NVIDIA UFM High-Availability User Guide v5.6.0

Table of contents

About This Document

This document describes NVIDIA® UFM High-Availability (HA) Architecture, connectivity, configuration options, and monitoring procedures.

Software Download

To download the latest UFM High-Availability software package, please visit NVIDIA's Licensing Portal.

Related Documents

Pacemaker	• https://wiki.clusterlabs.org/wiki/Pacemaker • https://clusterlabs.org/pacemaker/doc/deprecated/en-US/Pacemaker/2.0/pdf/Clusters_from_Scratch/Pacemaker-2.0-Clusters_from_Scratch-en-US.pdf
DRBD	• https://linbit.com/drbd/
Split-Brain	• https://xahteiwi.eu/resources/hints-and-kinks/solve-drbd-split-brain-4-steps/

Technical Support

Customers who purchased NVIDIA products directly from NVIDIA are invited to contact us through the following methods:

- E-mail: enterprisesupport@nvidia.com
- Enterprise Support page: https://www.nvidia.com/en-us/support/enterprise

Customers who purchased NVIDIA M-1 Global Support Services, please see your contract for details regarding Technical Support.

Customers who purchased NVIDIA products through an NVIDIA-approved reseller should first seek assistance through their reseller.

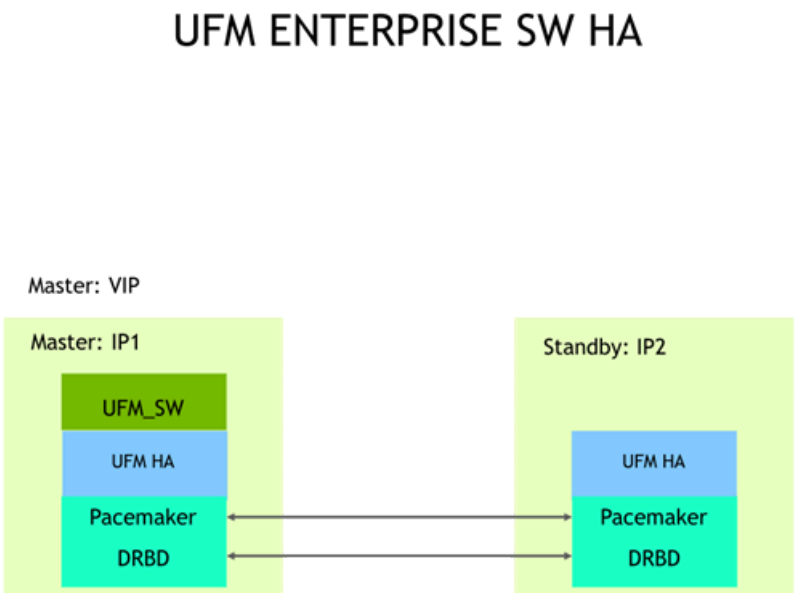
Document Revision History

For the list of changes made to this document, refer to Document Revision History.

Overview

UFM HA provides High-Availability on the host level for UFM products (UFM Enterprise/UFM Appliance Gen 3.0 and UFM Cyber-AI). The solution is based on Pacemaker to monitor host resources, services, and applications; and DRBD to sync file-system states. The HA package can be used with both bare-metal and Dockerized UFM deployments.

UFM HA should be installed on the master and standby nodes. The below figure describes the UFM Enterprise HA Architecture.



UFM State

The below files are replicated between the master and standby nodes:

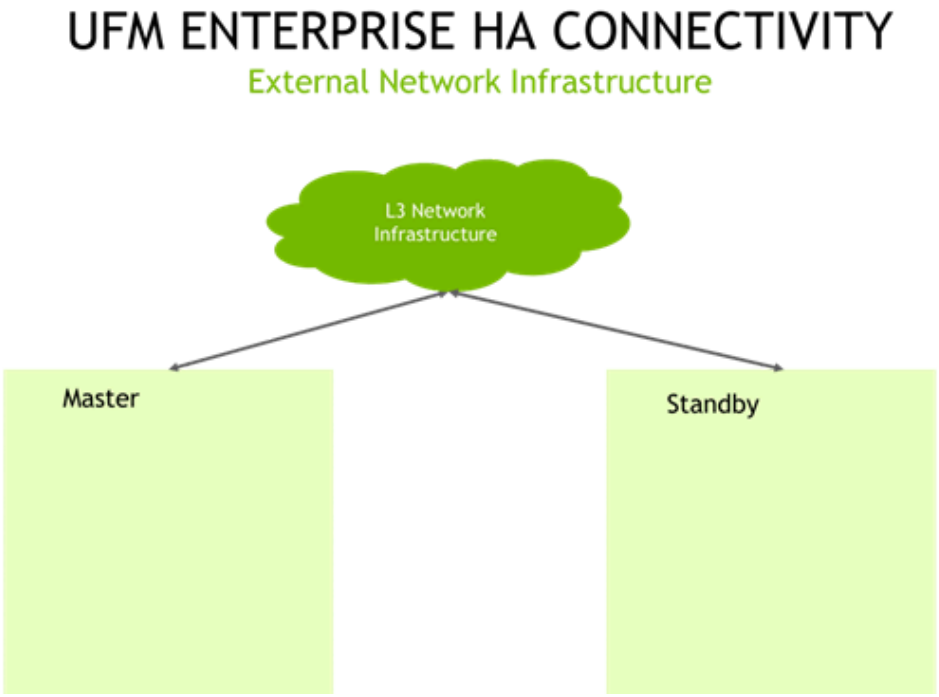
```
/opt/ufm/files/*
```

Examples: log files, events, SQLite DB files (configuration, Telemetry history, persistent states topology groups).

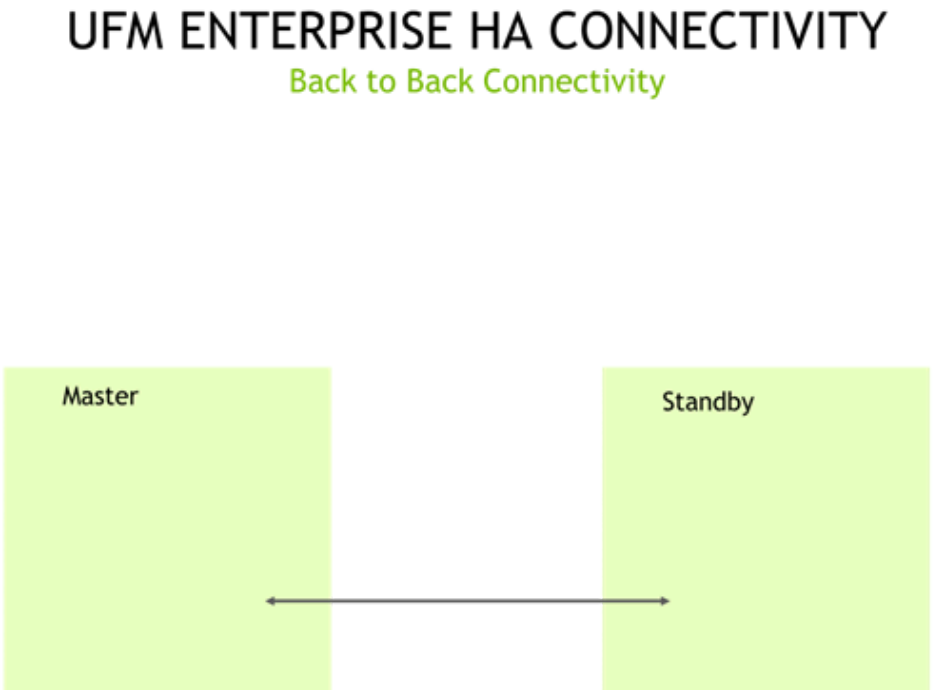
Connectivity Options

The master and standby nodes communicate with each other to establish and monitor a High-Availability solution. This connectivity is used by both the Pacemaker and DRBD. Below are connectivity options:

1. Cloud Connectivity. The following figure describes the external network infrastructure.



2. Back-to-back Connectivity, described in the following figure.

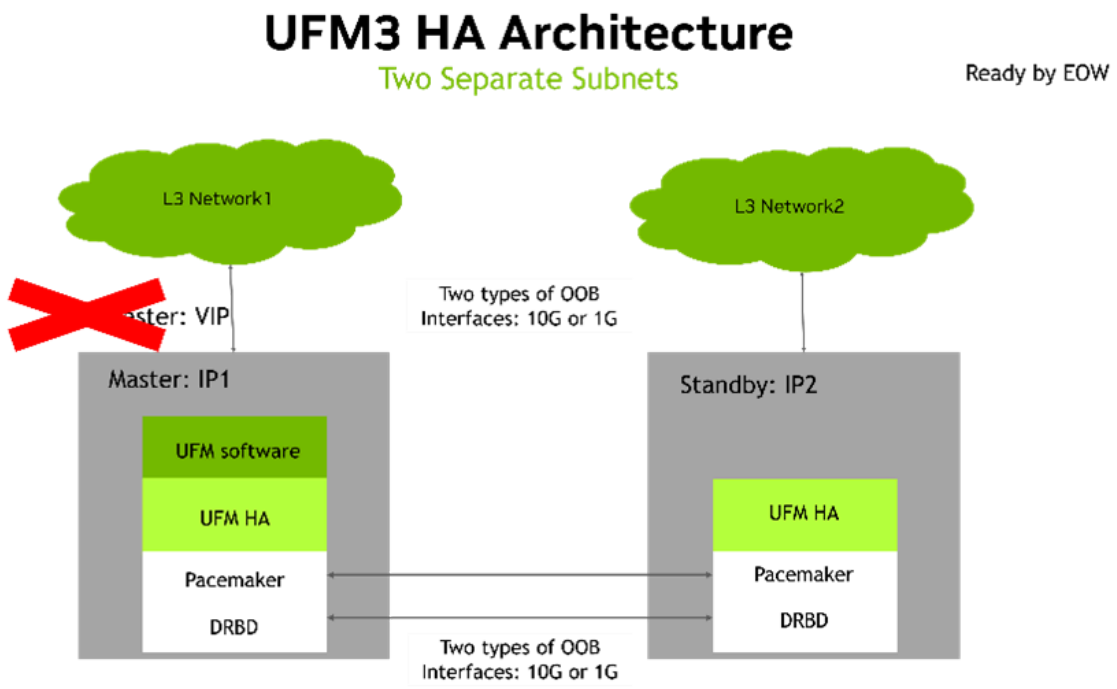


UFM-HA employs a dual-link configuration comprising primary and secondary connections to enhance system stability while mitigating the risk of connectivity challenges. It leverages two prioritized IP addresses, primary and secondary, which the Pacemaker utilizes to establish two connectivity links. Notably, DRBD utilizes the primary IP address to synchronize data. It is recommended to utilize this IP address for interfaces with high transfer rates such as InfiniBand interfaces for optimal performance (IP over IB) and rapid DRBD synchronization. On the other hand, the secondary connectivity link may be effected via the management interface, typically an Ethernet interface.

DRBD and Pacemaker can use the same network interface or utilize different interfaces. For example, while the Pacemaker connectivity can be done through the management interface (usually an Ethernet interface), the DRBD synchronization could be done on an InfiniBand interface for better performance (IP over IB).

See below the configuration options for selecting a dif:

1. No VIP Connectivity Option



For some constrained network environments, the no VIP Connectivity option is supported. In this architecture,every UFM node has two physical IP addresses, primary and secondary. There is no VIP (floating) IP representing the whole cluster. This option allows two cluster nodes to lay in different subnets. In such a setup, clients who communicate with the UFM cluster should be aware of the active node status or constantly try to access both nodes.

