Installing UFM Server Software

Table of contents

The default UFM® installation directory is */opt/ufm*.

For instructions on installing the UFM server software, please refer to following instructions per desired installation mode.

- Installing UFM Server on Bare Metal Server
    - Installing UFM on Bare Metal Server- Standalone Mode
    - Installing UFM on Bare Metal Server - High Availability Mode
- Installing UFM Docker Container Mode
    - Installing UFM on Docker Container - Standalone Mode
    - Installing UFM on Docker Container - High Availability Mode

The following processes might be interrupted during the installation process:

- httpd (Apache2 in Ubuntu)
- dhcpd

> **Note**
>
> To install UFM over static IPv4 configuration (instead of DHCP) please refer to Configuring UFM Over Static IPv4 Address before installation.

After installation:

1. Activate the software license
2. Perform initial configuration

> **Note**

Before you run UFM, ensure that all ports used by the UFM server for internal and external communication are open and available. For the list of ports, see Appendix – Used Ports.

Prerequisites for UFM Server Software Installation

Verify that a supported version of Linux is installed on your machine. For details, see UFM System Requirements.

The following table lists the packages that must be installed on your machine (according to the system OS) before you install the UFM server software.

RedHat 7	RedHat 8	RedHat 9	Ubuntu 18.04	Ubuntu 20.04	Ubuntu 22.04
acl	acl	acl	acl	acl	acl
apr-util-openssl	apr-util-openssl	apr-util-openssl	apache2	apache2	apache2
bc	bc	bc	bc	bc	bc
cairo	gnutls	gnutls	chrpath	chrpath	chrpath
gnutls	httpd	httpd	cron	cron	cron
httpd	iptables	iptables-nft	gawk	gawk	gawk
iptables	jansson	jansson	lftp	lftp	lftp
lftp	lftp	lftp	libcurl4	libcurl4	libcurl4
libxml2	libnsl	libnsl	logrotate	logrotate	logrotate
libxslt	libxml2	libxml2	python3	python3	python3
mod_session	libxslt	libxslt	qperf	qperf	qperf
mod_ssl	mod_session	mod_session	rsync	rsync	rsync
net-snmp	mod_ssl	mod_ssl	snmpd	snmpd	snmpd
net-snmp-libs	net-snmp	net-snmp	sqlite3	sqlite3	sqlite3

RedHat 7	RedHat 8	RedHat 9	Ubuntu 18.04	Ubuntu 20.04	Ubuntu 22.04
net-snmp-utils	net-snmp-libs	net-snmp-libs	sshpass	sshpass	sshpass
net-tools	net-snmp-utils	net-snmp-utils	ssl-cert	ssl-cert	ssl-cert
php	net-tools	net-tools	sudo	sudo	sudo
psmisc	php	php	telnet	telnet	telnet
python3	psmisc	psmisc	zip	zip	zip
python3-libs	python36	python3			
qperf	qperf	qperf			
rsync	rsync	rsync			
sqlite	sqlite	sqlite			
sshpass	sshpass	sshpass			
sudo	sudo	sudo			
telnet	telnet	telnet			
zip	zip	zip			

Note

On some Ubuntu OSs, Docker is installed via SNAP, which might lead to errors when trying to use UFM Plugins.

To solve this issue, perform the following:

1. Remove Docker installed via SNAP, run:

```
snap remove --purge docker
```

2. Update the local package index, run :

```
apt update
```

3. Install native Docker, run:

```
apt install-y docker.io
```

In addition, ensure the following before you begin installation:

- The computer hostname is not defined as 127.0.0.1 and localhost is defined as 127.0.0.1.
- The hostname must NOT appear on the loopback address line. An example of the loopback address is: 127.0.0.1 localhost.localdomain localhost.
- Disable the firewall service (/etc/init.d/iptables stop), or ensure that the required ports are open (see the prerequisite script, refer to Used Ports).
- Disable SELinux.
- If more than one fabric is managed by different UFM instances, set up different management network spaces for each fabric (not the same LAN).
- Uninstall any previously installed Subnet Manager from the UFM server machine.
- MLNX_OFED 5.x version is installed prior to installing UFM.
- As of UFM v.6.12.0, it is **NOT mandatory** to configure the IPoIB fabric interface with an IP address.

  In cases where the IP is configured, it is **mandatory** that the IP is permanently configured and that it starts automatically upon server reboot (the IPoIB fabric interface should be active even if the network is down).

