

Decrease Capacity with Heat Orchestration

OPERATING INSTRUCTIONS

Copyright

© Ericsson AB 2018, 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	Decrease Capacity with Heat Orchestration	1
2.2	Configure Graceful Scale-In with Heat Orchestration	2
2.3	Configure Forceful Scale-In with Heat Orchestration	8



Decrease Capacity with Heat Orchestration



1 Description

This instruction describes how to decrease the Call Session Control Function (CSCF) cluster capacity by performing a graceful scale-in operation with heat orchestration, which removes a Virtual Machine (VM) from the cluster.



Attention!

Risk of data loss or data corruption.

Do not use this procedure to remove resources created manually by openstack commands (**nova**, **neutron**), or from Horizon or Atlas Dashboard as it can corrupt the database of openstack. When openstack is used to create a VNF instance, do not use Horizon, the Atlas Dashboard, or the openstack commands **nova** or **neutron** for the scaling. To repair a faulty resource of a heat stack, use the following openstack commands on the stack: `openstack stack check`, `openstack stack resource list`, and `openstack stack update`. Detailed descriptions of these procedures are beyond the scope of this instruction. For more information about openstack, see [OpenStackClient](#).

This document always refers to horizontal scaling, where the scalability of the system is provided by multiple instances to distribute the load in parallel for having the capacity needed. Vertical scaling is not considered in this document.

The scaling function does not require a license.

Note: Even though the PL-3 and PL-4 Virtual Machines (VMs) are considered to be part of the scaling domain, they cannot be scaled in.

2 Procedure

2.1 Decrease Capacity with Heat Orchestration

Prerequisites

— This instruction references the following documents:

- CSCF Health Check



- Ericsson Command-Line Interface User Guide
- No tools are required.
- The following conditions must apply:
 - The procedure must only be performed by support personnel with experience of Cloud and the CSCF.
 - No other upgrade or maintenance activity must be performed during the procedure.
 - Before starting these procedures, the user performing the operations must have access to the System Controller (SC) nodes.
 - Signaling Manager Command-Line Interface (CLI) or Graphical User Interface (GUI) must be closed before the start of the Scaling Operations. Manual updates of the configurations during Scaling Operations are not allowed.
 - A Virtual Infrastructure Manager (VIM) is available.
 - An Ericsson Command-Line Interface (ECLI) session in Exec mode is in progress.

Steps

1. Select an action based on the type of scale-in:
 - Decrease capacity through graceful scale-in, see Section 2.2 Configure Graceful Scale-In with Heat Orchestration on page 2.
 - Decrease capacity through forceful scale-in, see Section 2.3 Configure Forceful Scale-In with Heat Orchestration on page 8.

2.2 Configure Graceful Scale-In with Heat Orchestration



Attention!

Risk of data loss or data corruption.

Do not use this procedure to remove resources created manually by openstack commands (**nova**, **neutron**), or from Horizon or Atlas Dashboard as it can corrupt the database of openstack. When openstack is used to create a VNF instance, do not use Horizon, the Atlas Dashboard, or the openstack commands **nova** or **neutron** for the scaling. To repair a faulty resource of a heat stack, use the following openstack commands on the stack: `openstack stack check`,



`openstack stack resource list`, and `openstack stack update`. Detailed descriptions of these procedures are beyond the scope of this instruction. For more information about openstack, see [OpenStackClient](#).

Steps

1. Before any scaling-related activities are performed, create a system backup. See [Create Backup](#).
2. Check that the status of the CSCF stack is `CREATE_COMPLETE` or `UPDATE_COMPLETE`:

```
openstack stack list
```

If the status of the stack is not `CREATE_COMPLETE` or `UPDATE_COMPLETE`, stop the scaling procedure. For information on how to identify and correct the stack status, see the VIM documentation.