HA Cluster Resources

The cluster software monitors the following HA cluster resources:

- **UFM Enterprise**

  A systemd service runs and monitors all UFM Enterprise processes.

- **UFM HA Watcher**

  The **ufm-ha-watcher service monitors the health status of the UFM Enterprise and performs a failover in case the ufm-health process decides to perform a failover.**

- **Virtual IP**

Also known as Cluster IP, a virtual IP is a unique IP resource allocated on the master node. The virtual IP address should be reachable from any machine that uses it (REST API or UI). Virtual IP is not a mandatory configuration and can be omitted.

- **DRBD and File System**

  DRBD needs its block device on each node. This can be a physical disk partition or a logical volume. The volume size planning should be done according to specific cluster sizing. The UFM-HA creates a DRBD resource and a filesystem resource with primary/secondary states based on the node if it is a master of a standby node.

Cluster Network Access

Cluster Network Access must consume UFM REST APIs and UI or performs management or monitoring tasks (ssh, scp, syslog, etc.).

For access to the UFM cluster, the below five IP addresses should be configured:

- Primary Physical IP1 – For the master node
- Secondary Physical IP1 – For the master node
- Primary Physical IP2 – For the standby node
- Secondary Physical IP2 – For the standby node
- Virtual (floating) IP (VIP)

Each two IP addresses of the same class should be configured in the same subnet and accessible (routable) by both cluster nodes. A virtual IP address should be in the subnet of one of the classes. The cluster manages the virtual IP address state. By default, the VIP is assigned to the master node. In case of failure of the master node, the VIP is moved by the cluster SW to the standby node. Network failures from the client to the UFM cluster are not monitored or handled by the HA cluster. Network infrastructure redundancy is out of the UFM HA solution scope. UFM HA cluster components utilize L3 and communication protocols (TCP/IP) for their internal communication and are agnostic to underlying L2 networking infrastructure.

Supported platforms

UFM HA is supported on the following Linux distributions:

1. Ubuntu 18.04, 20.04 and 22.04
2. CentOS7.7-9
3. CentOS8 Stream, RHEL8.5
4. CentOS9 Stream, RHEL9.X (2023)

Prerequisites

The following packages should be installed.

Pacemaker Packages

Pacemaker Package	Supported Versions
pacemaker	1.1.18 and 2.1.3
pcs	0.9.x, 0.10.x and 0.11.x
Corosync	2.4.3 and 3.1.5

DRBD

In the default sync mode, DRBD must be installed by the user. However, if NFS is chosen as the synchronization mechanism, DRBD is not mandatory.

DRBD	Supported Versions
DRBD utils	8.x.x,and 9.x.x

Installation and Configuration

Installation

The UFM HA package can be downloaded by running the following command:

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

The UFM HA package should be installed on both machines (**Master** and **Standby**) and the required UFM products. Installation order does not matter. To install the UFM-HA package:

- Untar the ufm-ha package:

  ```
  tar xvzf ufm-ha-<version>.tgz
  ```

- Go to the directory you extracted and run the installation script. For example:

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

  For NFS support, run the following installation script. For example:

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

Option	Description
-l	Sync Files Location. Must be always /opt/ufm/files/
-d	Disk name for DRBD. For example /dev/sda5 (in case of using DRBD). Note that the `-d` option is not needed in case of NFS.
-p	Product Name. Must use “enterprise” to UFM Enterprise

Info

In cases where you have a previous installation of ufm_ha and you want to upgrade to the newer version, run the following command:

```
./install.sh -u
```

Info

UFM HA scripts are installed under /usr/bin.

Configuration

There are two methods to configure the HA cluster:

- Configure HA with SSH Trust - Requires passwordless SSH connection between the servers.
- Configure HA without SSH Trust - Does not require passwordless SSH connection between the servers, but asks you to run configuration commands on both servers.