Note

The user can set a persistent IP address using Netplan (mainly for Ubuntu systems) or modifying the interface network script (RedHat systems).

- The default MLNX_OFED installation includes opensm. Remove the MLNX_OFED opensm before UFM installation like the following examples:

  RedHat:

  ```
  rpm -e opensm-3.3.9.MLNX_20111006_e52d5fc-0.1
  ```

  Ubuntu:

  ```
  apt purge opensm
  ```

  By default, ib0 and eth0 are configured as primary access points for the UFM management. If different management and/or InfiniBand interfaces (including bond interfaces) are used as the primary access points, you should modify the configuration file by running the script */opt/ufm/scripts/change_fabric_config.sh* as described in the section Configuring General Settings in gv.cfg.

  Change the UFM Agent interface to the Ethernet and/or IPoIB interfaces used for communication with UFM Agent:

  ```
  ufma_interfaces = ib0,eth0
  ```

Additional Prerequisites for UFM High Availability Installation

- Reliable and high-capacity out-of-band IP connectivity between the UFM Primary and Secondary servers (1 Gb Ethernet is recommended). This connectivity is used for DRBD synchronization.

- Format two identical servers with dedicated disk partitions for UFM replication. Since the UFM configuration file is replicated to the standby server, both master and standby servers must have the same interfaces.

- Allocate exactly the same size partition on both servers (master and slave) for the replicated data. See UFM Server Requirements for the recommended partition size.

  Partitions should not be mounted and must be zeroed (the file system should not be installed on the partitions). For disk partitioning, see the Linux user manual (man fdisk).

- We recommend establishing a passwordless SSH (via /root/.ssh/authorized_keys file) between the two servers before the installation.

- In fabrics consisting of multiple tiers of switches, it is recommended that the management ports (ib0) of the primary and secondary UFM server be connected to different fabric switches on the same tier (the outermost edge in CLOS 5 designs).

  This is because by default, UFM manages the IB fabric via ib0, port 1 of the HCA. Failure or disconnect of ib0, the IB management port, causes a failure condition in UFM resulting in HA failover.

  When the management ports (ib0) of the primary and secondary UFM server are connected to the same switch, a failure of this switch will result in a disconnect of both UFMs from the fabric, and therefore UFM will not be able to manage the fabric.

Note

Subnet Manager is running over the native InfiniBand layer, therefore bonding the IpoIB interfaces will not provide high availability. For additional information, please refer to section UFM Failover to Another Port.

The UFM installation includes the InfiniBand Performance Management module (IBPM). This module is responsible for reporting performance information back to UFM and upper layer applications. When available, this process is offloaded to the non-management port (default ib1) of the UFM server. Failure or disconnect of the non-management port (ib1) on the primary UFM server will not cause UFM to failover. By default, the UFM Health Monitoring process is

configured to try to restart the IBPM. For more information, see UFM Health Configuration in the UFM User Manual.

Installing UFM Server on Bare Metal Server

Installing UFM server on Bare Metal server can be done with the following modes:

- Installing UFM on Bare Metal Server- Standalone Mode
- Installing UFM on Bare Metal Server - High Availability Mode

Installing UFM on Bare Metal Server - High Availability Mode

Before installing UFM server software in High-Availability mode, ensure that the Additional Prerequisites for UFM High Availability Installation are met.

The UFM High-Availability configuration requires dual-link connectivity based on two separate interfaces between the two UFM HA nodes. This configuration comprises of a primary link that is exclusively reserved for DRBD operations and a secondary link designated for backup purposes. Crucially, it is imperative that communication between the servers is established in a bidirectional manner across both interfaces and validated through user-initiated testing, such as a 'ping' command or other suitable alternatives before HA configuration can be implemented. In cases where only one link is available among the two UFM HA nodes/servers, manually configure UFM with a single link. Refer to Configure HA without SSH Trust (Single Link Configuration).

Note

UFM HA package requires a dedicated partition with the same name for DRBD on both servers. This guide uses `/dev/sda5` as an example.

1. On both servers, Install UFM Enterprise in Stand Alone (SA) mode.

   > **Note**
   >
   > Do not start UFM service.