3. Check that the cluster is in a healthy state, see [CSCF Health Check](#).
4. Navigate to the CrM MO, for example:

```
>dn ManagedElement=1,SystemFunctions=1,SysM=1,CrM=1
```

5. Verify that the VM to be scaled in is scalable:

```
(CrM=1)>show -r
```

The following is an example output:



```

CrM=1
  autoRoleAssignment=ENABLED
  ComputeResourceRole=PL-3
    adminState=UNLOCKED
    instantiationState=INSTANTIATED
    operationalState=ENABLED
    provides="ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, Role=PLs"
    uses="ManagedElement=1, Equipment=1, ComputeResource=PL-3"
  ComputeResourceRole=PL-4
    adminState=UNLOCKED
    instantiationState=INSTANTIATED
    operationalState=ENABLED
    provides="ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, Role=PLs"
    uses="ManagedElement=1, Equipment=1, ComputeResource=PL-4"
  ComputeResourceRole=SC-1
    adminState=UNLOCKED
    instantiationState=INSTANTIATED
    operationalState=ENABLED
    provides="ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, Role=SCs"
    uses="ManagedElement=1, Equipment=1, ComputeResource=SC-1"
  ComputeResourceRole=SC-2
    adminState=UNLOCKED
    instantiationState=INSTANTIATED
    operationalState=ENABLED
    provides="ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, Role=SCs"
    uses="ManagedElement=1, Equipment=1, ComputeResource=SC-2"
  ComputeResourceRole=PL-7
    adminState=UNLOCKED
    instantiationState=INSTANTIATED
    operationalState=ENABLED
    provides="ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, Role=PLs"
    uses="ManagedElement=1, Equipment=1, ComputeResource=PL-7"
  ComputeResourceRole=PL-8
    adminState=UNLOCKED
    instantiationState=INSTANTIATED
    operationalState=ENABLED
    provides="ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, Role=PLs"
    uses="ManagedElement=1, Equipment=1, ComputeResource=PL-8"
  ComputeResourceRole=PL-6
    adminState=UNLOCKED
    instantiationState=INSTANTIATED
    operationalState=ENABLED
    provides="ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, Role=PLs"
    uses="ManagedElement=1, Equipment=1, ComputeResource=PL-6"
  ComputeResourceRole=PL-5
    adminState=UNLOCKED
    instantiationState=INSTANTIATED
    operationalState=ENABLED
    provides="ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, Role=PLs"
    uses="ManagedElement=1, Equipment=1, ComputeResource=PL-5"
  Role=SCs
    isProvidedBy
      "ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, ComputeResourceRole=SC-1"
      "ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, ComputeResourceRole=SC-2"
    scalability=NON_SCALABLE
  Role=PLs
    isProvidedBy
      "ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, ComputeResourceRole=PL-3"
      "ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, ComputeResourceRole=PL-4"
      "ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, ComputeResourceRole=PL-5"
      "ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, ComputeResourceRole=PL-6"
      "ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, ComputeResourceRole=PL-7"
      "ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, ComputeResourceRole=PL-8"
    scalability=SCALABLE

```

- Retrieve the Universally Unique Identifier (UUID) for the VM to be scaled in and write it down for use in Step a of Step 17, for example:

```

(CrM=1)>show ManagedElement=1,Equipment=1,\
ComputeResource=PL-8

```

The following is an example output:



```
ComputeResource=PL-8
macAddress
  "fa:16:3e:5f:a7:2b"
  "fa:16:3e:b7:5c:45"
  "fa:16:3e:8e:f0:70"
  "fa:16:3e:bf:ef:94"
uuid="a4dcda89-cf95-4bf6-81bd-99d47fde9eef"
```

7. Navigate to the `ComputeResourceRole` MO for the VM to be scaled in, for example:

```
(CrM=1)>ComputeResourceRole=PL-8
```

8. Enter Config mode:

```
(ComputeResourceRole=PL-8)>configure
```

9. Prepare the scale-in operation:

```
(config-ComputeResourceRole=PL-8)>no provides
```

10. Navigate to the `CrM` MO:

```
(config-ComputeResourceRole=PL-8)>up
```

11. Perform the scale-in:

```
(config-CrM=1)>commit
```

Note: To cancel the scale-in, run `abort`.

12. Verify that the scale-in process has started:

```
(CrM=1)>show -r
```

The following is an example output:

```
ComputeResourceRole=PL-8
adminState=SHUTTINGDOWN
instantiationState=UNINSTANTIATING
operationalState=ENABLED
uses="ManagedElement=1,Equipment=1,ComputeResource=PL-8"
```

