

Onboard VNF Package on VNF-LCM MTAS

OPERATING INSTRUCTIONS

Copyright

© Ericsson AB 2019. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Trademark List

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.



Contents

1	Description	1
2	Procedure	1
2.1	Onboard VNF Package on VNF-LCM	1
2.2	Troubleshooting	8





1 Description

This instruction describes the onboarding procedure for deployment of a Virtual Network Function (VNF) using VNF-LCM.

Performing this procedure is a prerequisite for life cycle operations.

For more information about the VNF-LCM, see [MTAS VNF Lifecycle Management](#).

2 Procedure

2.1 Onboard VNF Package on VNF-LCM

Prerequisites

- No tools are required.
- The following condition must apply:
 - VNF deployment parameters are prepared. For more information, see Sections [Create VNF Package](#) and [Onboard VNF Workflows and Configurations to VNF-LCM](#) in [MTAS SW Installation](#).
 - The VNF-LCM is available using either the Operations Support System for Radio and Core (OSS-RC) or the Ericsson Network Manager (ENM).

Note:

- The following commands are to be executed on the VNF-LCM's Services VM.
- For OpenStack, the onboarding process has changed to VNF package upload and unpack. For VMware it is still the old legacy way!

Steps

1. Decompress the vMTAS workflow pack `CXP9034815_1-<R-state>.tar.gz` where `<R-state>` defines the revision state of the package.


```
cd /home/cloud-user
```

```
tar -zxvf CXP9034815_1-<R-state>.tar.gz
```
2. Install the vIMS workflow bundle:



- a. Log on to vnflaf-services VM as cloud-user and switch to root user.
- b. Verify that the bundle is not already installed by running the List command.

```
wfmgr bundle list
```

If the version of vIMSWorkflows is older than the version included in the vMTAS Workflow Pack CXP9034815_1-*<R-state>*.tar.gz, install the newer version.

- c. Install the workflow bundle package by running the install command.

```
wfmgr bundle install --package=\
<workflow_bundle_rpm_file _path>
```

For more information on workflow bundle installation in VNF-LCM, see ENM/OSS-RC documentation.

3. OpenStack only onboarding steps:

- a. Decompress vMTAS VNF package *<VNF package>*.zip (which was created in MTAS SW Installation, and uploaded to VFN-LCM):

```
unzip /home/cloud-user/<VNF package>.zip -d
/vnflcm-ext/current/vnf_package_repo/
```

- b. Copy the VNF configuration files which are previously created. See MTAS SW Installation:

```
cd /vnflcm-ext/current/vnf_package_repo/\
<VNF package>/configurations/<VNF_1 configuration>

cp -r /home/cloud-user/<VNF_1 configuration> .
```

Note: Each directory in configurations contains a VNF instance-specific environment file (*env.yaml*), configuration files *evip_cli.txt* for eVIP, *ss7.conf* for SS7, and *pdb_bundle.zip* for application configuration. Each directory also contains *id_rsa.pub* public key of the VNF-LCM's *jboss_user*. The files *evip_cli.txt*, *ss7.conf*, and *pdb_bundle.zip* are optional.

To use the same VNF package for instantiate another VNF (VNF2), the following steps are needed:

- Create the VNF2 specific configuration files as It is described in Section Example for Instantiating More VNFs with the Same VNF Package in MTAS SW Installation.
- Create the configuration directory for VNF2:

```
mkdir /vnflcm-ext/current/vnf_package_repo/\
```



```
<VNF package>/configurations/<VNF_2 configuration>
```

- Copy the VNF2 specific configuration files to the configuration directory:

```
cd /vnflcm-ext/current/vnf_package_repo/\
<VNF package>/configurations/<VNF_2 configuration>

cp -r /home/cloud-user/<VNF_2 configuration> .
```

4. VMware only onboarding steps:

- a. Decompress vMTAS LCM scripts CXP9034788_1-<R-state>.tar.gz where <R-state> defines the revision state of the package:

```
mkdir /vnflcm-ext/current/vnf_package_repo/\
<VNFTYPE__VNFVersion>

cd /vnflcm-ext/current/vnf_package_repo/\
<VNFTYPE__VNFVersion>

tar -zxvf /home/cloud-user/CXP9034788_1-<R-state>.tar.gz
```

Note: Follow the naming convention: VNF type and VNF version separated by __. (Where “__” is a double underscore.)

