

MTAS Scaling Management Guide

MTAS

USER GUIDE

Copyright

© Ericsson AB 2016–2018. 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	Introduction	1
1.1	Prerequisites	1
2	Overview	3
2.1	Scaling Terminology	3
2.2	Limitations	3
2.3	Subfunctions	3
3	Procedures	7
3.1	Preparation	7
3.2	Auto Scale-Out	8
3.3	Graceful Scale-In	10
3.4	Forceful Scale-In	13
3.5	Scaling Management from Cloud with Heat Orchestration	14





1 Introduction

This document describes the scalability functions of the MTAS cluster as a distributed system. It also gives instructions on how to do expansion or contraction of the cluster by these functions.

If scaling type is not mentioned, 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.

1.1 Prerequisites

This section describes the prerequisites that must be fulfilled before expanding or contracting the MTAS cluster.

1.1.1 Licenses

The scaling function does not require a license.

1.1.2 Documents

Before starting these procedures, the following documents must be available:

- *MTAS Health Check*
- *Ericsson Command-Line Interface User Guide*

1.1.3 Conditions

Before starting this procedure, ensure that the following conditions are met:

- The procedure must only be performed by support personnel with experience of Cloud and MTAS.
- No other upgrade or maintenance activity must be performed during the procedure.





2 Overview

This section provides an overview of the scaling procedures. For the operational steps, see Section 3 on page 7.

2.1 Scaling Terminology

Throughout this document the following terminology is used:

Node	Refers to a compute resource and can be a physical hardware blade or a virtual machine (VM) instantiation.
Fixed Domain	The set of nodes that cannot be subject of a scaling operation. Fixed domain of MTAS consists of SC-1 and SC-2 nodes permanently. The domain cannot be changed.
Scaling Domain	The set of nodes that can be subject of a scaling operation. MTAS scaling domain consists of all traffic nodes (PL-3, PL-4, PL-5 ... PL-N).

2.2 Limitations

This section summarizes the limitations relating to scaling functions.

2.2.1 PL-3 and PL-4 Nodes Are Not Scalable

Even though PL-3 and PL-4 nodes are considered to be part of the scaling domain, they cannot be scaled in.

2.2.2 Traffic Handling

The scaling operation involves planned reconfiguration of distribution units. This activity is performed in the quickest possible manner with high priority, hence load regulation-related alarms can appear during scaling operation. Such alarm is not expected to be present for longer time than 2-4 seconds. The effect is minimal on traffic handling capability.

2.3 Subfunctions

This section describes the subfunctions related to the scalability of the cluster.

2.3.1 Auto Scale-Out

Auto Scale-Out is an operation when one or more new compute resources are launched, see Figure 1. The system automatically detects, configures, and brings up the nodes as a member of the scaling domain of the cluster. See Figure 2 for an example when one new compute node is added to the cluster.

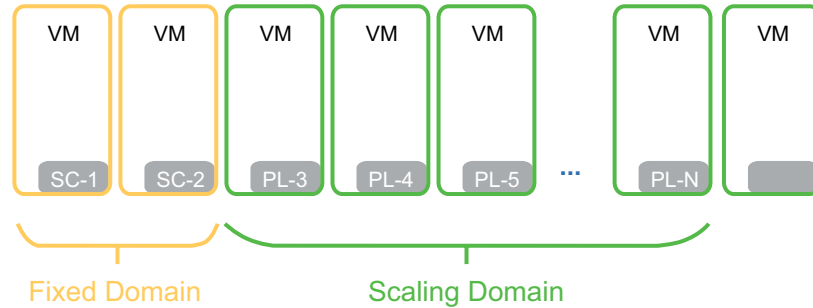


Figure 1 New Compute Resource Spawned and Available

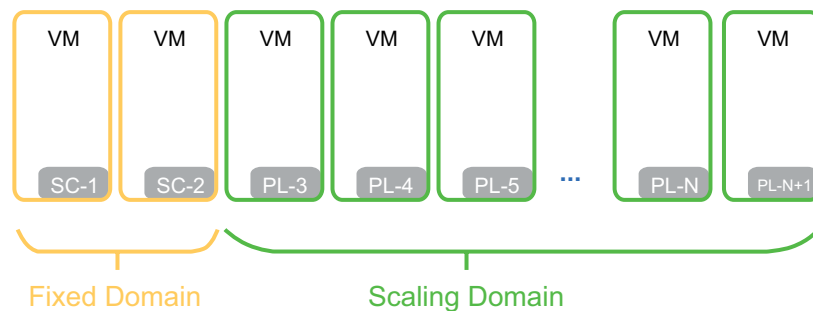


Figure 2 After Auto Scale-Out New Resource Is Added to Cluster

2.3.2 Graceful Scale-In

Graceful Scale-In is an operation where one or more compute resources, part of the scaling domain of the cluster (see Figure 3) are removed from the cluster (see Figure 4) to free up resources.