13. If a failure occurs during the scale-in, see [CSCF Troubleshooting Guideline](#).

14. Verify that the VM is scaled in:

```
(CrM=1)>show -r
```

The following is an example output showing that the VM `ComputeResourceRole=PL-8` is no longer running:



```

CrM=1
  autoRoleAssignment=ENABLED
  ComputeResourceRole=PL-3
    adminState=UNLOCKED
    instantiationState=INSTANTIATED
    operationalState=ENABLED
    provides="ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, Role=PLs"
    uses="ManagedElement=1, Equipment=1, ComputeResource=PL-3"
  ComputeResourceRole=PL-4
    adminState=UNLOCKED
    instantiationState=INSTANTIATED
    operationalState=ENABLED
    provides="ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, Role=PLs"
    uses="ManagedElement=1, Equipment=1, ComputeResource=PL-4"
  ComputeResourceRole=SC-1
    adminState=UNLOCKED
    instantiationState=INSTANTIATED
    operationalState=ENABLED
    provides="ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, Role=SCs"
    uses="ManagedElement=1, Equipment=1, ComputeResource=SC-1"
  ComputeResourceRole=SC-2
    adminState=UNLOCKED
    instantiationState=INSTANTIATED
    operationalState=ENABLED
    provides="ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, Role=SCs"
    uses="ManagedElement=1, Equipment=1, ComputeResource=SC-2"
  ComputeResourceRole=PL-7
    adminState=UNLOCKED
    instantiationState=INSTANTIATED
    operationalState=ENABLED
    provides="ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, Role=PLs"
    uses="ManagedElement=1, Equipment=1, ComputeResource=PL-7"
  ComputeResourceRole=PL-6
    adminState=UNLOCKED
    instantiationState=INSTANTIATED
    operationalState=ENABLED
    provides="ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, Role=PLs"
    uses="ManagedElement=1, Equipment=1, ComputeResource=PL-6"
  ComputeResourceRole=PL-5
    adminState=UNLOCKED
    instantiationState=INSTANTIATED
    operationalState=ENABLED
    provides="ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, Role=PLs"
    uses="ManagedElement=1, Equipment=1, ComputeResource=PL-5"
  Role=SCs
    isProvidedBy
      "ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, ComputeResourceRole=SC-1"
      "ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, ComputeResourceRole=SC-2"
    scalability=NON_SCALABLE
  Role=PLs
    isProvidedBy
      "ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, ComputeResourceRole=PL-3"
      "ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, ComputeResourceRole=PL-4"
      "ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, ComputeResourceRole=PL-5"
      "ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, ComputeResourceRole=PL-6"
      "ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, ComputeResourceRole=PL-7"
    scalability=SCALABLE

```

15. If more VMs need to be scaled in, repeat Step 6 to Step 14.

16. Check the value of the parameter `number_of_scaled_out_PL_VMs`:

```

openstack stack show <CSCF stack name> | \
grep number_of_scaled_out_PL_VMs

```

The following is an example output:

```

"number_of_scaled_out_PL_VMs": "4"

```

17. Find the value for parameter `PL_to_be_scaled_in`:



- a. Check the UUID that is retrieved in Step 6, and write down its position:

```
openstack stack show <CSCF stack name>
```

In the following example, `output_value` of `scaled_VMs_UUID` shows that the position of the example UUID `a4dcda89-cf95-4bf6-81bd-99d47fde9eef` that is retrieved in Step 6 is the second in the list:

```
| outputs | - description: MAC address of VM SC-1 vNIC eth0
|         |   output_key: SC-1_eth0_mac_address
|         |   output_value: fa:16:3e:8f:03:15
|         | - description: MAC address of VM PL-3 vNIC eth0
|         |   output_key: PL-3_eth0_mac_address
|         |   output_value: fa:16:3e:17:4c:9e
|         | - description: UUIDs of the PL VMs in the ResourceGroup
|         |   output_key: scaled_VMs_UUID
|         |   output_value: 55ab4245-8b18-49ca-8ded-1cfca49a4d89 \
|         |                   a4dcda89-cf95-4bf6-81bd-99d47fde9eef \
|         |                   baa95296-b433-4801-a2fc-d6ea23520287 \
|         |                   9ab5e3a4-aa31-4bc4-8e0a-d31f7f28c138
```

- b. Retrieve the names of the scaled out VMs from the stack, using one of the following options:

- After a Maiden Installation of the CSCF:

```
openstack stack show <CSCF stack name> | \
grep <CSCF vnf name>_VM
```

The following is an example output:

```
"output_value": "cscf1_VM-0 cscf1_VM-9 cscf1_VM-10 \
cscf1_VM-11"
```

- After an Upgrade of the CSCF:

```
openstack stack show <CSCF stack name> | \
grep <CSCF stack name>_scaled_out_VM
```

The following is an example output:

```
"output_value": "cscf-104_scaled_out_VM-0 \
cscf-104_scaled_out_VM-9 cscf-104_scaled_out_VM-10 \
cscf-104_scaled_out_VM-11"
```

- c. Find the name of the VM to be scaled in at the position from Step a in the list from Step b, and write down the number at the end of the VM name.

