK<VIM]
[GENERAL.Version==constant:MITAKA]}
SET.Status ==                                  ACTIVE.
EXECUTE.Workflow ==
                                     “WF_TS_ACTIVATE_VPORT”
EXECUTE.Inactive ==                           false
ROLLBACK.Behaviour_on_error ==                STOP
ROLLBACK.Number_of_retries ==                 0
DATA.Lock ==                                  true
  
```

The Workflow present in EXECUTE.Workflow attribute it is going to seek a VIRTUAL_PORT that match the FIND.Condition which means that the VIM which it is connected to is a MITAKA and the port itself is in Status INSTANTIATED in the DDBB, notice that we are not trying to get a VIRTUAL_MACHINE in status INSTANTIATED. The query it is going to use the Path present in the category FIND. Once found , the WF will start the activation, if the activation it is successful we set the status of the artifact as the SET.Status attribute dictates. The attribute SET.Running_Status concern about the temporal status that the artifact it is going to maintain until the final change of status that comes from SET.Status.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a “STOP” set as behavior, so no Rollback it is going to be initiated, so the execution it is going to end here in case of error.

Due to that the value of the attribute DATA.Lock is true, when the Task Definition has finished the artifact that was used in the workflow executed will be set as “Locked”.

5.17 TLD ACTIVATE: ACTIVATE_VM

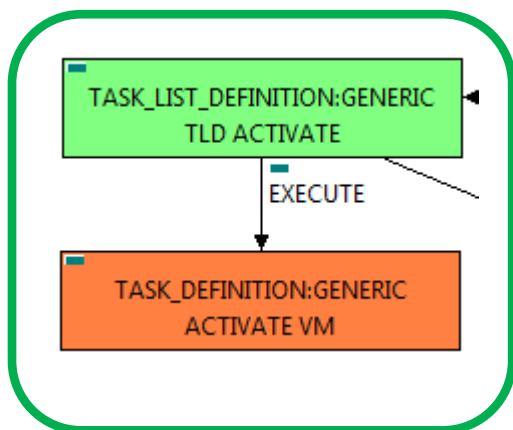


Figure: 22 Activation of Virtual Machine related to the VNF.

The TDs that have present in the their names “Activate”, are Task Definitions responsible of the activation in the platform targeted and the updating of the status in the platform and the DDBB, in this case, the artifact that is going to be activated is a “VIRTUAL_MACHINE, this means, when this workflow finish, we will have a VIRTUAL_MACNIHE with status ACTIVE.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

```

GENERAL.Name ==                Activate Virtual Machine
FIND.MainArtifact==
VNF>VNF_COMPONENT>
VIRTUAL_MACHINE@status=INSTANTIATED
SET.Running_Status ==          INSTANTIATED.
Set.Status ==                  ACTIVE.
EXECUTE.Workflow ==
    “WF_TS_ACTIVATE_VM”
EXECUTE.Inactive==             else
ROLLBACK.Behaviour_on_error == STOP
ROLLBACK.Number_of_retries == 0
DATA.Lock ==                   true
INPUT_MAPPING.MAPPING_LIST ==
ALLOCATION OWNER=VIM
  
```

The Workflow present in EXECUTE.Workflow attribute it is going to seek a “VIRTUAL_MACHINE” in Status INSTANTIATED in the DDBB . Once found , the WF will start the activation of the Virtual Machine, if the activation it is successful we set the status of the artifact as the SET.Status attribute dictates. The attribute SET.Running_Status concern about the temporal status that the artifact it is going to maintain until the final change of status that comes from SET.Status.

This TD could launch different workflows depending on the type of the VM that it is going to be activated, the main kinds of our VIRTUAL_MACHINES are CG and HELION, so two of the WFs that are going to be used in this activation are: “WF_TS_ACTIVATE_VM_CARRIER_GRADE”, “WF_TS_ACTIVATE_VM_NO_PORTS” and “WF_TS_ACTIVATE_VM_HELION”. In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a “STOP” set as behavior, so no Rollback it is going to be initiated, so the execution it is going to end here in case of error.

Due to that the value of the attribute DATA.Lock is true, when the Task Definition has finished the artifact that was used in the workflow executed will be set as “Locked”.

5.18 TLD ATTACH: ATTACH VOLUME

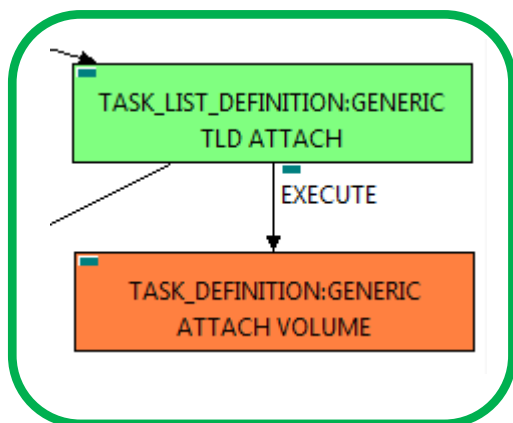


Figure: 23 Attach Volume.

The TDs that have present in the their names “Attach”, are Task Definitions responsible of the connection between artifacts, this means, this TDs will attach a VOLUME to a specific VIM, this specific VIM could change, so the workflow implied in this TD it will launch a custom WF for each kind of VIM. The VOLUMES that are activated by this TD have two final uses, directly linked with a VIM, or used as External Storage.

Once finished, we should have a number of VOLUMES activated, liable to a VIM or acting as external Storage.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

GENERAL.Name ==	Attach Volume
FIND.MainArtifact==	
VNF>VNF_COMPONENT>	
VIRTUAL_MACHINE>VIRTUAL_LUN@status=CREATED	
SET.Running_Status ==	CREATED.
Set.Status ==	ACTIVE.
EXECUTE.Workflow ==	
“WF_TS_ATTACH_VOLUME”	
EXECUTE.Inactive==	false
ROLLBACK.Behaviour_on_error ==	STOP
ROLLBACK.Number_of_retries ==	0
DATA.Lock ==	true

In case of error during the execution, the TD will jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a “STOP” as value set for behavior, so no Rollback it is going to be initiated, the execution will stop.

Due to that the value of the attribute DATA.Lock is true, when the Task Definition has finished the artifact that was used in the workflow executed will be set as “Locked”.

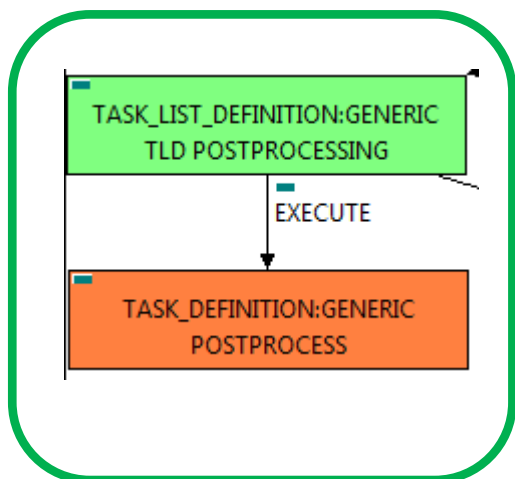


Figure: 24 Deploying Post-Processing policies.

This TD is responsible of the provision in the right order of the artifacts referenced by the Post-Processing policies, these policies allow the user to treat a number of elements that should be taken in consideration after the execution of some TD in a specific order, in other case, the execution will fail depending on the event occurred.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

FIND.MainArtifact ==	POLICY:POSTPRE_PROCESSING
FIND.Condition ==	
PROCESSING_JOB.Job_type==constant:POST&&	
PROCESSING_JOB.Operation==constant:DEPLOY	
EXECUTE.OrderBy ==	PROCESSING_JOB.OrderBy
ROLLBACK.Behaviour_on_error ==	STOP
ROLLBACK.Number_of_retries ==	0
DATA.Lock ==	false

In this TD there is not a workflow to be executed, the target of this TD is process in the correct order the PostProcessing policies present in the VNF, these policies should be executed in a specific order to make the changes or configurations properly, in other case an error will be launched

If the TD ends successfully, the Post-Processing policies will have been applied adequately.

In case of error during the execution, the TD will jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a “STOP” as value set for behavior, so no Rollback it is going to be initiated, the execution will stop.

Due to that the value of the attribute DATA.Lock is false, when the Task Definition has finished the artifact that was used in the workflow executed will remain unlocked.

5.20 • TLD START MONITORS: MONITOR START.

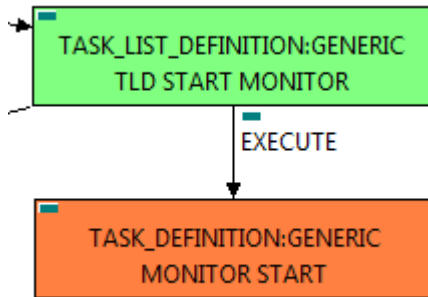


Figure: 25 Starting Monitor.

The TDs that have present in the their names “Start” are Task Definitions responsible of the launching of the component in the platform targeted and the updating of the status in the platform and the DDBB , these deployments are slightly different to the ones we launch for our entities, as a rule, they are small components as the MONITORS. In this case, the artifact that is going to be deployed is a “MONITOR”, this means, when this workflow finish, we will have a MONITOR deployed with status STARTED ready to monitories.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

GENERAL.Name ==	Monitor Start
FIND.MainArtifact ==	MONITOR
FIND.Condition ==	status==constant:DEPLOYED
SET.Running_Status ==	DEPLOYED
SET.Status ==	STARTED.
EXECUTE.Workflow ==	“WF_TS_MONITOR_START”
EXECUTE.Inactive==	false
ROLLBACK.Behaviour_on_error ==	STOP
ROLLBACK.Number_of_retries ==	0
DATA.Lock ==	true

The Workflow present in EXECUTE.Workflow attribute it is going to seek a MONITOR with Status DEPLOYED.

Once found , the WF will start the activation, if the activation it is successful we set the status of the artifact as the SET.Status attribute dictates. The attribute SET.Running_Status concern about the temporal status that the artifact it is going to maintain until the final change of status that comes from SET.Status.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a “STOP” set as behavior, so no Rollback it is going to be initiated, so the execution it is going to end here in case of error.

Due to that the value of the attribute DATA.Lock is true, when the Task Definition has finished the artifact that was used in the workflow executed will be set as “Locked”.

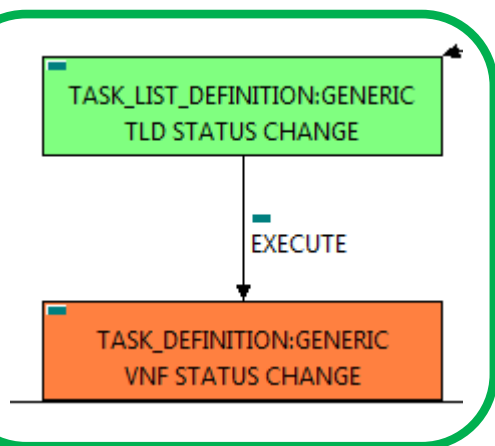


Figure: 26 Changing the status of the VNF.

The TDs that have present in their names “Status Change”, are Task Definitions responsible of the change in the status of the entity associated, in this case a VNF. When the WF has finished we will have an VNF with status **ACTIVE** in case of successful execution, or status **ERROR** in case of error, or simply not any change in the status because a **ROLLBACK** during the execution.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

GENERAL.Name ==	VNF STATUS CHANGE
FIND.MainArtifact ==	
VNF>VNF_COMPONENT>VIRTUAL_MACHINE@status=ACTIVE	
FIND.Condition ==	status==constant:INSTANTIATED
SET.Running_Status ==	INSTANTIATED.
SET.Status ==	ACTIVE.
EXECUTE.Inactive==	false
ROLLBACK.Behaviour_on_error ==	STOP
ROLLBACK.Number_of_retries ==	0
DATA.Lock ==	true

The **TASK_DEFINITION** do not execute any workflow, with the attributes present in the categories it is enough to change the status of the entity.

In case of error during the execution, the workflow jump to the **ROLLBACK** category, If the “Behaviour_on_error” attribute its set on “**ROLLBACK**” the WF will start the execution of the Workflow present in the attribute with the same name in the category **ROLLBACK**, but in this case, we have a “**STOP**” set as behavior, so no Rollback it is going to be initiated, so the execution it is going to end here in case of error.

The **TLD** will finish correctly once the execution reaches this point, the VNF will change its status to “**ACTIVE**”.

Chapter 6 Scale In of a VNF – Bottom-Up.

From now on, and to make easier the understanding of the TLDs, we are going to explain the functionality of each set of TASK_LIST_DEFINITION:GENERIC, and the number of TASK_DEFINITION:GENERIC children of the previously mentioned TASK_LIST_DEFINITION:GENERIC.

Basically, the TASK_LIST_DEFINITION:GENERIC connect what we can consider “units of execution”, those are the TASK_DEFINITION:GENERIC, that have a WORKFLOW assigned to be executed when the execution of the TLD reach them.

If you like to have a more deep knowledge about the workflows mentioned through this document please refer to the specific document.

If in the category FIND, the attribute Path is present, the attribute FIND.ArtifactType will be the starting artifact for the Path, but the FIND.Status attribute refers to the last artifact on the Path.

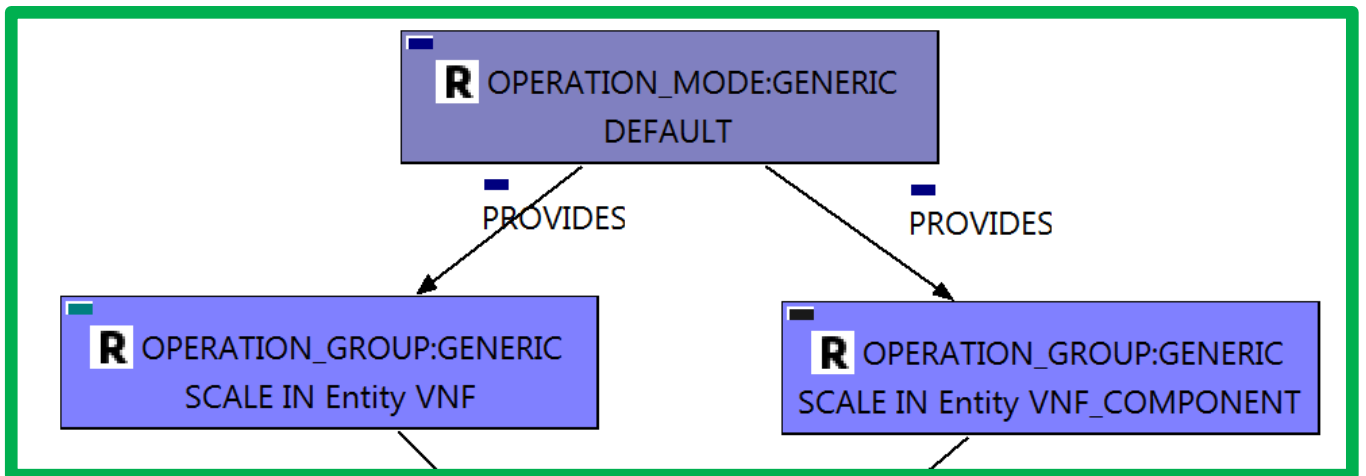
```
FIND.ArtifactType == VIRTUAL_MACHINE
FIND.Status==      INSTANTIATED
FIND.Path ==
    VIRTUAL_MACHINE>VIRTUAL_CORE<CORE<CPU<SERVER<AVAILABILITY_ZONE<REGION>
    COMPUTE>FLAVOR
```

In this example, we are looking for a FLAVOR in status INSTANTIATED, we do not expect to get a VIRTUAL_MACHINE, in status INSTANTIATED.

If during the use of the TLDs, the “Regenerate UUIDs” option is used, the user should check the Id of the tree that brings all the elements of the TLD, this “id” is specific and it will be the same for all the tree groups in all the TLDs.

The two modes available are “Default” and “Simulated”, the second one is only available if it is configured previously, by defect, and the mode that will be used is “Default”.

The Scale In TLD has some specifications that should be explained, when we open the TLD of the operation, the first we will see is the following:



This TLD starts with one Operation_Mode (as usual), but it has two Operation_Group, each one of the elements is “RootArtifact” of its own tree, this division of groups is justified in order to cover all the possible levels of the escalation policy, one group responds to the policies that has been set between the VNF and the VNF Component, and the second one respond to the policies set it in the level between the VNF Component and the Virtual Machine.

6.1 Specific Elements of the TLD Scale In VNF.

In this chapter the different elements of the specific TLD will be explained conscientiously.

6.2 TLD SCALE IN: SELECT COMPONENTS.

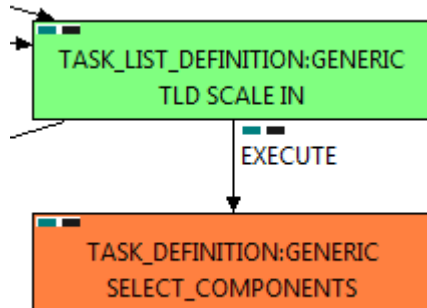


Figure: 27 Selection of the components for the Scale In.

The Scale In over a VNF is a process that delete a number of Virtual Machines that were escalated previously over a specific VNF, this means that a Scale In is only applicable to a VNF that suffered a Scale Out before.

The WF present in this TD will check for all the artifacts and relationship created for the Virtual Machine that was scaled out also will manage to leave the components that are connected with the ones that are going to be deleted in an adequate state to be used.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

GENERAL.Name == Select Components

EXECUTE.Workflow ==

“WF_TS_SCALE_IN_SELECTED”

EXECUTE.Inactive==

false

ROLLBACK.Behaviour_on_error ==

ROLLBACK

ROLLBACK.Rollback_Status ==

ACTIVE

ROLLBACK.Number_of_retries ==

0

DATA.Lock ==

false

The Workflow present in EXECUTE.Workflow it is going to get whatever we have passed to the job and start. This workflow change the status of the number of the immediate children and recursive of all the POLICY:ENTITY_RANGE found equals to its SCALE.DEFAULT_SCALE_IN field.

In case of error during the execution, the workflow jump to the ROLLBACK category, if the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, the attribute “number_of_retries” set the number of rollback attempts. In this case, the behavior is set as “ROLLBACK” without a rollback workflow assigned, in case of error the operation will rollback the transaction.

Due to that the value of the attribute DATA.Lock is false, once the TD has finished, no artifact or element will be set as “locked”.

6.3 SCALE IN PRE TLD: PREPROCESS.

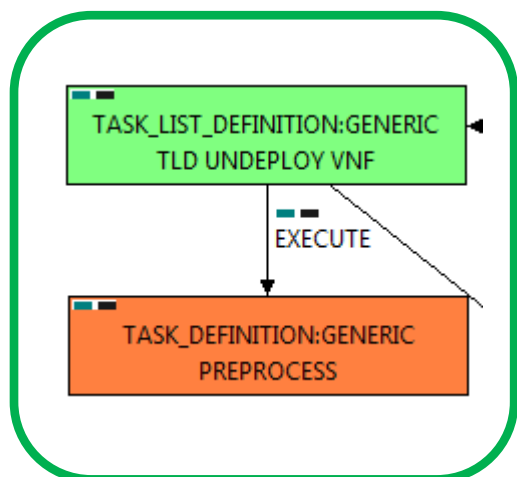


Figure: 28 Undeployment of Pre-Processing policies

This TD is responsible of the provision in the right order of the artifacts referenced by the PreProcessing policies, these policies allow the user to set a number of elements that should be taken in consideration in a certain order, in other case, the execution will fail depending on the event occurred.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

