

SAPC VNF Descriptor Generator Tool

Ericsson Service-Aware Policy Controller

USER GUIDE

Copyright

© Ericsson España, S.A., 2017, 2018. 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 | SAPC VNF Descriptor Generator Tool Overview | 1 |
| 2 | SAPC VNF Descriptor Generator Tool Prerequisites | 3 |
| 3 | SAPC VNF Descriptor Generator Tool Preparations | 5 |
| 3.1 | Virtual Delivery Package | 5 |
| 3.2 | Cluster Adaptation | 7 |
| 3.3 | Initial Configuration and Provisioning | 7 |
| 3.4 | Descriptor Generator Tool Configuration File | 8 |
| 4 | SAPC VNF Descriptor Generation | 23 |
| 4.1 | Package Generation | 23 |
| 4.2 | OVF Package Generation Output | 24 |
| 4.2.1 | ECM over CEE | 24 |
| 4.2.2 | VMware vSphere Client | 25 |
| 4.2.3 | VMware vCloud Director | 26 |
| 4.3 | HOT Package Generation Output | 28 |
| 4.3.1 | OpenStack | 28 |
| 5 | SAPC VNF Descriptor Generator Tool Examples | 31 |
| 5.1 | Example of the adapt_cluster.cfg File | 31 |
| 5.2 | Example of the Initial Configuration File | 32 |
| 5.3 | Example of the Initial Provisioning File | 32 |
| 5.4 | Example of the SAPC.cfg Configuration File | 35 |
| 5.5 | Example of the ECM Flavors XML File | 36 |
| 5.6 | Example of the Script to Upload Flavors in OpenStack | 38 |





List of Tables

| Table | Title | Page |
|-------|---|------|
| 1 | Virtual Delivery Package Content | 5 |
| 2 | SAPC.cfg [cloud] Section Parameters | 8 |
| 3 | SAPC.cfg [vsphere] Section Parameters | 9 |
| 4 | SAPC.cfg [vcloud_director] Section Parameters | 9 |
| 5 | SAPC.cfg [cee] Section Parameters | 9 |
| 6 | SAPC.cfg [ecm] Section Parameters | 10 |
| 7 | SAPC.cfg [scaling] Section Parameters | 10 |
| 8 | SAPC.cfg [sapc] Section Parameters | 12 |
| 9 | Possible Values for deployment_type in SAPC.cfg | 21 |





1 SAPC VNF Descriptor Generator Tool Overview

The purpose of this document is to describe how to set up and execute the Descriptor Generator Tool, which generates the SAPC Virtual Network Function (VNF) Descriptor needed as input for the SAPC to be deployed in cloud environments. The VNF Descriptor for the SAPC can be written in the Open Virtualization Format (OVF), generating an OVF package, or in Heat Orchestration Template (HOT), generating a HOT package.

The SAPC VNF Descriptor details all the information needed to deploy the SAPC as a Virtual Application (vAPP) or stack running in the cloud. For example, the Virtual Machines to be deployed and their resources, the virtual networks, and the startup order. For further details, refer to the [Open Virtualization Format specification and White Paper](#) or to the [Heat Orchestration Template \(HOT\) specification](#).

The Descriptor Generator Tool generates the OVF/HOT main file required for the SAPC deployment, and the OVF/HOT package including it and some additional files. The input required for the Descriptor Generator Tool is a configuration file with the parameters needed to generate the resulting OVF/HOT package. For instance, the target cloud system or the flavors related to the Virtual Machines (basically number of vCPUs, memory, and disk).





2 SAPC VNF Descriptor Generator Tool Prerequisites

Prerequisites

- General knowledge about Linux systems and shell commands.
- General knowledge about virtualization and cloud concepts.
- General knowledge about networking.
- Good understanding and knowledge about the SAPC networking in cloud deployments. Refer to [SAPC VNF Network Configuration Guide](#).
- A machine referred in this document as "preparation server".
 - The preparation server must be a Linux machine.
- The Virtual Delivery Package (VDP) for the SAPC is previously downloaded from the Software Gateway. The VDP is made available after extraction in the preparation server.
- The Descriptor Generator Tool main executable is `SAPC_descriptor_generator` inside the SAPC VDP. It requires the presence of the `SAPC.cfg` file which is the configuration file needed as input for the Descriptor Generator Tool, and the `adapt_cluster_template.cfg` file which is also needed for the computation executed by the Descriptor Generator Tool. All these files and directories are present in the VDP.





3 SAPC VNF Descriptor Generator Tool Preparations

3.1 Virtual Delivery Package

The VDP for the SAPC deployments in OpenStack is named `vdp_sapc_qcow2_cxp9032849_<revision>.tar.gz`.

The VDP for the SAPC deployments in VMware platforms is named `vdp_sapc_vmdk_cxp9032850_<revision>.tar.gz`.

To generalize, from now on the VDP file is referred as: `vdp_sapc_<image_format>_<cxp_number>_<revision>.tar.gz`

Once the SAPC VDP has been downloaded from the Software Gateway and copied to the preparation server, its content is decompressed and extracted:

```
PreparationServer:# tar xzf vdp_sapc_<image_format>_<cxp_number>_<revision>.tar.gz
```

Then, navigate to the extracted folder:

```
PreparationServer:# cd vdp_sapc_<image_format>_<cxp_number>_<revision>
```

The VDP content is available, and some differences can be found depending on the target cloud system.

Table 1 Virtual Delivery Package Content

| Filename | OpenStack | VMware | Description |
|---------------------------|-----------|--------|---|
| SAPC_descriptor_generator | Yes. | Yes. | The Descriptor Generator Tool . It is a binary executable used to generate the SAPC OVF or HOT package. See Section 4.1 on page 23. |
| SAPC.cfg | Yes. | Yes. | Configuration file used as input by all the tools delivered with the VDP. See Section 3.4 on page 7. |



| Filename | OpenStack | VMware | Description |
|--|-------------------------|--|---|
| bin/ | Yes. | Yes. | <p>Directory containing the following binary executable files:</p> <ul style="list-style-type: none">• SAPC_orchestrator: Tool used during the Decommissioning and Scale-out/Scale-In procedures in OpenStack deployments. For further details, refer to SAPC VNF Decommissioning Instruction for OpenStack, Configure Scale-Out and Configure Graceful Scale-In documents.• SAPC_authentication: Tool used to configure an SSH-key pair between the VNF-LCM and the SAPC, required for some workflows as Scale-In.• upgrade: Tool used during the Upgrade By Replacement procedure. For further details, refer to VNF Upgrade by Replacement Instruction. <p>Additional details for each tool can be found with the help parameter.</p> |
| templates/ | Yes. | Yes. | Directory containing the different templates for the cluster adaptation. Refer to Adapt Cluster Tool. |
| examples/ | Yes. | Yes. | Directory containing files as simple examples for the initial configuration and provisioning. See Section 3.3 on page 7. |
| sapc_sc_<product_number>_<revision>.<format> | Yes, <format> is qcow2. | Yes, <format> is vmdk. | Virtual Machine image for the SC-1 and SC-2 VMs. |
| sapc_vr_<product_number>_<revision>.<format> | Yes, <format> is qcow2. | Yes for VMware vSphere Client, <format> is vmdk. | Virtual Machine image for the VR-1, VR-2, VR-3 and VR-4 VMs. |
| sapc_vr-1_<product_number>_<revision>.<format> | No, <format> is qcow2. | Yes, <format> is vmdk. | Virtual Machine image for the VR-1 VM. |
| sapc_vr-2_<product_number>_<revision>.<format> | No, <format> is qcow2. | Yes, <format> is vmdk. | Virtual Machine image for the VR-2 VM. |
| sapc_vr-3_<product_number>_<revision>.<format> | No, <format> is qcow2. | Yes, <format> is vmdk. | Virtual Machine image for the VR-3 VM. |
| sapc_vr-4_<product_number>_<revision>.<format> | No, <format> is qcow2. | Yes, <format> is vmdk. | Virtual Machine image for the VR-4 VM. |
| sapc_pxe_<product_number>_<revision>.<format> | Yes, <format> is qcow2. | Yes, <format> is vmdk. | Virtual Machine image for PXE (Preboot Execution Environment) boot. |



3.2 Cluster Adaptation

The SAPC configuration is customized during the vAPP or stack deployment to be adapted to the customer network. It is possible to modify, for example, the VIP addresses for O&M and traffic, the configuration of the SAPC as Diameter peer, or the IP addresses for the NTP servers.