2. Install the latest pcs and drbd-utils drivers on both servers.

   For Ubuntu:

   ```
   apt install pcs pacemaker drbd-utils
   ```

   For CentOS/Red Hat:

   ```
   yum install pcs pacemaker drbd84-utils kmod-drbd84
   ```

   OR

   ```
   yum install pcs pacemaker drbd90-utils kmod-drbd90
   ```

3. Download UFM-HA latest package from using this command:

   ```
   wget https://www.mellanox.com/downloads/UFM/ufm_ha/5.6.0/ufm_ha_5.6.0-4.tgz
   ```

   For Sha256:

   ```
   wget https://download.nvidia.com/ufm/ufm_ha/5.6.0/ufm_ha_5.6.0-4.sha256
   ```

> **Note**
>
> For more information on the UFM-HA package and all installation and configuration options, please refer to UFM High-Availability User Guide.

4. Extract the downloaded UFM-HA package on both servers under /tmp/.

5. Go to the directory you extracted /tmp/ufm_ha_XXX and run the installation script. For example, if your DRBD partition is /dev/sda5 run:

```
./install.sh -l /opt/ufm/files/ -d /dev/sda5 -p enterprise
```

6. Configure the HA cluster. There are the three methods:

- Configure HA with SSH Trust (Dual Link Configuration) - Requires passwordless SSH connection between the servers.
- Configure HA without SSH Trust (Dual Link Configuration) - Does not require passwordless SSH connection between the servers, but asks you to run configuration commands on both servers.
- Configure HA without SSH Trust (Single Link Configuration) - Can be used in cases where only one link is available among the two UFM HA nodes/servers.

Configure HA with SSH Trust (Dual Link Configuration)

1.

    1. On the **master server only**, configure the HA nodes. To do so, from /tmp, run the configure_ha_nodes.sh command as shown in the below example

```
configure_ha_nodes.sh \
--cluster-password 12345678 \
--master-primary-ip 10.10.10.1 \
--standby-primary-ip 10.10.10.2 \
```

```
--master-secondary-ip 192.168.10.1 \
--standby-secondary -ip 192.168.10.2 \
--no-vip
```

Note

The script configure_ha_nodes.sh is is located under /usr/local/bin/, therefore, by default, you do not need to use the full path to run it.

Note

The --cluster-password must be at least 8 characters long.

Note

To set up a Virtual IP for UFM and gain access to UFM through this IP, regardless of which server is running UFM, you may employ the --no-vip OR --virtual-ip command and provide an IP address as an argument. This can be achieved by navigating to https://<Virtual-IP>/ufm on your web browser.

Note

When using back-to-back ports with local IP addresses for HA sync interfaces, ensure that you add your IP addresses and hostnames to the /etc/hosts file. This is needed to allow

the HA configuration to resolve hostnames correctly based on the IP addresses you are using.

Note

configure_ha_nodes.sh requires SSH connection to the standby server. If SSH trust is not configured, then you are prompted to enter the SSH password of the standby server during configuration runtime

2. Depending on the size of your partition, wait for the configuration process to complete and DRBD sync to finish.

Configure HA without SSH Trust (Dual Link Configuration)

If you cannot establish an SSH trust between your HA servers, you can use **ufm_ha_cluster directly to configure HA. To configure HA, follow the below instructions:**

Note

Please change the variables in the commands below based on your setup.

1.

    1. [On Standby Server] Run the following command to configure Standby Server:

```
ufm_ha_cluster config -r standby \
--local-primary-ip 10.10.50.1 \
--peer-primary-ip 10.10.50.2 \
--local-secondary-ip 192.168.10.1 \
```

```
--peer-secondary-ip 192.168.10.2 \
--hacluster-pwd 123456789 \
--no-vip
```

2. [On Master Server] Run the following command to configure Master Server:

```
ufm_ha_cluster config -r master --local-primary-ip 10.10.50.1 \
--peer-primary-ip 10.10.50.2 \
--local-secondary-ip 192.168.10.1 \
--peer-secondary-ip 192.168.10.2 \
--hacluster-pwd 123456789 \
--no-vip
```

You must wait until after configuration for DRBD sync to finish, depending on the size of your partition. To check the DRBD sync status, run:

```
ufm_ha_cluster status
```

Configure HA without SSH Trust (Single Link Configuration)

> ⚠ **Warning**
>
> This is not the recommended configuration and, in case of network failure, it might cause HA cluster split brain.

If you cannot establish an SSH trust between your HA servers, you can use **ufm_ha_cluster directly to configure HA. To configure HA, follow the below instructions:**

> ⓘ **Note**

Please change the variables in the commands below based on your setup.

1.

