

# Upgrade Guide

Virtual Multimedia Resource Function

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.

# Contents

1	About This Document	1
2	Manual Upgrade Methods for vMRF	2
2.1	Network-Redundant Upgrade	2
2.2	vMRF In-Service Upgrade	5
3	Rollback Procedure	11





# 1 About This Document

This document describes the manual vMRF upgrade and rollback process on a cloud service. During the manual upgrade process, the user must perform the upgrade-related tasks manually, using application scripts and the deployment-related functions in the cloud environment. The manual upgrade process is an alternative to the fully automated upgrade VNF life cycle operation which involves deployment of a new VNF, migrating the configuration, and the possibility to rollback to the earlier version if needed.

This document is written for vMRF operator personnel who are responsible for upgrading vMRF. The vMRF operator is assumed to be a cloud service consumer, that is, an end user on a cloud service. The end user is also referred to as a tenant.



## 2 Manual Upgrade Methods for vMRF

The network-redundant upgrade process can be performed when two VNFs are available in parallel during normal operation, as described in [Network-Redundant Upgrade](#) on page 2.

Alternatively, manual in-service upgrade process, as described in [vMRF In-Service Upgrade](#) on page 5, requires that temporarily two VNFs are running in parallel, and there is no traffic impact during the upgrade process.

### 2.1 Network-Redundant Upgrade

This procedure describes how to upgrade vMRF with network redundancy available. This procedure can be performed when two VNFs are available in parallel during normal operation. [Figure 1](#) shows the network-redundant upgrade process.

1. Export data  
Configuration data is exported from the vMRF.
2. The vMRF is locked. After lock, the vMRF does not handle sessions, so it stops processing traffic.
3. The vMRF is scaled-in and removed.
4. A new version of vMRF is deployed.

The configuration data exported previously is imported into the new version. After that, the new version of vMRF already starts processing traffic.