The adaptation is done automatically during deployment time through the Adapt Cluster injected configuration file. Several configuration files `adapt_cluster_template.cfg` serving as examples are contained inside the VDP. Refer to [Adapt Cluster Tool](#) to follow the steps for performing the customization.

Refer to Section 5.1 on page 31 to see an example of the resulting `adapt_cluster.cfg` file after executing the Descriptor Generator Tool.

3.3 Initial Configuration and Provisioning

This subsection is optional. If it is not desired to perform an initial configuration or provisioning of the SAPC during deployment time, skip this subsection and go to Section 3.4 on page 7.

The SAPC is able to be initially configured or provisioned during deployment time. To achieve this, some files must be created and copied inside the path where the VDP is placed before executing the Descriptor Generator Tool:

- Write the initial configuration and store it in a file called `initial_configuration.xml` inside the path where the VDP is placed. This file has to contain an XML document with the operations to be sent to the NETCONF interface, like, for example, the configuration of a Diameter peer and its controls. See Section 5.2 on page 31 to check a basic example of the `initial_configuration.xml` file (the file containing this same example is provided in the `examples/configuration/` directory inside the VDP, see Section 3.1 on page 5).
- Write the initial provisioning and store it in a file called `initial_provisioning.rest` inside the path where the VDP is placed. This file contains the commands to provision the internal database available in the SAPC, with, for example, subscribers, groups, policies, and rules. See Section 5.3 on page 32 to check a basic example of the `initial_provisioning.rest` file (the file containing this same example is provided in the `examples/provisioning/` directory inside the VDP, see Section 3.1 on page 5).

If one or both of the previous files have been created and placed inside the VDP path, the Descriptor Generator Tool detects them when it is executed, and includes them in the OVF/HOT package. They are injected to the SAPC during deployment. Finally, the SAPC detects the injected files and perform automatically the O&M tasks to configure and provision them.



3.4 Descriptor Generator Tool Configuration File

The Descriptor Generator Tool needs a configuration file as input for its execution. This configuration file is called `SAPC.cfg`.

The `SAPC.cfg` configuration file uses `.ini` format. During this chapter, the sections and parameters to be configured in the `SAPC.cfg` are detailed. See Section 5.4 on page 35 to check an example of this file.

— [cloud] Section

This section is mandatory in the `SAPC.cfg` file.

Table 2 SAPC.cfg [cloud] Section Parameters

| Attribute | Mandatory (M), Optional (O), Conditional (C) | Description | Values | Default Value |
|----------------------------------|--|--|---|--|
| <code>target_cloud_system</code> | M | <p>Specifies target cloud system. (the NFV Infrastructure, NFV Orchestrator or cloud orchestrator, and VIM combination):</p> <ul style="list-style-type: none">• <code>ecm_cee</code> when ECM with vimZone CEE is used for deployment (OVF package to be generated).• <code>openstack</code> when Atlas/Director with CEE or RHOSP is used for deployment (HOT package shall be generated). Additionally, it generates a HOT package for ECM too. <p>Specifies target VMware DataCenter:</p> <ul style="list-style-type: none">• <code>vsphere</code>• <code>vcloud_director</code> | <ul style="list-style-type: none">• <code>ecm_cee</code>• <code>openstack</code>• <code>vsphere</code>• <code>vcloud_director</code> | If the parameter is not configured, the default value is <code>ecm_cee</code> . ⁽¹⁾ |

(1) Default value should be changed in case a different deployment configuration is used.

— [vsphere] Section

This section is conditional. It is mandatory only if the `target_cloud_system` value is `vsphere` or `vcloud_director`.



Table 3 SAPC.cfg [vsphere] Section Parameters

| Attribute | Mandatory (M), Optional (O), Conditional (C) | Description | Values | Default Value |
|----------------|--|----------------------------------|---|---|
| version | M | The VMware vSphere version used. | <ul style="list-style-type: none"> • 5.5 • 6.0 • 6.5 | If the parameter is not configured, the default value is 6.5. |

— [vcloud_director] Section

This section is conditional. It is mandatory only if the **target_cloud_system** value is **vcloud_director**.

Table 4 SAPC.cfg [vcloud_director] Section Parameters

| Attribute | Mandatory (M), Optional (O), Conditional (C) | Description | Values | Default Value |
|----------------|--|--|---|---|
| version | M | The VMware vCloud Director version used. | <ul style="list-style-type: none"> • 8 | If the parameter is not configured, the default value is 8. |

— [cee] Section

This section is conditional. It is mandatory only if the **target_cloud_system** value is **ecm_cee** or **openstack**.

Table 5 SAPC.cfg [cee] Section Parameters

| Attribute | Mandatory (M), Optional (O), Conditional (C) | Description | Values | Default Value |
|----------------|--|-----------------------|--|--|
| version | M | The CEE version used. | <ul style="list-style-type: none"> • R6 | If the parameter is not configured, the default value is R6. |

— [ecm] Section

This section is conditional. It is mandatory only if the cloud orchestrator used is ECM.

This section is not required for the Descriptor Generator Tool execution. It is required only for the scaling related activities and for decommissioning. Refer to *Configure Scale-Out*, *Configure Graceful Scale-In*, and *SAPC VNF Decommissioning Instruction for OpenStack*.



Table 6 SAPC.cfg [ecm] Section Parameters

| Attribute | Mandatory (M), Optional (O), Conditional (C) | Description | Default Value |
|------------------|--|---|--|
| version | M | The ECM version used. | If the parameter is not configured, the default value is 17.1 . Possible values: <ul style="list-style-type: none">• 16.0• 17.0• 17.1• 18.0 |
| url | M | The Ericsson Cloud Manager (ECM) hostname(FQDN) which is needed to compose the API URL offered by the ECM web service. | If the parameter is not configured, the execution of the tools for scalability and decommissioning ends. |
| tenant | M | The ECM tenant name. | If the parameter is not configured, the execution of the tools for scalability and decommissioning ends. |
| subtenant | O | The ECM subtenant name belonging to the previously defined tenant . Subtenants support is available from ECM 15.2-ICP1 and later versions. | If the parameter is not configured, no subtenant is used, only the tenant name. |
| user | M | The ECM username. | If the parameter is not configured, the execution of the tools for scalability and decommissioning ends. |
| password | O | The ECM password. | If the parameter is not configured, the user is prompted to provide the password, when needed. |

— [scaling] Section

This section is conditional. It is mandatory only if the manager used is ECM.

This section is not required for the Descriptor Generator Tool execution. It is required only for the scaling related activities and for decommissioning. Refer to [Configure Scale-Out](#), [Configure Graceful Scale-In](#), and [SAPC VNF Decommissioning Instruction for OpenStack](#).