```

FIND.MainArtifact == POLICY:POSTPRE_PROCESSING
FIND.Condition ==
PROCESSING_JOB.Job_type==constant:PRE&&
PROCESSING_JOB.Operation==constant:SCALEIN
EXECUTE.OrderBy == PROCESSING_JOB.OrderBy
ROLLBACK.Behaviour_on_error== ROLLBACK
ROLLBACK.Number_of_retries == 0
DATA.Lock == false
  
```

In this TD there is not a workflow to be executed, the target of this TD is process in the correct order the PreProcessing policies present in the VNF, these policies should be executed in a specific order to make the changes or configurations properly, in other case an error will be launched

If the TD ends successfully, the Pre-Processing policies will have been processed adequately.

In case of error during the execution, the TD will jump to the ROLLBACK category, If the "Behaviour_on_error" attribute its set on "ROLLBACK" the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK.

Due to that the value of the attribute DATA.Lock is false, when the Task Definition has finished the artifact that was used in the workflow executed will remain unlocked.

6.4 TLD STOP MONITOR: STOP MONITOR

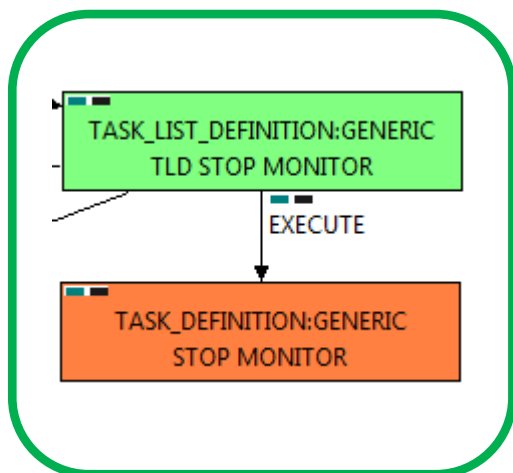


Figure: 29 Stopping monitor.

The TDs that have present in the their names “Stop”, are Task Definitions responsible of the stopping of an specific artifact or element, in this case of the MONITORS, the TLD it is going to stop the MONITOR element given.

Once finished, our VNF should have the MONITOR given in status DEPLOYED, having taken in consideration all the rules for the stopping.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

GENERAL.Name ==	Stop Monitor
FIND.MainArtifact==	MONITOR
FIND.Condition==	status==constant:TO_BE_STOPPED
SET.Running_Status ==	TO_BE_STOPPED
Set.Status ==	TO_BE_UNDEPLOYED
EXECUTE.Workflow ==	“WF_TS_MONITOR_STOP”
EXECUTE.Inactive==	false
EXECUTE.OrderBy==	GENERAL.Order
ROLLBACK.Behaviour_on_error ==	ROLLBACK
ROLLBACK.Number_of_retries ==	0
DATA.Lock ==	false

The Workflow present in EXECUTE.Workflow it is going to seek a MONITOR in Status STARTED in the DDBB, when the WF find it, it will start. This workflow stop the given MONITOR needed by the VNF to get a successful Undeploy.

In case of error during the execution, the TD will jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a “STOP” as value set for behavior, so no Rollback it is going to be initiated, the execution will stop.

Due to that the value of the attribute DATA.Lock is false, when the Task Definition has finished the artifact that was used in the workflow executed will remain unlocked.

6.5 TLD STOP MONITOR: DETACH VOLUME

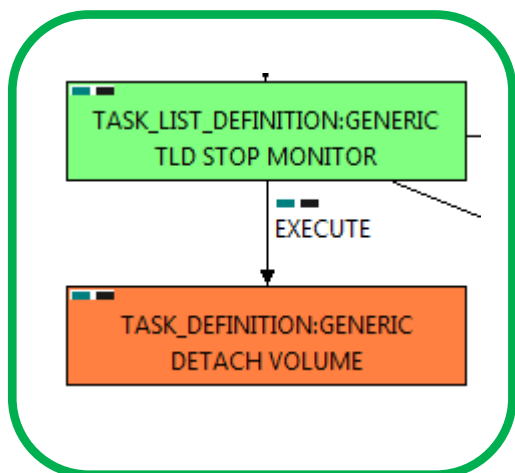


Figure: 30 Detach Volume.

The TDs that have present in the their names “Detach”, are Task Definitions responsible of the detaching of an specific artifact or element, in this case of the VOLUME, the TLD it is going to detach the VOLUME element given.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

```

GENERAL.Name == DETACH VOLUME
FIND.MainArtifact==
VNF>VNF_COMPONENT>VIRTUAL_MACHINE>
VIRTUAL_LUN@status=TO_BE_DELETED,VNF_COMPONENT>
VIRTUAL_MACHINE>VIRTUAL_LUN@status=TO_BE_DELETED
EXECUTE.Inactive== false
EXECUTE.OrderBy== GENERAL.Order
ROLLBACK.Behaviour_on_error == ROLLBACK
ROLLBACK.Number_of_retries == 0
ROLLBACK.Workflow== “WF_TS_ATTACH_VOLUME”
DATA.Lock == false
  
```

This Task is kind of special, as it does not have an execute workflow. This Task is not going to execute anything.

In case of error during the execution, the TD will jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a “ROLLBACK” as value set for behavior, and as the ROLLBACK.Workflow category is filled, it is going to execute the WF_TS_ATTACH_VOLUME workflow.

Due to that the value of the attribute DATA.Lock is false, when the Task Definition has finished the artifact that was used in the workflow executed will remain unlocked.

6.6 TLD DEACTIVATE VM: DEACTIVATE VM.

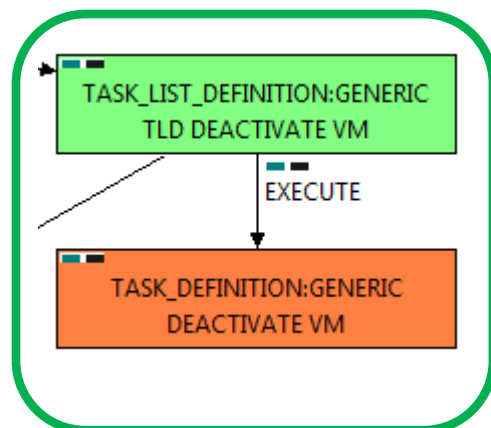


Figure: 31 Deactivation of a Virtual Machine.

The TDs that have present in the their names “Deactivate”, are Task Definitions responsible of the deactivation in the platform targeted and the updating of the status in the platform and the DDBB, in this case, the artifact that is going to be deactivated is a “VIRTUAL_MACHINE with status “STOPPED”, this means, when this workflow finish, we will have the VIRTUAL_MACHINE given with a new status INSTANTIATED.

Targets of the TASK DEFINITION:
 STATUS of the TD: ENABLED
 Categories:

```

FIND.MainArtifact ==
VNF>VNF_COMPONENT>
VIRTUAL_MACHINE@status=TO_BE_DEACTIVATED,
VNF_COMPONENT>
VIRTUAL_MACHINE@status=TO_BE_DEACTIVATED
SET.Running_Status == TO_BE_DEACTIVATED.
SET.Status == TO_BE_DELETED.
EXECUTE.Workflow == "WF_TS_DEACTIVATE_VM"
EXECUTE.Inactive== false
ROLLBACK.Behaviour_on_error == ROLLBACK
ROLLBACK.Number_of_retries == 0
DATA.Lock == false
  
```

The Workflow present in EXECUTE.Workflow attribute it is going to deactivate a “VIRTUAL_MACHINE” in Status TO_BE_DELETED in the DDBB. Once found, the WF will start the deactivation, if the deactivation it is successful we set the status of the artifact as the SET.Status attribute dictates. The attribute SET.Running_Status concern about the temporal status that the artifact it is going to maintain until the final change of status that comes from SET.Status.

This TD could launch different workflows depending on the type of the VM that it is going to be deactivated, the main kinds of our VIRTUAL_MACHINES are HELION, and thereupon two of the WFs that could be used in this deactivation are: “WF_NFVD_DEACTIVATE_VM_HELION” and “WF_NFVD_DEACTIVATE_VM_OPENSTACK”.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, in this case it will be “WF_TS_ACTIVATE_VM”, in this case we have a “ROLLBACK” set as behavior, so it is going to try to ACTIVATE again the Virtual Machine.

Due to that the value of the attribute DATA.Lock is false, when the Task Definition has finished the artifact that was used in the workflow executed will remain unlocked.

6.7 TLD UNDEPLOY POST: UNDEPLOY POST.

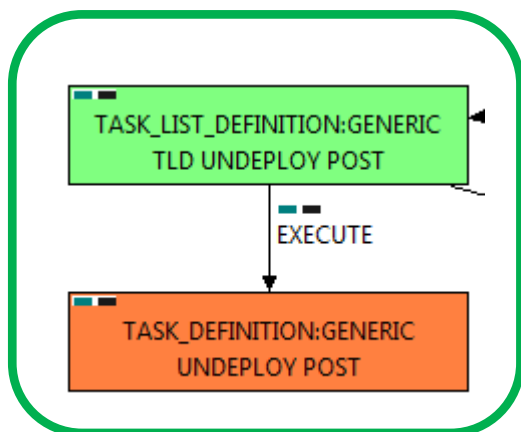


Figure: 32 Undeploying Post-Processing policies.

This TD is responsible of the provision in the right order of the artifacts referenced by the Post-Processing policies, these policies allow the user to treat a number of elements that should be taken in consideration after the execution of some TD in a specific order, in other case, the execution will fail depending on the event occurred.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

FIND.MainArtifact ==	POLICY:POSTPRE_PROCESSING
FIND.Condition ==	
PROCESSING_JOB.Job_type==constant:POST&&	
PROCESSING_JOB.Operation==constant:SCALEIN	
EXECUTE.OrderBy ==	PROCESSING_JOB.OrderBy
EXECUTE.Inactive==	false
ROLLBACK.Behaviour_on_error ==	STOP
ROLLBACK.Number_of_retries ==	0
DATA.Lock ==	false

In this TD there is not a workflow to be executed, the target of this TD is process in the correct order the PostProcessing policies present in the VNF, these policies should be executed in a specific order to make the changes or configurations properly, in other case an error will be launched

If the TD ends successfully, the Post-Processing policies will have been applied adequately.

In case of error during the execution, the TD will jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, as we have a “ROLLBACK” as value set for behavior, we are going to rollback the transaction.

Due to that the value of the attribute DATA.Lock is false, when the Task Definition has finished the artifact that was used in the workflow executed will remain unlocked.

The TDs that have present in their names “Delete”, are Task Definitions that delete an artifact or element from the DDBB or from the platforms targeted, in this case, the artifact that it is going to be deleted is the VIRTUAL_PORT given.

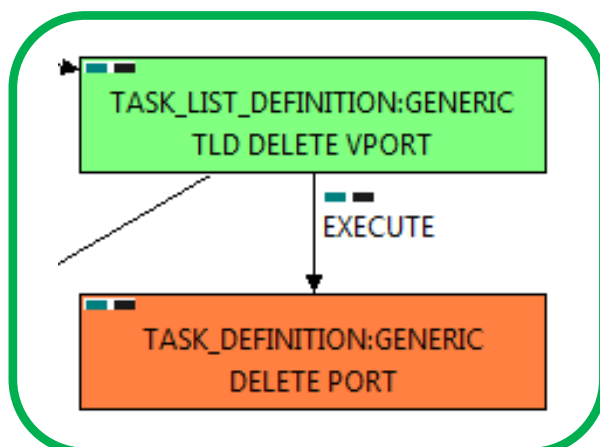


Figure: 33 Deletion of a Virtual Port.

Targets of the TASK:DEFINITION: **STATUS of the TD:** **ENABLED**
Categories:

```

FIND.MainArtifact==
VNF>VNF_COMPONENT>VIRTUAL_MACHINE>VIRTUAL_PORT@status=TO_BE_DELETED,VNF_COMPONENT>VI
RTUAL_MACHINE>VIRTUAL_PORT@status=TO_BE_DELETED
EXECUTE.Workflow ==                    "WF_TS_DEACTIVATE_PORT"
EXECUTE.Inactive==                    false
ROLLBACK.Behaviour_on_error ==                    ROLLBACK
ROLLBACK.Number_of_retries ==                    0
DATA.Lock ==                                            true
  
```

The Workflow present in EXECUTE.Workflow it is going to seek a VIRTUAL_PORT in Status ACTIVE in the DDBB, when the WF find it, it will start. This workflow will start one more, this last one, it is going to be named after the VIRTUAL_PORT that the TD it is trying to delete.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, as we have a “ROLLBACK” set as behavior, it is going to rollback the transaction.

Due to that the value of the attribute DATA.Lock is true, when the Task Definition has finished the artifact that was used in the workflow executed will be set as “Locked”.

6.9 TLD UNDEPLOY MONITOR VOLUME: DELETE VOLUME

The TDs that have present in the their names “Delete”, are Task Definitions responsible of the deletion of an artifact in the platform targeted and the updating of the status in the platform and the DDBB, in this case, the artifact that is going to be deleted is a “VOLUME”, this means, when this workflow finish, we are going to have a volume less.

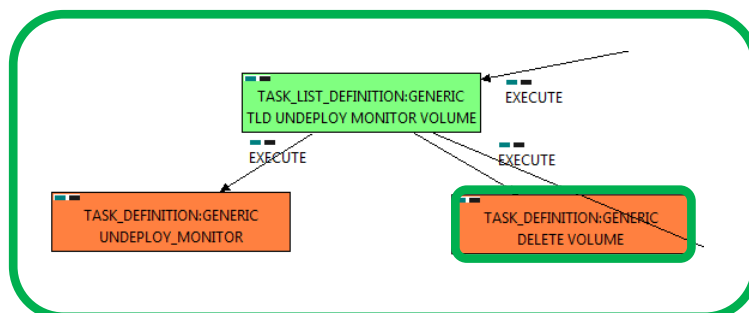


Figure: 34 Deletion of the Volume

Targets of the TASK:DEFINITION:
Categories:

STATUS of the TD: ENABLED

```

GENERAL.Name == Delete Volume
FIND.mainArtifact == VNF>VNF_COMPONENT>VIRTUAL_MACHINE>VIRTUAL_LUN@status=TO_BE_DELETED,
VNF_COMPONENT>VIRTUAL_MACHINE>VIRTUAL_LUN@status=TO_BE_DELETED.
SET.Running_Status == TO_BE_DELETED.
SET.Status == TO_BE_DELETED
EXECUTE.Workflow == "WF_TS_DELETE_VOLUME"
ROLLBACK.Behaviour_on_error == STOP
ROLLBACK.Number_of_retries == 0
DATA.Lock == false
  
```

The Workflow present in EXECUTE.Workflow attribute it is going to seek a VIRTUAL_LUN in the DDBB . Once found , the WF will start the deleting, if the deletion it is successful we set the status of the artifact as the SET.Status attribute dictates. The attribute SET.Running_Status concern about the temporal status that the artifact it is going to maintain until the final change of status that comes from SET.Status.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, in this case it will be “WF_TS_CREATE_VOLUME”.

The attribute DATA.Lock is set with a value of “false”, this means no element will be locked at the of the TD’s execution.

The TDs that have present in the their names “Undeploy” are Task Definitions responsible of the undeployment in the platform targeted and the updating of the status in the platform and the DDBB , these deployments are slightly different to the ones we launch for our entities, as a rule, they are small components as the MONITORS. In this case, the artifact that is going to be deployed is a “MONITOR”, this means, when this workflow finish, we will have a MONITOR deployed with status DEPLOYED.

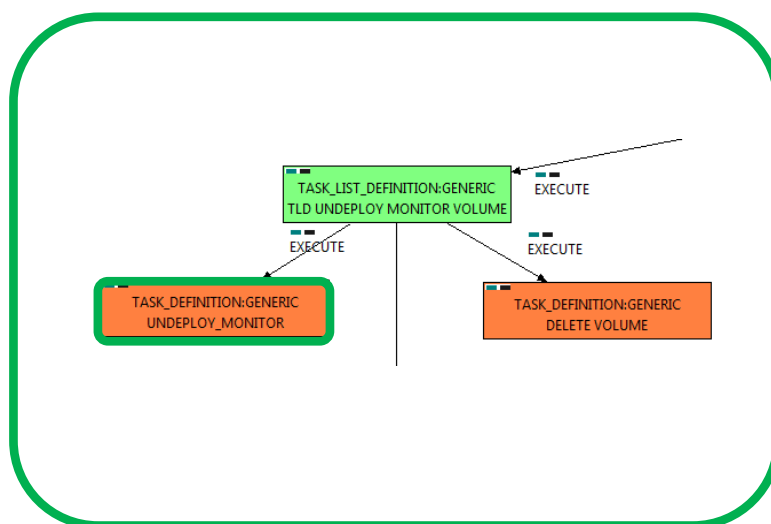


Figure: 35 Undeploy Monitor

Targets of the TASK:DEFINITION:
Categories:

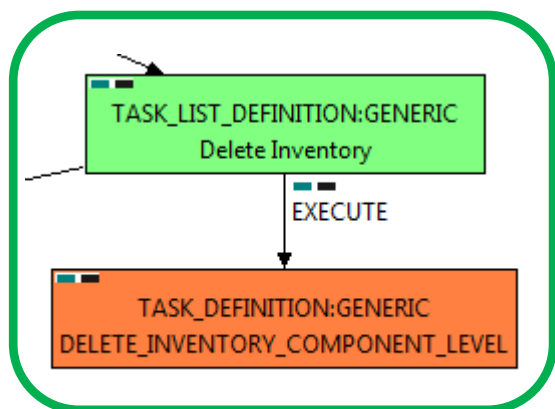
STATUS of the TD: ENABLED

GENERAL.Name==	UNDEPLOY MONITOR
FIND.ArtifactType ==	MONITOR.
FIND.Condition ==	status==constant:TO_BE_UNDEPLOYED
SET.Running_Status ==	TO_BE_UNDEPLOYED.
SET.Status ==	TO_BE_DELETED
EXECUTE.Workflow ==	“WF_TS_MONITOR_UNDEPLOY”
ROLLBACK.Behaviour_on_error ==	STOP
ROLLBACK.Number_of_retries ==	0
DATA.Lock ==	false

The Workflow present in EXECUTE.Workflow attribute it is going to seek a MONITOR in Status DEPLOYED in the DDBB . Once found, the WF will start the deployment, if the deployment is successful we set the status of the artifact as the SET.Status attribute dictates. The attribute SET.Running_Status concern about the temporal status that the artifact it is going to maintain until the final change of status that comes from SET.Status.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a “STOP” set as behavior, so no Rollback it is going to be initiated, so the execution it is going to end here in case of error.

Due to that the value of the attribute DATA.Lock is false, when the Task Definition has finished the artifact that was used in the workflow executed will remain unlocked.



The TDs that have present in their names “Delete Inventory”, are Task Definitions responsible of deleting the entity associated, in this case responsible of the deletion of a specific entity by ID, these entities are the components related to the Virtual Machine that is going to be deleted during the Scale In process of the VNF. When the WF has finished we will have an VNF with status ACTIVE in case of successful execution, or status ERROR in case of error, or simply not any change in the status because a ROLLBACK during the execution.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

Figure: 36 Deletion at component level.

GENERAL.Name ==	DELETE VNF
FIND.MainArtifact ==	
VNF>POLICY:ENTITY_RANGE>	
VNF_COMPONENT@status=TO_BE_DELETED	
EXECUTE.Inactive==	false
EXECUTE.Workflow ==	
“WF_TS_DELETE_INSTANCE_TREE_BY_ARTIFACT_ID”	
ROLLBACK.Behaviour_on_error ==	STOP
ROLLBACK.Number_of_retries ==	0
DATA.Lock ==	false

The TASK_DEFINITION do not execute any workflow, with the attributes present in the categories it is enough to change the status of the entity.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, as we have a “ROLLBACK” set as behavior, it will rollback the transaction.

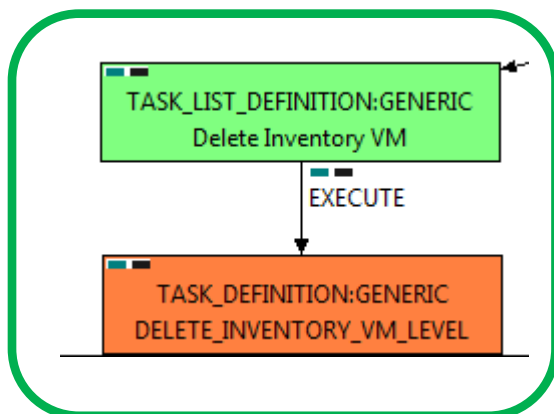


Figure: 37 Delete Virtual Machine from scaled in VNF.

The TDs that have present in their names “Delete Inventory”, are Task Definitions responsible of deleting the entity associated, in this case responsible of the deletion of a specific entity by ID, these entities are the components related to the Virtual Machine that is going to be deleted during the Scale In process of the VNF. When the WF has finished we will have an VNF with status ACTIVE in case of successful execution, or status ERROR in case of error, or simply not any change in the status because a ROLLBACK during the execution.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

```

GENERAL.Name ==                               DELETE_INVENTORY_VM
FIND.MainArtifact ==
VNF>VNF_COMPONENT>POLICY:ENTITY_RANGE>
VIRTUAL_MACHINE@status=TO_BE_DELETED,
VNF_COMPONENT>POLICY:ENTITY_RANGE>
VIRTUAL_MACHINE@status=TO_BE_DELETED
FIND.Condition ==                             status==constant:INSTANTIATED
SET.Running_Status ==                         INSTANTIATED.
SET.Status ==                                 ACTIVE.
EXECUTE.Workflow==
        “WF_TS_DELETE_INSTANCE_TREE_BY_ARTIFACT_ID”
EXECUTE.Inactive==                            false
ROLLBACK.Behaviour_on_error ==                STOP
  
```

The TASK_DEFINITION do not execute any workflow, with the attributes present in the categories it is enough to change the status of the entity.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, as we have a “ROLLBACK” set as behavior, it will rollback the transaction.

Chapter 7 Scale Out of a VNF – Bottom-Up.

From now on, and to make easier the understanding of the TLDs, we are going to explain the functionality of each set of TASK_LIST_DEFINITION:GENERIC, and the number of TASK_DEFINITION:GENERIC children of the previously mentioned TASK_LIST_DEFINITION:GENERIC.

Basically, the TASK_LIST_DEFINITION:GENERIC connect what we can consider “units of execution”, those are the TASK_DEFINITION:GENERIC, that have a WORKFLOW assigned to be executed when the execution of the TLD reach them.

If you like to have a more deep knowledge about the workflows mentioned through this document please refer to the specific document.

If in the category FIND, the attribute Path is present, the attribute FIND.ArtifactType will be the starting artifact for the Path, but the FIND.Status attribute refers to the last artifact on the Path.