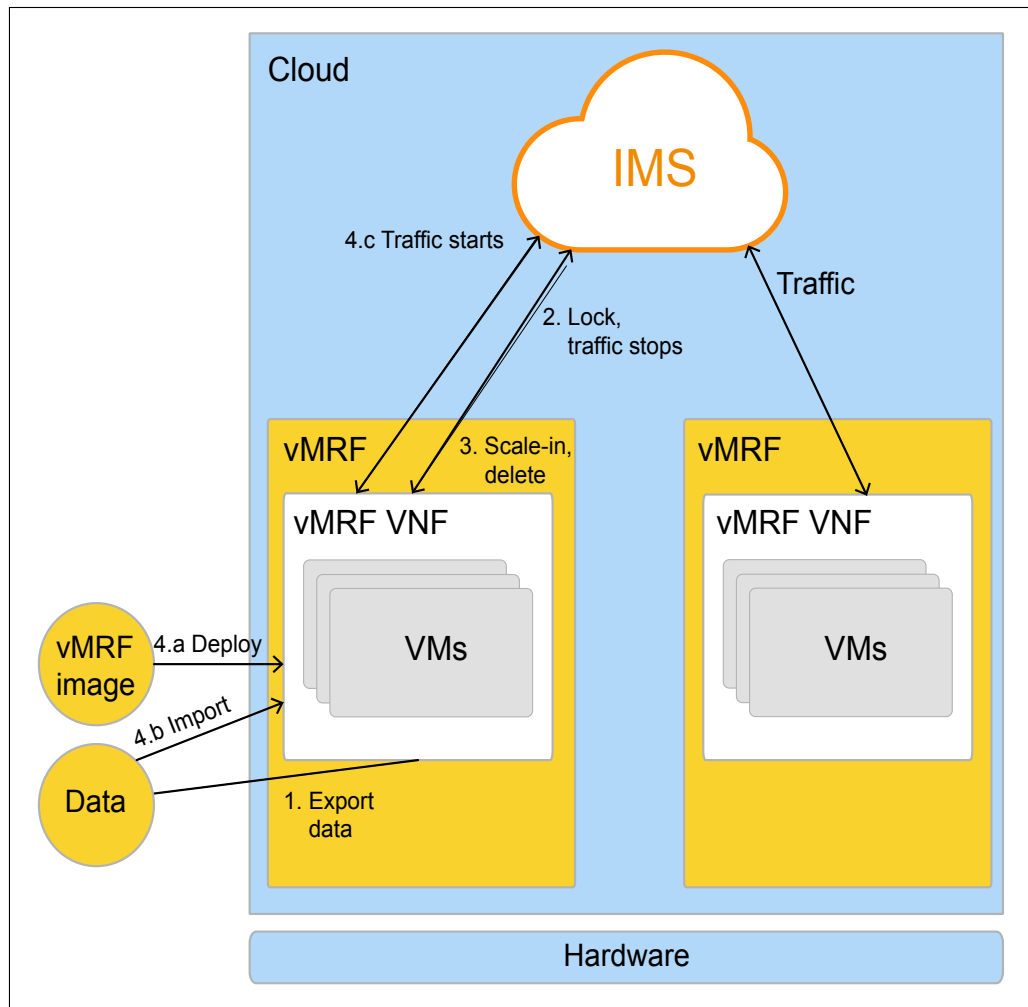


Figure 1 vMRF Network-Redundant Upgrade

### 2.1.1 Export Configuration Data from the Old Version

#### Steps

1. Open an SSH connection to the O&M IP address of the old version of the vMRF VNF instance using the following command:

```
ssh <user ID>@<O&M IP address>
```

2. Run the following command:

```
/opt/mrf_director/mrf_export_conf.py /home/<user ID>/<output file without extension>
```

**Result:** The configuration data is exported into a specified .tar.gz archive file (the default and recommended format).



3. Copy the exported configuration file out of the file system of the VNF using, for example, `scp`:

```
scp <user ID>@<O&M IP address>:/home/<user ID>/mrf_conf.tar.gz .
```

**Result:** The configuration file `mrf_conf.tar.gz` is copied from the `/home/<user ID>/` folder in the file system of the vMRF VNF to the current directory.

## 2.1.2 Lock and Scale-in VMs

### Steps

1. Lock all the deployed VMs in the old version of the vMRF VNF instance. Consider graceful locking through MTAS configuration by gracefully locking the MRFP nodes in MTAS. For more information, refer to section Deactivate Gracefully in MTAS Media Control Management Guide, Reference [1]. Otherwise continue with the following steps:

**Note:** In the procedure below, after modifying the `administrativeState` attribute, the VMs are immediately locked and all ongoing traffic on the VMs stops.

- a. Open an SSH connection to the O&M IP address of the vMRF VNF using the following command:

```
ssh <user ID>@<O&M IP address>
```

- b. Start a session by issuing the `cliss` command.
- c. Navigate to the `MrfInstance` MO that represents the VM and enter configure mode:

```
>ManagedElement=1,MediaResourceFunction=1,MrfResource=1,MrfInstance=<mrfInstanceId>
```

```
(MrfInstance=<mrfInstanceId>)>configure
```

- d. Modify the value of the `administrativeState` attribute:

```
(config-MrfInstance=<mrfInstanceId>)>administrativeState=<LOCKED>
```

- e. Commit the changes:

```
(config-MrfInstance=<mrfInstanceId>)>commit
```

- f. Repeat steps from [Step 1.b](#) to [Step 1.e](#) for all VMs.