Table 7 SAPC.cfg [scaling] Section Parameters

| Attribute | Mandatory (M), Optional (O), Conditional (C) | Description | Default Value |
|------------|--|---|--|
| vdc | M | The name of the ECM Virtual Data Center where the SAPC VNF is deployed. | If the parameter is not configured, the execution of the tools for scalability and decommissioning ends. |



| Attribute | Mandatory (M), Optional (O), Conditional (C) | Description | Default Value |
|-------------|--|--|--|
| vapp | M | The name of the ECM vAPP where the SAPC VNF is deployed. | If the parameter is not configured, the execution of the tools for scalability and decommissioning ends. |
| pls | M | The number of the SAPC PL Virtual Machines to scale-out or scale-in. | If the parameter is not configured, the default and strongly recommended value is 1 PL Virtual Machine. |
| azs | O | The availability zones configured for the number of pls to be scaled-out. | <p>The default value is empty (no availability zones configuration).</p> <p>Possible values:</p> <ul style="list-style-type: none"> The value can be left empty, which means that no availability zone is provided for the number of new pls to be scaled-out. Then the cloud infrastructure decides the availability zone to be used for one or more new Virtual Machines. The name of one availability zone (which matches with one already configured in the cloud infrastructure and manager) to be configured for the number of new pls to be scaled-out. In case that the pls parameter is configured to a greater value than 1, the same availability zone is applied to all the number of PayLoads to be scaled-out. The name of several availability zones (which match with several already configured in the cloud infrastructure and manager) to be configured for the number of new pls to be scaled-out. In case that the numbers of pls configured do not match with the number of azs provided, the execution ends. <p>Exception: if the deployment_type parameter in the [sapc] section is set to single_box (refer to Table 8 and Table 9), azs parameter is ignored and the PayLoads to be scaled-out are created using the same availability zone configured for the existing PayLoad Virtual Machines.</p> |

— [sapc] Section

This section is mandatory in the `SAPC.cfg` file.

The disk size defined, by default, for the SC Virtual Machines are 40 GB. This value cannot be modified.



Table 8 SAPC.cfg [sapc] Section Parameters

| Attribute | Mandatory (M), Optional (O), Conditional (C) | Description | Default Value |
|-----------------|--|---|---|
| deployment_type | M | Specifies the deployment type. See Table 9 to check its possible values. This field only applies to the SAPC on ecm_cee and openstack to determine the correct distribution of Virtual Machines on compute nodes (the definition of affinity_groups and anti_affinity_groups). For the SAPC on VMware platforms, the distribution is manually done from the cloud orchestrator. The default value is compact . | If the parameter is not configured, the default value is compact . |
| create_ova_file | O | This parameter only applies an OVF package is to be generated. If true , the output files are packaged in an OVA container (OVF package) ready to deploy the SAPC vAPP through the cloud orchestrator. | The recommended value (and default value if the parameter is not configured) is true . |
| sc_vcpus | M | The number of SC Virtual Machines vCPUs. | The default and minimal value is 2 vCPUs. |
| sc_mem | M | The amount of SC RAM in MB. | The default and minimal value is 6144MB . |
| sc_disk | M | The amount of SC disk in GB. | The default and minimal value is 40GB . If the parameter is configured to a value lower than 40GB , the tool automatically takes this minimal value for the generation. |
| initial_pls | M | The number of PL Virtual Machines for the initial deployment. Recommended, minimal and default value is 2 PL. | The default and minimal value is 2 PL Virtual Machines. The maximal number of PLs in the SAPC cluster is 70 . When more than 2 PLs are required, deploy only 2 and then, scale out to the maximum following the instruction in <i>Configure Scale-Out</i> . |
| pl_vcpus | M | The number of PL Virtual Machines vCPUs. | The default and minimal value is 2 vCPUs. The maximal value is 20 vCPUs. |
| pl_mem | M | The amount of PL RAM in MB. | The default and minimal value is 10240MB . |
| vr_vcpus | O | The number of VR Virtual Machines vCPUs. | The default and minimal value is 2 vCPUs. |
| vr_mem | O | The amount of VR RAM in MB. | The default and minimal value is 1024MB . |



| Attribute | Mandatory (M), Optional (O), Conditional (C) | Description | Default Value |
|------------------------|--|---|---|
| vms_prefix_name | O | The prefix to be added to every Virtual Machine name. Example of Virtual Machines names if the value is my_sapc_ : my_sapc_SC-1 , my_sapc_PL-3 , my_sapc_VR-4 . | If the parameter is not configured, the default value is an empty string. |
| scs_az | O | <p>This field only applies to the SAPC on OpenStack deployments.</p> <p>The availability zone where the SC Virtual Machines must be deployed.</p> | <p>If the parameter is not configured, and the default availability zone is the only one configured in the cloud, that default availability zone is considered by the cloud orchestrator.</p> <p>If the parameter is not configured, but there are several availability zones configured in the cloud, the cloud orchestrator requires to configure the parameter manually through its GUI.</p> <p>Configuration examples:</p> <ul style="list-style-type: none"> • No availability zone is provided for SC-1 and SC-2 (value is left empty): scs_az = . • If only one availability zone is configured, all the SC Virtual Machines are configured to be deployed in that same availability zone: scs_az = az1. Availability zone az1 is configured for both SC-1 and SC-2. • If each one of the two SC Virtual Machines is deployed in a certain availability zone, the two availability zones are defined semicolon separated: scs_az = az1;az2. Availability zone az1 is configured for SC-1 and availability zone az2 for SC-2. • If more than two availability zones are configured semicolon separated, the execution ends taking into account that there are only two SC Virtual Machines. <p>Exception: if the deployment_type parameter in the [sapc] section is set to single_box (refer to Table 8 and Table 9), no availability zones or just one for both SC-1 and SC-2 must be configured.</p> |



| Attribute | Mandatory (M), Optional (O), Conditional (C) | Description | Default Value |
|---------------------|--|---|--|
| <code>pls_az</code> | O | <p>This field only applies to the SAPC on OpenStack deployments.</p> <p>The availability zone where the PL Virtual Machines must be deployed.</p> | <p>If the parameter is not configured, and the default availability zone is the only one configured in the cloud, that default availability zone is considered by the cloud orchestrator.</p> <p>If the parameter is not configured, but there are several availability zones configured in the cloud, the cloud orchestrator requires to configure the parameter manually through its GUI.</p> <p>Configuration examples:</p> <ul style="list-style-type: none">• No availability zone is provided for the PL Virtual Machines (value is left empty): <code>pls_az = .</code>• If only one availability zone is configured, all the PL Virtual Machines are configured to be deployed in that same availability zone: <code>pls_az = az1</code>. Availability zone <code>az1</code> is configured for all the PL Virtual Machines: <code>PL-3</code>, <code>PL-4</code>, and also <code>PL-N</code>, in case that <code>pls</code> parameter is configured to a greater value than 2.• If each one of the PL Virtual Machines is deployed in a certain availability zone, several availability zones are defined semicolon separated. Considering that the <code>pls</code> parameter is set to the default and recommended value 2: <code>pls_az = az1;az2</code>. Availability zone <code>az1</code> is configured for <code>PL-3</code> and availability zone <code>az2</code> for <code>PL-4</code>. In case that the <code>pls</code> parameter is set to 6, for example: <code>pls_az = az1;az2;az3;az4;az5;az6</code> means that <code>az1</code> is configured for <code>PL-3</code>, <code>az2</code> for <code>PL-4</code>, <code>az3</code> for <code>PL-5</code>, <code>az4</code> for <code>PL-6</code>, <code>az5</code> for <code>PL-7</code> and <code>az6</code> for <code>PL-8</code>; or for example: <code>pls_az = az1;az2;az3;az1;az2;az3</code> means that <code>az1</code> is configured for <code>PL-3</code> and <code>PL-6</code>, <code>az2</code> for <code>PL-4</code> and <code>PL-7</code>, and <code>az3</code> for <code>PL-5</code> and <code>PL-8</code>.• If the number of availability zones configured semicolon separated does not match the <code>pls</code> parameter value, the execution ends. <p>Exception: if the <code>deployment_type</code> parameter in the <code>[sapc]</code> section is set to <code>single_box</code> (refer to Table 8 and Table 9), no availability zones or just one for all the PL must be configured.</p> |