```
FIND.ArtifactType == VIRTUAL_MACHINE
FIND.Status==      INSTANTIATED
FIND.Path ==
    VIRTUAL_MACHINE>VIRTUAL_CORE<CORE<CPU<SERVER<AVAILABILITY_ZONE<REGION>
    COMPUTE>FLAVOR
```

In this example, we are looking for a FLAVOR in status INSTANTIATED, we do not expect to get a VIRTUAL_MACHINE, in status INSTANTIATED.

If during the use of the TLDs, the “Regenerate UUIDs” option is used, the user should check the Id of the tree that brings all the elements of the TLD, this “id” is specific and it will be the same for all the tree groups in all the TLDs.

The two modes available are “Default” and “Simulated”, the second one is only available if it is configured previously, by defect, and the mode that will be used is “Default”.

The Scale Out TLD has some specifications that should be explained, when we open the TLD of the operation, the first we will see is the following:

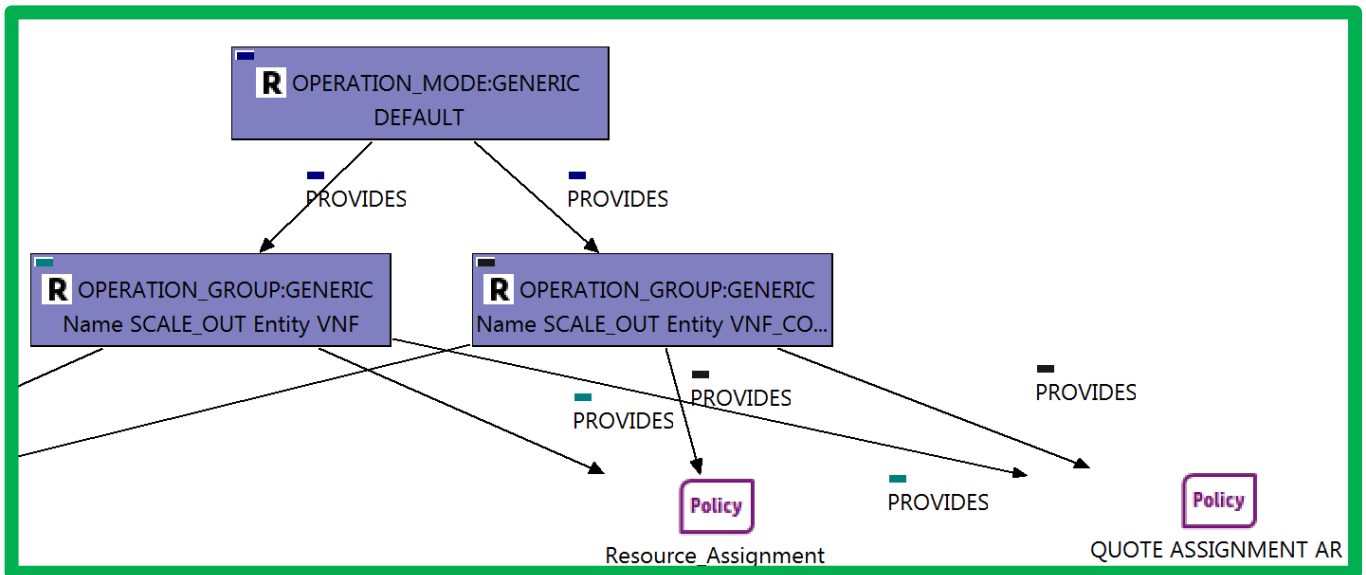


Figure: 38 Scale Out Operations MODE&GROUP

This TLD starts with one Operation_Mode (as usual), but it has two Operation_Group, each one of the elements is “RootArtifact” of its own tree, the assignation of resources and quotas is equal for both of them, this division of groups is justified in order to cover all the possible levels of the escalation policy, one group responds to the policies that has been set between the VNF and the VNF Component, and the second one respond to the policies set it in the level between the VNF Component and the Virtual Machine.

7.1 Specific Elements of the TLD Scale Out of a VNF.

In this chapter the different elements of the specific TLD will be explained conscientiously.

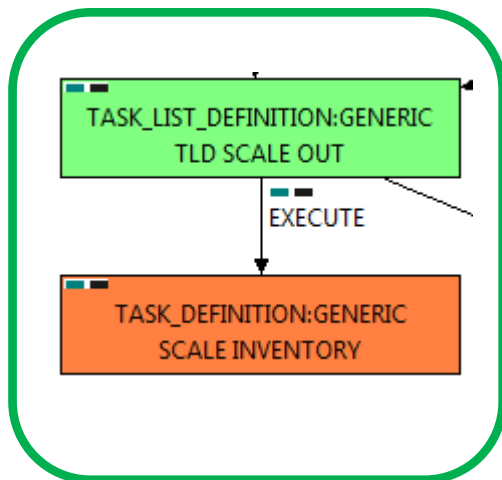


Figure: 39 Scale Inventory Task.

7.2 TLD SCALE OUT: SCALE INVENTORY.

This TD is the responsible to list and harvest all the artifacts and relationships of the Virtual Machine that is going to be escalated, inside this TD another workflow will be thrown, "WF_NFVD_CREATE_INSTANCES_FROM_TEMPLATE_ROOT", it will be the responsible of the creation of the new instances related to the new Virtual Machine as from the templates.

Once finished, our Scale should have the template to start the creation and deployment processes of the newly escalated Virtual Machine.

Targets of the TASK DEFINITION: STATUS of the TD:
ENABLED

```

GENERAL.Name == Scale Inventory Task
FIND.MainArtifact ==
VNF>POLICY:ENTITY_RANGE,
VNF>VNF_COMPONENT>POLICY:ENTITY_RANGE,
VNF_COMPONENT>POLICY:ENTITY_RANGE
EXECUTE.Workflow ==
"WF_TS_SCALE_OUT_COMPONENT"
EXECUTE.Inactive == false
ROLLBACK.Behaviour_on_error == ROLLBACK
ROLLBACK.Number_of_retries == 0
DATA.Lock == false
  
```

The Workflow present in EXECUTE.Workflow it is going to seek an ENTITY_RANGE in Status INSTANTIATED in the DDBB, when the WF find it, it will start. This workflow assign all the resources needed by the newly instantiated Virtual Machine to get a successful Deploy, it will check the available resources and decide which one should be assigned.

The creation of instances for the newly created Virutal Machine it uses another WF that it is called from our workflow, "WF_NFVD_CREATE_INSTANCES_FROM_TEMPLATE_ROOT".

In case of error during the execution, the workflow jump to the ROLLBACK category, if the "Behaviour_on_error" attribute its set on "ROLLBACK" the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, the attribute "number_of_retries" set the number of rollback attempts. In this case, the TLD has not assigned a rollback workflow, so in this case the TD will only change the status of the artifact which is being used.

Due to that the value of the attribute DATA.Lock is false, once the TD has finished, no element used in the previous execution will be set as "Locked".

7.3 TLD QUOTA ASSIGNMENT: QUOTA ASSIGNMENT.

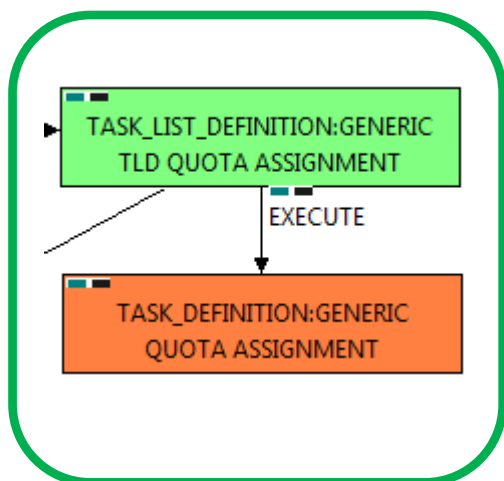


Figure: 40 Quota Assignment

The TDs that have present in the their names “Assignment”, are Task Definitions responsible of the assignation of resources for an specific artifact, in the case of the quotas, the TLD it is going to assign an amount of each resource needed for the correct execution of the deployment.

Once finished, our VNF should have every quota needed for a successful deployment assigned, having taken in consideration all the rules for the assignment. This is crucial, because our component consume quotas during the execution of the TLD.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

```

GENERAL.Name ==                               Quota_Assignment Task
FIND.MainArtifact ==                           VNF,VNF_COMPONENT<VNF
EXECUTE.Workflow ==                             “WF_NFVD_CREATE_CONSUMED_FOR_VNF”
EXECUTE.Inactive==                             false
ROLLBACK.Behaviour_on_error ==                 ROLLBACK
ROLLBACK.Number_of_retries ==                  0
DATA.Lock ==                                   true
INPUT_MAPPING.MAPPING_LIST ==                  resourceTreeID=nfvd#quotaResourceID
  
```

The Workflow present in EXECUTE.Workflow it is going to seek a VNF in Status INSTANTIATED in the DDBB, when the WF find it, it will start. This workflow assign all the resources needed by the VNF to get a successful Deploy, it will check the quotas and make relationships with them.

In case of error during the execution, the workflow jump to the ROLLBACK category, if the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, the attribute “number_of_retries” set the number of rollback attempts. In this case, the TLD has not assigned a rollback workflow, so in this case the TD will only change the status of the artifact which is being used.

Due to that the value of the attribute DATA.Lock is true, once the TD has finished, the Quota recently assigned, will be blocked.

7.4 TLD RESOURCE ASSIGNMENT: ALLOCATE RESOURCES VNF WITH AFFINITY.

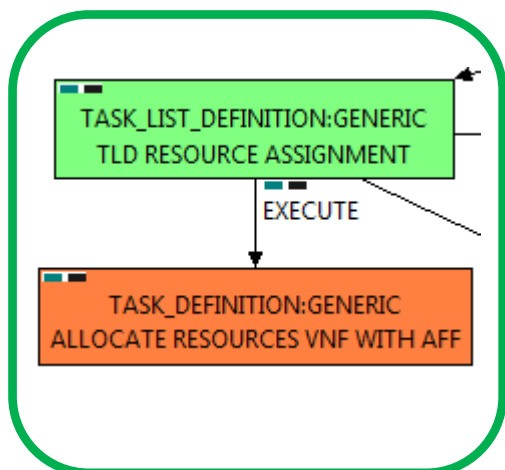


Figure: 41 Assignment of the resources with affinity.

The TDs that have present in their names “Assignment”, are Task Definitions responsible of the assignation of resources for a specific artifact or deploy, in this case, we are looking for a VNF to assign the resources needed for the future deployment. In order to have a successful assignation we must have in our TLD Deploy VNF an artifact POLICY: ASSIGMENT_RELATIONSHIP, with a GENERAL.Name==”Resource_Assignment”, also this artifact must be related with the OPERATION_GROUP: GENERIC of our TLD with a relationship of type PROVIDES and status ENABLED.

Once finished, our VNF should have every resources needed for a successful deployment assigned, having taken in consideration all the rules for the assignment.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

```

GENERAL.Name ==                Allocate Resources with Affinity
FIND.MainArtifact ==           VNF,VNF_COMPONENT<VNF.
FIND.Condition ==
{[VNF>POLICY:AFFINITY][VNF>POLICY:ANTI_AFFINITY]
[GENERAL.Name!=""]}
EXECUTE.Workflow ==
“WF_NFVD_RESOURCE_ALLOCATION_WITH_AFFINITIES”
EXECUTE.Inactive==                false
ROLLBACK.Behaviour_on_error ==    ROLLBACK
ROLLBACK.Number_of_retries ==     0
DATA.Lock ==                      true
INPUT_MAPPING.MAPPING_LIST ==
assignmentRelationshipID=Resource_Assignment;
resourceTreeID=resourceArtifactID;
def_exclusion_list=TENANT:OPENSTACK,NETWORKING,
COMPUTE,IMAGE_STORAGE

```

The Workflow present in EXECUTE.Workflow it is going to seek a VNF in the DDBB with the path given by the FIND.MainArtifact attribute, and with the condition that it has to find an affinity or anti_affinity; when the WF find it, it will start. This workflow assign all the resources needed by the VNF to get a successful Deploy, it will check the available resources and decide which one should be assigned.

The Workflow also check the affinity policies , in case our TLD has it, the way the assignation it is going to behave depends also of this policies, once checked, we launch the assignation of resources.The assignation of resources it uses another WF that it is called from our workflow, “WF_NFVD_ASSIGN_RESOURCES”.

In case of error during the execution, the workflow jump to the ROLLBACK category, if the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, the attribute “number_of_retries” set the number of rollback attempts. In this case, the TLD has not assigned a rollback workflow, so in this case the TD will only change the status of the artifact which is being used.

Due to that the value of the attribute DATA.Lock is true, once the TD has finished, the artifact recently assigned, will be blocked.

7.5 TLD RESOURCE ASSIGNMENT: ALLOCATE RESOURCES VNF.

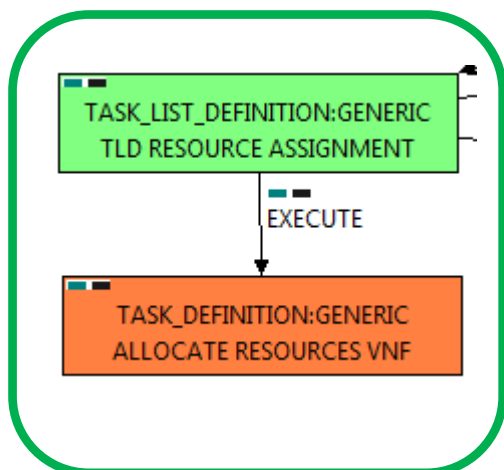


Figure: 42 Assignment of the resources without affinity.

The TDs that have present in their names “Assignment”, are Task Definitions responsible of the assignation of resources for a specific artifact or deploy, in this case, we are looking for a VNF to assign the resources needed for the future deployment. In order to have a successful assignation we must have in our TLD Deploy VNF an artifact POLICY: ASSIGMENT_RELATIONSHIP, with a GENERAL.Name==”Resource_Assignment”, also this artifact must be related with the OPERATION_GROUP: GENERIC of our TLD with a relationship of type PROVIDES and status ENABLED.

Once finished, our VNF should have every resources needed for a successful deployment assigned, having taken in consideration all the rules for the assignment.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

```

GENERAL.Name ==                               Allocate Resources
FIND.MainArtifact ==                           VNF,VNF_COMPONENT<VNF.
FIND.Condition==
!{[VNF>POLICY:AFFINITY][VNF>POLICY:ANTI_AFFINITY]
[GENERAL.Name!=""]}
EXECUTE.Workflow ==
        “WF_NFVD_RESOURCE_ALLOCATION”
EXECUTE.Inactive==                             false
ROLLBACK.Behaviour_on_error ==                 ROLLBACK
ROLLBACK.Number_of_retries ==                   0
DATA.Lock ==                                   true
INPUT_MAPPING.MAPPING_LIST ==
assignmentRelationshipID=Resource_Assignment;
resourceTreeID=resourceArtifactID;
def_exclusion_list=TENANT:OPENSTACK,NETWORKING,
COMPUTE,IMAGE_STORAGE

```

The Workflow present in EXECUTE.Workflow it is going to seek a VNF in the DDBB with the path given by the FIND.MainArtifact attribute, and with no affinity nor anti_affinity policies when the WF find it, it will start. This workflow assign all the resources needed by the VNF to get a successful Deploy, it will check the available resources and decide which one should be assigned.

The Workflow also check the affinity policies , in case our TLD has it, the way the assignation it is going to behave depends also of this policies, once checked, we launch the assignation of resources.The assignation of resources it uses another WF that it is called from our workflow, “WF_NFVD_ASSIGN_RESOURCES”.

In case of error during the execution, the workflow jump to the ROLLBACK category, if the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, the attribute “number_of_retries” set the number of rollback attempts. In this case, the TLD has not assigned a rollback workflow, so in this case the TD will only change the status of the artifact which is being used.

Due to that the value of the attribute DATA.Lock is true, once the TD has finished, the artifact recently assigned, will be blocked.

7.6 TLD IMAGE: GET IMAGE.

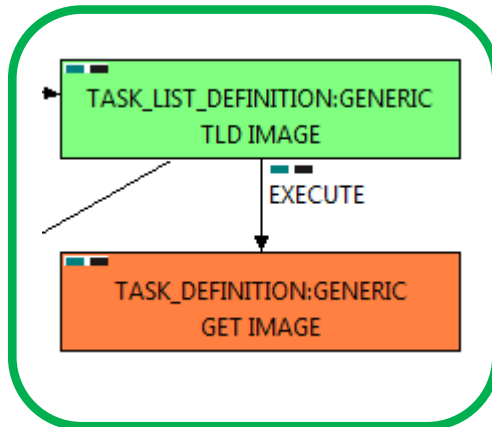


Figure: 43 Checking the image permissions.

The TDs that have present in the their names “Provision”, are Task Definitions responsible of the creation and store of an artifact in DDBB, in this case, the artifact that it is going to be provisioned it is an “TENANT:OPENSTACK”, this means, when this workflow finish, we will have a new artifact “TENANT:OPENSTACK” in our DDBB, as well, due to the nature of the artifact, the artifact will be prepared to be activated in the platform Openstack when will be required.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

```

GENERAL.Name ==                               Get Image
FIND.MainArtifact ==
VNF>VNF_COMPONENT>
VIRTUAL_MACHINE@status=INSTANTIATED,
VNF_COMPONENT>
VIRTUAL_MACHINE@status=INSTANTIATED
SET.Running_Status ==                         INSTANTIATED.
SET.Status ==                                 INSTANTIATED.
EXECUTE.Workflow ==
        “WF_TS_CHECK_VM_IMAGE”
EXECUTE.Inactive==                             false
ROLLBACK.Behaviour_on_error ==                 ROLLBACK
ROLLBACK.Number_of_retries ==                  0
DATA.Lock ==                                   true
  
```

The Workflow present in EXECUTE.Workflow it is going to seek a VIRTUAL_MACHINE in Status INSTANTIATED in the DDBB, when the WF find it, it will start. This workflow will start another two more, the one that check the permissions of the IMAGE, “WF_TS_CHECK_IMAGE_PERMISSIONS”, and the one that will deploy our IMAGE if it is not deployed, “WF_TS_DEPLOY_IMAGE”.

In case of IMAGE managed by VIM, “WF_TS_CHECK_IMAGE_PERMISSIONS will work the same way, however in that case “WF_TS_DEPLOY_IMAGE” couldn’t work the same way, the image can be reused it previously has been deployed in the VIM, but can not generate a new deployed IMAGE.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, the attribute “number_of_retries” set the number of rollback attempts. If an error take place in this TD, no action will be taken, the execution of the TLD will try to start a rollback workflow but there is not a workflow to be executed in the attribute ROLLBACK.Workflow, so the execution will continue.

Due to that the value of the attribute DATA.Lock is true, once the TD has finished, the artifact recently checked, will be blocked.

7.7 TLD VIM SPECS: GET KEYPAIR.

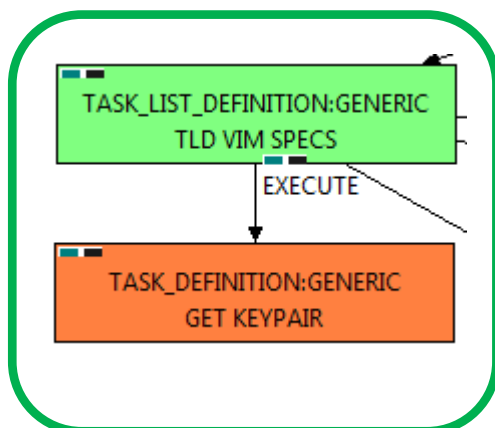


Figure: 44 Creation the Keypair element

The TDs that have present in the their names “Activate”, are Task Definitions responsible of the activation in the platform targeted and the updating of the status in the platform and the DDBB, in this case, the artifact that is going to be activated is a “TENANT:OPENSTACK”, this means, when this workflow finish, we will have a TENANT:OPENSTACK with status ACTIVE in our Openstack platform, also the TD will update the status and other attributes of the instance that represents the artifact TENANT:OS in the DDBB and in the platform, creating all the relationships needed for a correct activation.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

```

GENERAL.Name ==                               Get KeyPair
FIND.MainArtifact ==
VNF>VNF_COMPONENT>VIRTUAL_MACHINE@status=INSTANTIATED,
VNF_COMPONENT>VIRTUAL_MACHINE@status=INSTANTIATED
FIND.Condition ==
    KEYPAIR.Pubkey_Data != null || KEYPAIR.Pubkey_Path != null
EXECUTE.Workflow ==
    “WF_TS_NFVD_CREATE_KEY_PAIR_INVENTORY”
EXECUTE.Inactive==                             false
EXECUTE.OrderBy ==
ROLLBACK.Behaviour_on_error ==                 ROLLBACK
ROLLBACK.Number_of_retries ==                   0
DATA.Lock ==                                    true
  
```

The Workflow present in EXECUTE.Workflow it is going to seek a VIRTUAL_MACHINE in Status INSTANTIATED in the DDBB, also the artifact which we are looking for have to match the FIND.Condition, means, that our VM must have as KEYPAIR.Pubkey_Data a not null value, neither can be null the value in KEYPAIR.Pubkey_Path, if the TD find some artifact that fill all the conditions, the WF will start the creation of the KEY_PAIR.

In case of error during the execution, the workflow jump to the ROLLBACK category. If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, the attribute “number_of_retries” set the number of rollback attempts.

In this case, there is not a workflow designated for the Rollback process, so in case of error the TD will change the status of the artifact identified by the specific ID which it is been used during the execution of the Workflow.

The attribute “DATA.Lock” is set with a true value, so when the WF has finished its execution, the TLD will lock the artifact identified by the ID used in the execution of the workflow.

7.8 TLD VIM SPECS: GET FLAVOR.

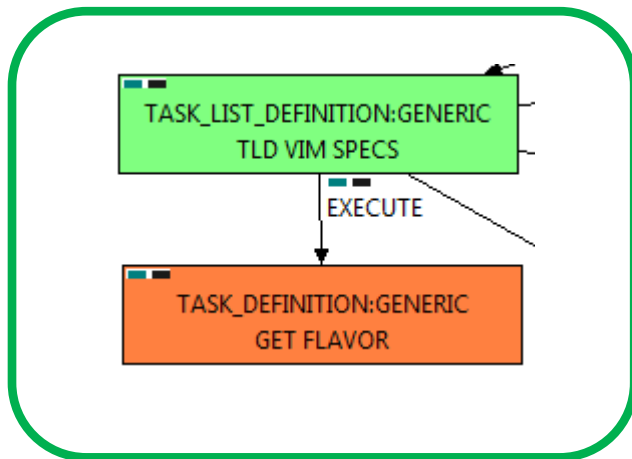


Figure: 45 Get Flavor.

This TD it is going to create the FLAVORS needed for each VMs to be activated later on, **this means, the WFs implied in this TLD are going to check each element of our VMs to gather all the information needed to create a specific FLAVOR artifact, during the execution of the TD, the ENTITY_SCALE Policies are going to be consulted, the situation of these policies are required for the correct creation of the FLAVOR.**

Once finished, we will have a number of FLAVORS bonded to a VM or VMs, prepare to be activated with these FLAVORS.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