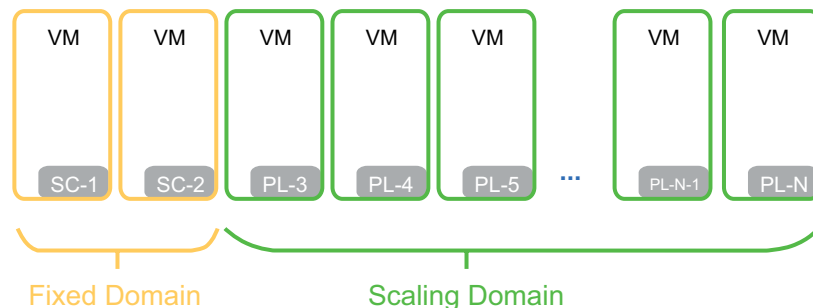


Figure 3 Node Named PL-(N-1) Is Part of Cluster

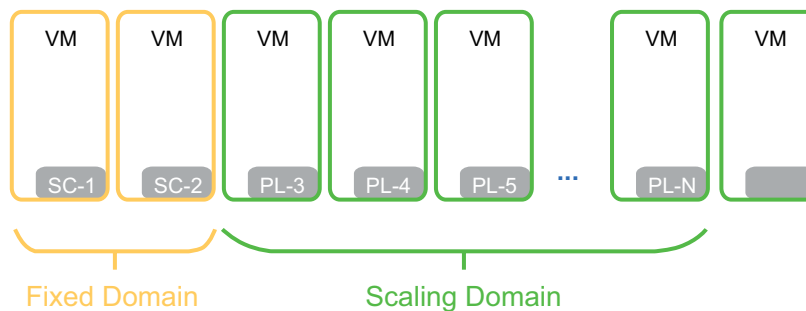


Figure 4 Node Named $PL-(N-1)$ Is Removed from Cluster and Its Resources Can Be Released

Note: The Graceful Scale-In operation can be rejected by the cluster if, according to the automatic estimation of the system, the target size of the cluster does not have the memory resources to serve the needed memory capabilities for the ongoing traffic.

2.3.3

Forceful Scale-In

Forceful Scale-In is, similarly to Graceful Scale-In, an operation to remove one or more nodes from the scaling domain of the cluster. The only difference is that in this case, the node is not available (see Figure 5) either because it already freed up its resources or because of a failure. Therefore the removal is only an administrative operation, see Figure 6.

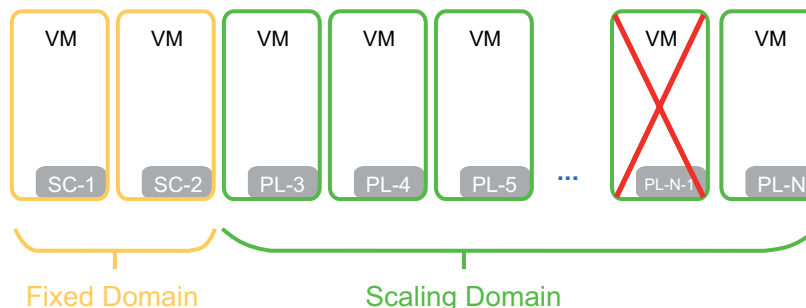


Figure 5 Node Named $PL-(N-1)$ in the Cluster Scaling Domain Is Unavailable

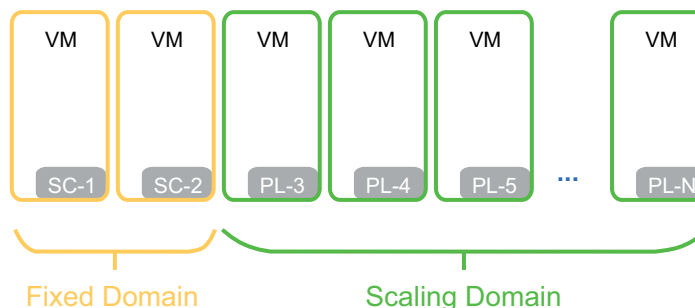


Figure 6 Node Named $PL-(N-1)$ Is Removed Administratively from Cluster





3 Procedures

This section describes the procedures of preparation; Auto Scale-Out, Graceful Scale-In, and Forceful Scale-In.