**Result:** The VM is locked immediately.



2. Scale in all the deployed VMs of the instance, as described in [vMRF Configuration Management](#).

### 2.1.3

## Deploy the New Version

### Steps

1. Using the proper [deployment instructions](#), deploy the new version of vMRF with **one or two VMs**, and check that it is running properly. Ensure that the new version connects to the same external networks as the old version. It is recommended to import the configuration data during deployment.
2. If you have imported configuration data during deployment, continue with [Step 6](#). Otherwise, continue with the next step.
3. Open an SSH connection to the O&M IP address of the new version of the vMRF VNF instance using the following command:

```
ssh <user_ID>@<O&M_IP_address>
```

4. Copy the configuration data file exported from the **old** version to the file system of the **new** version using, for example, `scp`:

```
scp mrf_conf.tar.gz <user_ID>@<O&M_IP_address>:/home/<user_ID>
```

**Result:** The configuration file `mrf_conf.tar.gz` is copied from the current directory to the `/home/<user_ID>/` folder in the file system of the vMRF VNF.

5. Run the following command:

```
/opt/mrf_director/mrf_import_conf.py /home/<user ID>/  
mrf_conf.tar.gz
```

6. Check that traffic processing in the new version of the vMRF VNF instance is working properly.

### Results

The new version starts processing traffic. If there are problems with the new version that cannot be solved and that are considered unacceptable, continue with [Rollback Procedure](#) on page 11.

## 2.2

## vMRF In-Service Upgrade

The vMRF in-service manual upgrade process involves deployment of a new VNF and migrating the configuration. This method requires that temporarily two VNFs are running in parallel.