| Attribute | Mandatory (M), Optional (O), Conditional (C) | Description | Default Value |
|-------------------|--|---|---|
| vrs_oam_az | O | <p>This field only applies to the SAPC on OpenStack deployments.</p> <p>The availability zone where the VR for OAM Virtual Machines must be deployed.</p> | <p>If the parameter is not configured, and the default availability zone is the only one configured in the cloud, that default availability zone is considered by the cloud orchestrator.</p> <p>If the parameter is not configured, but there are several availability zones configured in the cloud, the cloud orchestrator requires to configure the parameter manually through its GUI.</p> <p>Configuration examples:</p> <ul style="list-style-type: none"> • No availability zone is provided for VR-1 and VR-2 (value is left empty): vrs_oam_az = . • If only one availability zone is configured, all the VR OAM Virtual Machines are configured to be deployed in that same availability zone: vrs_oam_az = az1. Availability zone az1 is configured for both VR-1 and VR-2. • If each one of the two VR OAM Virtual Machines is deployed in a certain availability zone, the two availability zones are defined semicolon separated: vrs_oam_az = az1;az2. Availability zone az1 is configured for VR-1 and availability zone az2 for VR-2. • If more than two availability zones are configured semicolon separated, the execution ends taking into account that there are only two VR OAM Virtual Machines. <p>Exception: if the deployment_type parameter in the [sapc] section is set to single_box (refer to Table 8 and Table 9), no availability zones or just one for both VR-1 and VR-2 must be configured.</p> |



| Attribute | Mandatory (M), Optional (O), Conditional (C) | Description | Default Value |
|-----------------------|--|---|--|
| vrs_traffic_az | O | <p>This field only applies to the SAPC on OpenStack deployments.</p> <p>The availability zone where the VR for traffic Virtual Machines must be deployed.</p> | <p>If the parameter is not configured, and the default availability zone is the only one configured in the cloud, that default availability zone is considered by the cloud orchestrator.</p> <p>If the parameter is not configured, but there are several availability zones configured in the cloud, the cloud orchestrator requires to configure the parameter manually through its GUI.</p> <p>Configuration examples:</p> <ul style="list-style-type: none">• No availability zone is provided for VR-3 and VR-4 (value is left empty): vrs_traffic_az = .• If only one availability zone is configured, all the VR Traffic Virtual Machines are configured to be deployed in that same availability zone: vrs_traffic_az = az1. Availability zone az1 is configured for both VR-3 and VR-4.• If each one of the two VR Traffic Virtual Machines is deployed in a certain availability zone, the two availability zones are defined semicolon separated: vrs_traffic_az = az1;az2. Availability zone az1 is configured for VR-3 and availability zone az2 for VR-4.• If more than two availability zones are configured semicolon separated, the execution ends taking into account that there are only two VR Traffic Virtual Machines. <p>Exception: if the deployment_type parameter in the [sapc] section is set to single_box (refer to Table 8 and Table 9), no availability zones or just one for both VR-3 and VR-4 must be configured.</p> |



| Attribute | Mandatory (M), Optional (O), Conditional (C) | Description | Default Value |
|------------------------|--|---|---|
| affinity_groups | C | <p>This field only applies to the SAPC on OpenStack deployments and when the deployment_type parameter is set to custom.</p> <p>List of affinity groups semicolon separated. Each group defined is a comma-separated list of Virtual Machine names that are in the same affinity group. Multiple instances of parameter needed if multiple affinity groups required.</p> <p>The wild character * in Virtual Machine name must be supported to allow name expansion, for example, SC* matches all Virtual Machines with Virtual Machine name starting with SC.</p> <p>affinity_groups = * generates a single_box deployment.</p> <p>For the specific case of the Descriptor Generator Tool used to generate a HOT descriptor, if custom deployment is configured, then all the total number of possible PL virtual machines in the SAPC must be configured here (check the pls parameter detailed in this table), or just none of them (this is, empty affinity groups for the PL virtual machines), but it is not possible to configure only part of the total number of PLs.</p> | If the parameter is not configured, the default value is an empty list of groups. |



| Attribute | Mandatory (M), Optional (O), Conditional (C) | Description | Default Value |
|-----------------------------------|--|---|--|
| anti_affinity_groups | C | <p>This field only applies to the SAPC on OpenStack deployments and when the deployment_type parameter is set to custom.</p> <p>List of anti-affinity groups semicolon separated. Each group defined is a comma-separated list of Virtual Machine names that are in the same anti-affinity group. Multiple instances of parameter needed if multiple anti-affinity groups required.</p> <p>The wild character * in Virtual Machine name must be supported to allow name expansion, for example, SC* matches all Virtual Machines with Virtual Machine name starting with SC.</p> <p>anti_affinity_groups = * generates a multiple_host deployment.</p> <p>For the specific case of the Descriptor Generator Tool used to generate a HOT descriptor, if custom deployment is configured, then all the total number of possible PL virtual machines in the SAPC must be configured here (check the pls parameter detailed in this table), or just none of them (this is, empty anti-affinity groups for the PL virtual machines), but it is not possible to configure only part of the total number of PLs.</p> | If the parameter is not configured, the default value is: SC*;PL*;VR-1,VR-2;VR-3,VR-4 . |
| vnets_exist ⁽¹⁾ | C | <p>This field only applies to the SAPC on CEE deployments using ECM as cloud orchestrator.</p> <p>If the parameter is set to false, the virtual networks must be created in deployment time by the ECM. In case the virtual networks exist at ECM level, the value to be configured is true.</p> | If the parameter is not configured, the default value is false . |



| Attribute | Mandatory (M), Optional (O), Conditional (C) | Description | Default Value |
|---|--|---|---|
| external_networks ⁽¹⁾ | C | <p>This field only applies to HOT descriptors.</p> <p>Taking into account the number of external networks defined for the SAPC deployment (ExtOAM and ExtTraffic0 for the most basic scenario), it defines a semicolon separated list of network IDs or names for the external networks already created in the VIM as provider networks. The order in which the values are considered is like: ExtOAM, ExtTraffic0, ExtTraffic1, and so on.</p> <p>If this parameter is left empty, the HOT template creates the missing external networks in the infrastructure.</p> <p>external_networks =</p> <p>Example 1: the SAPC has two external networks already created in the VIM, one for OAM traffic with name ExtOAM0 and another one for payload traffic with name ExtTraffic0:</p> <p>external_networks = ExtOAM0;ExtTraffic0</p> <p>Example 2: the SAPC has three external networks already created in the VIM, one of them for OAM traffic and two of them for payload traffics. Their provider network IDs in OpenStack are respectively:</p> <p>ExtOAM <ID1> --> e31856ab-b3f4-405f-b135-0a0368afae4c</p> <p>ExtTraffic0 <ID2> --> 1eaa98c-c17e-461a-b9d4-8f554d86ef7f</p> <p>ExtTraffic1 <ID3> --> 4a159026-a292-41c0-85f6-53a1429e28c7</p> <p>Then, the configuration is:</p> <p>external_networks = <ID1>;<ID2>;<ID3></p> | If the parameter is not configured, the default value is empty (no provider network IDs or names are provided), and the SAPC external networks are created by the HOT template. |



| Attribute | Mandatory (M), Optional (O), Conditional (C) | Description | Default Value |
|-----------------------|--|--|---|
| ha-policy | C | <p>This field only applies to the SAPC on CEE R6 (through ECM or Atlas) deployments.</p> <p>If the parameter is set to unmanaged, no action is performed by CEE on this VM at Host recovery.</p> <p>If the parameter is set to managed-on-host, VM starts up with the host and shuts down with it. In case of failure, the VM is not moved to another host, but it is restarted, when the node is restarted.</p> <p>If the parameter is set to ha-offline (High Availability with offline migration), VM is evacuated in case of failure and moved to another host.</p> | <p>The default value is managed-on-host.</p> <p>If the parameter is set to ha-offline, it will not be applied to VMs acting as System Controller because of OpenStack limitations; for these VMs, it is internally set to managed-on-host instead.</p> |
| paravirtual | C | <p>This parameter is conditional to the use of vsphere or vcloud_director as target_cloud_system in SAPC.cfg.</p> <p>It allows to configure the virtual machine's disks to use VMware Paravirtual SCSI (PVSCSI) adapters.</p> <p>If the parameter is set to true, the VMware Paravirtual adapter is used.</p> <p>If the parameter is set to false, the LSI Logic Parallel adapter is used.</p> | The default value is false . |
| port_security | C | <p>This field applies only to the SAPC on OpenStack deployments. It sets port_security_enabled attribute value for every port created in OpenStack for the VNF. Possible values are true and false.</p> <p>Value true is supported only for CEE non-SDN deployments configured with IPv4.</p> | If the attribute is not configured, the default value is false . |
| security_group | C | <p>This field applies only to the SAPC on OpenStack deployments. It is used only when the port_security attribute is true. It sets the security-group attribute to target value for every port created in OpenStack for the VNF. The security-group target value is required for the vApp deployment in OpenStack NFVI.</p> | If the attribute is not configured, the default value is default . |

(1) If the networks already exist at deployment time, they are created according to the constraints explained in the eVIP Configuration Overview in SAPC VNF Network Configuration Guide.