3.1 Preparation

This section describes preparation for the procedure.

3.1.1 Prerequisites

The following prerequisites must be met:

- Before starting these procedures, the user performing the operations must have access to the System Controller (SC) nodes.
- Scaling must only be performed after site-specific initial configuration is applied on the node. For more details on scaling, refer to *MTAS Hardening Guide*.
- Signaling Manager CLI or GUI must be closed before the start of the Scaling Operations. Manual updates of the configurations during Scaling Operations are not allowed.

3.1.2 Enable Scaling Feature

To enable the scaling feature:

1. Connect to one of the SC nodes:

```
ssh <user>@<system management IP address>
```

2. Check the operational state of the scaling feature:

```
SC-1: ~ # cmw-configuration --status SCALING
```

The following is an example output:

```
Disable
```

3. If the result is `Disable`, enable scaling functionality:

```
SC-1: ~ # cmw-configuration --enable SCALING
```



3.1.3 Create Backup

Before any scaling-related activities are performed, create a backup. Refer to *Create Backup*.

3.2 Auto Scale-Out

The guide for the Auto Scale-Out procedure is detailed in this section.

3.2.1 Prerequisites

Before starting these procedures, ensure that the following conditions are met:

- The cluster is in a healthy state, refer to *MTAS Health Check*.
- The target size of the cluster does not exceed the maximum cluster size supported, refer to *vMTAS Characteristics Specification*.
- The user monitoring the Scale-Out procedure has access to the Ericsson Command-Line Interface (ECLI).

3.2.2 Create One or More Compute Resources

Creating a compute resource in the Virtual Network Function (VNF) is outside the scope for this application. Follow the instructions given by the cloud management system about how to create a Virtual Machine (VM) instance or use the Heat orchestration based-scaling method described in Section 3.5 on page 13.

The Scale-Out procedure is triggered automatically once the new resource is available and launched.

Note: The newly created VM or VMs must have the same number of Virtual CPUs, the same amount of RAM, and the same number of ports as the other Payload (PL) VMs in the cluster.

3.2.3 Monitor the Scale-Out Progress

To monitor the Scale-Out progress of the operation on one of the COM CLIs:

1. Connect to the cluster through ECLI:

```
ssh -p 830 -t -s <user>@<OAM VIP> cli
```

2. Navigate to the CrM Managed Object (MO), for example:

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

3. Verify that the scale-out process has started:



```
(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=Default-Role"
  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=Default-Role"
  uses="ManagedElement=1, Equipment=1, ComputeResource=PL-4"
ComputeResourceRole=PL-5
  adminState=UNLOCKED
  instantiationState=INSTANTIATING
  operationalState=DISABLED
  provides="ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, Role=Default-Role"
  uses="ManagedElement=1, Equipment=1, ComputeResource=PL-5"
ComputeResourceRole=SC-1
  adminState=UNLOCKED
  instantiationState=INSTANTIATED
  operationalState=ENABLED
  provides="ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, Role=SYSTEM"
  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=SYSTEM"
  uses="ManagedElement=1, Equipment=1, ComputeResource=SC-2"
Role=Default-Role
  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"
  scalability=SCALABLE
Role=SYSTEM
  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
```

This example shows that `instantiationState` has changed to `INSTANTIATING` for node `PL-5`. It means that the scale-out has started.

4. Continue to monitor the progress until the scale-out process has ended and the added node has joined the cluster:

```
(CrM=1) >show -m ComputeResourceRole -p instantiationState,operationalState
```

The following example output shows the final result:



```
ComputeResourceRole=PL-3
  instantiationState=INSTANTIATED
  operationalState=ENABLED
ComputeResourceRole=PL-4
  instantiationState=INSTANTIATED
  operationalState=ENABLED
ComputeResourceRole=PL-5
  instantiationState=INSTANTIATED
  operationalState=ENABLED
ComputeResourceRole=SC-1
  instantiationState=INSTANTIATED
  operationalState=ENABLED
ComputeResourceRole=SC-2
  instantiationState=INSTANTIATED
  operationalState=ENABLED
```

This example shows that `instantiationState` has changed to `INSTANTIATED` for node PL-5. It means that PL-5 is added to the cluster. The example also shows that `operationalState` has changed to `ENABLED` for node PL-5. It means that node PL-5 has joined the cluster.