```

GENERAL.Name ==                CREATE_FLAVOR
FIND.MainArtifact ==
VNF,VNF_COMPONENT<VNF
EXECUTE.Workflow ==
    "WF_NFVD_CREATE_FLAVOR_INSTANCES"
EXECUTE.Inactive==             false
ROLLBACK.Behaviour_on_error == ROLLBACK
ROLLBACK.Number_of_retries==   0
DATA.Lock ==                   true
  
```

The Workflow present in EXECUTE.Workflow it is going to seek a VNF with Running_Status INSTANTIATED in the DDBB, if the WF find some artifact that fill all the conditions, it will start.

This workflow will start another two more, the one that check if the FLAVOR needs Extra_Specs, a special set of configurations for the FLAVOR, "WF_NFVD_CREATE_FLAVOR_EXTRA_INVENTORY", and the one that will create the Flavor Instance in OPENSTACK platform, "WF_NFVD_CREATE_FLAVOR_OS_INSTANCE".

In case of error during the execution, the workflow jump to the ROLLBACK category, If the "Behaviour_on_error" attribute its set on "ROLLBACK" the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, the attribute "number_of_retries" set the number of rollback attempts. If an error take place in this TD , no action will be taken, the execution of the TLD will try to start a rollback workflow but there is not a workflow to be executed in the attribute ROLLBACK.Workflow, so the execution will continue without error.

Due to that the value of the attribute DATA.Lock is true, once the TD has finished, the artifact recently created, will be blocked.

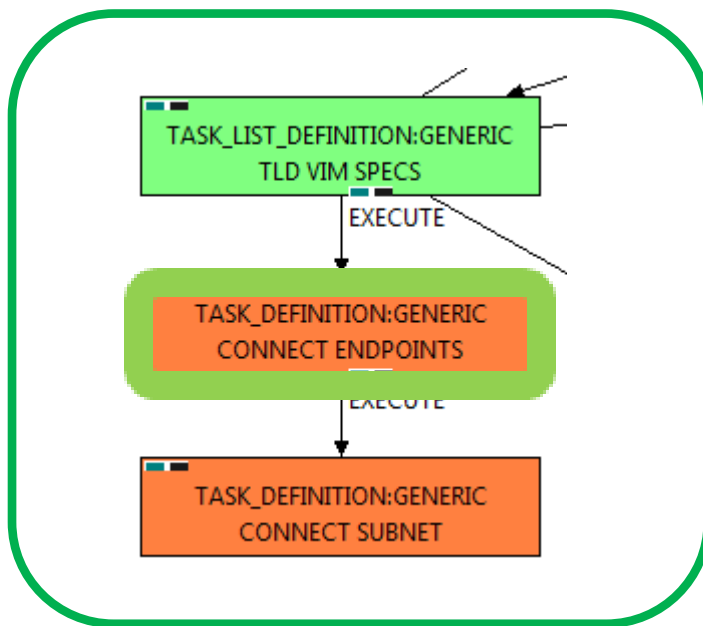


Figure: 46 Connection of the Endpoints.

This TD it is going to check and manage all the new EndPoints needed by the newly created Virtual Machine, during the Scale Out Process the Virtual Machine origin will be harvested to know how much End Points exist, and where they are connected. This TD is the responsible of this specific task.

Once finished, we will have a number of End-Points correctly set and connected to the adequate elements and artifacts.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

```

GENERAL.Name == Connect_Endpoint
FIND.MainArtifact ==
VNF,
VNF_COMPONENT<VNF
EXECUTE.Workflow ==
  "WF_NFVD_CONNECT_VNF_ENDPOINT"
EXECUTE.Inactive== false
ROLLBACK.Behaviour_on_error == ROLLBACK
ROLLBACK.Number_of_retries == 0
DATA.Lock == true
  
```

The Workflow present in EXECUTE.Workflow it is going to take the specific End Point origin in the artifact origin, if the WF find some artifact that fill all the conditions, it will start.

The workflow will check and create all the artifacts and relationship needed in order to make the new Virtual Machien accessible from the other components. Once finished, the Escalated Virtual Machine will have all the Eps of the old machine cloned, properly configurated in the new machine.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the "Behaviour_on_error" attribute its set on "ROLLBACK" the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, the attribute "number_of_retries" set the number of rollback attempts. If an error take place in this TD , no action will be taken, the execution of the TLD will try to start a rollback workflow but there is not a workflow to be executed in the attribute ROLLBACK.Workflow, so the execution will continue without error.

Due to that the value of the attribute DATA.Lock is true, once the TD has finished, the artifact recently created, will be blocked.

7.10 TLD VIM SPECS: CONNECT_SUBNET.

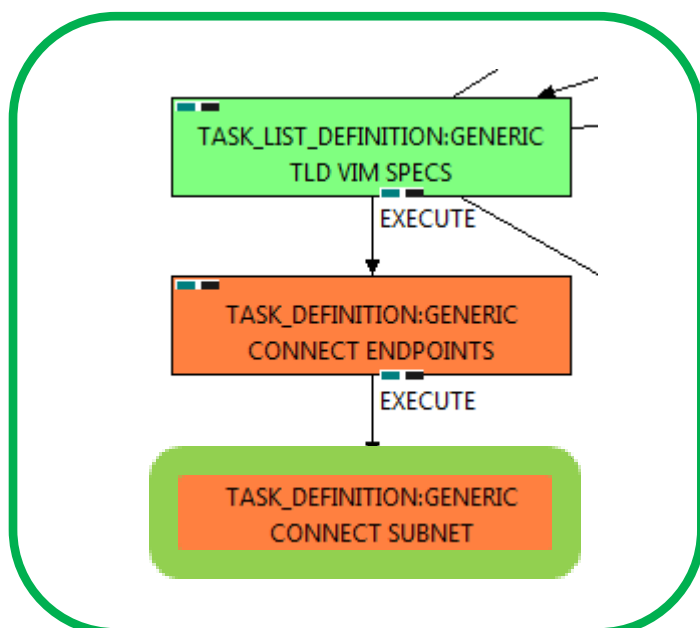


Figure: 47 Connection of the Subnetwork to the VPort

The TDs that have present in the their names “Connect”, are Task Definitions responsible of the connection between artifacts, this means, this TDs will create relationship of specific kind between concrete artifacts, in this case, the WF it is going to query the DDBB looking for all the VIRTUAL_LINK:END_POINTS with Status “TO_BE_CONNECTED”, once the WF has the VL:EP, it will query for all the SUBNETWORKS, NETWORKS and IPADDRESS of the VL, when the WF reach this point, it will query for the VPORTS related to these artifacts, after that, it is going to evaluate the relationships between the previously mentioned artifacts and the VPORTS, creating VPORTs and new relationships of type ALLOCATED and USES depending on the artifacts which are going to be related, mainly, SUBNETWORKs and VPORTs. The last thing this WF will do is change the status of the relationship between VL:EPs and the VNF:EP to CONNECTED.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

```

GENERAL.Name ==                CONNECT SUBNET
FIND.MainArtifact ==
VNF>VNF_ENDPOINT,
VNF_COMPONENT<VNF>VNF_ENDPOINT
SET.Running_Status ==          INSTANTIATED.
SET.Status ==                  INSTANTIATED.
EXECUTE.Workflow ==
                                “WF_TS_CONNECT_VM_SUBNET”
EXECUTE.Inactive==             false
ROLLBACK.Behaviour_on_error == ROLLBACK
ROLLBACK.Number_of_retries == 0
DATA.Lock ==                    true
  
```

The Workflow present in EXECUTE.Workflow it is going to seek a VNF_ENDPOINT with Running_Status INSTANTIATED in the DDBB, if the WF find some artifact that fill all the conditions, it will start.

In case of error during the execution, the workflow jump to the ROLLBACK category, if the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, the attribute “number_of_retries” set the number of rollback attempts. If an error take place in this TD, no action will be taken, the execution of the TLD will try to start a rollback workflow but there is not a workflow to be executed in the attribute ROLLBACK.Workflow, so the execution will continue without error.

Due to that the value of the attribute DATA.Lock is true, once the TD has finished, the artifact recently created, will be blocked.

7.11 TLD PREPROCESSING: PREPROCESS.

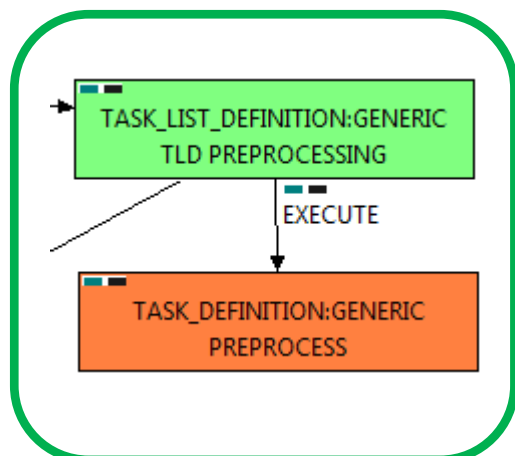


Figure: 48 Deploying Pre-Process policies.

This TD is responsible of the provision in the right order of the artifacts referenced by the PreProcessing policies, these policies allow the user to set a number of elements that should be taken in consideration in a certain order, in other case, the execution will fail depending on the event occurred.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

FIND.MainArtifact ==	POLICY:POSTPRE_PROCESSING
FIND.Condition ==	
PROCESSING_JOB.Job_type==constant:PRE&&	
PROCESSING_JOB.Operation==constant:SCALEOUT	
EXECUTE.OrderBy ==	PROCESSING_JOB.OrderBy
ROLLBACK.Behaviour_on_error ==	STOP
ROLLBACK.Number_of_retries ==	0
DATA.Lock ==	false

In this TD there is not a workflow to be executed, the target of this TD is process in the correct order the PreProcessing policies present in the VNF, these policies should be executed in a specific order to make the changes or configurations properly, in other case an error will be launched

If the TD ends successfully, the Pre-Processing policies will have been processed adequately.

In case of error during the execution, the TD will jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a “STOP” as value set for behavior, so no Rollback it is going to be initiated, the execution will stop.

Due to that the value of the attribute DATA.Lock is false, when the Task Definition has finished the artifact that was used in the workflow executed will remain unlocked.

The TDs that have present in the their names “Activate”, are Task Definitions responsible of the activation in the platform targeted and the updating of the status in the platform and the DDBB, in this case, the artifact that is going to be activated is a “KEYPAIR”, this means, when this workflow finish, we will have a KEYPAIR with status ACTIVE.

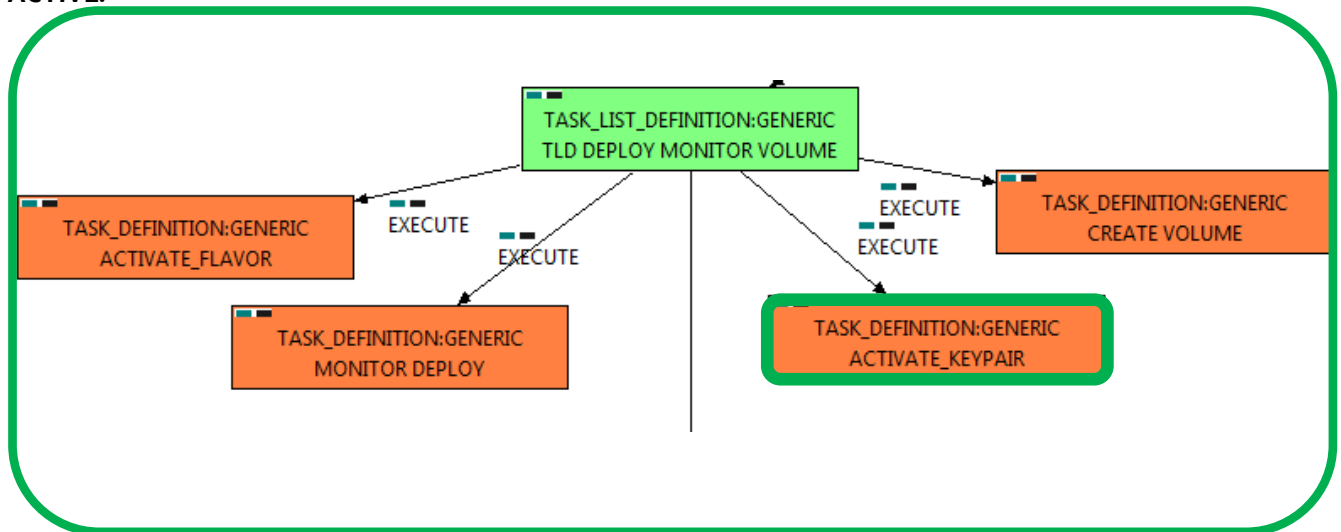


Figure: 49 Activation of the Keypair.

Targets of the TASK DEFINITION:

STATUS of the TD:

ENABLED

```

GENERAL.Name ==                                ACTIVATE_KEYPAIR
FIND.MainArtifact==
VNF>VNF_COMPONENT>VIRTUAL_MACHINE>VIRTUAL_CORE<CORE<CPU<SERVER<AVAILABILITY_ZONE<REGION>COMPUTE>KEY_PAIR,
VNF_COMPONENT>VIRTUAL_MACHINE>VIRTUAL_CORE<CORE<CPU<SERVER<AVAILABILITY_ZONE<REGION>COMPUTE>KEY_PAIR
FIND.Condition==                                status==constant:INSTANTIATED
SET.Running_Status ==                            INSTANTIATED.
SET.Status ==                                    ACTIVE.
EXECUTE.OrderBy ==                               GENERAL.order
EXECUTE.Workflow==                               “WF_TS_CREATE_KEY_PAIR”
EXECUTE.Inactive==                               false
ROLLBACK.Behaviour_on_error ==                   ROLLBACK
ROLLBACK.Number_of_retries ==                    0
DATA.Lock ==                                     true
  
```

The Workflow present in EXECUTE.Workflow attribute is going to seek a KEYPAIR that match the FIND.Condition attribute with value “KEYPAIR.Pubkey_Data==%GENERAL.Pubkey_Data%” also given by the path represented by the attribute FIND.Path :

“VNF>VNF_COMPONENT>VIRTUAL_MACHINE>VIRTUAL_CORE<CORE<CPU<SERVER<AVAILABILITY_ZONE<REGION>COMPUTE>KEY_PAIR@status=INSTANTIATED,VNF_COMPONENT>VIRTUAL_MACHINE>VIRTUAL_CORE<CORE<CPU<SERVER<AVAILABILITY_ZONE<REGION>COMPUTE>KEY_PAIR@status=INSTANTIATED” in Status INSTANTIATED in the DDBB, notice that we are not trying to get a VNF or VNF_COMPONENT in status INSTANTIATED.

Once found, the WF will start the activation, if the activation it is successful we set the status of the artifact as the SET.Status attribute dictates. The attribute SET.Running_Status concern about the temporal status that the artifact it is going to maintain until the final change of status that comes from SET.Status.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, in this case, we have a “ROLLBACK” set as behavior, so the rollback process will start when the TD reaches this point, it will throw an error due there is no workflow assigned to be executed during the rollback.

Due to that the value of the attribute DATA.Lock is true, when the Task Definition has finished the artifact that was used in the workflow executed will be set as “Locked”.

The TDs that have present in the their names “Activate”, are Task Definitions responsible of the activation in the platform targeted and the updating of the status in the platform and the DDBB, in this case, the artifact that is going to be activated is a “FLAVOR”, this means, when this workflow finish, we will have a FLAVOR with status ACTIVE associated to the VIRTUAL_MACHINE that it is going to use it in the activation.

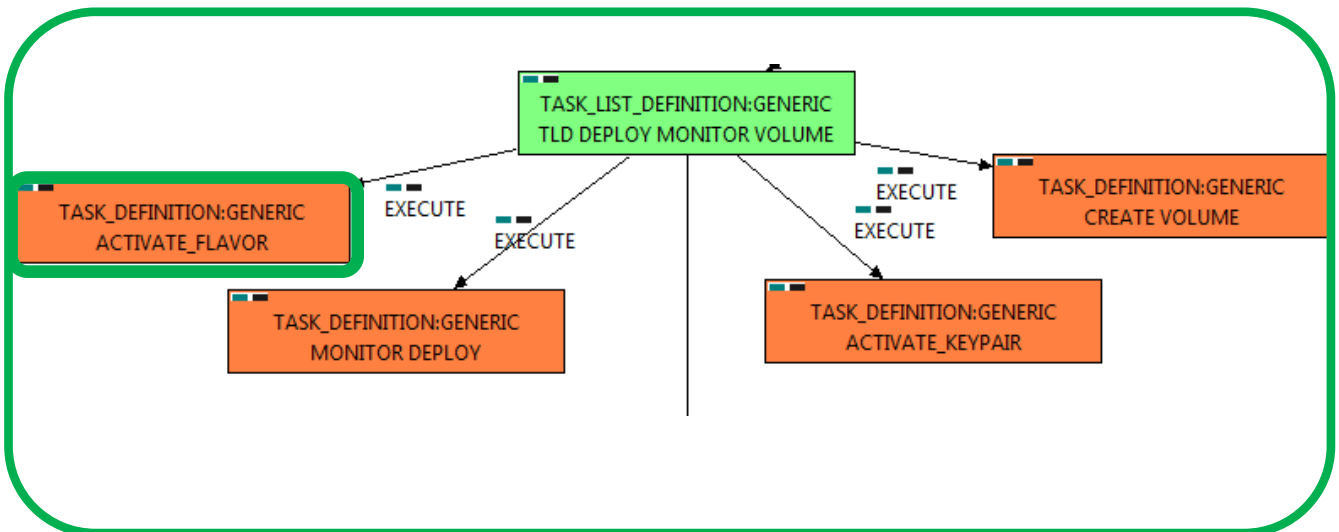


Figure: 50 Activation of the Flavor.

Targets of the TASK DEFINITION:

STATUS of the TD:

ENABLED