The number is the index of the VM and is used as the value of parameter `PL_to_be_scaled_in`. For example, write down 9 that is the number at the end of the VM names in the example from Step b.



Note: The index starts with 0.

18. If multiple VMs are scaled in, repeat Step 17.

19. Update the stack:

— For non-Cinder Environment:

```
openstack stack update <CSCF stack name> \
-t vcscf_hot.yaml -e vcscf_env.yaml --parameter \
number_of_scaled_out_PL_VMs=<number_of_scaled_out_PL_VMs> \
--parameter PL_to_be_scaled_in=<index of VMs>
```

— For Cinder-Supported Environment:

```
openstack stack update <CSCF stack name> \
-t vcscf_hot_sio.yaml -e vcscf_env.yaml --parameter \
number_of_scaled_out_PL_VMs=<number_of_scaled_out_PL_VMs> \
--parameter PL_to_be_scaled_in=<index of VMs>
```

The value for the parameter `number_of_scaled_out_PL_VMs` is the value obtained in Step 16 minus the number of VMs to scale in. The value for the parameter `PL_to_be_scaled_in` is a comma-separated list of the values obtained in Step 17 (repeat Step 17 for multiple VMs to scale in).

According to the example in Step a and Step b, the parameter `PL_to_be_scaled_in` is set to 9.

20. Check that the cluster is in a healthy state, see [CSCF Health Check](#).

2.3 Configure Forceful Scale-In with Heat Orchestration



Attention!

Risk of data loss or data corruption.

Do not use this procedure to remove resources created manually by openstack commands (**nova**, **neutron**), or from Horizon or Atlas Dashboard as it can corrupt the database of openstack. To repair a faulty resource of a heat stack, use the following openstack commands on the stack: `openstack stack check`, `openstack stack resource list`, and `openstack stack update`. Detailed descriptions of these procedures are beyond the scope of this instruction. For more information about openstack, see [OpenStackClient](#).

Steps



1. Before any scaling-related activities are performed, create a system backup. See [Create Backup](#).
2. Check that the status of the CSCF stack is CREATE_COMPLETE or UPDATE_COMPLETE:

```
openstack stack list
```

If the status of the stack is not CREATE_COMPLETE or UPDATE_COMPLETE, stop the scaling procedure. For information on how to identify and correct the stack status, see the VIM documentation.

3. Check that the cluster is in a healthy state, see [CSCF Health Check](#).
4. Check the value of the parameter number_of_scaled_out_PL_VMs:

```
openstack stack show <CSCF stack name> | \
grep number_of_scaled_out_PL_VMs
```

The following is an example output:

```
"number_of_scaled_out_PL_VMs": "4"
```

5. Update the stack:

— For non-Cinder Environment:

```
openstack stack update -t vcscf_hot.yaml \
-e vcscf_env.yaml <CSCF stack name> --parameter \
number_of_scaled_out_PL_VMs=<number_of_scaled_out_PL_VMs>
```

— For Cinder-Supported Environment:

```
openstack stack update -t vcscf_hot_sio.yaml \
-e vcscf_env.yaml <CSCF stack name> --parameter \
number_of_scaled_out_PL_VMs=<number_of_scaled_out_PL_VMs>
```

The value for the parameter number_of_scaled_out_PL_VMs is the value obtained in Step 4 minus the number of VMs to scale in.