Table 9 Possible Values for deployment_type in SAPC.cfg

| deployment_type ⁽¹⁾ Name | Description |
|-------------------------------------|---|
| compact | <p>Used for deployments with minimum possible servers and that have hardware redundancy. A redundant hardware system is defined as a system where one server can fail and users on other servers are not affected. Typically, multiple Virtual Machines execute on one server in a compact deployment to make the setup compact. This means that the application handles multiple Virtual Machine failures if there is a server outage without impacting users on other servers.</p> <p>As described in Page 17 and Page 18, the affinity and anti-affinity related parameters are internally calculated and defined to guarantee this kind of deployment, considering that the SAPC SC Virtual Machines (SC-1 and SC-2) have anti-affinity between them, the PL Virtual Machines (PL-3, PL-4, PL-N) have anti-affinity among them, the VR OAM Virtual Machines have anti-affinity between them (VR-1 and VR-2), and the VR Traffic Virtual Machines (VR-3 and VR-4) have anti-affinity between them.</p> <p>When a PL scale-out is executed, the new PL has anti-affinity with all the rest of SAPC PL Virtual Machines that belongs to the SAPC instance.</p> |
| single_box | <p>Deployment on a single physical server.</p> <p>Values described in Page 17 and Page 18 are internally calculated and defined to guarantee this kind of deployment, considering all the SAPC Virtual Machines in the same physical host.</p> <p>When a PL scale-out is executed, the new PL has affinity with all the rest of SAPC Virtual Machines that belongs to the SAPC instance.</p> |
| multiple_host | <p>Single Virtual Machine per physical server.</p> <p>Values described in Page 17 and Page 18 are internally calculated and defined to guarantee this kind of deployment, considering each SAPC Virtual Machine in a different physical host.</p> <p>When a PL scale-out is executed, the new PL has anti-affinity with all the rest of SAPC Virtual Machines that belongs to the SAPC instance.</p> |
| custom | <p>Custom deployment, using the user-defined affinity/anti-affinity settings. See Table 8 and check Page 17 and Page 18 for definitions.</p> <p>For the specific case of deployments over CEE using ECM, when a PL scale-out is executed, each new pair of scaled-out PLs has anti-affinity between them. Example: the SAPC initial deployment consists of 2 PLs (PL-3 and PL-4). Then, PL-5, PL-6, PL-7, and PL-8 are scaled-out. The result is that PL-5 and PL-6 have anti-affinity between them, and PL-7 and PL-8 have anti-affinity between them.</p> <p>For the specific case of deployments using Atlas/Director, the affinities and anti-affinities for all the possible PLs (the ones in the initial deployment and the ones to be added through a scale-out procedure) are defined in a static way in the HOT template to be generated for the initial deployment.</p> |

(1) This parameter, and therefore the whole information provided in this table, only applies to deployments over CEE, both using ECM or Atlas/Director, and over RHOSP.





4 SAPC VNF Descriptor Generation

The SAPC VNF Descriptor can be generated in two different formats:

- OVF descriptor for the SAPC deployments over:
 - CEE using ECM.
 - VMware platforms using VMware vSphere or VMware vCloud Director.
- HOT descriptor for the SAPC deployments over CEE and RHOSP using Atlas/Director. Additionally, a specific HOT descriptor is provided for ECM usage.

The Descriptor Generator Tool, when executed using as input the default `SAPC.cfg` file delivered inside the SAPC VDP, generates a default VNF Descriptor for a deployment consisting of:

- 2 Virtual Machines as System Controllers.
- 2 Virtual Machines as PayLoads.
- 4 Virtual Machines as Virtual Routers (optional).

In the same way, the `SAPC.cfg` configuration file contains minimum values for CPU, memory, and so on, for each of the previously mentioned Virtual Machines.

4.1 Package Generation

The executable `SAPC_descriptor_generator` (over x86-64 architecture) is the tool which generates the SAPC VNF Descriptor package. To execute it, the following arguments are available:

- **--help** or **-h**: if this argument is present, an explanation for every argument is shown and the execution finishes.
- **--version**: if this argument is present, the version is shown and the execution finishes.
- **--cfg <CFG>** or **-c <CFG>**: mandatory argument. It indicates where the `SAPC.cfg` configuration file is. This file is the input for the tool and it has been explained in Section 3.4 on page 7. The default `SAPC.cfg` configuration file is placed in the root directory of the VDP.
- **--vdp <VDP>**: this argument indicates the path where the VDP is stored in the file system.
- **--vnf-config <VNF_CONFIG>**: mandatory argument. It indicates the path where the `adapt_cluster_template.cfg` template file is stored in the file system.



- **--output**: optional argument. It indicates the path where the result package file is stored. If omitted, the current path is used.
- **--debug**: optional argument for debugging and troubleshooting purposes.
- **--log <LOG>**: optional argument to log the messages to a file instead of showing them into the standard output.

Then, an example of the `SAPC_descriptor_generator` execution to generate the OVF package is:

```
PreparationServer:# bin/SAPC_descriptor_generator --cfg SAPC.cfg  
--vdp ./ --vnf-config ./templates/adapt_cluster_template.cfg
```

The previous command considers that:

- It has been executed in a shell where the current directory is the VDP root directory.
- The `SAPC_descriptor_generator` executable file is placed in the `bin` folder under the root directory of the VDP.
- The `SAPC.cfg` config file is placed inside the root directory of the VDP.
- The `adapt_cluster_template.cfg` template file is placed in the `templates` folder under the root directory of the VDP.
- If an `initial_configuration.xml` file has been prepared, it is placed in the root directory of the VDP.
- If an `initial_provisioning.rest` file has been prepared, it is placed in the root directory of the VDP.

The tool will automatically select the appropriate output according to the input read from `SAPC.cfg`.

4.2 OVF Package Generation Output

This section details the output files generated by the Descriptor Generator Tool after its execution when creating an OVF package. The proper use of these output files is detailed during the deployment procedure.