    1. [On Standby Server] Run the following command to configure Standby Server:

```
ufm_ha_cluster config \
-r standby \
-e 10.212.145.5 \
-l 10.212.145.6 \
--enable-single-link
```

    2. [On Master Server] Run the following command to configure Master Server:

```
ufm_ha_cluster config -r master \
-e 10.212.145.6 \
-l 10.212.145.5 \
-i 10.212.145.50 \
--enable-single-link
```

    You must wait until after configuration for DRBD sync to finish, depending on the size of your partition. To check the DRBD sync status, run:

```
ufm_ha_cluster status
```

Starting HA Cluster

- To start UFM HA cluster:

```
ufm_ha_cluster start
```

- To check UFM HA cluster status:

```
ufm_ha_cluster status
```

Stopping UFM HA cluster:

```
ufm_ha_cluster stop
```

> **Note**
>
> For complete details on high availability, refer to NVIDIA UFM High-Availability User Guide.

Installing UFM on Bare Metal Server-Standalone Mode

To install the UFM server software as a standalone for InfiniBand:

1. Create a temporary directory (for example */tmp/ufm*).
2. Open the UFM software zip file that you downloaded. The zip file contains the following installation files:
    - RedHat 7/CentOS 7/OEL 7: ufm-X.X-XXX.el7.x86_64.tgz
    - RedHat 8/Centos 8: ufm-X.X-XXX.el8.x86_64.tgz
    - Ubuntu 18.04: ufm-X.X-XXX.Ubuntu18.x86_64.tgz
    - Ubuntu 20.04: ufm-X.X-XXX.Ubuntu20.x86_64.tgz

- Ubuntu 22.04: ufm-X.X-XXX.Ubuntu22.x86_64.tgz

3. Extract the installation file for your system's OS to the temporary directory that you created.

4. From within the temporary directory, run the following command as root:

```
./install.sh
```

> **Note**
>
> Running with the option "-o ib" is no longer required. For automatic installation, use the -q flag.
>
> For “quiet” installation -q flag can be added (automatically answer yes for each question the installer asks).

> **Note**
>
> Export MULTISUBNET_CONSUMER=1 environment variable before running the installation script to install the UFM server in Multisubnet Consumer mode.

The UFM software is installed. You can now remove the temporary directory.

Installing UFM Docker Container Mode

General Prerequisites

- MLNX_OFED must be installed on the server that will run UFM Docker
- For UFM to work, you must have an InfiniBand port configured with an IP address and in "up" state.

> **Note**
>
> For InfiniBand support, please refer to NVIDIA Inbox Drivers , or MLNX_OFED guides.

- Make sure to stop the following services before running UFM Docker container, as it utilizes the same default ports that they do: Pacemaker, httpd, OpenSM, and Carbon.
- If firewall is running on the host, please make sure to add an allow rule for UFM used ports (listed below):

> **Note**
>
> If the default ports used by UFM are changed in UFM configuration files, make sure to open the modified ports on the host firewall.

- 80 (TCP) and 443 (TPC) are used by WS clients (Apache Web Server)
- 8000 (UDP) is used by the UFM server to listen for REST API requests (redirected by Apache web server)
- 6306 (UDP) is used for multicast request communication with the latest UFM Agents
- 8005 (UDP) is used as a UFM monitoring listening port
- 8888 (TCP) is used by DRBD to communicate between the UFM Primary and Standby servers
- 2022 (TCP) is used for SSH

Prerequisites for Upgrading UFM Docker Container

- Supported versions for upgrade are UFM v.6.10.0 and above.
- UFM files directory from previous container version mounted on the host.

Step 1: Loading UFM Docker Image

To load the UFM docker image, pull the latest image from docker hub:

```
docker pull mellanox/ufm-enterprise:latest
```

Note

You can see full usage screen for ufm-installation by running the container with `-h` or `-help` flag:

```
docker run --rm mellanox/ufm-enterprise-installer:latest -h
```

If an Internet connection is not available, perform the following:

- Copy the UFM image to your machine.
- Load the image from the file using this command:

```
docker image load -i <image-path>
```

Step 2: Installing UFM Docker

Installation Command Usage

```
docker run -it --name=ufm_installer --rm \
-v /var/run/docker.sock:/var/run/docker.sock \
-v /etc/systemd/system/:/etc/systemd_files/ \
-v /opt/ufm/files/:/installation/ufm_files/ \
-v [LICENSE_DIRECTORY]:/installation/ufm_licenses/ \
mellanox/ufm-enterprise:latest \
--install [OPTIONS]
```

Modify the variables in the installation command as follows:

- [UFM_LICENSES_DIR]: UFM license file or files location.