Configure HA with SSH Trust

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 \
   --virtual-ip 10.10.10.5
```

> **Note**
>
> The script configure_ha_nodes.sh is located under /usr/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 ensure effective HA sync interface functionality for PCS version 0.9.X, employing back-to-back ports with local IP addresses, it is crucial to incorporate the relevant IP addresses and hostnames into the /etc/hosts file. This step is necessary to enable the HA configuration to accurately resolve hostnames based on the specific IP addresses in use.

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

Note

While configuring UFM HA on Oracle Linux, make sure the SELinux is disabled. You can check SELinux status with sestatus.

If it is enabled, follow the below steps to disable it:

- Run vi /etc/selinux/config
- Add SELINUX=disabled
- Reboot the machine
- Verify SELinux is disabled with the command sestatus.

Option	Description
--cluster-password	UFM HA cluster password for authentication by the pacemaker.
--master-ip	Master (main) server IP address
--standby-ip	Standby server IP address
--virtual-ip **OR** --no-vip	UFM HA cluster Virtual IP or configure HA without virtual IP

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

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
```

Usage:

```
ufm_ha_cluster config [<options>]
```

Option		Description
-r	--role <node role>	Node role (master or standby).
-e	--peer-primary-ip <ip address>	Peer node primary IP address (mandatory).
-l	--local-primary-ip <ip address>	Local node primary IP address (mandatory).
-E	--peer-secondary-ip <ip address>	Peer node secondary IP address (mandatory).
-L	--local-secondary-ip <ip address>	Local node primary IP address (mandatory).
-i	--virtual-ip <virtual-ip> --virtual-ip6 <virtual-ip6>	Cluster virtual IP(v4). Cluster virtual IP(v4).
-p	--hacluster-pwd <pwd>	HA cluster user password.
-h	--help	Show this message
-N	--no-vip	Configure HA without virtual IP
-M	--ignore-mgmt-failure	Ignore management interface status if VIP is configured. Will not failover if master node's secondary IP is down.

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 primary ip address> -l <local primary ip address> -E <peer secondary ip address> -L <local secondary 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 primary ip address> -l <local primary ip address> -E <peer secondary ip address> -L <local secondary ip address> -p -i <virtual ip address>
```

NFS File Sharing

NFS synchronization mechanism can be used instead of DRBD. Multi-Nodes Support can be used with NFS synchronization mechanism only, as described in the following section. To activate this functionality, users must define the following parameters:

- Mode: NFS
- NFS Server
- Shared Folder

Ensure that the NFS version supports nfs4. It is recommended that the NFS server is not one of the UFM-HA nodes. Refer to the section below for details on configuring the file.

Multi-Nodes Support

The UFM-HA cluster can comprise of more than two nodes. Among these nodes, one will serve as the master, while the others will operate in standby mode.

To configure multiple nodes, users must populate the configuration file '/etc/ufm_ha/ha_nodes.cfg' on all nodes (ensuring that the file is identical across all nodes).

This file contains details about each participating node, including:

- Role: Master/Standby
- Primary IP address
- Secondary IP address

Using File Configuration

The '/etc/ufm_ha/ha_nodes.cfg' file contains all the necessary information for HA configuration and can serve as a replacement for command-line configuration. The only configuration not saved in the file is the password for security reasons.

To configure, use the following command (should be executed after setting the configuration):

```
ufm_ha_cluster config –p <password>
```

Info

The standby nodes must be configured at first, with the last node being set as the master node.

Configuration File

The sample configuration file includes up to three sections for nodes, but users can add additional sections as needed.

```
[General]
# Connection mode
# in case dual_link is true, each node must have primary and secondary IPs
dual_link = true

[Node.1]
# valid role options: master/standby
role = master
# Mandatory
primary_ip =
# Mandatory if dual_link = true
secondary_ip =

[Node.2]
role = standby
primary_ip =
secondary_ip =

[Node.3]
role = standby
primary_ip =
secondary_ip =

# Add other Node.x sections if needed.

[Virtual]
# If virtual IP should not be added, set `no_vip = true`
no_vip =

virtual_ip =

virtual_ip6 =

ignore_mgmt_failure = false
# when using BGP virtual IP, you must use the loopback interface, set `interface = lo`
# in other cases we let the pcs to decide on the relevant network interface.
interface =