4.2.1 ECM over CEE

After executing the Descriptor Generator Tool, the output is:

- The `SAPC_cxp9032849_<revision>.ova` OVF package. To get this OVF package, the parameter `create_ova_file` is set to `true` (default and recommended value) in the `SAPC.cfg` file used as input for the Descriptor Generator Tool. This file is mandatory and required to deploy the SAPC vAPP. It is the input for the cloud orchestrator to create **all** the necessary



resources (virtual networks, Virtual Machines, vNICs, and so on) in the cloud infrastructure. This OVF package contains internally:

- The `SAPC_cxp9032849_<revision>.ovf` OVF file. This file is mandatory to deploy the SAPC vAPP.
 - The `adapt_cluster.cfg` file. This file is mandatory to deploy the SAPC vAPP. It is generated by the Descriptor Generator Tool taking as input the parameters configured in the `adapt_cluster_template.cfg` file.
 - The `initial_configuration.xml` file if present when the Descriptor Generator Tool was executed. This file is optional.
 - The `initial_provisioning.rest` file if present when the Descriptor Generator Tool was executed. This file is optional.
 - The `SAPC_cxp9032849_<revision>.mf` manifest file (containing the SHA1 hash for all the previous files).
- The `ecm_VMHD_sapc_cxp9032849_<revision>.xml` file containing the needed flavors for the deployment when ECM 16.0 or older version is the cloud orchestrator. For those cases, this file is mandatory to deploy the SAPC vAPP. The values included in the resulting `SAPC_cxp9032849_<revision>.ovf` and `openstack_flavors_sapc_cxp9032849_<revision>.sh` files are fully aligned with this flavors file. See Section 5.5 on page 36 and Section 5.6 on page 37 to check an example of these files. From ECM 17.0 onwards, flavors are created on ECM GUI and pushed to the VIM.
- The `openstack_flavors_sapc_cxp9032849_<revision>.sh` script used to perform configuration related to flavors when CEE is used as cloud infrastructure. This file is mandatory to deploy the SAPC vAPP, unless ECM 17.0 or newer is the cloud orchestrator. The values included in the resulting `SAPC_cxp9032849_<revision>.ovf` and `ecm_VMHD_sapc_cxp9032849_<revision>.xml` files are fully aligned with the flavors that this script configures in the cloud infrastructure.

4.2.2 VMware vSphere Client

After executing the Descriptor Generator Tool, the output for deployments using VMware vSphere Client is.

- The `SAPC_cxp9032850_<revision>.ova` OVF package. To get this OVF package, the parameter `create_ova_file` is set to `true` (default and recommended value) in the `SAPC.cfg` file used as input for the Descriptor Generator Tool. This file is mandatory and required to deploy the SAPC vAPP. It is the input for the cloud orchestrator to create the necessary resources (Virtual Machines, vNICs, and so on) in the cloud infrastructure. This OVF package contains internally:
- The `SAPC_cxp9032850_<revision>.ovf` OVF file. This file is mandatory to deploy the SAPC vAPP.



- One ISO file for each virtual machine which requires it for internal customization:

- `adapt_cluster_SC-1.iso`

- `adapt_cluster_VR.iso`

The ISO file for the SC-1 is mandatory to deploy the SAPC vAPP. The ISO files for the VRs are only needed in case the SAPC deployment includes Virtual Routers. Each ISO image file contains internally:

- The `adapt_cluster.cfg` file. At least, this file must be contained in the ISO image file. This file is mandatory to deploy the SAPC vAPP. It is generated by the Descriptor Generator Tool taking as input the parameters configured in the `adapt_cluster_template.cfg` file.

- The `initial_configuration.xml` file if present when the Descriptor Generator Tool was executed. This file is optional.

- The `initial_provisioning.rest` file if present when the Descriptor Generator Tool was executed. This file is optional.

- The `SAPC_cxp9032850_<revision>.mf` manifest file (containing the SHA1 hash for all the files contained in the OVF package).

- The following VMDK files delivered inside the VDP. Their presence is required for a successful OVF package generation. These files are mandatory to deploy the SAPC vAPP since they are the disks for the Virtual Machines to be created.

- `sapc_sc_cxp9032850_<revision>.vmdk`

- `sapc_pxe_cxp9032850_<revision>.vmdk`

- `sapc_vr_cxp9032850_<revision>.vmdk`

4.2.3

VMware vCloud Director

After executing the Descriptor Generator Tool, the output for deployments using VMware vCloud Director are:

- The `SAPC_cxp9032850_<revision>.ova` OVF package. To get this OVF package, the parameter `create_ova_file` is set to `true` (default and recommended value) in the `SAPC.cfg` file used as input for the Descriptor Generator Tool. This file is mandatory and required to deploy the SAPC vAPP. It is the input for the cloud orchestrator to create the necessary resources (Virtual Machines, vNICs, and so on) in the cloud infrastructure. This OVF package contains internally:

- The `SAPC_cxp9032850_<revision>.ovf` OVF file. This file is mandatory to deploy the SAPC vAPP.



- The `SAPC_cxp9032850_<revision>.mf` manifest file (containing the SHA1 hash for all the files contained in the OVF package).
- The following VMDK files delivered inside the VDP. Their presence is required for a successful OVF package generation. These files are mandatory to deploy the SAPC vAPP since they are the disks for the Virtual Machines to be created.
 - `sapc_sc_cxp9032850_<revision>.vmdk`
 - `sapc_pxe_cxp9032850_<revision>.vmdk`
 - `sapc_vr-1_cxp9032850_<revision>.vmdk`
 - `sapc_vr-2_cxp9032850_<revision>.vmdk`
 - `sapc_vr-3_cxp9032850_<revision>.vmdk`
 - `sapc_vr-4_cxp9032850_<revision>.vmdk`
- The `adapt_cluster.iso` file. This file is mandatory to deploy the SAPC vAPP. This ISO image file contains internally:
 - The `adapt_cluster.cfg` file. At least, this file must be contained in the ISO image file. This file is mandatory to deploy the SAPC vAPP. It is generated by the Descriptor Generator Tool taking as input the parameters configured in the `adapt_cluster_template.cfg` file.
 - The `initial_configuration.xml` file if present when the Descriptor Generator Tool was executed. This file is optional.
 - The `initial_provisioning.rest` file if present when the Descriptor Generator Tool was executed. This file is optional.

The `adapt_cluster.iso` file is not contained inside the OVF package. It is uploaded to the VMware vCloud Director catalog in a separated way during the deployment procedure.
- The `SAPC_cxp9032850_<revision>_PL_ScaleOut_Template.ova` file. This file is mandatory to scale-out the SAPC vAPP. This OVF package contains internally:
 - The `SAPC_cxp9032850_<revision>_PL_ScaleOut_Template.ovf` OVF file. This file is mandatory to scale-out the SAPC vAPP.
 - The `SAPC_cxp9032850_<revision>_PL_ScaleOut_Template.mf` manifest file (containing the SHA1 hash for all the files contained in the OVF package).
 - The `sapc_pxe_cxp9032850_<revision>.vmdk` VMDK file for the PL virtual machines delivered inside the VDP. Its presence is required for a successful OVF package generation. This file is mandatory to scale-out the SAPC vAPP since it is the disk for the PL Virtual Machine to be created.



4.3 HOT Package Generation Output

This section details the output files generated by the Descriptor Generator Tool after its execution. The proper use of these output files is detailed during the deployment procedure.

4.3.1 OpenStack

After executing the Descriptor Generator Tool, the output for deployments with OpenStack CLI is:

- The `SAPC_cxp9032849_<revision>.zip` HOT package. This file is mandatory and required to deploy the SAPC stack. It is the input for the cloud orchestrator to create **all** the necessary resources (virtual networks, Virtual Machines, vNICs, and so on) in the cloud infrastructure. This HOT package contains internally (once the `.zip` file is extracted) a root directory `SAPC_cxp9032849_<revision>/`. Inside this directory, the following structure is created:
 - The `SAPC_cxp9032849_<revision>.yaml` HOT file in **yaml** format, which contains the general information about the stack (the SAPC) to be deployed and its associated resources to be created in the cloud infrastructure by the cloud orchestrator (Virtual Machines, ports, and so on).
 - The `Resources/` subdirectory which contains some additional subdirectories:
 - The `EnvironmentFiles/` subdirectory which contains a **yaml** file (`SAPC_cxp9032849_<revision>_params.yaml`) where some configurable parameters can be modified and adapted for a specific stack and to be aligned with the ones preconfigured in the VIM (for example the names for flavors and images) instead of using the values dumped by default by the Descriptor Generator Tool.
 - The `HotFiles/` subdirectory which contains a **yaml** file (`SAPC_cxp9032849_<revision>_pl.yaml`) which is referenced from the main **yaml** which defines the stack. This nested **yaml** file contains the definition of the resource group used for the PL Virtual Machines scalability.
 - The `Images/` subdirectory which contains all the **qcow2** files delivered inside the VDP. Their presence is required for a successful HOT package generation. These files are mandatory to deploy the SAPC stack since they are the disks for the Virtual Machines to be created.
 - `sapc_sc_cxp9032849_<revision>.qcow2`
 - `sapc_pxe_cxp9032849_<revision>.qcow2`
 - `sapc_vr_cxp9032849_<revision>.qcow2`