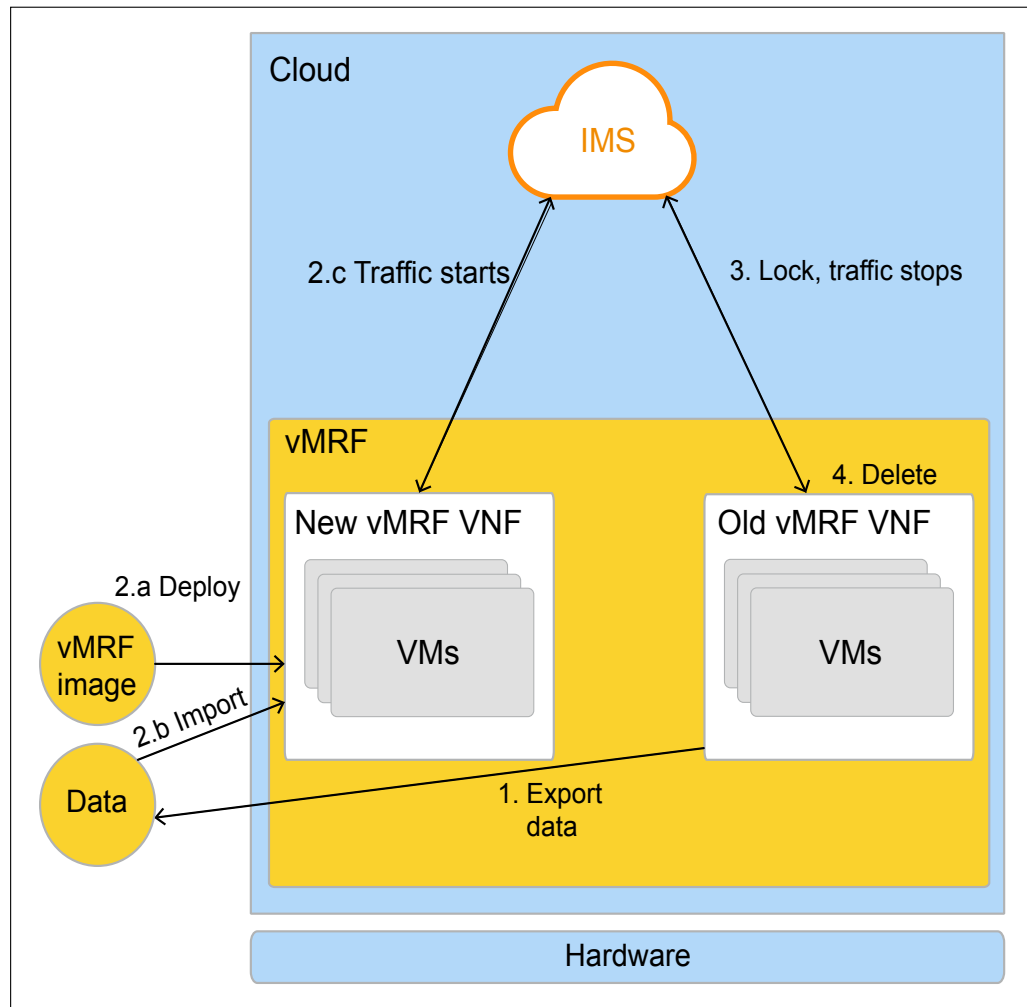


Figure 2 vMRF In-Service Upgrade

### 1. Export data

Configuration data is exported from the old version of vMRF.

### 2. Deploy the new version of vMRF

The new version is deployed with only a few VMs to minimize the potential impact on traffic of a software fault in the new version. The configuration data exported from the old version is imported into the new version. After that, the new version of vMRF already starts processing traffic.

It is recommended to monitor the new version of vMRF. If the new version has any severe problems, the upgrade must be rolled back.

### 3. Commit to using the new version of vMRF



The new version is scaled out to the actual number of VMs and the old version is locked. After the lock, the old version of vMRF does not handle sessions, so it stops processing traffic.

It is recommended to monitor the new version of vMRF until it has fully taken over the traffic. If any severe problems are found, the upgrade must be rolled back.

#### 4. Remove the old version of vMRF

If the new version is considered to be operating on a sufficient level, the old version can be removed. It is also possible to keep the old version and run the old and new versions in parallel if, for example, there is a requirement to have a longer testing period for the new version.

## 2.2.1 Export Configuration Data from the Old Version

### Steps

1. Open an SSH connection to the O&M IP address of the old version of the vMRF VNF instance using the following command:

```
ssh <user ID>@<O&M IP address>
```

2. Run the following command:

```
/opt/mrf_director/mrf_export_conf.py /home/<user ID>/<output file without extension>
```

**Result:** The configuration data is exported into a specified .tar.gz archive file (the default and recommended format).

3. Copy the exported configuration file out of the file system of the VNF using, for example, scp:

```
scp <user ID>@<O&M IP address>:/home/<user ID>/mrf_conf.tar.gz .
```

**Result:** The configuration file mrf\_conf.tar.gz is copied from the /home/<user ID>/ folder in the file system of the vMRF VNF to the current directory.

## 2.2.2 Deploy the New Version

### Prerequisites

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

- The vMRF VMs to be scaled-out are configured as MRFP (Media Resource Function Processor) nodes in the MTAS.



A vMRF VM identifies itself to the MTAS with a Message Id (MId) that contains the vMRF VM signaling IP address and SCTP port. Typically, the whole range of signaling IP addresses, that is, the signaling subnet, that has been configured in the OpenStack for the vMRF VNF, is configured as MRFP nodes in the MTAS.

For more information on adding an MRFP node in MTAS, refer to section Add MRFP in *MTAS Media Control Management Guide*, Reference [1].

## Steps

1. Using the proper deployment instructions, deploy the new version of vMRF with **one or two VMs**, and check that it is running properly. Ensure that the new version connects to the same external networks as the old version. It is recommended to import the configuration data during deployment.

**Note:** In OSS-RC, make sure to create a new VNF as well, due to the different O&M IP addresses used for the old and the new VNFs.

2. If you have imported configuration data during deployment, continue with [Step 6](#). Otherwise, continue with the next step.
3. Open an SSH connection to the O&M IP address of the new version of the vMRF VNF instance using the following command:

```
ssh <user ID>@<O&M IP address>
```

4. Copy the configuration data file exported from the **old** version to the file system of the **new** version using, for example, `scp`:

```
scp mrf_conf.tar.gz <user ID>@<O&M IP address>:/home/<user ID>
```

**Result:** The configuration file `mrf_conf.tar.gz` is copied from the current directory to the `/home/<user ID>/` folder in the file system of the vMRF VNF.