- b. In /vnflcm-ext/current/vnf_package_repo/<VNFTYPE__VNFVersion>, create a configurations subdirectory with read permission for the jboss_user and a child directory for each VNF configuration (that is, VNF instance), and copy the VNF configuration to the child directory:

```
mkdir /vnflcm-ext/current/vnf_package_repo/\
<VNFTYPE__VNFVersion>/configurations

mkdir /vnflcm-ext/current/vnf_package_repo/\
<VNFTYPE__VNFVersion>/configurations/<VNF_1
configuration>

cd /vnflcm-ext/current/vnf_package_repo/\
<VNFTYPE__VNFVersion>/configurations/<VNF_1
configuration>

cp -r /home/cloud-user/<VNF_1 configuration> .
```

Note: Each directory in configurations contains a VNF instance-specific environment file (env-vcd.yaml), configuration files evip_cli.txt for eVIP, ss7.conf for SS7, and pdb_bundle.zip for application configuration. Each directory also contains id_rsa.pub public key of the VNF-LCM jboss_user. The files evip_cli.txt, ss7.conf, and pdb_bundle.zip are optional.



5. If the SSH key is not available yet for jboss, create it using the following commands:

```
su jboss_user  
  
ssh-keygen -t rsa  
  
exit
```

Note: The use of encrypted private keys is not supported in the current release (that is, do not use passphrase).

6. Copy the public SSH key into the configuration directory of VNF instance:

```
cp /home/jboss_user/.ssh/id_rsa.pub /vnflcm-ext/current/vnf_package_repo/<VNF package>\  
/configurations/<VNF_1 configuration>/
```

Note: The public key must be added in the configuration directory for each instance.

7. Verify the structure of the /vnflcm-ext/current/vnf_package_repo/<VNF package> directory:

In OpenStack based Cloud environment:



```

`-- vMTAS-<version>-flavorId
    |-- configurations
    |   |-- <VNF_1 configuration>
    |   |   |-- env.yaml
    |   |   |-- evip_cli.txt*
    |   |   |-- ss7.conf*
    |   |   |-- instance1_pdb_bundle.zip*
    |   |   |-- id_rsa.pub
    |   |-- <VNF_2 configuration>
    |   |   |-- env.yaml
    |   |   |-- evip_cli.txt*
    |   |   |-- ss7.conf*
    |   |   |-- instance1_pdb_bundle.zip*
    |   |   |-- id_rsa.pub
    |-- mtas_hot.yaml
    |-- Resources
    |   |-- HotFiles
    |   |   |-- mtas_hot_scaling.yaml
    |   |-- LcmScripts
    |   |   |-- common.py
    |   |   |-- lcmUtilities
    |   |   |   |-- __init__.py
    |   |   |   |-- lcm_decorators.py
    |   |   |   |-- lcm_output_manager.py
    |   |   |   |-- lcm_task_config.py
    |   |   |   |-- lcm_task_config.pyi
    |   |   |-- post_heal.py
    |   |   |-- post_instantiation.py
    |   |   |-- post_scale_in.py
    |   |   |-- post_scale_out.py
    |   |   |-- pre_heal.py
    |   |   |-- pre_instantiation.py
    |   |   |-- pre_scale_in.py
    |   |   |-- pre_scale_out.py
    |   |   |-- pre_termination.py
    |   |   |-- pre_upgrade.py
    |   |   |-- scale_base.py
    |   |   |-- version
    |   |-- VnfdWrapperFiles
    |   |   |-- VNFD_Wrapper.json
    |   |   |-- vnflcmOperationsConfiguration.json

```

In VMware based cloud environment:

```

`-- vMTAS__<x.y>
    |-- vnflcmOperationsConfiguration.json
    |-- configurations
    |   |-- <VNF_1 configuration>
    |   |   |-- env-vcd.yaml
    |   |   |-- id_rsa.pub
    |   |   |-- evip_cli.txt*

```



```

|   |   |-- ss7.conf*
|   |   `-- instance1_pdb_bundle.zip*
|   `-- <VNF_2 configuration>
|       |-- env-vcd.yaml
|       |-- id_rsa.pub
|       |-- evip_cli.txt*
|       |-- ss7.conf*
|       `-- instance1_pdb_bundle.zip*
`-- lcmScripts

```

The files marked with * are optional.

8. If the default LCMhooks time-out needs to be changed, open the `vnfLcmOperationsConfiguration.json` file and edit the details using the following instructions:

Note: The default time-out value is 1800 seconds (30 minutes).

— Instantiation

Add a new property under the `instantiateVnfOpConfig` item with the following syntax:

```
workflow_lcmScriptTimeout:<seconds>
```

— Termination

Add a new property under the `terminateVnfOpConfig` with the following syntax:

```
workflow_lcmScriptTimeout:<seconds>
```

9. Only for OpenStack based Cloud environment: If automatic invocation of Heal VNF WF (so called auto-heal WF) is to be enabled:

Note: For CEE deployments, this step is not necessarily needed, since the CEE CM-HA feature can be primarily used to Heal a VNF.