- The `UserConfigurationFiles/` subdirectory which contains the `adapt_cluster.cfg` file to be injected during the SAPC deployment.
 - `adapt_cluster.cfg`
- The `VnfdWrapperFiles/` subdirectory which contains the SAPC wrapper file for the implementation of the Or-Vnfm MANO interface. This file is used by the ECM Virtual Network Function Lifecycle Manager (VNF-LCM).
 - `VNFD_Wrapper_sapc.json`
- The `openstack_flavors_sapc_cxp9032849_<revision>.sh` script used to perform configuration related to flavors when OpenStack is used as cloud infrastructure. The flavors need to be created before deploying the SAPC stack. The values included by default in the resulting `SAPC_cxp9032849_<revision>_params.yaml` file are fully aligned with the flavors that this script configures in the cloud infrastructure. In case of RHOSP, flavors metadata is ignored.
- The `SAPC_cxp9032849_<revision>_ECM.zip` HOT package. This file is mandatory and required to deploy the SAPC stack towards the ECM application. This HOT package can be uploaded to the ECM for a later deployment of the SAPC and contains the following structure and files:
 - The `SAPC_cxp9032849_<revision>.yaml` HOT file in **yaml** format, which contains general information about the stack (the SAPC) to be deployed and its associated resources.
 - The `Resources/` subdirectory which contains some additional subdirectories:
 - The `EnvironmentFiles/` subdirectory which contains a **yaml** file (`SAPC_cxp9032849_<revision>_params.yaml`) where some configurable parameters can be modified and adapted for a specific stack and to be aligned with the ones preconfigured in the VIM.
 - The `HotFiles/` subdirectory which contains a **yaml** file (`SAPC_cxp9032849_<revision>_pl.yaml`) which is referenced from the main **yaml** which defines the stack. This nested **yaml** file contains the definition of the resource group used for the PL Virtual Machines scalability.
 - The `UserConfigurationFiles/` subdirectory which contains the `adapt_cluster.cfg` file to be injected during the SAPC deployment.
 - `adapt_cluster.cfg`
 - The `VnfdWrapperFiles/` subdirectory which contains the SAPC wrapper file for the implementation of the Or-Vnfm MANO interface. This file is used by the ECM VNF-LCM.
 - `VNFD_Wrapper_sapc.json`





5 SAPC VNF Descriptor Generator Tool Examples

5.1 Example of the adapt_cluster.cfg File

```
[Customer]
TIMEZONE = Europe/Madrid
NTP_SERVER_IP = 192.168.1.1

[Cluster]
INITIAL_PLS = 2

[Interface]
SC_IFACE_COUNT = 3
PL_IFACE_COUNT = 3
SC_MAC_PREFIX = 02:10:20:3C
PL_MAC_PREFIX = 02:10:40:3C

[Network]
SAPC_INT_SH_NETWORK = 172.16.100.0/24

OAM_FEE_NODE = 1 2
OAM_FEE_IFZ_INDEX = 2
OAM_FEE_NETWORK = 172.16.213.0/29

TRF_FEE_NODE = 3 4 5 6 7 8
TRF_FEE_IFZ_INDEX = 2
TRF_FEE_NETWORK = 172.16.113.0/28

OAM_VIP = 10.58.31.7 ALB_OAM
GX_VIP = 10.58.31.137 ALB_TRF
```

Example 1 The adapt_cluster.cfg File Generated from the adapt_cluster_template.cfg File



5.2 Example of the Initial Configuration File

```
<?xml version="1.0" encoding="UTF-8"?>
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
    <capability>urn:ietf:params:netconf:base:1.0</capability>
  </capabilities>
</hello>
]]>]]>
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="1" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running />
    </target>
    <config>
      <ManagedElement xmlns="urn:com:ericsson:ecim:ComTop">
        <managedElementId>1</managedElementId>
        <dnPrefix>dc=ManagedElement</dnPrefix>
        <networkManagedElementId>1</networkManagedElementId>
        <userLabel>Managed Element</userLabel>
        <PolicyControlFunction xmlns="urn:com:ericsson:ecim:sapcmom">
          <policyControlFunctionId>1</policyControlFunctionId>
          <Network xmlns="urn:com:ericsson:ecim:networkmom">
            <networkId>1</networkId>
            <DiameterNodes>
              <diameterNodesId>1</diameterNodesId>
              <DiameterNode xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" nc:operation="m">
                <diameterNodeId>vEPC</diameterNodeId>
                <clusterPattern>ericsson.se</clusterPattern>
                <controls>IP_CAN_SESSION_ACCESS</controls>
                <controls>SERVICE_ACCESS_PCRF_TOD</controls>
                <controls>BEARER_QOS</controls>
                <controls>CONTENT_FILTERING</controls>
                <controls>SUBSCRIBER_CHARGING</controls>
                <controls>SERVICE_CHARGING</controls>
                <controls>USAGE_REPORTING</controls>
                <dynamicServiceSupport>1</dynamicServiceSupport>
              </DiameterNode>
            </DiameterNodes>
          </Network>
          <AppConfig xmlns="urn:com:ericsson:ecim:appconfigmom">
            <appConfigId>1</appConfigId>
            <PccConfig xmlns="urn:com:ericsson:ecim:mobileconfigmom" xmlns:nc="urn:ietf:params:">
              <pccConfigId>1</pccConfigId>
              <subsIdType>MSISDN</subsIdType>
            </PccConfig>
          </AppConfig>
        </PolicyControlFunction>
      </ManagedElement>
    </config>
  </edit-config>
</rpc>
]]>]]>
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="3" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <close-session/>
</rpc>
]]>]]>
```

Example 2 The initial_configuration.xml File

5.3 Example of the Initial Provisioning File

PUT /rules/rule_QoS_trottled

```
{
  "condition" : "((AccessData.subscriber.accumulatedUsage.reportingGroup[\"total\"]).isLimits
  "outputAttributes" :
  [
    {
```



```

        "attrName" : "min-qos",
        "attrValue" : "BearerQosProfile[\\"Qos_ThrottledBearer\\"]",
        "result" : "permit"
    },
    {
        "attrName" : "max-qos",
        "attrValue" : "BearerQosProfile[\\"Qos_ThrottledBearer\\"]",
        "result" : "permit"
    }
],
"ruleName" : "rule_QoS_trottled"
}

PUT /rules/rule_QoS1
{
    "condition" : "(1==1)",
    "outputAttributes" :
    [
        {
            "attrName" : "min-qos",
            "attrValue" : "BearerQosProfile[\\"Qos_DefaultBearer\\"]",
            "result" : "permit"
        },
        {
            "attrName" : "max-qos",
            "attrValue" : "BearerQosProfile[\\"Qos_DefaultBearer\\"]",
            "result" : "permit"
        }
    ],
    "ruleName" : "rule_QoS1"
}

PUT /policies/QoS_throttled
{
    "policyName" : "QoS_throttled",
    "ruleCombiningAlgorithm" : "permit-overrides",
    "rules" : [ "rule_QoS_trottled", "rule_QoS1" ]
}

PUT /dataplan/GroupFUP/locators/resources/ip-can-session/contexts/qos
{
    "policies" : [ "QoS_throttled" ]
}

PUT /dataplan/GroupFUP
{
    "dataplanName" : "GroupFUP",
    "usageLimits" :
    [
        {
            "absoluteLimits" :
            {
                "dlVolume" : [ 10240 ],
                "resetPeriod" :
                {
                    "volume" : "monthly"
                }
            },
            "description" : "Total traffic Limit",
            "name" : ""
        }
    ]
}

PUT /dataplan/GroupThrottled
{
    "dataplanName" : "GroupThrottled",