6. Navigate to the CrM MO, for example:
7. Identify the ComputeResourceRole where the adminState is LOCKED and operationalState is DISABLED:

```
(CrM=1)>show -r
```

The following is an example output:



```
ComputeResourceRole=PL-8
  adminState=LOCKED
  instantiationState=INSTANTIATED
  operationalState=DISABLED
  provides="ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, Role=PLs"
  uses="ManagedElement=1, Equipment=1, ComputeResource=PL-8"
```

8. Enter Config mode:

```
(ComputeResourceRole=PL-8)>configure
```

9. Prepare the scale-in operation:

```
(config-ComputeResourceRole=PL-8)>no provides
```

10. Navigate to the CrM MO:

```
(config-ComputeResourceRole=PL-8)>up
```

11. Perform the scale-in:

```
(config-CrM=1)>commit
```

12. Verify that the scaling-in process has started:

```
(CrM=1)>show -r
```

The following is an example output:

```
CrM=1
[...]
```

ComputeResourceRole=PL-8
adminState=SHUTTINGDOWN
instantiationState=UNINSTANTIATING

```
[...]
```

13. If a failure occurs during the scale-in, see [CSCF Troubleshooting Guideline](#).

14. Verify that the VM is scaled in:

```
(CrM=1)>show -r
```

The following is an example output showing that the VM ComputeResourceRole=PL-8 is no longer running:



```
CrM=1
autoRoleAssignment=ENABLED
ComputeResourceRole=PL-3
  adminState=UNLOCKED
  instantiationState=INSTANTIATED
  operationalState=ENABLED
  provides="ManagedElement=1,SystemFunctions=1,SysM=1,CrM=1,Role=PLs"
  uses="ManagedElement=1,Equipment=1,ComputeResource=PL-3"
ComputeResourceRole=PL-4
  adminState=UNLOCKED
  instantiationState=INSTANTIATED
  operationalState=ENABLED
  provides="ManagedElement=1,SystemFunctions=1,SysM=1,CrM=1,Role=PLs"
  uses="ManagedElement=1,Equipment=1,ComputeResource=PL-4"
ComputeResourceRole=SC-1
  adminState=UNLOCKED
  instantiationState=INSTANTIATED
  operationalState=ENABLED
  provides="ManagedElement=1,SystemFunctions=1,SysM=1,CrM=1,Role=SCs"
  uses="ManagedElement=1,Equipment=1,ComputeResource=SC-1"
ComputeResourceRole=SC-2
  adminState=UNLOCKED
  instantiationState=INSTANTIATED
  operationalState=ENABLED
  provides="ManagedElement=1,SystemFunctions=1,SysM=1,CrM=1,Role=SCs"
  uses="ManagedElement=1,Equipment=1,ComputeResource=SC-2"
ComputeResourceRole=PL-7
  adminState=UNLOCKED
  instantiationState=INSTANTIATED
  operationalState=ENABLED
  provides="ManagedElement=1,SystemFunctions=1,SysM=1,CrM=1,Role=PLs"
  uses="ManagedElement=1,Equipment=1,ComputeResource=PL-7"
ComputeResourceRole=PL-6
  adminState=UNLOCKED
  instantiationState=INSTANTIATED
  operationalState=ENABLED
  provides="ManagedElement=1,SystemFunctions=1,SysM=1,CrM=1,Role=PLs"
  uses="ManagedElement=1,Equipment=1,ComputeResource=PL-6"
ComputeResourceRole=PL-5
  adminState=UNLOCKED
  instantiationState=INSTANTIATED
  operationalState=ENABLED
  provides="ManagedElement=1,SystemFunctions=1,SysM=1,CrM=1,Role=PLs"
  uses="ManagedElement=1,Equipment=1,ComputeResource=PL-5"
Role=SCs
  isProvidedBy
    "ManagedElement=1,SystemFunctions=1,SysM=1,CrM=1,ComputeResourceRole=SC-1"
    "ManagedElement=1,SystemFunctions=1,SysM=1,CrM=1,ComputeResourceRole=SC-2"
  scalability=NON_SCALABLE
Role=PLs
  isProvidedBy
    "ManagedElement=1,SystemFunctions=1,SysM=1,CrM=1,ComputeResourceRole=PL-3"
    "ManagedElement=1,SystemFunctions=1,SysM=1,CrM=1,ComputeResourceRole=PL-4"
    "ManagedElement=1,SystemFunctions=1,SysM=1,CrM=1,ComputeResourceRole=PL-5"
    "ManagedElement=1,SystemFunctions=1,SysM=1,CrM=1,ComputeResourceRole=PL-6"
    "ManagedElement=1,SystemFunctions=1,SysM=1,CrM=1,ComputeResourceRole=PL-7"
  scalability=SCALABLE
```

15. If more VMs need to be scaled in, repeat Step 7 to Step 14.

16. Check that the cluster is in a healthy state, see [CSCF Health Check](#).