5. Run the following command:

```
/opt/mrf_director/mrf_import_conf.py /home/<user ID>/  
mrf_conf.tar.gz
```

6. Check that traffic processing in the new version of the vMRF VNF instance is working properly.
7. If the operation of the new version is considered acceptable, continue with [Commit to Using the New Version](#) on page 8.

If there are problems with the new version that cannot be solved and that are considered unacceptable, do not proceed with the upgrade, continue with [Rollback Procedure](#) on page 11.

## 2.2.3 Commit to Using the New Version



## Steps

1. Scale out the new version of the VNF by increasing the number of VMs to the full capacity of the VNF.

**Note:** If there are not enough resources to scale out the new instance while the old instance still exists, scale in the old instance, as described in [vMRF Configuration Management](#). Always keep one VM in the old VNF.

If there are problems with the new version during or after scaling out that cannot be solved and that are considered unacceptable, do not proceed with the upgrade, continue with [Rollback Procedure](#) on page 11 to roll back the upgrade.

2. Lock all the deployed VMs in the old version of the vMRF VNF instance. Consider graceful locking through MTAS configuration by gracefully locking the MRFP nodes in MTAS. For more information, refer to section Deactivate Gracefully in MTAS Media Control Management Guide, Reference [1]. Otherwise continue with the following steps:

**Note:** In the procedure below, after modifying the `administrativeState` attribute, the VMs are immediately locked and all ongoing traffic on the VMs stops.

- a. Open an SSH connection to the O&M IP address of the vMRF VNF using the following command:

```
ssh <user ID>@<O&M IP address>
```

- b. Start a session by issuing the `cliss` command.
- c. Navigate to the `MrfInstance` MO that represents the VM and enter configure mode:

```
>ManagedElement=1,MediaResourceFunction=1,MrfResource=1,MrfInstance=<mrflInstanceId>
```

```
(MrfInstance=<mrflInstanceId>)>configure
```

- d. Modify the value of the `administrativeState` attribute:

```
(config-MrfInstance=<mrflInstanceId>)>administrativeState=LOCKED>
```

- e. Commit the changes:

```
(config-MrfInstance=<mrflInstanceId>)>commit
```

- f. Repeat steps from [Step 2.c](#) to [Step 2.e](#) for all VMs.

**Result:** The VM is locked immediately.



## 2.2.4 Remove the Old Version

It is possible to keep the old version and run the old and new version in parallel if, for example, there is a requirement to have a longer testing period for the new version. If the old version is no longer needed, remove it.



## 3 Rollback Procedure

If there are problems with the new version that cannot be solved and that are considered unacceptable, the latest upgrade must be rolled back.

### Steps

1. If any severe problems are found in the new vMRF VNF instance while the old VNF still exists, lock the new version by locking all the deployed VMs.

**Note:** In the procedure below, after modifying the `administrativeState` attribute, the VMs are immediately locked and all ongoing traffic on the VMs stops.

- a. Open an SSH connection to the O&M IP address of the vMRF VNF using the following command:

```
ssh <user_ID>@<O&M_IP_address>
```

- b. Start a session by issuing the `cliss` command.
- c. Navigate to the `MrfInstance` MO that represents the VM and enter configure mode:

```
>ManagedElement=1,MediaResourceFunction=1,MrfResource=1,MrfInstance=<mrfInstanceId>
```

```
(MrfInstance=<mrfInstanceId>)>configure
```

- d. Modify the value of the `administrativeState` attribute:

```
(config-MrfInstance=<mrfInstanceId>)>administrativeState=<LOCKED>
```

- e. Commit the changes:

```
(config-MrfInstance=<mrfInstanceId>)>commit
```

2. Create a report, and attach troubleshooting data according to the [Data Collection Guideline for vMRF](#). Send the report to the Ericsson support organization.
3. If required by the Ericsson support organization, keep the new version of vMRF for debugging purposes. Otherwise, remove the new version.

**Result:** The upgrade is rolled back, you can exit this procedure.