3.2.4 Check State of the Cluster

The Scale-Out procedure can be considered as successfully finished if the cluster is in healthy state after the operation, refer to *MTAS Health Check*.

3.3 Graceful Scale-In

This section describes a step by step guide for the Graceful Scale-In procedure.

3.3.1 Prerequisites

Before starting these procedures, ensure that the following conditions are met:

- The cluster is in a healthy state, refer to *MTAS Health Check*.
- The user performing the operations has access to the ECLI.
- SC-1, SC-2, PL-3, and PL-4 nodes cannot be subject of a Scale-In operation.

3.3.2 Scale-In One PL

To remove a PL from the cluster:

1. Connect to the cluster through the ECLI:

```
ssh -p 830 -t -s <user>@<OAM VIP> cli
```



2. Remove one or more PL nodes by navigating to the corresponding `ComputeResourceRole` MO in configure mode and removing the `provides` attribute.

The following is an example of removing PL-5:

```
>ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, ComputeResourceRole=PL-5
(ComputeResourceRole=PL-5) >configure
(config-ComputeResourceRole=PL-5) >no provides
(config-ComputeResourceRole=PL-5) >up
(config-CrM=1) >commit
```

3.3.2.1 Cancel Scale-In

The Scale-In procedure can be ended before committing the operation, using the `abort` command.

The following is an example of ending a multiple Scale-In procedure in the ECLI:

```
>ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, ComputeResourceRole=PL-5
(ComputeResourceRole=PL-5) >configure
(config-ComputeResourceRole=PL-5) >no provides
(config-ComputeResourceRole=PL-5) >up
(config-CrM=1) >ComputeResourceRole=PL-6
(config-ComputeResourceRole=PL-6) >no provides
(config-ComputeResourceRole=PL-6) >abort
```

3.3.3 Monitor Scale-In Progress

To monitor the progress of the Scale-In operation through the ECLI:

1. Verify that the Scale-In process has started:

```
>ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1
(CrM=1) >show -m ComputeResourceRole -p instantiationState,operationalState
```

The following is an example output when the node PL-5 is subject of Scale-In:

```
ComputeResourceRole=PL-3
  instantiationState=INSTANTIATED
  operationalState=ENABLED
ComputeResourceRole=PL-4
  instantiationState=INSTANTIATED
  operationalState=ENABLED
ComputeResourceRole=PL-5
  adminState=SHUTTINGDOWN
  instantiationState=UNINSTANTIATING
  operationalState=ENABLED
ComputeResourceRole=SC-1
  instantiationState=INSTANTIATED
  operationalState=ENABLED
ComputeResourceRole=SC-2
  instantiationState=INSTANTIATED
  operationalState=ENABLED
```

The PL-5 node attributes `instantiationState=INSTANTIATED` and `operationalState=UNINSTANTIATING` show that the graceful Scale-In has started.

2. The Scale-In procedure can only be considered successfully finished if the compute resource entry cannot be found through the ECLI.

The following is an example where PL-5 was scaled in:

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

The following is an example output where `ComputeResourceRole=PL-5` does not exist any more.

```
CrM=1
  autoRoleAssignment=ENABLED
  ComputeResourceRole=PL-3
  ComputeResourceRole=PL-4
  ComputeResourceRole=SC-1
  ComputeResourceRole=SC-2
  Role=Default-Role
  Role=SYSTEM
```

3.3.4 Remove Compute Resource

Removing a compute resource from the VNF is out of the scope for this application. Follow the instructions given by the cloud management system on how to remove VMs from the VNF or use the Heat orchestration-based scaling method described in section Section 3.5 on page 13.



3.3.5 Check State of the Cluster

The Graceful Scale-In procedure can be considered as successfully finished if the cluster is in healthy state after the operation, refer to *MTAS Health Check*.

3.3.6 Troubleshoot Scale-In Failures

If the Scale-In operation is unsuccessful, refer to *MTAS Troubleshooting Guideline*.

3.4 Forceful Scale-In

This section provides a step by step guide for the Forceful Scale-In procedure.

3.4.1 Prerequisites

Before starting these procedures, ensure that the following conditions are met:

- The user performing the operations has access to the ECLI interfaces and to the SC nodes.
- PL-3 and PL-4 nodes cannot be a subject of a Scale-In operation
- One or more nodes are unavailable, which results in faulty state of the cluster, refer to *MTAS Health Check*.