> **Note**
>
> Example: If your license file or files are located under /downloads/ufm_license_files/ then you must set this volume to be -v /downloads/ufm_license_files/:/installation/ufm_licenses/

- [OPTIONS]: UFM installation options. For more details see the table below.

Command Options

Flag	Description	Default Value
-f | --fabric-interface	IB fabric interface name.	ib0
-g | --mgmt-interface	Management interface name.	eth0
-h | --help	Show help	N/A
-m | --multisubnet-consumer	UFM Multisubnet Consumer mode	N/A

Installation Modes

UFM Enterprise installer supports several deployment modes:

- Installing UFM on Docker Container - High Availability Mode
- Installing UFM on Docker Container - Standalone Mode

Installing UFM on Docker Container - High Availability Mode

Pre-Deployments Requirements

- Install **pacemaker**, **pcs**, and **drbd-utils** on both servers

  For Ubuntu:

  ```
  apt install pcs pacemaker drbd-utils
  ```

  For CentOS/Red Hat:

  ```
  yum install pcs pacemaker drbd84-utils kmod-drbd84
  ```

OR

```
yum install pcs pacemaker drbd90-utils kmod-drbd90
```

- A partition for DRBD on each server (**with the same name** on both servers) such as /dev/sdd1. Recommended partition size is 10-20 GB, otherwise DRBD sync will take a long time to complete.
- CLI command hostname -i must return the IP address of the management interface used for pacemaker sync correctly (update /etc/hosts/ file with machine IP)
- Create the directory on each server under /opt/ufm/files/ with read/write permissions on each server. This directory will be used by UFM to mount UFM files, and it will be synced by DRBD.
- Disable the firewall service (/etc/init.d/iptables stop), or ensure that the required ports are open (see the prerequisite script).
- Disable SELinux.

Installing UFM Containers

On the main server, install UFM Enterprise container with the command below:

```
docker run -it --name=ufm_installer --rm \
-v /var/run/docker.sock:/var/run/docker.sock \
-v /etc/systemd/system/:/etc/systemd_files/ \
-v /opt/ufm/files/:/installation/ufm_files/ \
-v /tmp/license_file/:/installation/ufm_licenses/ \
mellanox/ufm-enterprise:latest \
--install
```

On the standby (secondary) server, install the UFM Enterprise container like the following example with the command below:

```
docker run -it --name=ufm_installer --rm \
-v /var/run/docker.sock:/var/run/docker.sock \
```

```
-v /etc/systemd/system/:/etc/systemd_files/ \
-v /opt/ufm/files/:/installation/ufm_files/ \
mellanox/ufm-enterprise:latest \
--install
```

Downloading UFM HA Package

Download the UFM-HA package on both servers using the following command:

```
wget https://www.mellanox.com/downloads/UFM/ufm_ha/5.6.0/ufm_ha_5.6.0-4.tgz
```

For Sha256:

```
wget https://download.nvidia.com/ufm/ufm_ha/5.6.0/ufm_ha_5.6.0-4.sha256
```

Installing UFM HA Package

For more information on the UFM-HA package and all installation and configuration options, please refer to UFM High Availability User Guide.

1. [On Both Servers] Extract the downloaded UFM-HA package under /tmp/
2. [On Both Servers] Go to the extracted directory /tmp/ufm_ha_XXX and run the installation script. For example, if your DRBD partition is /dev/sda5 run the following command:

   ```
   ./install.sh -l /opt/ufm/files/ -d /dev/sda5 -p enterprise
   ```

Configuring UFM HA

There are the three methods to configure the HA cluster:

- Configure HA with SSH Trust (Dual Link Configuration) - Requires passwordless SSH connection between the servers.
- Configure HA without SSH Trust (Dual Link Configuration) - Does not require passwordless SSH connection between the servers, but asks you to run configuration commands on both servers.
- Configure HA without SSH Trust (Single Link Configuration) - Can be used in cases where only one link is available among the two UFM HA nodes/servers.

Configure HA with SSH Trust (Dual Link Configuration)

1. On the **master server only**, configure the HA nodes. To do so, from /tmp, run the configure_ha_nodes.sh command as shown in the below example

```
configure_ha_nodes.sh \
--cluster-password 12345678 \
--master-primary-ip 10.10.50.1 \
--standby-primary-ip 10.10.50.2 \
--master-secondary-ip 192.168.10.1 \
--standby-secondary-ip 192.168.10.2 \
--no-vip
```

> **Note**
>
> The script configure_ha_nodes.sh is is located under /usr/local/bin/, therefore, by default, you do not need to use the full path to run it.