- a. Verify the presence of file `vIMS-autostart-rules.xml` in services VM of VNF-LCM:

```
ll /vnflcm-ext/current/workflows/auto-start-rules/vIMS-autostart-rules.xml
```

- b. If the file is not present, copy the template `autostart-rule` file to the directory in the previous step, and remove all existing template `workflowTriggerRule`:

```
sed '24,45d' /opt/ericsson/ERICvIMSWorkflowsworkflows/autostart-rules/vIMS-autostart-rules.xml \
> /vnflcm-ext/current/workflows/auto-start-rules/vIMS-autostart-rules.xml
```



Note: It can be needed to verify that the first template workflowTriggerRule starts in line 24 and the last workflowTriggerRule ends in line 45.

- c. The file /vnflcm-ext/current/workflows/auto-start-rules/vIMS-autostart-rules.xml must contain the following workflowTriggerRule:

```
<wft:workflowTriggerRule name="vMTAS-AutoHealing" id="vMTAS-AutoHealing"
type="FM_ALARM">
  <wft:triggerCondition>
    <wft:attributeName>specificProblem</wft:attributeName>
    <wft:triggerConditionOperator>EQUALS</wft:triggerConditionOperator>
    <wft:attributeValue>COM SA, CLM Cluster Node Unavailable</wft:attributeValue>
  </wft:triggerCondition>
  <wft:triggerAction WorkflowDefinitionName="Heal VNF"
WorkflowBundleName="vIMSWorkflows">
    <wft:param name="additionalParams" value="{}/">
  </wft:triggerAction>
</wft:workflowTriggerRule>
```

- d. Verify that the content of /vnflcm-ext/current/workflows/auto-start-rules/vIMS-autostart-rules.xml is similar to the following example: (it can contain additional workflowTriggerRule):

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!--
```

```
*****
```

Brief description for the elements of the sample rule shown below:

```
*****
```

<wft:workflowTriggerRule> : This is the rule element, which can have multiple \ triggerConditions and a triggerAction. *

And its mandatory attributes are id, name, type.

<wft:triggerCondition> : Defines the condition, based on which the \ action will be performed. Each triggerCondition *

has the attributeName, triggerConditionOperator, \ attributeValue as child elements. *

<wft:attributeName> : This element is mapped with the Alarms/Event \ attributes. *

<wft:triggerConditionOperator>: Describes the condition matching operation on the \ attribute values specified. Valid values *

for the element are EQUALS, NOTEQUALS and CONTAINS.

<wft:attributeValue> : The exact values to be evaluated.

<wft:triggerAction> : Defines the action to be taken when the condition is matched. Its mandatory attributes are *

WorkflowDefinitionName, and WorkflowBundleName.

```
*****
```

For more details, follow the below link :

<https://arm101-eiffel004.lmera.ericsson.se:8443/nexus/content/sites/tor/vnflaf-sdk/latest/workflow-log-feature.html> *



```

*****
-->
<wft:workflowTriggerRules name="" type=""
    version="" xmlns:wft="urn:com:ericsson:schema:xml:oss:wft" xmlns:xsi=\
"http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:com:ericsson:schema:xml:oss:wft auto-start-wf-rules.xsd ">
    <wft:workflowTriggerRule name="vMTAS-AutoHealing" id="vMTAS-AutoHealing"
type="FM_ALARM">
    <wft:triggerCondition>
    <wft:attributeName>specificProblem</wft:attributeName>
    <wft:triggerConditionOperator>EQUALS</wft:triggerConditionOperator>
    <wft:attributeValue>COM SA, CLM Cluster Node Unavailable</wft:attributeValue>
    </wft:triggerCondition>
    <wft:triggerAction WorkflowDefinitionName="Heal VNF"
WorkflowBundleName="vIMSWorkflows">
    <wft:param name="additionalParams" value="{ }"/>
    </wft:triggerAction>
    </wft:workflowTriggerRule>
</wft:workflowTriggerRules>

```

Note: If it is needed to enable auto-healing only for a particular VNF instance, the `workflowTriggerRule` can be refined by adding additional `triggerConditions` to it.

10. Only for OpenStack-based Cloud environment: If time-based automatic invocation of Scale VNF WF (so-called time-based auto-scale WF) is to be enabled:

- a. Copy `vmtas-time-based-scaling-autostart-rule.xml` included in the vMTAS Workflow Pack `CXP9034815_1-<R-state>.tar.gz` at `autostart-rules` folder to `/vnflcm-ext/current/workflows/autostart-rules`
- b. Restart Jboss:

```
sudo service jboss restart
```

2.2 Troubleshooting

If the workflow execution fails, inspect the relevant logs to identify the cause of the failure.

Steps

1. Increase the log level from INFO to DEBUG. For information on how to change log level, see VNF-Lifecycle Manager System Administration Guide, 1543-APR 901 0578.
2. Inspect the following logs to identify the cause of the failure:



- Jboss Server log: `/ericsson/3pp/jboss/standalone/log/server.log`
 - System log: `/var/log/messages`
 - Workflow log: the **Workflow Log** view in the VNF-LCM
3. If a problem cannot be solved, consult the next level of maintenance support and provide the logs. Further actions are outside the scope of this instruction.