```

GENERAL.Name == ACTIVATE_FLAVOR
FIND.MainArtifact==
VNF>VNF_COMPONENT>VIRTUAL_MACHINE>
VIRTUAL_CORE<CORE<CPU<SERVER<AVAILABILITY_ZONE<REGION>COMPUTE>FLAVOR,
VNF_COMPONENT>VIRTUAL_MACHINE>
VIRTUAL_CORE<CORE<CPU<SERVER<AVAILABILITY_ZONE<REGION>COMPUTE>FLAVOR
SET.Running_Status == INSTANTIATED.
SET.Status == ACTIVE.
EXECUTE.Workflow == "WF_TS_ACTIVATE_FLAVOR"
EXECUTE.Inactive == false
ROLLBACK.Behaviour_on_error == ROLLBACK
ROLLBACK.Number_of_retries == 0
DATA.Lock == true
  
```

The Workflow present in EXECUTE.Workflow attribute it is going to seek a FLAVOR in Status INSTANTIATED in the DDBB. Notice that we are not trying to get a VIRTUAL_MACHINE in status INSTANTIATED. The query it is going to use the Path present in the category FIND. Once found , the WF will start the activation. Flavor activation is only possible if we have the admin role in openstack. If the activation it is successful we set the status of the artifact as the SET.Status attribute dictates. The attribute SET.Running_Status concern about the temporal status that the artifact it is going to maintain until the final change of status that comes from SET.Status.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, in this case, we have a “ROLLBACK” set as behavior, so the rollback process will start when the TD reaches this point, it will throw an error due there is no workflow assigned to be executed during the rollback.

Due to that the value of the attribute DATA.Lock is true, when the Task Definition has finished the artifact that was used in the workflow executed will be set as “Locked”.

The TDs that have present in the their names “Deploy” are Task Definitions responsible of the deployment in the platform targeted and the updating of the status in the platform and the DDBB , these deployments are slightly different to the ones we launch for our entities, as a rule, they are small components as the MONITORS. In this case, the artifact that is going to be deployed is a “MONITOR”, this means, when this workflow finish, we will have a MONITOR deployed with status DEPLOYED.

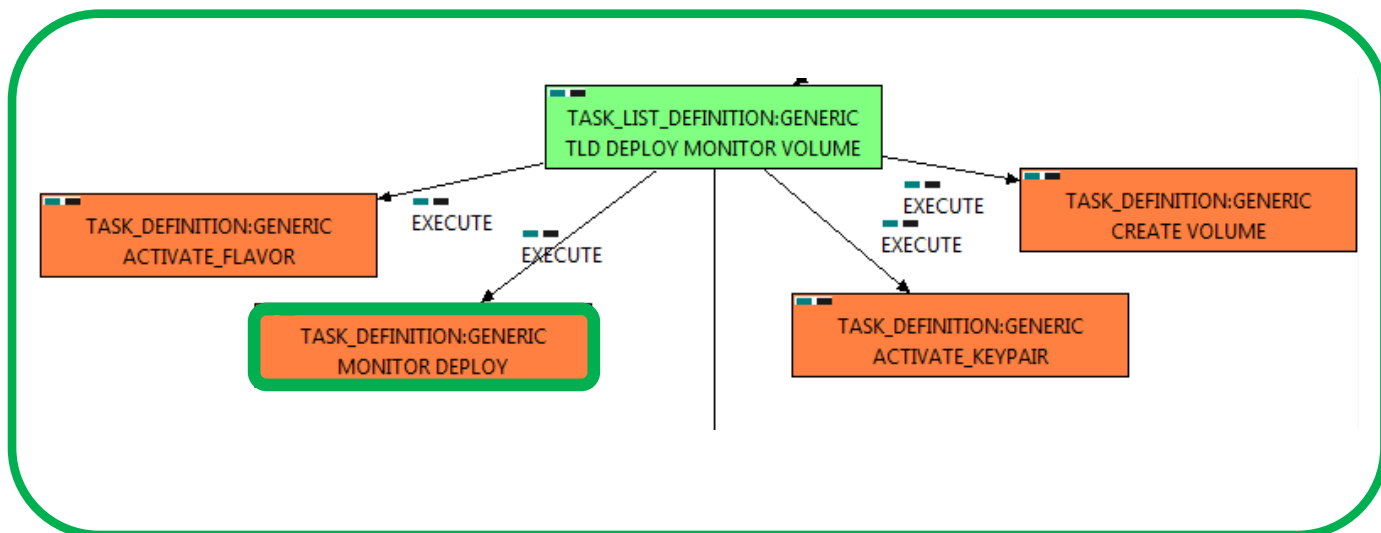


Figure: 51 Deployment of a Monitor.

Targets of the TASK DEFINITION:

STATUS of the TD: ENABLED

GENERAL.Name ==	Deploy Monitor
FIND.Condition==	status==constant:INSTANTIATED
SET.Running_Status ==	INSTANTIATED.
SET.Status ==	DEPLOYED.
EXECUTE.OrderBy ==	GENERAL.order
EXECUTE.Workflow==	“WF_TS_MONITOR_DEPLOY”
EXECUTE.Inactive==	false
ROLLBACK.Behaviour_on_error ==	ROLLBACK
ROLLBACK.Number_of_retries ==	0
ROLLBACK.Workflow ==	“WF_TS_MONITOR_UNDEPLOY”
DATA.Lock ==	true

The Workflow present in EXECUTE.Workflow attribute it is going to seek a MONITOR in Status INSTANTIATED in the DDBB. Once found, the WF will start the deployment, if the deployment is successful we set the status of the artifact as the SET.Status attribute dictates. The attribute SET.Running_Status concern about the temporal status that the artifact it is going to maintain until the final change of status that comes from SET.Status.

In case of error during the execution, the TD will jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, in this case the workflow will be “WF_TS_MONITOR_UNDEPLOY” the TD will initiate the rollback process launching the previous workflow.

Due to that the value of the attribute DATA.Lock is true, when the Task Definition has finished the artifact that was used in the workflow executed will be set as “Locked”.

7.15 TLD DEPLOY MONITOR VOLUME: CREATE VOLUME.

The TDs that have present in the their names “Activate”, are Task Definitions responsible of the activation in the platform targeted and the updating of the status in the platform and the DDBB, in this case, the artifact that is going to be activated is a “FLAVOR”, this means, when this workflow finish, we will have a FLAVOR with status ACTIVE associated to the VIRTUAL_MACHINE that it is going to use it in the activation.

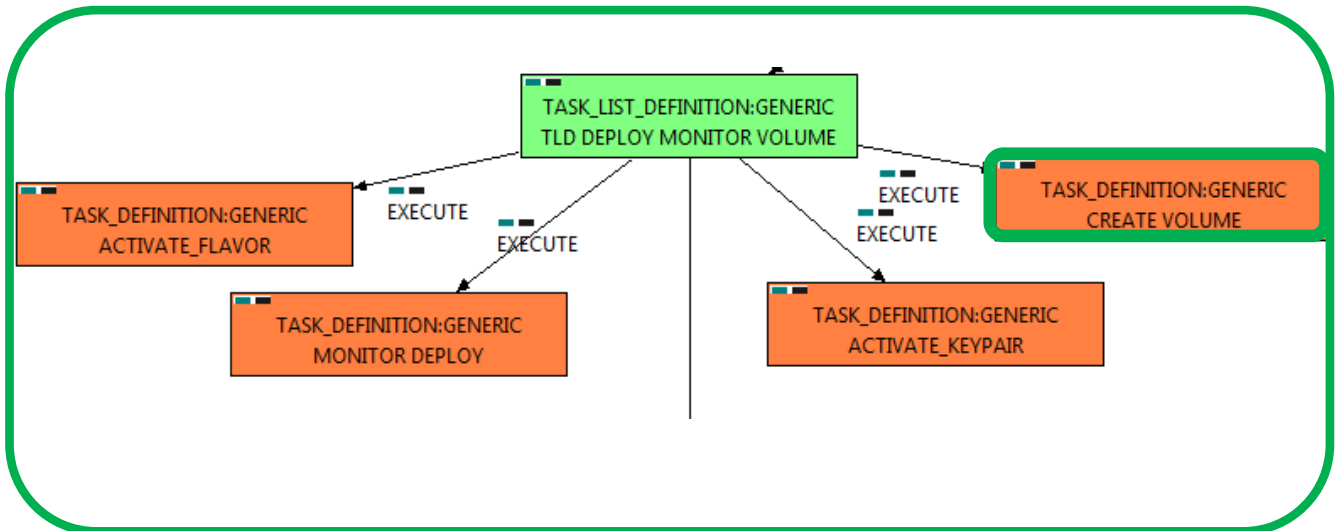


Figure: 52 Activation of the Volume.

Targets of the TASK DEFINITION:

STATUS of the TD:

ENABLED

```

GENERAL.Name ==                               Create Volume
FIND.MainArtifact==
VNF>VNF_COMPONENT>VIRTUAL_MACHINE>VIRTUAL_LUN@status=INSTANTIATED,
VNF_COMPONENT>VIRTUAL_MACHINE>VIRTUAL_LUN@status=INSTANTIATED
SET.Running_Status ==                         INSTANTIATED.
SET.Status ==                                 CREATED.
EXECUTE.Workflow ==                           “WF_TS_CREATE_VOLUME”
EXECUTE.Inactive ==                           false
ROLLBACK.Behaviour_on_error ==                ROLLBACK
ROLLBACK.Number_of_retries ==                 0
DATA.Lock ==                                  true
  
```

The Workflow present in EXECUTE.Workflow attribute it is going to seek a VIRTUAL_LUN in Status INSTANTIATED in the DDBB. Once found, the WF will start the activation, if the activation it is successful we set the status of the artifact as the SET.Status attribute dictates. The attribute SET.Running_Status concern about the temporal status that the artifact it is going to maintain until the final change of status that comes from SET.Status.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, in this case, we have a “ROLLBACK” set as behavior, so the rollback process will start when the TD reaches this point, it will throw an error due there is no workflow assigned to be executed during the rollback.

Due to that the value of the attribute DATA.Lock is true, when the Task Definition has finished the artifact that was used in the workflow executed will be set as “Locked”.

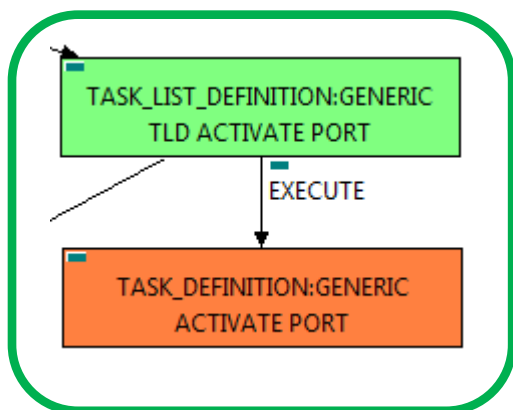


Figure: 53 Activation of a Virtual Port.

The TDs that have present in their names “Activate”, are Task Definitions responsible of the activation in the platform targeted and the updating of the status in the platform and the DDBB, in this case, the artifact that is going to be activated is a “VIRTUAL PORT”, this means, when this workflow finish, we will have a VIRTUAL PORT with status ACTIVE.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

```

GENERAL.Name ==                                Activate Port
FIND.MainArtifact==
VNF>VNF_COMPONENT>VIRTUAL_MACHINE>VIRTUAL_PORT@status=INSTANTIATED,
VNF_COMPONENT>VIRTUAL_MACHINE>VIRTUAL_PORT@status=INSTANTIATED
FIND.Condition == {[VIRTUAL_PORT<SUBNETWORK<NETWORK
<TENANT:OPENSTACK<VIM]
[GENERAL.Version==constant:MITAKA]}
SET.Status ==                                ACTIVE.
EXECUTE.Workflow ==
                                “WF_TS_ACTIVATE_VPORT”
EXECUTE.Inactive ==                            false
ROLLBACK.Behaviour_on_error ==                STOP
ROLLBACK.Number_of_retries ==                  0
DATA.Lock ==                                  true
  
```

The Workflow present in EXECUTE.Workflow attribute it is going to seek a VIRTUAL_PORT that match the FIND.Condition which means that the VIM which it is connected to is a MITAKA and the port itself is in Status INSTANTIATED in the DDBB, notice that we are not trying to get a VIRTUAL_MACHINE in status INSTANTIATED. The query it is going to use the Path present in the category FIND. Once found, the WF will start the activation, if the activation it is successful we set the status of the artifact as the SET.Status attribute dictates. The attribute SET.Running_Status concern about the temporal status that the artifact it is going to maintain until the final change of status that comes from SET.Status.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a “STOP” set as behavior, so no Rollback it is going to be initiated, so the execution it is going to end here in case of error.

Due to that the value of the attribute DATA.Lock is true, when the Task Definition has finished the artifact that was used in the workflow executed will be set as “Locked”.

7.18 TLD ATTACH: ATTACH VOLUME

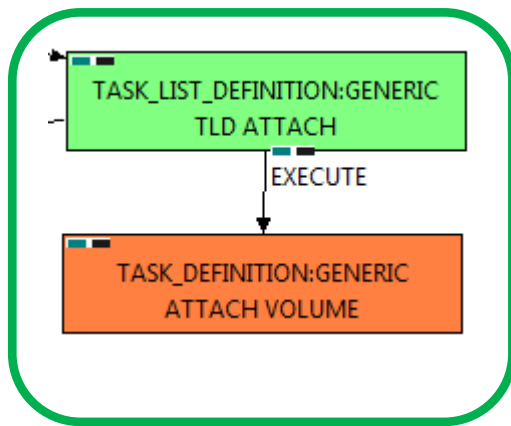


Figure: 55 Attaching of the Volume used.

The TDs that have present in the their names “Attach”, are Task Definitions responsible of the connection between artifacts, this means, this TDs will attach a VOLUME to a specific VIM, this specific VIM could change, so the workflow implied in this TD it will launch a custom WF for each kind of VIM. The VOLUMES that are activated by this TD have two final uses, directly linked with a VIM, or used as External Storage.

Once finished, we should have a number of VOLUMES activated, liable to a VIM or acting as external Storage.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

```

GENERAL.Name == ATTACH_VOLUME
FIND.MainArtifact==
VNF>VNF_COMPONENT>VIRTUAL_MACHINE>
VIRTUAL_LUN@status=CREATED,VNF_COMPONENT>
VIRTUAL_MACHINE>VIRTUAL_LUN@status=CREATED
SET.Running_Status == ACTIVE.
Set.Status == ACTIVE.
EXECUTE.Workflow ==
    "WF_TS_ATTACH_VOLUME"
EXECUTE.Inactive== false
ROLLBACK.Behaviour_on_error == ROLLBACK
ROLLBACK.Number_of_retries == 0
DATA.Lock == true
  
```

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, in this case, we have a “ROLLBACK” set as behavior, so the rollback process will start when the TD reaches this point, it will throw an error due there is no workflow assigned to be executed during the rollback.

Due to that the value of the attribute DATA.Lock is true, when the Task Definition has finished the artifact that was used in the workflow executed will be set as “Locked”.

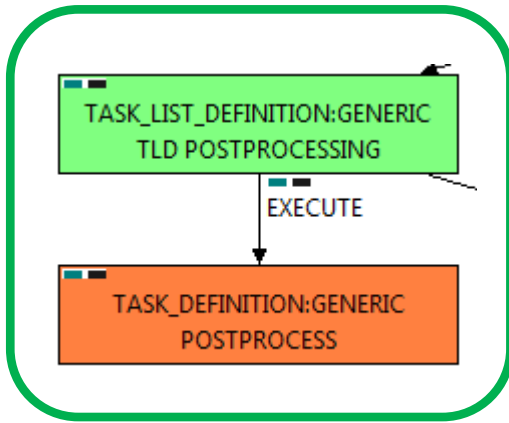


Figure: 56 Deploying Post-Processing policies.

This TD is responsible of the provision in the right order of the artifacts referenced by the Post-Processing policies, these policies allow the user to treat a number of elements that should be taken in consideration after the execution of some TD in a specific order, in other case, the execution will fail depending on the event occurred.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

```

FIND.MainArtifact == POLICY:POSTPRE_PROCESSING
FIND.Condition ==
PROCESSING_JOB.Job_type==constant:POST&&
PROCESSING_JOB.Operation==constant:SCALEOUT
EXECUTE.OrderBy == PROCESSING_JOB.OrderBy
ROLLBACK.Number_of_retries == 0
DATA.Lock == false
  
```

In this TD there is not a workflow to be executed, the target of this TD is process in the correct order the PostProcessing policies present in the VNF, these policies should be executed in a specific order to make the changes or configurations properly, in other case an error will be launched

If the TD ends successfully, the Post-Processing policies will have been applied adequately.

In case of error during the execution, the TD will jump to the ROLLBACK category, If the "Behaviour_on_error" attribute its set on "ROLLBACK" the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK.

Due to that the value of the attribute DATA.Lock is false, when the Task Definition has finished the artifact that was used in the workflow executed will remain unlocked.

7.20 • TLD START MONITOR: MONITOR START

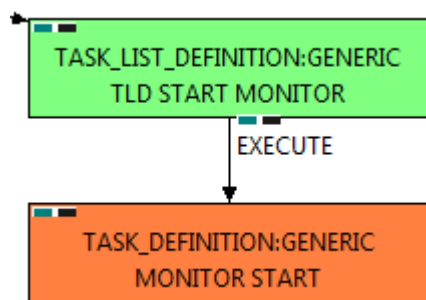


Figure: 57 Starting Monitor.

The TDs that have present in the their names “Start” are Task Definitions responsible of the launching of the component in the platform targeted and the updating of the status in the platform and the DDBB , these deployments are slightly different to the ones we launch for our entities, as a rule, they are small components as the MONITORS. In this case, the artifact that is going to be deployed is a “MONITOR”, this means, when this workflow finish, we will have a MONITOR deployed with status STARTED ready to monitories.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

```

GENERAL.Name == START_MONITOR
FIND.MainArtifact == MONITOR
FIND.Condition == status==constant:DEPLOYED
SET.Running_Status == DEPLOYED
SET.Status == STARTED.
EXECUTE.OrderBy == GENERAL.order
EXECUTE.Workflow == "WF_TS_MONITOR_START"
EXECUTE.Inactive== false
ROLLBACK.Behaviour_on_error == ROLLBACK
ROLLBACK.Number_of_retries == 0
DATA.Lock == true
  
```

The Workflow present in EXECUTE.Workflow attribute it is going to seek a MONITOR with Status DEPLOYED.

Once found, the WF will start the activation, if the activation it is successful we set the status of the artifact as the SET.Status attribute dictates. The attribute SET.Running_Status concern about the temporal status that the artifact it is going to maintain until the final change of status that comes from SET.Status.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, in this case, we have a “ROLLBACK” set as behavior, so the rollback process will start when the TD reaches this point, it will throw an error due there is no workflow assigned to be executed during the rollback.

Due to that the value of the attribute DATA.Lock is true, when the Task Definition has finished the artifact that was used in the workflow executed will be set as “Locked”.

Chapter 8 Start VM – Bottom-Up.

From now on, and to make easier the understanding of the TLDs, we are going to explain the functionality of each set of `TASK_LIST_DEFINITION:GENERIC`, and the number of `TASK_DEFINITION:GENERIC` children of the previously mentioned `TASK_LIST_DEFINITION:GENERIC`.

Basically, the `TASK_LIST_DEFINITION:GENERIC` connect what we can consider “units of execution”, those are the `TASK_DEFINITION:GENERIC`, that have a `WORKFLOW` assigned to be executed when the execution of the TLD reach them.

If you like to have a more deep knowledge about the workflows mentioned through this document please refer to the specific document.

If in the category `FIND`, the attribute `Path` is present, the attribute `FIND.ArtifactType` will be the starting artifact for the `Path`, but the `FIND.Status` attribute refers to the last artifact on the `Path`.

```
FIND.ArtifactType == VIRTUAL_MACHINE
FIND.Status==      INSTANTIATED
FIND.Path ==
    VIRTUAL_MACHINE>VIRTUAL_CORE<CORE<CPU<SERVER<AVAILABILITY_ZONE<REGION>
    COMPUTE>FLAVOR
```

In this example, we are looking for a `FLAVOR` in status `INSTANTIATED`, we do not expect to get a `VIRTUAL_MACHINE`, in status `INSTANTIATED`.

If during the use of the TLDs, the “Regenerate UUIDs” option is used, the user should check the `Id` of the tree that brings all the elements of the TLD, this “`id`” is specific and it will be the same for all the tree groups in all the TLDs.

The two modes available are “Default” and “Simulated”, the second one is only available if it is configured previously, by defect, and the mode that will be used is “Default”.

8.1 Specific Elements of the TLD Start VM.

In this chapter the different elements of the specific TLD will be explained conscientiously.

8.2 TLD START VM: START_VM

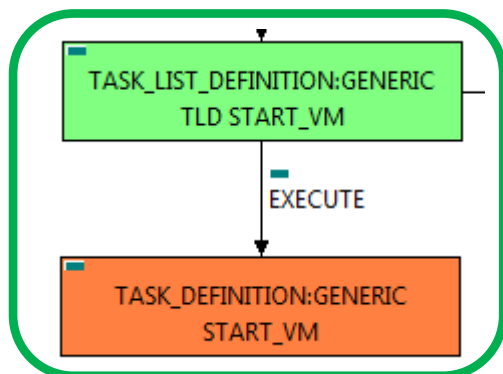


Figure: 58 Start Virtual Machine

The TDs that have present in their names “Start”, are Task Definitions responsible of the starting a specific artifact, in this case, start the virtual machine.

Once finished, the Virtual Machine will be available and with an ACTIVE status.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

```

GENERAL.Name == START_VM
FIND.Condition == status==constant:STOPPED
SET.Running_Status == STOPPED
SET.Status == ACTIVE
EXECUTE.Workflow ==
    "WF_TS_START_VM"
EXECUTE.Inactive == false
ROLLBACK.Behaviour_on_error == STOP
ROLLBACK.Number_of_retries == 0
DATA.Lock == true
INPUT_MAPPING.MAPPING_LIST ==
assignmentRelationshipID=Quota_Assignment;
resourceTreeID=nfvd#currentArtifactID
  
```

The Workflow present in EXECUTE.Workflow it is going to get the virtual machine passed in the DDBB, when the WF is found, it will start. This workflow is going to start the virtual machine and put it to an ACTIVE status.

In case of error during the execution, the workflow jump to the ROLLBACK category, if the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, the attribute “number_of_retries” set the number of rollback attempts. In this case, the TLD has not assigned a rollback workflow, so in this case the TD will only change the status of the artifact which is being used.

Due to that the value of the attribute DATA.Lock is true, once the TD has finished, the Quota recently assigned, will be blocked.

8.3 TLD START MONITOR: START MONITOR.

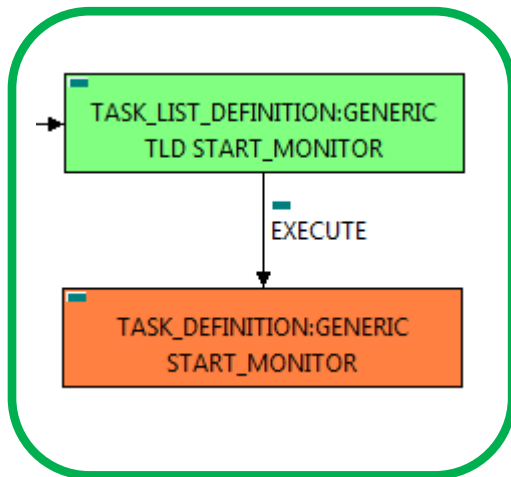


Figure: 59 Start Monitor

The TDs that have present in their names “Start”, are Task Definitions responsible of the starting a specific artifact, in this case, start the monitor.

Once finished, the Monitor will be available and with a STARTED status.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

FIND.MainArtifact ==	MONITOR
FIND.Condition ==	
status==constant:DEPLOYED	
EXECUTE.Workflow==	
“WF_TS_MONITOR_START”	
ROLLBACK.Behaviour_on_error==	STOP
ROLLBACK.Number_of_retries ==	0
DATA.Lock ==	false

The Workflow present in EXECUTE.Workflow it is going to get the monitor passed in the DDBB, when the WF is found, it will start. This workflow is going to start the monitor and put it to an STARTED status.

In case of error during the execution, the TD will jump to the ROLLBACK category, if the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK.

Due to that the value of the attribute DATA.Lock is false, when the Task Definition has finished the artifact that was used in the workflow executed will remain unlocked.

Chapter 9 Stop VM – Bottom-Up.

From now on, and to make easier the understanding of the TLDs, we are going to explain the functionality of each set of TASK_LIST_DEFINITION:GENERIC, and the number of TASK_DEFINITION:GENERIC children of the previously mentioned TASK_LIST_DEFINITION:GENERIC.

Basically, the TASK_LIST_DEFINITION:GENERIC connect what we can consider “units of execution”, those are the TASK_DEFINITION:GENERIC, that have a WORKFLOW assigned to be executed when the execution of the TLD reach them.

If you like to have a more deep knowledge about the workflows mentioned through this document please refer to the specific document.

If in the category FIND, the attribute Path is present, the attribute FIND.ArtifactType will be the starting artifact for the Path, but the FIND.Status attribute refers to the last artifact on the Path.