> **Note**
>
> The --cluster-password must be at least 8 characters long.

> **Note**
>
> When using back-to-back ports with local IP addresses for HA sync interfaces, ensure that you add your IP addresses and hostnames to the /etc/hosts file. This is needed to allow the HA configuration to resolve hostnames correctly based on the IP addresses you are using.

> **Note**
>
> configure_ha_nodes.sh requires SSH connection to the standby server. If SSH trust is not configured, then you are prompted to enter the SSH password of the standby server during configuration runtime

2. Depending on the size of your partition, wait for the configuration process to complete and DRBD sync to finish. To check the DRBD sync status, run:

```
ufm_ha_cluster status
```

Configure HA without SSH Trust (Dual Link Configuration)

If you cannot establish an SSH trust between your HA servers, you can use ufm_ha_cluster directly to configure HA. You can see all the options for configuring HA in the Help menu:

```
ufm_ha_cluster config -h
```

To configure HA, follow the below instructions:

> **Note**
>
> Please change the variables in the commands below based on your setup.

1. [On **Standby Server**] Run the following command to configure **Standby Server**:

```
ufm_ha_cluster config -r standby -e <peer ip address> -l <local ip address> -p <cluster_password>
```

2. [On **Master Server**] Run the following command to configure **Master Server**:

```
ufm_ha_cluster config -r master -e <peer ip address> -l <local ip address> -p <cluster_password>
-i <virtual ip address>
```

Configure HA without SSH Trust (Single Link Configuration)

> **Warning**
>
> This is not the recommended configuration and, in case of network failure, it might cause HA cluster split brain.

If you cannot establish an SSH trust between your HA servers, you can use **ufm_ha_cluster directly to configure HA. To configure HA, follow the below instructions:**

> **Note**

Please change the variables in the commands below based on your setup.

1.

    1. [On **Standby Server**] Run the following command to configure **Standby Server**:

```
ufm_ha_cluster config \
-r standby \
-e 10.212.145.5 \
-l 10.212.145.6 \
--enable-single-link
```

    2. [On **Master Server**] Run the following command to configure **Master Server**:

```
ufm_ha_cluster config -r master \
-e 10.212.145.6 \
-l 10.212.145.5 \
-i 10.212.145.50 \
--enable-single-link
```

    You must wait until after configuration for DRBD sync to finish, depending on the size of your partition. To check the DRBD sync status, run:

```
ufm_ha_cluster status
```

    IPv6 Example:

```
ufm_ha_cluster config -r standby -l fcfc:fcfc:209:224:20c:29ff:fee7:d5f2 -e
fcfc:fcfc:209:224:20c:29ff:fecb:4962 --enable-single-link -p some_secret
```

Starting HA Cluster

- To start UFM HA cluster:

```
ufm_ha_cluster start
```

- To check UFM HA cluster status:

```
ufm_ha_cluster status
```

- To stop UFM HA cluster:

```
ufm_ha_cluster stop
```

- To uninstall UFM HA, first stop the cluster and then run the uninstallation command as follows:

```
/opt/ufm/ufm_ha/uninstall_ha.sh
```

Installing UFM on Docker Container - Standalone Mode

1. Copy only your UFM license file(s) to a temporary directory which we're going to use in the installation command. For example: /tmp/license_file/

2. Run the UFM installation command according to the following example which will also configure UFM fabric interface to be ib1:

```
docker run -it --name=ufm_installer --rm \
-v /var/run/docker.sock:/var/run/docker.sock \
-v /etc/systemd/system/:/etc/systemd_files/ \
-v /opt/ufm/files/:/installation/ufm_files/ \
-v /tmp/license_file/:/installation/ufm_licenses/ \
mellanox/ufm-enterprise:latest \
--install \
--fabric-interface ib1
```

3. Reload systemd:

```
systemctl daemon-reload
```

4. To Start UFM Enterprise service run:

```
systemctl start ufm-enterprise
```

Replacing the Standby Node

- Install the HA package for the new node (standby).
- Disconnect the standby node (the old standby) and run the following command on the master node:

```
ufm_ha_cluster detach
```

- Configure the new standby node; please refer to the relevant section depending on the installation
- Connect the new standby to the cluster by running the command on the master node:

```
ufm_ha_cluster attach -l <local primary ip address> -e <peer primary ip address> -E <peer secondary ip address> -p <clust
```