```



```
        "staticQualification" :
        {
            "maxBearerQosProfileId" : "Qos_ThrottledBearer",
            "minBearerQosProfileId" : "Qos_ThrottledBearer"
        }
    }

PUT /dataplan/MWCGroup
{
    "dataplanName" : "MWCGroup",
    "staticQualification" :
    {
        "maxBearerQosProfileId" : "Qos_DefaultBearer",
        "minBearerQosProfileId" : "Qos_DefaultBearer"
    }
}

PUT /profiles/ip-can-session-qos/Qos_DefaultBearer
{
    "arpPci" : true,
    "arpPriorityLevel" : 3,
    "arpPvi" : false,
    "mbrDownlink" : 5120,
    "mbrUplink" : 512,
    "profileId" : "Qos_DefaultBearer",
    "qci" : 6
}

PUT /profiles/ip-can-session-qos/Qos_ThrottledBearer
{
    "arpPci" : true,
    "arpPriorityLevel" : 3,
    "arpPvi" : false,
    "mbrDownlink" : 512,
    "mbrUplink" : 256,
    "profileId" : "Qos_ThrottledBearer",
    "qci" : 6
}

PUT /subscribers/46701940410
{
    "dataplan" :
    [
        {
            "dataplanName" : "GroupFUP"
        }
    ],
    "subscriberId" : "46701940410"
}

PUT /subscribers/46701940418
{
    "dataplan" :
    [
        {
            "dataplanName" : "GroupFUP"
        }
    ],
    "subscriberId" : "46701940418"
}

PUT /subscribers/46701940462
{
    "dataplan" :
```




```
[
    {
        "dataplanName" : "GroupFUP"
    },
    "subscriberId" : "46701940462"
]
```

Example 3 The initial_provisioning.rest File

5.4 Example of the SAPC.cfg Configuration File

Configuration for SAPC config (OVF, HOT, etc.) generation

```
[cloud]
target_cloud_system = openstack

[vsphere]
version = 6.5

[vcloud_director]
version = 8
url =
organization =
catalog =
template =
user =
password =
vApp

[cee]
version = R6

[ecm]
version = 17.1
url = sapc03ecm.eld.es.eu.ericsson.se
tenant = ECM
subtenant =
user = username
password = password

[scaling]
vdc = Virtual_Data_Center_003
vapp = SAPC
pls = 1
azs =

[sapc]
deployment_type = compact

create_ova_file = true

sc_vcpus = 2
sc_mem = 6144
sc_disk = 40

pl_vcpus = 2
pl_mem = 10240

vr_vcpus = 2
vr_mem = 1024

initial_pls = 2

vms_prefix_name =

scs_az =
pls_az =
vrs_oam_az =
vrs_traffic_az =
```



```

affinity_groups =
anti_affinity_groups = SC*;PL*;VR-1,VR-2;VR-3,VR-4

vnets_exist = false

external_networks =

ha_policy = managed-on-host

paravirtual = false

```

Example 4 The SAPC.cfg File

5.5 Example of the ECM Flavors XML File

```

<?xml version="1.0"?>
<InstanceData>
  <DataItems>
    <Data>
      <type>Virtual Machine Hardware Definition</type>
      <name>2vcpu_6144MBmem_40GBdisk</name>
      <provisioningStatus>ACTIVE</provisioningStatus>
      <Udas>
        <Uda>
          <group>VMHD Attributes</group>
          <name>VMHD Name</name>
          <value>2vcpu_6144MBmem_40GBdisk</value>
        </Uda>
        <Uda>
          <group>VMHD Attributes</group>
          <name>Disk Size</name>
          <value>40 GB</value>
        </Uda>
        <Uda>
          <group>VMHD Attributes</group>
          <name>Number Of CPUs</name>
          <value>2</value>
        </Uda>
        <Uda>
          <group>VMHD Attributes</group>
          <name>RAM</name>
          <value>6144 MB</value>
        </Uda>
      </Udas>
    </Data>
    <Data>
      <type>Virtual Machine Hardware Definition</type>
      <name>2vcpu_10240MBmem_0GBdisk</name>
      <provisioningStatus>ACTIVE</provisioningStatus>
      <Udas>
        <Uda>
          <group>VMHD Attributes</group>
          <name>VMHD Name</name>
          <value>2vcpu_10240MBmem_0GBdisk</value>
        </Uda>
        <Uda>
          <group>VMHD Attributes</group>
          <name>Disk Size</name>
          <value>0 GB</value>
        </Uda>
        <Uda>
          <group>VMHD Attributes</group>
          <name>Number Of CPUs</name>
          <value>2</value>
        </Uda>
        <Uda>
          <group>VMHD Attributes</group>
          <name>RAM</name>
          <value>10240 MB</value>
        </Uda>
      </Udas>
    </Data>
  </DataItems>
</InstanceData>

```



```

</Data>
<Data>
  <type>Virtual Machine Hardware Definition</type>
  <name>2vcpu_1024MBmem_4GBdisk</name>
  <provisioningStatus>ACTIVE</provisioningStatus>
  <Udas>
    <Uda>
      <group>VMHD Attributes</group>
      <name>VMHD Name</name>
      <value>2vcpu_1024MBmem_4GBdisk</value>
    </Uda>
    <Uda>
      <group>VMHD Attributes</group>
      <name>Disk Size</name>
      <value>4 GB</value>
    </Uda>
    <Uda>
      <group>VMHD Attributes</group>
      <name>Number Of CPUs</name>
      <value>2</value>
    </Uda>
    <Uda>
      <group>VMHD Attributes</group>
      <name>RAM</name>
      <value>1024 MB</value>
    </Uda>
  </Udas>
</Data>
</DataItems>
</InstanceData>

```

Example 5 The ecm_VMHD_sapc_cxp9032849_<revision>.xml File



5.6 Example of the Script to Upload Flavors in OpenStack

```
#!/bin/bash

function createFlavor
{
    name="$1"
    mem="$2"
    disk="$3"
    vcpus="$4"

    nova flavor-list | grep -i ${name} > /dev/null
    if [ $? -eq 1 ]
    then
        nova flavor-create ${name} auto ${mem} ${disk} ${vcpus} > /dev/null
        if [ $? -eq 0 ]
        then
            echo "Flavor ${name} successfully created."
        else
            echo "Error creating flavor ${name}."
            echo "Execution aborted."
            exit 1
        fi
    else
        echo "Flavor ${name} already exists in the system."
    fi

    nova flavor-key ${name} set hw:mem_page_size='1048576' > /dev/null
    if [ $? -eq 0 ]
    then
        echo "Flavor ${name} successfully modified: hw:mem_page_size is 1048576."
    else
        echo "Error modifying flavor ${name}: hw:mem_page_size not set to 1048576."
        echo "Execution aborted."
        exit 1
    fi

    nova flavor-key ${name} set hw:cpu_policy='dedicated' > /dev/null
    if [ $? -eq 0 ]
    then
        echo "Flavor ${name} successfully modified: hw:cpu_policy is dedicated."
    else
        echo "Error modifying flavor ${name}: hw:cpu_policy not set to dedicated."
        echo "Execution aborted."
        exit 1
    fi

    nova flavor-key ${name} set hw:watchdog_action='reset' > /dev/null
    if [ $? -eq 0 ]
    then
        echo "Flavor ${name} successfully modified: hw:watchdog_action is reset."
    else
        echo "Error modifying flavor ${name}: hw:watchdog_action not set to reset."
        echo "Execution aborted."
        exit 1
    fi
fi

createFlavor 2vcpu_6144MBmem_40GBdisk 6144 40 2
createFlavor 2vcpu_10240MBmem_0GBdisk 10240 0 2
createFlavor 2vcpu_1024MBmem_4GBdisk 1024 4 2
```

Example 6 The openstack_flavors_sapc_cxp9032849_<revision>.sh File