3.4.2 Scale-In Unavailable PL

This step is equivalent with the corresponding step of the Forceful Scale-In procedure, see Section 3.3.2 on page 10.

3.4.3 Monitor Scale-In Progress

This step is equivalent with the corresponding step of the Forceful Scale-In procedure, see Section 3.3.3 on page 11.

3.4.4 Check State of the Cluster

The Forceful Scale-In procedure can be considered as successfully finished if the cluster is in healthy state after the operation, refer to *MTAS Health Check*.

3.4.5 Troubleshoot Scale-In Failures

This step is equivalent with the corresponding step of the Forceful Scale-In procedure, see Section 3.3.6 on page 13.



3.5 Scaling Management from Cloud with Heat Orchestration

After the scaling feature is enabled and the node is instantiated with Heat Orchestration Templates (HOT) that support scaling, node scaling can be performed on the cloud through Heat Orchestration.

Note:

- Because of an OpenStack limitation, scaling is to be initiated from the original installation directory where all the HOT files are located. For more information about installation directory and HOT files, refer to *MTAS SW Installation*.
- It is important to highlight that there is no direct connection between MTAS and OpenStack therefore the name (number) of the VM present in OpenStack differs from the ComputeResource present in MTAS. To correlate a compute resource with a VM, use the Universally Unique Identifier (UUID).

To scale in or out the current node with a specific number of VMs, do the following:

1. Verify the status of the MTAS stack.

```
heat --os-tenant-name <tenant name> stack-list
```

Status of the stack is either `CREATE_COMPLETE` or `UPDATE_COMPLETE`, otherwise do not continue with the scaling procedure.

2. Verify the value of parameter `number_of_scaled_out_VMs`.

```
heat --os-tenant-name <tenant name> stack-show <MTAS  
stack name> | grep <number_of_scaled_out_VMs>
```

3. Calculate the new value of parameter `number_of_scaled_out_VMs`.

- To scale out, the value of parameter `number_of_scaled_out_VMs` is to be increased with the number of VMs one would like to create.

For example: The current value of the parameter `number_of_scaled_out_VMs` is 1 (meaning: beyond the initial size of 2+2; the cluster contains an extra VM/PL, so the size of the node is actually 2+3). To increase the size of the cluster to 2+5, that is, scale out by 2 VMs, the new value of the parameter should be 3.

- To scale in, the value of parameter `number_of_scaled_out_VMs` is to be decreased with the number of VMs one would like to remove. Typically, the VMs are removed in reverse chronological order.

4. Was the node instantiated from VNF-LCM by means of a workflow:
 - a. Yes



Verify the list of tags associated with the stack and update the stack:

```
heat --os-tenant-name <tenant name> stack-show
<MTAS stack name>|grep -A 2 tags
```

```
heat --os-tenant-name <tenant name> stack-update
-f mtas_hot.yaml -P number_of_scaled_out_VMs=<nu
mber_of_scaled_out_VMs> -x <MTAS stack name> --tags
<list of tags>
```

b. No

Update stack:

```
heat --os-tenant-name <tenant name> stack-update
-f mtas_hot.yaml -P number_of_scaled_out_VMs=<num
ber_of_scaled_out_VMs> -x <MTAS stack name>
```

5. Monitor the progress of the stack-update:

```
heat --os-tenant-name <tenant name> stack-list
```

A successful stack-update is indicated by a `UPDATE_COMPLETE` stack status. If there is unsuccessful stack-update, check the reason for failure by doing the following:

```
heat --os-tenant-name <tenant name> stack-show <MTAS
stack name>
```

Troubleshoot the issue and then repeat Step 4. (If the failed operation was scale out, another option can be to repeat the stack-update with the value of `number_of_scaled_out_VMs` parameter decreased to its original value).



Attention!

Risk of data loss or data corruption.

Do not remove resources created by Heat manually by commands (`nova`, `neutron`), or from Horizon/Atlas dashboard as it can corrupt the database of Heat.

To repair a faulty resource of a heat stack, use the following Heat commands on the stack:

- `heat action-check`



- `heat resource-list`
- `heat stack-update`

Detailed descriptions of these procedures are beyond the scope of this document. For more information about `Heat`, refer to <https://wiki.openstack.org/wiki/Heat>.