```
FIND.ArtifactType == VIRTUAL_MACHINE
FIND.Status==      INSTANTIATED
FIND.Path ==
    VIRTUAL_MACHINE>VIRTUAL_CORE<CORE<CPU<SERVER<AVAILABILITY_ZONE<REGION>
    COMPUTE>FLAVOR
```

In this example, we are looking for a FLAVOR in status INSTANTIATED, we do not expect to get a VIRTUAL_MACHINE, in status INSTANTIATED.

If during the use of the TLDs, the “Regenerate UUIDs” option is used, the user should check the Id of the tree that brings all the elements of the TLD, this “id” is specific and it will be the same for all the tree groups in all the TLDs.

The two modes available are “Default” and “Simulated”, the second one is only available if it is configured previously, by defect, and the mode that will be used is “Default”.

9.1 Specific Elements of the TLD Stop VM.

In this chapter the different elements of the specific TLD will be explained conscientiously.

9.2 TLD STOP MONITOR: STOP MONITOR.

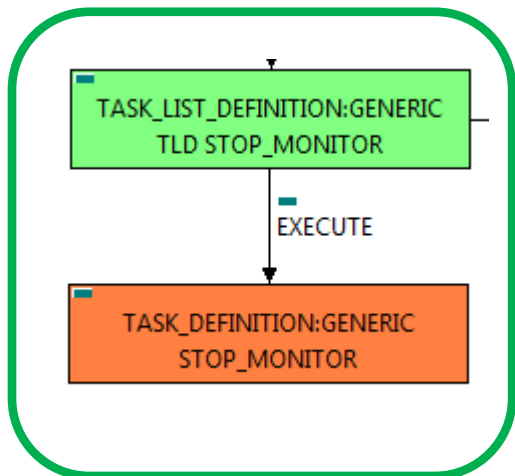


Figure: 60 Stop Monitor

The TDs that have present in their names “Stop”, are Task Definitions responsible of stopping a specific artifact, in this case, stop the monitor.

Once finished, the Monitor will be stopped and with a DEPLOYED status.

Targets of the TASK DEFINITION:
ENABLED

STATUS of the TD:

FIND.MainArtifact ==	MONITOR
FIND.Condition ==	
status==constant:STARTED	
EXECUTE.Workflow==	
“WF_TS_MONITOR_STOP”	
ROLLBACK.Behaviour_on_error==	STOP
ROLLBACK.Number_of_retries ==	0
DATA.Lock ==	false

The Workflow present in EXECUTE.Workflow it is going to get the monitor passed in the DDBB, when the WF is found, it will start. This workflow is going to stop the monitor and put it to a DEPLOYED status.

In case of error during the execution, the TD will jump to the ROLLBACK category, if the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK.

Due to that the value of the attribute DATA.Lock is false, when the Task Definition has finished the artifact that was used in the workflow executed will remain unlocked.

Chapter 10 Undeploy of an Organization – Bottom-Up.

From now on, and to make easier the understanding of the TLDs, we are going to explain the functionality of each set of TASK_LIST_DEFINITION:GENERIC, and the number of TASK_DEFINITION:GENERIC children of the previously mentioned TASK_LIST_DEFINITION:GENERIC.

Basically, the TASK_LIST_DEFINITION:GENERIC connect what we can consider “units of execution”, those are the TASK_DEFINITION:GENERIC, that have a WORKFLOW assigned to be executed when the execution of the TLD reach them.

If you like to have a more deep knowledge about the workflows mentioned through this document please refer to the specific document.

If in the category FIND, the attribute Path is present, the attribute FIND.ArtifactType will be the starting artifact for the Path, but the FIND.Status attribute refers to the last artifact on the Path.

FIND.ArtifactType == VIRTUAL_MACHINE.

FIND.Status== INSTANTIATED.

FIND.Path==

VIRTUAL_MACHINE>VIRTUAL_CORE<CORE<CPU<SERVER<AVAILABILITY_ZONE<REGION>COMPUTE>FLAVOR

In this example, we are looking for a FLAVOR in status INSTANTIATED, we do not expect to get a VIRTUAL_MACHINE, in status INSTANTIATED.

If during the use of the TLDs, the “Regenerate UUIDs” option is used, the user should check the Id of the tree that brings all the elements of the TLD, this “id” is specific and it will be the same for all the tree groups in all the TLDs.

The two modes available are “Default” and “Simulated”, the second one is only available if it is configured previously, by defect, the mode that will be used is “Default”.

10.1 Specific Elements of the TLD Undeploy Organization.

In this chapter the different elements of the specific TLD will be explained conscientiously.

10.2 TLD UNDEPLOY ORGANIZATION: Undeploy Check

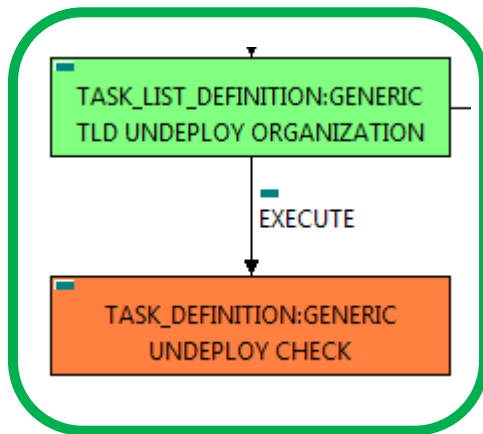


Figure: 62Undeploy check.

This TD it is going to assure the scenario in order to delete a specific Organization, this means that during the execution the TD is going to check if all the children of the Organization are still or were properly deleted before launch the undeploy of an Organization.

Targets of the TASK DEFINITION:

STATUS of the TD: ENABLED

Categories:

GENERAL.Name ==	Undeploy check
FIND.Condition ==	status==constant:ACTIVE
SET.Status ==	INSTANTIATED
EXECUTE.Workflow ==	“WF_TS_UNDEPLOY_CHECK_CHILDREN”
EXECUTE.Inactive==	false
ROLLBACK.Behaviour_on_error ==	ROLLBACK
ROLLBACK.Number_of_retries ==	0
DATA.Lock ==	true

The Workflow present in EXECUTE.Workflow attribute it is going to seek for the children entities of the Organization, in case the TD find some the execution of the TD will fail, the goal of this TD is to guarantee that the Organization has no children and also is in the proper conditions to be set as an entity with status INSTANTIATED.

Once found, the TD would execute the WF present in EXECUTE.Workflow, in this case, the Wf is “WF_TS_UNDEPLOY_CHECK_CHILDREN“, the workflow will develop the task previously explained.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a “STOP” set as behavior, so no Rollback it is going to be initiated, so the execution it is going to end here in case of error.

The attribute DATA.Lock is set with the value “true”, this means once the TD ends its execution the element which is being used by the TD will be locked.

10.3 TLD DELETE TREE: Delete Organization Tree.

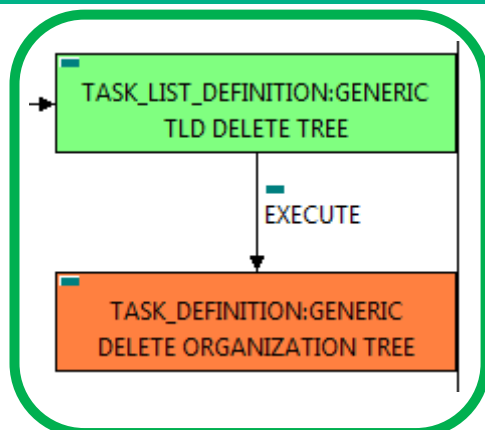


Figure: 63 Organization Delete.

The TDs that have present in their names “Delete”, are Task Definitions responsible of the deletion of the artifact given, in this case, this TD it is going to delete an ORGANIZATION, notice the workflow used in this TD, “WF_TS_DELETE_INSTANCE_TREE”, all the components and elements below the entity that it is going to be deleted, are going to be eliminated as well, in other case, this elements will remain unreachable, that is not desirable.

Targets of the TASK:DEFINITION:

STATUS of the TD: ENABLED

Categories:

```

EXECUTE.Workflow ==
    "WF_TS_DELETE_INSTANCE_TREE"
ROLLBACK.Behaviour_on_error ==      STOP
ROLLBACK.Number_of_retries ==      0
DATA.Lock ==                        true
  
```

The Workflow present in EXECUTE.Workflow attribute it is going to seek an ORGANIZATION in the DDBB. Once found, the WF will start the deleting.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a “STOP” set as behavior, so no Rollback it is going to be initiated, so the execution it is going to end here in case of error.

Chapter 11 Undeploy of a Tenant – Bottom-Up.

From now on, and to make easier the understanding of the TLDs, we are going to explain the functionality of each set of TASK_LIST_DEFINITION:GENERIC, and the number of TASK_DEFINITION:GENERIC children of the previously mentioned TASK_LIST_DEFINITION:GENERIC.

Basically, the TASK_LIST_DEFINITION:GENERIC connect what we can consider “units of execution”, those are the TASK_DEFINITION:GENERIC, that have a WORKFLOW assigned to be executed when the execution of the TLD reach them.

If you like to have a more deep knowledge about the workflows mentioned through this document please refer to the specific document.

If in the category FIND, the attribute Path is present, the attribute FIND.ArtifactType will be the starting artifact for the Path, but the FIND.Status attribute refers to the last artifact on the Path.

FIND.ArtifactType == VIRTUAL_MACHINE.

FIND.Status== INSTANTIATED.

FIND.Path==

VIRTUAL_MACHINE>VIRTUAL_CORE<CORE<CPU<SERVER<AVAILABILITY_ZONE<REGION>COMPUTE>FLAVOR

In this example, we are looking for a FLAVOR in status INSTANTIATED, we do not expect to get a VIRTUAL_MACHINE, in status INSTANTIATED.

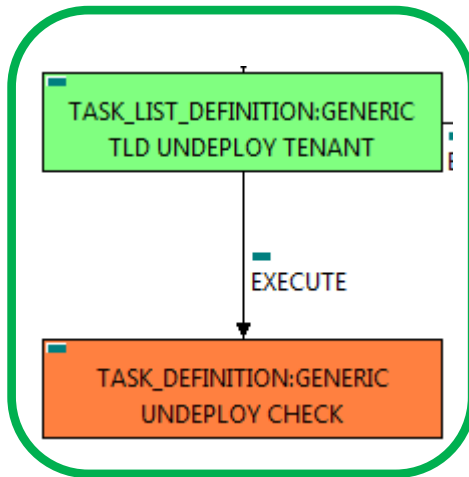
If during the use of the TLDs, the “Regenerate UUIDs” option is used, the user should check the Id of the tree that brings all the elements of the TLD, this “id” is specific and it will be the same for all the tree groups in all the TLDs.

The two modes available are “Default” and “Simulated”, the second one is only available if it is configured previously, by defect, the mode that will be used is “Default”.

11.1 Specific Elements of the TLD Undeploy Tenant.

In this chapter the different elements of the specific TLD will be explained conscientiously.

11.2 TLD UNDEPLOY TENANT: Undeploy Check.



This TD it is going to assure the scenario in order to delete a specific Tenant, this means that during the execution the TD is going to check if all the children of the Tenant were properly deleted before launch the undeploy of the Tenant.

Targets of the TASK DEFINITION:
STATUS of the TD: ENABLED
Categories:

```

FIND.Condition ==       status==constant:ACTIVE
EXECUTE.Workflow==       "WF_TS_UNDEPLOY_CHECK_CHILDREN"
ROLLBACK.Behaviour_on_error ==       STOP
ROLLBACK.Numbre_of_retries ==       0
DATA.Lock ==               true
  
```

Figure: 64 Undeploy Check.

The Workflow present in EXECUTE.Workflow attribute it is going to seek for the children entities of the Tenant, in case the TD find some the execution of the TD will fail, the goal of this TD is to guarantee that the Tenant has no children and also is in the proper conditions to be set as an entity with status INSTANTIATED.

Once found, the TD would execute the WF present in EXECUTE.Workflow, in this case, the Workflow is "WF_TS_UNDEPLOY_CHECK_CHILDREN", the workflow will develop the task previously explained.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the "Behaviour_on_error" attribute its set on "ROLLBACK" the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a "STOP" set as behavior, so no Rollback it is going to be initiated, so the execution it is going to end here in case of error.

The attribute DATA.Lock is set with the value "false".

11.3 TLD DELETE IDS: DELETE VDC ID OF OS TENANTS.

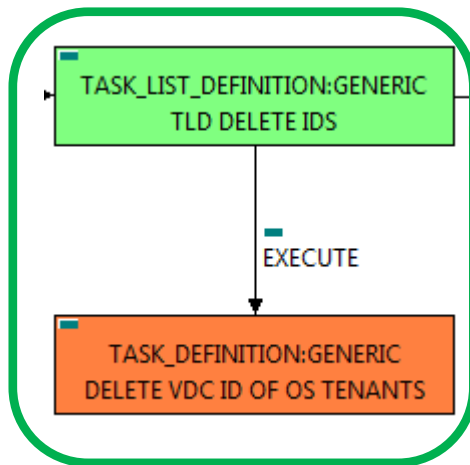


Figure: 65 Delete ID of TENANT OS

This TD it is going to unrelate our TENANT:OPENSTACK from the current VDC, **this means, the WF implied in this TLD is going to find a TENANT:OPENSTACK in status ACTIVE that fills the conditions present in the TD.**

Once finished, we will have an EGRESSACLENTY POLICY deactivated with all the relationship needed for the correct behavior of the artifact still present, prepare to be deleted when required.

Targets of the TASK DEFINITION: STATUS of the TD: ENABLED

Categories:

FIND.Path==

TENANT:GENERIC>RESOURCE_POOL>VIM>TENANT:OPENSTACK@status=ACTIVE,TENANT:GENERIC>RESOURCE_POOL>LOCATION>VIM>TENANT:OPENSTACK@status=ACTIVE,TENANT:GENERIC>RESOURCE_POOL>DATACENTER>VIM>TENANT:OPENSTACK@status=ACTIVE,TENANT:GENERIC>RESOURCE_POOL>SERVER<HYPERVISOR<VIM>TENANT:OPENSTACK@status=ACTIVE

FIND.Condition == GENERAL_VDC_id==%Id%

EXECUTE.Workflow==

"WF_TS_DELETE_TENANT_OS_VDC_ID"

ROLLBACK.Behaviour_on_error ==

STOP

ROLLBACK.Numbre_of_retries ==

0

DATA.Lock ==

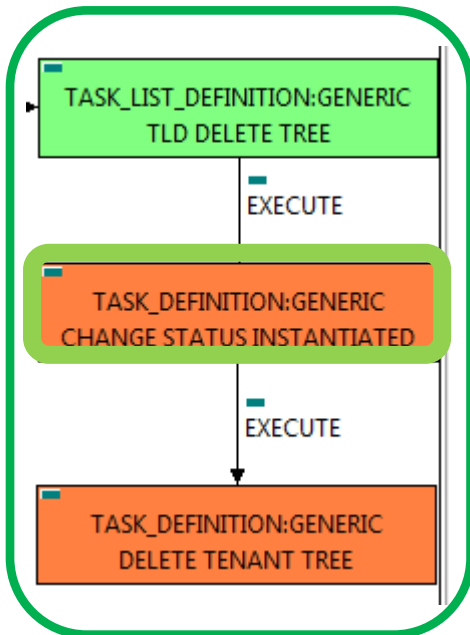
true

The Workflow present in EXECUTE.Workflow attribute it is going to take the TENANT:OPENSTACK given and clear the field GENERAL.VDC_id.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the "Behaviour_on_error" attribute its set on "ROLLBACK" the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a "STOP" set as behavior, so no Rollback it is going to be initiated, so the execution it is going to end here in case of error.

The attribute DATA.Lock is set with the value "true", this means once the TD ends its execution the element which is being used by the TD will be locked.

11.4 TLD DELETE TREE: Change Status Instantiated.



This TD it is going to change the status of VDC to INSTANTIATED to delete it in the next task. .

Targets of the TASK DEFINITION:

STATUS of the TD: ENABLED

Categories:

SET.Status == INSTANTIATED

ROLLBACK.Behaviour_on_error == STOP

ROLLBACK.Numbre_of_retries == 0

DATA.Lock== true

Figure: 66 Deactivating Egress policies for the Tenant.

The TASK_DEFINITION do not execute any workflow, with the attributes present in the categories it is enough to change the status of the entity.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a “STOP” set as behavior, so no Rollback it is going to be initiated, so the execution it is going to end here in case of error.

The attribute DATA.Lock is set with the value “true”, this means once the TD ends its execution the element which is being used by the TD will be locked.

11.5 TLD DELETE TREE: Delete Tenant Tree.

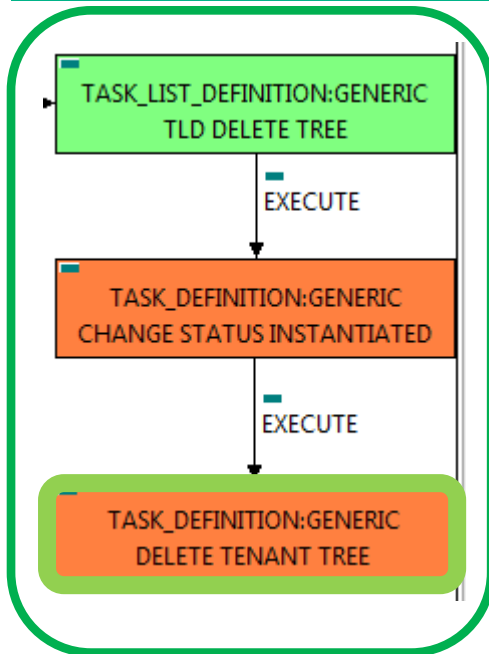


Figure: 67 Delete Tenant.

The TDs that have present in their names “Delete Inventory”, are Task Definitions responsible of the deletion of the artifact given, in this case, this TD it is going to delete a TENANT:GENERIC, notice the workflow used in this TD, “WF_TS_DELETE_INSTANCE_TREE”, all the components and elements below the entity that it is going to be deleted, are going to be eliminated as well, in other case, this elements will remain unreachable, that is not desirable.

Targets of the TASK:DEFINITION:
STATUS of the TD: ENABLED

Categories:

```

EXECUTE.Workflow ==
    "WF_TS_DELETE_INSTANCE_TREE"
ROLLBACK.Behaviour_on_error == STOP
ROLLBACK.Number_of_retries == 0
  
```

The Workflow present in EXECUTE.Workflow attribute it is going to seek a TENANT:GENERIC in the DDBB . Once found, the WF will start the deleting.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a “STOP” set as behavior, so no Rollback it is going to be initiated, so the execution it is going to end here in case of error.

Chapter 12 Undeploy of a Virtual Link – Bottom-Up.

From now on, and to make easier the understanding of the TLDs, we are going to explain the functionality of each set of TASK_LIST_DEFINITION:GENERIC, and the number of TASK_DEFINITION:GENERIC children of the previously mentioned TASK_LIST_DEFINITION:GENERIC.

Basically, the TASK_LIST_DEFINITION:GENERIC connect what we can consider “units of execution”, those are the TASK_DEFINITION:GENERIC, that have a WORKFLOW assigned to be executed when the execution of the TLD reach them.

If you like to have a more deep knowledge about the workflows mentioned through this document please refer to the specific document.

If in the category FIND, the attribute Path is present, the attribute FIND.ArtifactType will be the starting artifact for the Path, but the FIND.Status attribute refers to the last artifact on the Path.

FIND.ArtifactType == VIRTUAL_MACHINE.

FIND.Status== INSTANTIATED.

FIND.Path==

VIRTUAL_MACHINE>VIRTUAL_CORE<CORE<CPU<SERVER<AVAILABILITY_ZONE<REGION>COMPUTE>FLAVOR

In this example, we are looking for a FLAVOR in status INSTANTIATED, we do not expect to get a VIRTUAL_MACHINE, in status INSTANTIATED.

If during the use of the TLDs, the “Regenerate UUIDs” option is used, the user should check the Id of the tree that brings all the elements of the TLD, this “id” is specific and it will be the same for all the tree groups in all the TLDs.

The two modes available are “Default” and “Simulated”, the second one is only available if it is configured previously, by defect, the mode that will be used is “Default”.

12.1 Specific Elements of the TLD Undeploy Virtual Link.

In this chapter the different elements of the specific TLD will be explained conscientiously.

12.2 TLD UNDEPLOY VIRTUAL LINK : UNDEPLOY CHECK.

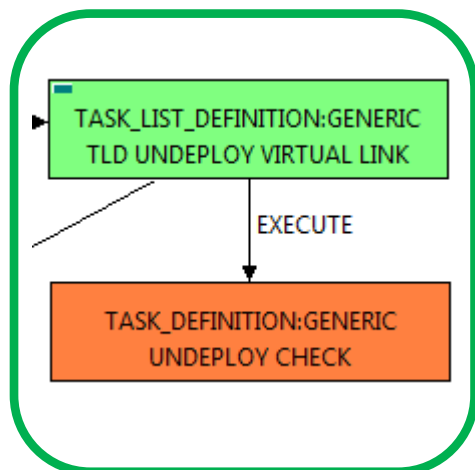


Figure: 68 Undeploy Check.

This TD it is going to assure the scenario in order to delete a specific Organization, this means that during the execution the TD is going to check if all the children of the Organization are still or were properly deleted before launch the undeploy of an Organization.

Targets of the TASK DEFINITION:
STATUS of the TD: ENABLED

Categories:

GENERAL.Name ==	Undeploy check
FIND.Condition ==	status==constant:ACTIVE
EXECUTE.Workflow ==	"WF_TS_UNDEPLOY_CHECK_CHILDREN"
EXECUTE.Inactive==	false
ROLLBACK.Behaviour_on_error ==	ROLLBACK
ROLLBACK.Number_of_retries ==	0
DATA.Lock ==	true

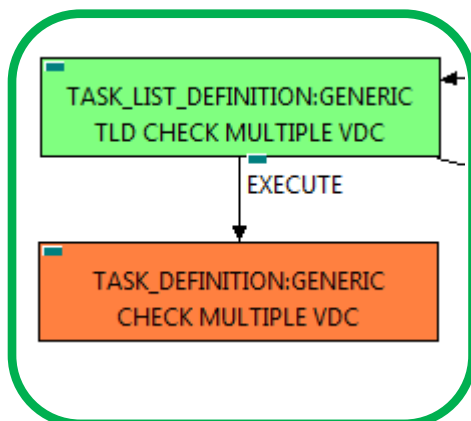
The Workflow present in EXECUTE.Workflow attribute it is going to seek for the children entities of the Virtual Link, in case the TD find some the execution of the TD will fail, the goal of this TD is to guarantee that the Virtual Link has no children and also is in the proper conditions to be set as an entity with status INSTANTIATED.

Once found, the TD would execute the WF present in EXECUTE.Workflow, in this case, the Wf is "WF_TS_UNDEPLOY_CHECK_CHILDREN", the workflow will develop the task previously explained.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the "Behaviour_on_error" attribute its set on "ROLLBACK" the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a "STOP" set as behavior, so no Rollback it is going to be initiated, so the execution it is going to end here in case of error.

The attribute DATA.Lock is set with the value "true", this means once the TD ends its execution the element which is being used by the TD will be locked.

12.3 TLD CHECK MULTIPLE VDC : CHECK MULTIPLE VDC.



This TD it is going to check if this Virtual Link is owned by more than one VDC. If it is true, it will detach it from the current VDC.

Once finished, we will have an `INGRESSACLENTRY POLICY` deactivated with all the relationship needed for the correct behavior of the artifact still present, prepare to be deleted when required.

Targets of the TASK DEFINITION:
 STATUS of the TD: ENABLED
 Categories:

```

GENERAL.Name==      Check Multiple VDC
EXECUTE.Workflow==
    "WF_TS_CHECK_MULTIPLE_VDC_VIRTUAL_LINK"
ROLLBACK.Behaviour_on_error ==      STOP
ROLLBACK.Numbre_of_retries ==      0
DATA.Lock ==          true
  
```

Figure: 69 Check Multiple VDC.

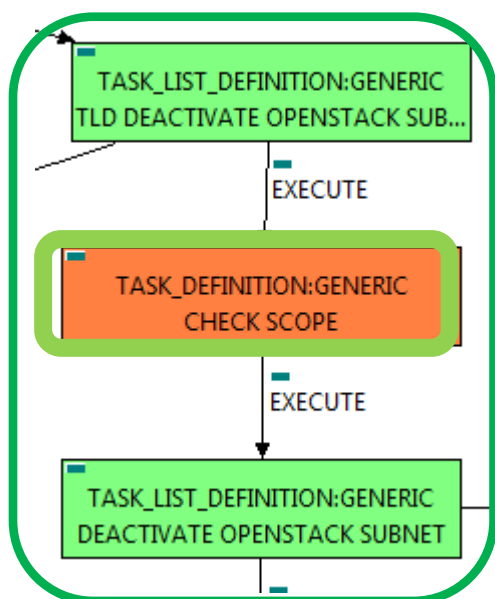
The Workflow present in `EXECUTE.Workflow` attribute it is going to get the Virtual Link passed to the job.

Once found, the WF will start the check, if the check is successful, whether it will detach or not the Virtual Link, we can continue.

In case of error during the execution, the workflow jump to the `ROLLBACK` category, If the "`Behaviour_on_error`" attribute its set on "`ROLLBACK`" the WF will start the execution of the Workflow present in the attribute with the same name in the category `ROLLBACK`, but in this case, we have a "`STOP`" set as behavior, so no Rollback it is going to be initiated, so the execution it is going to end here in case of error.

The attribute `DATA.Lock` is set with the value "`true`", this means once the TD ends its execution the element which is being used by the TD will be locked.

12.4 TLD DEACTIVATE OPENSTACK SUBNET: CHECK SCOPE.



This TD it is going to check the Scope of the Virtual Link. If it is VDC, this task will be generated. If it is not, it will not be generated at all.

Targets of the TASK DEFINITION:

STATUS of the TD: ENABLED

Categories:

```

FIND.Condition ==
GENERAL.Scope == constant:
ROLLBACK.Behaviour_on_error == STOP
ROLLBACK.Numbre_of_retries == 0
DATA.Lock == true
  
```

Figure: 70 Check Scope.

This Task does not have a workflow. It will only check the condition, hence it will be generated if it matches the condition.

Once found, the creation of next tasks will continue.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a “STOP” set as behavior, so no Rollback it is going to be initiated, so the execution it is going to end here in case of error.

The attribute DATA.Lock is set with the value “true”, this means once the TD ends its execution the element which is being used by the TD will be locked.

12.5 TLD DEACTIVATE OPENSTACK SUBNET: DEACTIVATE_SUBNETWORK_OPENSTACK.

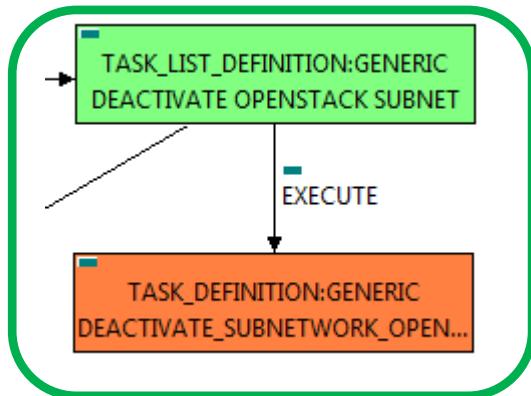


Figure: 71 Deactivate Subnetwork Openstack.

The TDs that have present in the their names “Deactivate”, are Task Definitions responsible of the deactivation in the platform targeted and the updating of the status in the platform and the DDBB, in this case, the artifact that is going to be deactivated is a “SUBNETWORK:OPENSTACK”, this means, when this workflow finish, we will have a SUBNETWORK:OPENSTACK with status INSTANTIATED, still present in the DDBB..

Targets of the TASK:DEFINITION:
STATUS of the TD: ENABLED
Categories:

```

FIND.MainArtifact==
VIRTUAL_LINK>NETWORK:GENERIC>
NETWORK:OPENSTACK>
SUBNETWORK:OPENSTACK@status=ACTIVE
SET.Running_Status == ACTIVE.
SET.Status ==                   INSTANTIATED.
EXECUTE.Workflow ==
    "WF_TS_DEACTIVATE_SUBNETWORK"
ROLLBACK.Behaviour_on_error ==    STOP
ROLLBACK.Number_of_retries ==    0
  
```

The Workflow present in EXECUTE.Workflow attribute it is going to seek a “SUBNETWORK:OPENSTACK” policy with Status ACTIVE, reachable by the Path given,

```

"VIRTUAL_LINK>NETWORK:GENERIC>NETWORK:OPENSTACK>SUBNETWORK:OPENSTACK@status=ACTIVE "
  
```

Once found, the WF will start the deactivating, if deactivation is successful we set the status of the artifact as the SET.Status attribute dictates. The attribute SET.Running_Status concern about the temporal status that the artifact it is going to maintain until the final change of status that comes from SET.Status.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a “STOP” set as behavior, so no Rollback it is going to be initiated, so the execution it is going to end here in case of error.

12.6 TLD DEACTIVATE OPENSTACK NET: DEACTIVATE_NETWORK_OPENSTACK

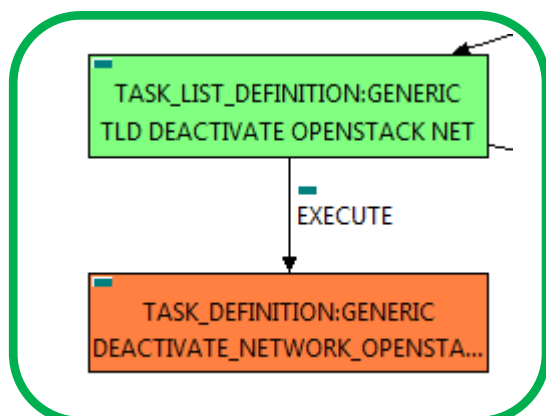


Figure: 72 Deactivate Network Openstack

The TDs that have present in the their names “Deactivate”, are Task Definitions responsible of the deactivation in the platform targeted and the updating of the status in the platform and the DDBB, in this case, the artifact that is going to be deactivated is a “NETWORK:OPENSTACK”, this means, when this workflow finish, we will have a NETWORK:OPENSTACK with status INSTANTIATED, still present in the DDBB.

Targets of the TASK:DEFINITION:
STATUS of the TD: ENABLED
Categories:

```

FIND.MainArtifact==
VIRTUAL_LINK>NETWORK:GENERIC>
NETWORK:OPENSTACK@status=ACTIVE
SET.Running_Status == ACTIVE.
SET.Status == INSTANTIATED.
EXECUTE.Workflow ==
    "WF_TS_DEACTIVATE_NETWORK"
ROLLBACK.Behaviour_on_error == STOP
ROLLBACK.Number_of_retries == 0
  
```

The Workflow present in EXECUTE.Workflow attribute it is going to seek a “NETWORK:OPENSTACK” policy with Status ACTIVE, reachable by the Path given,

“VIRTUAL_LINK>NETWORK:GENERIC>NETWORK:OPENSTACK@status=ACTIVE “

Once found, the WF will start the deactivating, if deactivation is successful we set the status of the artifact as the SET.Status attribute dictates. The attribute SET.Running_Status concern about the temporal status that the artifact it is going to maintain until the final change of status that comes from SET.Status.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a “STOP” set as behavior, so no Rollback it is going to be initiated, so the execution it is going to end here in case of error.

12.7 TLD INVENTORY DELETE NETWORKS: DELETE NETWORK.

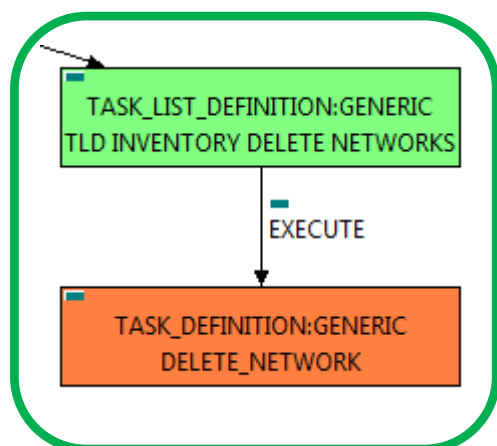


Figure: 73 Delete Network

The TDs that have present in the their names “Delete”, are Task Definitions responsible of the deletion in the platform targeted and in the DDBB, in this case, the artifacts that are going to be deleted are NETWORKS.

Once finished, the TD should have been deleted the NETWORKs artifacts mentioned above, this means, all NETWORKs both DCN and OPENSTACK from the DDBB.

Targets of the TASK:DEFINITION:
STATUS of the TD: ENABLED
Categories:

```

FIND.Condition ==          status==constant:ACTIVE
SET.Running_Status ==      ACTIVE.
SET.Status ==              INSTANTIATED.
EXECUTE.Workflow ==
    "WF_TS_DEPROVISION_NETWORK"
ROLLBACK.Behaviour_on_error == STOP
ROLLBACK.Numbre_of_retries == 0
  
```

The Workflow present in EXECUTE.Workflow attribute it is going to seek a VIRTUAL_LINK in Status ACTIVE in the DDBB . Once found , the WF will start the deleting, if deletion is successful we set the status of the artifact as the SET.Status attribute dictates. The attribute SET.Running_Status concern about the temporal status that the artifact it is going to maintain until the final change of status that comes from SET.Status. Notice that the TD is not going to change the status of the entity used for the deletion.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, the attribute “number_of_retries” set the number of rollback attempts.

12.8 TLD DELETE IDS: DELETE NETWORK SUBNETWORK IDS.

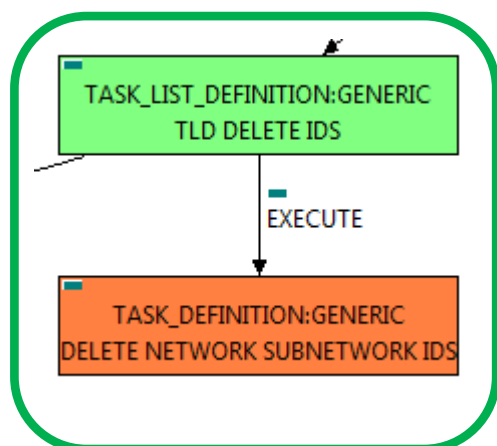


Figure: 74 Delete Network.

The TDs that have present in the their names “Delete”, are Task Definitions responsible of the deletion in the platform targeted and in the DDBB, in this case, the artifacts that are going to be deleted are NETWORKS.

Once finished, the TD should have been deleted the NETWORKS GENERAL.net_id field.

Targets of the TASK:DEFINITION:

STATUS of the TD: ENABLED

Categories:

```

FIND.MainArtifact==
VIRTUAL_LINK>NETWORK:GENERIC>NETWORK:OPENSTACK@status=ACTIVE
EXECUTE.Workflow ==
    "WF_TS_DELETE_NETWORK_SUBNETWORK_ID"
ROLLBACK.Behaviour_on_error == STOP
ROLLBACK.Numbre_of_retries == 0
  
```

The Workflow present in EXECUTE.Workflow attribute it is going to seek a NETWORK:OPENSTACK in Status ACTIVE in the DDBB . Once found, the WF will start the deleting. At the end, NETWORK OPENSTACK will have its GENERAL.net_id field empty. This task should only executed when the Virtual Link was created for shared networks and is attached to the last VDC.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, the attribute “number_of_retries” set the number of rollback attempts.

12.9 TLD INVENTORY DELETE VIRTUAL LINK: VIRTUAL LINK INVENTORY DELETE.

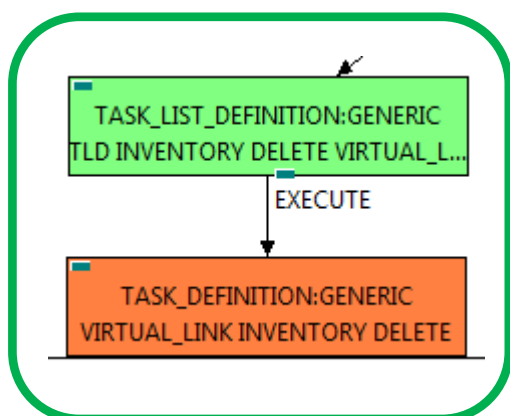


Figure: 75 Delete Virtual Link.

The TDs that have present in their names “Delete Inventory”, are Task Definitions responsible of the deletion of the artifact given, in this case, this TD it is going to delete a VNF:FW, notice the workflow used in this TD, “WF_TS_DELETE_INSTANCE_TREE”, all the components and elements below the entity that it is going to be deleted, are going to be eliminated as well.

Targets of the TASK:DEFINITION:

STATUS of the TD: ENABLED

Categories:

```

FIND.ArtifactType ==          VIRTUAL_LINK.
EXECUTE.Workflow ==
    "WF_TS_DELETE_INSTANCE_TREE"
ROLLBACK.Behaviour_on_error == STOP
ROLLBACK.Numbre_of_retries == 0
  
```

The Workflow present in EXECUTE.Workflow attribute it is going to seek a VIRTUAL_LINK in the DDBB . Once found, the WF will start the deleting.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a “STOP” set as behavior, so no Rollback it is going to be initiated, so the execution it is going to end here in case of error.

Chapter 13 Undeploy of a VNF Group – Bottom-Up.

From now on, and to make easier the understanding of the TLDs, we are going to explain the functionality of each set of TASK_LIST_DEFINITION:GENERIC, and the number of TASK_DEFINITION:GENERIC children of the previously mentioned TASK_LIST_DEFINITION:GENERIC.

Basically, the TASK_LIST_DEFINITION:GENERIC connect what we can consider “units of execution”, those are the TASK_DEFINITION:GENERIC, that have a WORKFLOW assigned to be executed when the execution of the TLD reach them.

If you like to have a more deep knowledge about the workflows mentioned through this document please refer to the specific document.

If in the category FIND, the attribute Path is present, the attribute FIND.ArtifactType will be the starting artifact for the Path, but the FIND.Status attribute refers to the last artifact on the Path.

FIND.ArtifactType == VIRTUAL_MACHINE.

FIND.Status== INSTANTIATED.

FIND.Path==

VIRTUAL_MACHINE>VIRTUAL_CORE<CORE<CPU<SERVER<AVAILABILITY_ZONE<REGION>COMPUTE>FLAVOR

In this example, we are looking for a FLAVOR in status INSTANTIATED, we do not expect to get a VIRTUAL_MACHINE, in status INSTANTIATED.

If during the use of the TLDs, the “Regenerate UUIDs” option is used, the user should check the Id of the tree that brings all the elements of the TLD, this “id” is specific and it will be the same for all the tree groups in all the TLDs.

The two modes available are “Default” and “Simulated”, the second one is only available if it is configured previously, by defect, the mode that will be used is “Default”.

13.1 Specific Elements of the TLD Undeploy VNF Group.

In this chapter the different elements of the specific TLD will be explained conscientiously.

13.2 TLD UNDEPLOY VNF GROUP: Undeploy Check

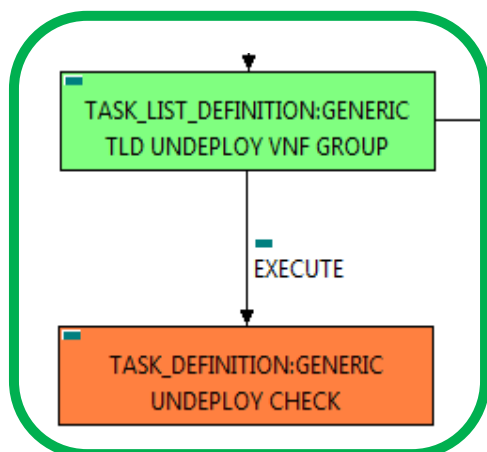


Figure: 76 Undeploy check.

This TD it is going to assure the scenario in order to delete a specific VNF Group, this means that during the execution the TD is going to check if all the children of the VNF Group are still or were properly deleted before launch the undeploy of an VNF Group.

Targets of the TASK DEFINITION:

STATUS of the TD: ENABLED

Categories:

GENERAL.Name ==	Undeploy check
FIND.Condition ==	status==constant:ACTIVE
SET.Status ==	INSTANTIATED
EXECUTE.Workflow ==	“WF_TS_UNDEPLOY_CHECK_CHILDREN”
EXECUTE.Inactive==	false
ROLLBACK.Behaviour_on_error ==	ROLLBACK
ROLLBACK.Number_of_retries ==	0
DATA.Lock ==	true

The Workflow present in EXECUTE.Workflow attribute it is going to seek for the children entities of the VNF Group, in case the TD find some the execution of the TD will fail, the goal of this TD is to guarantee that the VNF Group has no children and also is in the proper conditions to be set as an entity with status INSTANTIATED.

Once found, the TD would execute the WF present in EXECUTE.Workflow, in this case, the Wf is “WF_TS_UNDEPLOY_CHECK_CHILDREN“, the workflow will develop the task previously explained.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a “STOP” set as behavior, so no Rollback it is going to be initiated, so the execution it is going to end here in case of error.

The attribute DATA.Lock is set with the value “true”, this means once the TD ends its execution the element which is being used by the TD will be locked.

13.3 TLD DELETE VNF_GROUP: VNF_GROUP Delete.

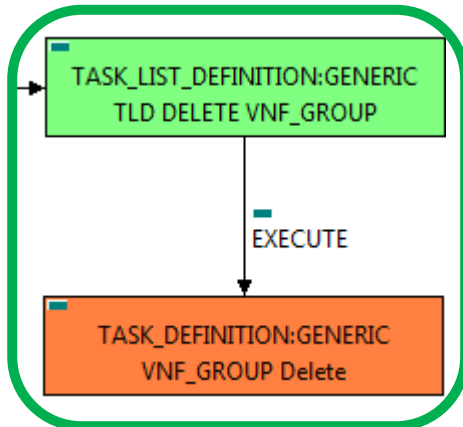


Figure: 77 VNF Group Delete.

The TDs that have present in their names “Delete”, are Task Definitions responsible of the deletion of the artifact given, in this case, this TD it is going to delete an VNF GROUP, notice the workflow used in this TD, “WF_TS_DELETE_INSTANCE_TREE”, all the components and elements below the entity that it is going to be deleted, are going to be eliminated as well, in other case, this elements will remain unreachable, that is not desirable.

Targets of the TASK:DEFINITION:

STATUS of the TD: ENABLED

Categories:

```

EXECUTE.Workflow ==
    "WF_TS_DELETE_INSTANCE_TREE"
ROLLBACK.Behaviour_on_error ==      STOP
ROLLBACK.Number_of_retries ==      0
DATA.Lock ==                        true
  
```

The Workflow present in EXECUTE.Workflow attribute it is going to seek a VNF GROUP in the DDBB. Once found, the WF will start the deleting.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a “STOP” set as behavior, so no Rollback it is going to be initiated, so the execution it is going to end here in case of error.

Chapter 14 Undeploy of a VNF - Bottom-Up.

From now on, and to make easier the understanding of the TLDs, we are going to explain the functionality of each set of TASK_LIST_DEFINITION:GENERIC, and the number of TASK_DEFINITION:GENERIC children of the previously mentioned TASK_LIST_DEFINITION:GENERIC.

Basically, the TASK_LIST_DEFINITION:GENERIC connect what we can consider “units of execution”, those are the TASK_DEFINITION:GENERIC, that have a WORKFLOW assigned to be executed when the execution of the TLD reach them.

If you like to have a more deep knowledge about the workflows mentioned through this document please refer to the specific document.

If in the category FIND, the attribute Path is present, the attribute FIND.ArtifactType will be the starting artifact for the Path, but the FIND.Status attribute refers to the last artifact on the Path.

FIND.ArtifactType == VIRTUAL_MACHINE.

FIND.Status== INSTANTIATED.

FIND.Path==

VIRTUAL_MACHINE>VIRTUAL_CORE<CORE<CPU<SERVER<AVAILABILITY_ZONE<REGION>COMPUTE>FLAVOR

In this example, we are looking for a FLAVOR in status INSTANTIATED, we do not expect to get a VIRTUAL_MACHINE, in status INSTANTIATED.

If during the use of the TLDs, the “Regenerate UUIDs” option is used, the user should check the Id of the tree that brings all the elements of the TLD, this “id” is specific and it will be the same for all the tree groups in all the TLDs.

The two modes available are “Default” and “Simulated”, the second one is only available if it is configured previously, by defect, the mode that will be used is “Default”.

14.1 Specific Elements of the TLD Undeploy VNF

In this chapter the different elements of the specific TLD will be explained conscientiously.

14.2 TLD UNDEPLOY VNF: PREPROCESS.

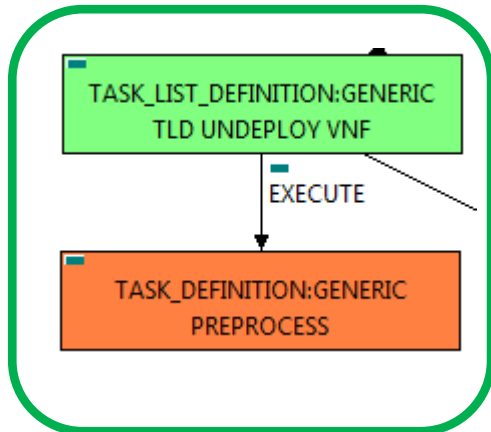


Figure: 78 Undeploy Preprocessing policies.

This TD is responsible of the deletion in the right order PreProcessing policies, these policies allow the user to set a number of elements that should be taken in consideration in a certain order, during the Undeploy process this policies should be taken down in the right order, in other way the Undeploy could be unsuccessful.

Targets of the TASK DEFINITION:

STATUS of the TD: ENABLED

Categories:

FIND.MainArtifact == POLICY:POSTPRE_PROCESSING.

FIND.Condition ==

PROCESSING_JOB.Job_type==constant:PRE&&

PROCESSING_JOB.Operation==constant:UNDEPLOY

EXECUTE.OrderBy == PROCESSING_JOB.OrderBy

In this TD there is not a workflow to be executed, the target of this TD is process in the correct order the PreProcessing policies present in the VNF, these policies should be executed in a specific order to make the changes or configurations properly, in other case an error will be launched

If the TD ends successfully, the Pre-Processing policies will have been deleted adequately.

In case of error during the execution, the TD will jump to the ROLLBACK category, If the "Behaviour_on_error" attribute its set on "ROLLBACK" the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a "STOP" as value set for behavior, so no Rollback it is going to be initiated, the execution will stop.

Due to that the value of the attribute DATA.Lock is false, when the Task Definition has finished the artifact that was used in the workflow executed will remain unlocked.

14.3 TLD STOP MONITOR: STOP MONITOR.

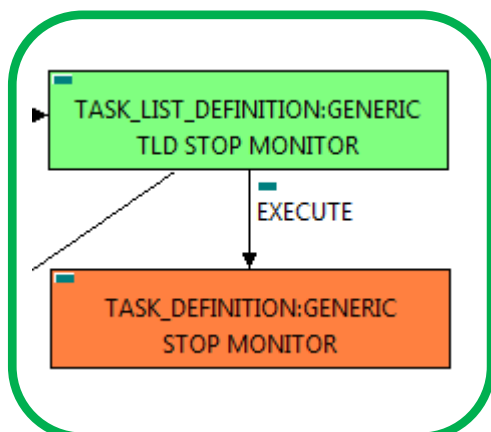


Figure: 79 Stop Monitor.

The TDs that have present in their names “Stop”, are Task Definitions responsible of the stopping of an specific artifact or element, in this case of the MONITORS, the TLD it is going to stop the MONITOR element given.

Once finished, our VNF should have the MONITOR given in status DEPLOYED, having taken in consideration all the rules for the stopping.

Targets of the TASK DEFINITION:
STATUS of the TD: ENABLED

Categories:

```

FIND.MainArtifact == MONITOR.
FIND.Condition == status==constant:STARTED
SET.Running_Status == STARTED.
SET.Status == DEPLOYED.
EXECUTE.OrderBy == GENERAL.order
EXECUTE.Workflow == "WF_TS_MONITOR_STOP"
ROLLBACK.Behaviour_on_error == STOP
ROLLBACK.Number_of_retries == 0
DATA.Lock == false
  
```

The Workflow present in EXECUTE.Workflow it is going to seek a MONITOR in Status STARTED in the DDBB, when the WF find it, it will start. This workflow stop the given MONITOR needed by the VNF to get a successful Undeploy.

In case of error during the execution, the TD will jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a “STOP” as value set for behavior, so no Rollback it is going to be initiated, the execution will stop.

Due to that the value of the attribute DATA.Lock is false, when the Task Definition has finished the artifact that was used in the workflow executed will remain unlocked.

14.4 TLD DEACTIVATE VM : DEACTIVATE VM.

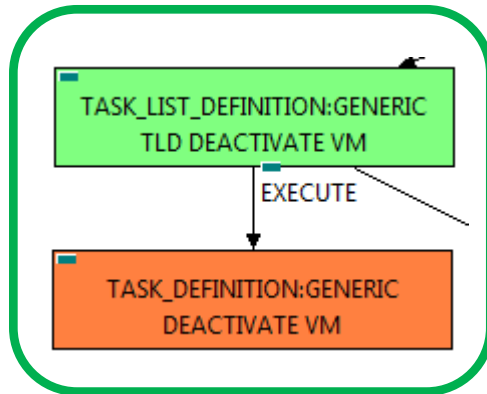


Figure: 80 Deactivation of a stopped VM.

The TDs that have present in the their names “Deactivate”, are Task Definitions responsible of the deactivation in the platform targeted and the updating of the status in the platform and the DDBB, in this case, the artifact that is going to be deactivated is a “VIRTUAL_MACHINE with status “STOPPED”, this means, when this workflow finish, we will have the VIRTUAL_MACNIHE given with a new status INSTANTIATED.

Targets of the TASK DEFINITION:
STATUS of the TD: ENABLED
Categories:

```

FIND.MainArtifact ==
VNF>VNF_COMPONENT>VIRTUAL_MACHINE
FIND.Condition==
status == constant:ACTIVE || status == constant:STOPPED
SET.Running_Status == ACTIVE.
SET.Status == TERMINATED.
EXECUTE.OrderBy = GENERAL.order
EXECUTE.Workflow == "WF_TS_DEACTIVATE_VM"
ROLLBACK.Behaviour_on_error == STOP
ROLLBACK.Number_of_retries == 0
ROLLBACK.Workflow== "WF_TS_ACTIVATE_VM"
DATA.Lock == false
  
```

The Workflow present in EXECUTE.Workflow attribute it is going to deactivate a “VIRTUAL_MACHINE” in Status ACTIVE in the DDBB. Once found, the WF will start the deactivation, if the deactivation it is successful we set the status of the artifact as the SET.Status attribute dictates. The attribute SET.Running_Status concern about the temporal status that the artifact it is going to maintain until the final change of status that comes from SET.Status.

This TD could launch different workflows depending on the type of the VM that it is going to be deactivated, the main kinds of our VIRTUAL_MACHINES are HELION, and thereupon two of the WFs that could be used in this deactivation are: “WF_NFVD_DEACTIVATE_VM_HELION” and “WF_NFVD_DEACTIVATE_VM_OPENSTACK”.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, in this case it will be “WF_TS_ACTIVATE_VM”, but in this case, we have a “STOP” set as behavior, so no Rollback it is going to be initiated, so the execution it is going to end here in case of error.

Due to that the value of the attribute DATA.Lock is false, when the Task Definition has finished the artifact that was used in the workflow executed will remain unlocked.

14.6 TLD DELETE VPORT: DELETE PORT.

The TDs that have present in their names “Delete”, are Task Definitions that delete an artifact or element from the DDBB or from the platforms targeted, in this case, the artifact that it is going to be delated is the VIRTUAL_PORT given.

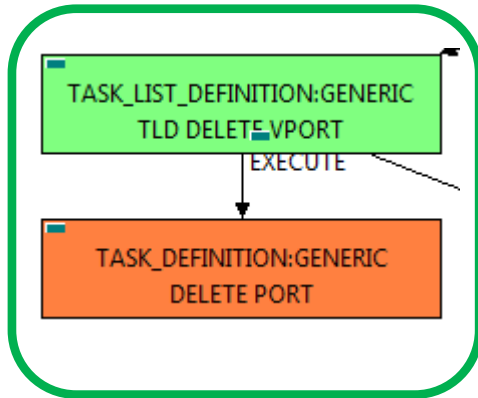


Figure: 82 Delete vPort.

Targets of the TASK:DEFINITION:

STATUS of the TD: ENABLED

Categories:

```

FIND.MainArtifact ==
VNF>VNF_COMPONENT>
VIRTUAL_MACHINE>VIRTUAL_PORT@status=ACTIVE
SET.Running_Status ==                ACTIVE.
SET.Status ==                        INSTANTIATED.
EXECUTE.Workflow ==                  "WF_TS_DEACTIVATE_PORT"
EXECUTE.Inactive ==                  false
ROLLBACK.Behaviour_on_error ==      STOP
ROLLBACK.Number_of_retries ==       0
DATA.Lock ==                         false
  
```

The Workflow present in EXECUTE.Workflow it is going to seek a VIRTUAL_PORT in Status ACTIVE in the DDBB, when the WF find it, it will start. This workflow will start one more, this last one, it is going to be named after the VIRTUAL_PORT that the TD it is trying to delete.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a “STOP” set as behavior, so no Rollback it is going to be initiated, so the execution it is going to end here in case of error.

Due to that the value of the attribute DATA.Lock is false, when the Task Definition has finished the artifact that was used in the workflow executed will remain unlocked.

14.7 LD Undeploy Monitor, Volume: UNDEPLOY_MONITOR

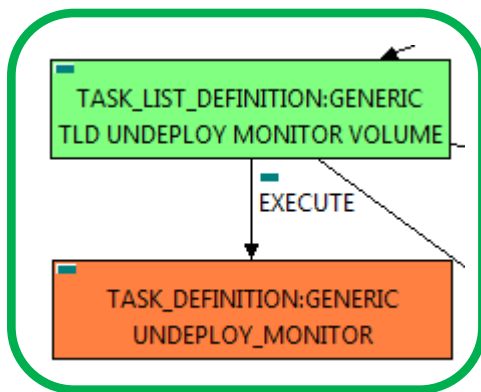


Figure: 83 Undeploy Monitor.

The TDs that have present in the their names “Deploy” are Task Definitions responsible of the deployment in the platform targeted and the updating of the status in the platform and the DDBB , these deployments are slightly different to the ones we launch for our entities, as a rule, they are small components as the MONITORS. In this case, the artifact that is going to be deployed is a “MONITOR”, this means, when this workflow finish, we will have a MONITOR deployed with status DEPLOYED.

Targets of the TASK:DEFINITION:
 STATUS of the TD: ENABLED
 Categories:

```

FIND.ArtifactType ==                      MONITOR.
FIND.Condition ==           status==constant:DEPLOYED
SET.Running_Status ==           DEPLOYED.
SET.Status ==                              INSTANTIATED
EXECUTE.Workflow ==
                                  "WF_TS_MONITOR_UNDEPLOY"
ROLLBACK.Behaviour_on_error ==        STOP
ROLLBACK.Number_of_retries ==         0
DATA.Lock ==                              false
  
```

The Workflow present in EXECUTE.Workflow attribute it is going to seek a MONITOR in Status DEPLOYED in the DDBB. Once found, the WF will start the deployment, if the deployment is successful we set the status of the artifact as the SET.Status attribute dictates. The attribute SET.Running_Status concern about the temporal status that the artifact it is going to maintain until the final change of status that comes from SET.Status.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a “STOP” set as behavior, so no Rollback it is going to be initiated, so the execution it is going to end here in case of error.

Due to that the value of the attribute DATA.Lock is false, when the Task Definition has finished the artifact that was used in the workflow executed will remain unlocked.

14.8 TLD Undeploy Monitor, Volume: DELETE VOLUME

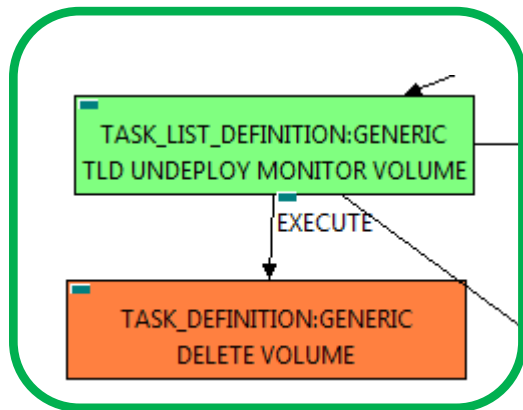


Figure: 84 Delete Volume.

The TDs that have present in the their names “Delete”, are Task Definitions responsible of the deletion of an artifact in the platform targeted and the updating of the status in the platform and the DDBB, in this case, the artifact that is going to be deleted is a “VOLUME”, this means, when this workflow finish, we are going to have a volume less.

Targets of the TASK:DEFINITION:
 STATUS of the TD: ENABLED
 Categories:

```

FIND.MainArtifact ==
VNF>VNF_COMPONENT>
VIRTUAL_MACHINE>VIRTUAL_LUN.
EXECUTE.Workflow ==
    "WF_TS_DELETE_VOLUME"
ROLLBACK.Behaviour_on_error ==    CONTINUE
ROLLBACK.Number_of_retries ==    0
DATA.Lock ==                      false
  
```

The Workflow present in EXECUTE.Workflow attribute it is going to seek a VIRTUAL_LUN in the DDBB . Once found, the WF will start the deleting, if the deletion it is successful we set the status of the artifact as the SET.Status attribute dictates. The attribute SET.Running_Status concern about the temporal status that the artifact it is going to maintain until the final change of status that comes from SET.Status.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, in this case the behavior is set as CONTINUE, this means, that the execution is going to continue no matter which error could be.

Due to that the value of the attribute DATA.Lock is false, when the Task Definition has finished the artifact that was used in the workflow executed will remain unlocked.

14.9 TLD VNF Inventory Delete: DELETE_INVENTORY.

The TDs that have present in their names “Delete Inventory”, are Task Definitions responsible of the deletion of the artifact given, in this case, this TD it is going to delete a VNF, notice the workflow used in this TD, “WF_TS_DELETE_INSTANCE_TREE”, all the components and elements below the entity that it is going to be deleted, are going to be eliminated as well.

Targets of the TASK:DEFINITION:
 STATUS of the TD: ENABLED
 Categories:

```
EXECUTE.Workflow ==
    "WF_TS_DELETE_INSTANCE_TREE"
ROLLBACK.Behaviour_on_error ==    STOP
ROLLBACK.Number_of_retries ==    0
```

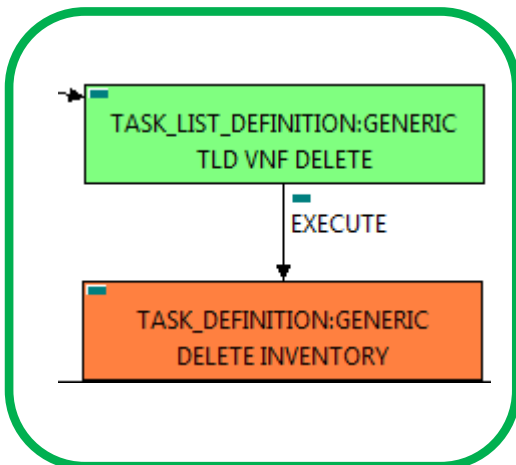


Figure: 85 Delete Inventory

The Workflow present in EXECUTE.Workflow attribute it is going to seek a VNF in the DDBB . Once found, the WF will start the deleting.

In case of error during the execution, the workflow jump to the ROLLBACK category, If the “Behaviour_on_error” attribute its set on “ROLLBACK” the WF will start the execution of the Workflow present in the attribute with the same name in the category ROLLBACK, but in this case, we have a “STOP” set as behavior, so no Rollback it is goin to be initiated, so the execution it is going to end here in case of error.

Abbreviations

Abbreviation	Definition
VM	Virtual Machine; virtualized computation environment that behaves very much like a physical computer/server
VNF	Virtual Network Function; the "application" that provides the functionality currently provided by devices
NS	Network Service; a composition of network functions (VNF or PNF) and defined by its functional and behavioral specification
NFV	Network Function Virtualization; the approach to building telecom services using virtualization approaches
VNFC	VNF Component; each VNF is composed of one or more components, often mapping to a VM
MANO	Management and Orchestration; addressing the functionality required to deal with the new abstractions; consists of NFVO, VNFM and VIM
NFVO	NFV Orchestrator; In charge of the orchestration and management of NFV Infrastructure and software resources, and realizing NS on NFVI.
VNFM	VNF Manager; responsible for VNF lifecycle management (such as Instantiation, update, query, scaling, termination). Can be implemented as part of the NFVO or supplied by the VNF provider.
VIM	Virtualized Infrastructure Manager; think OpenStack or Cloud OS
NFVI	NFV Infrastructure; the totality of all hardware and software components which build up the environment in which VNFs are deployed, managed and executed
EMS	Element Management System; performs the typical management functionality for one or several VNFs.
PNF	Physical Network Function; think today's devices.
CPU	Central Processing Unit; device in the compute node that provide the primary container interface
NF	Network Function; functional block within a network infrastructure that has well-defined external interfaces and well defined functional behavior
NIC	Network Interface Controller; device in a compute node that provides physical interface with the infrastructure network
SLA	Service Level Agreement; negotiated agreement between two or more parties, recording a common understanding about the service and/or service behavior.
CPE	Customer Premises Equipment
ETSI	European Telecoms Standards Institute
HA	High Availability
SDN	Software Defined Network
HPSA	HPE Service Activator
UCA EBC	Unified Correlation Analyzer for Event Based Correlation
JSON	JavaScript Object Notation
XML	Extensible Markup Language
API	Application Programmatic Interface
DNS	Domain Name System
DHCP	Dynamic Host Configuration Protocol
IaaS	Infrastructure-as-a-Service