[FileSync]
# valid options are: drbd/nfs
```

```
mode = nfs

[NFS]
# fill in case the FileSync.mode is nfs
nfs_server =
shared_folder =
```

UFM HA Cluster Operations

Show UFM HA version

Run the following command to show UFM HA version:

```
ufm_ha_cluster version
```

Starting UFM HA Cluster

> **Note**
>
> Before starting the UFM cluster, ensure that the DRBD sync is completed.

To start UFM HA cluster:

```
ufm_ha_cluster start
```

Checking UFM Cluster Status

To check UFM HA cluster status:

```
ufm_ha_cluster status
```

Stopping UFM HA Cluster

To stop UFM HA cluster:

```
ufm_ha_cluster stop
```

Takeover Services

The takeover command can be executed on the standby machine so that it will be the master.

```
ufm_ha_cluster takeover
```

Master Failover

The failover command can be executed on the master machine so that it will be the standby.

```
ufm_ha_cluster failover
```

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
```

- Config the new standby node; please refer to Configuration.
- 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 <cluster_password>
```

Uninstalling UFM HA

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

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

Monitoring and Troubleshooting

Check UFM Status	Run the below command on the master node: `/etc/init.d/ufmd status`

Check HA Status	Run the below command:
	`ufm_ha_cluster status`
	`pcs status`

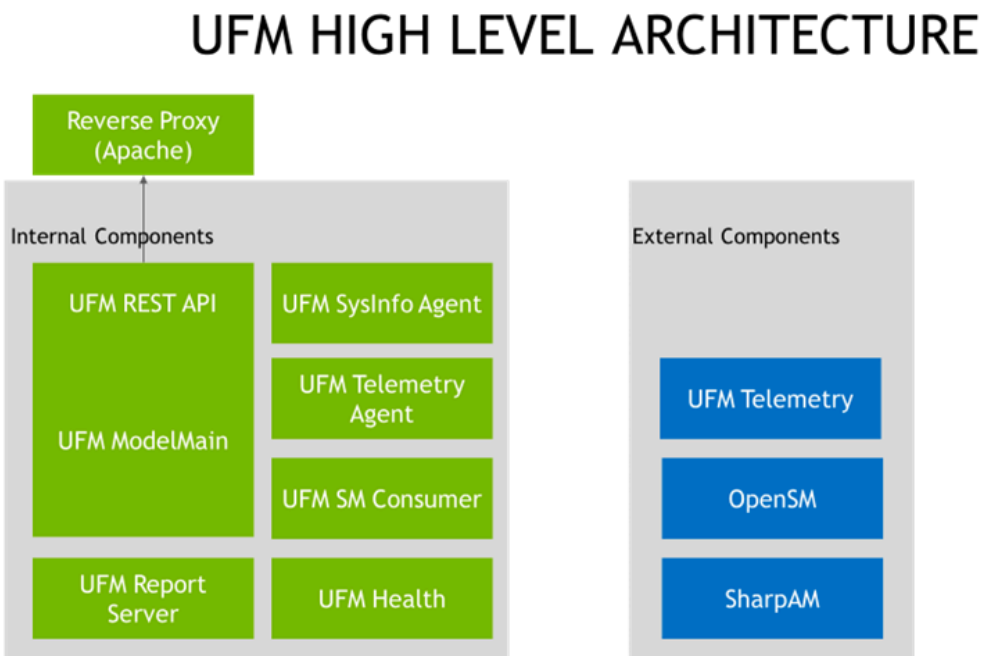
Check DRBD Status	Run the below command: `ufm_ha_cluster status`
Show DRBD Resource	Run the below command: `drbdadm sh-resources`
Show DRBD Disk State	Run the below command: `drbdadm dstate ha_data`
Show DRBD Role	Run the below command: `drbdadm role ha_data`
Show DRBD	Run the below command:

Check UFM Status	Run the below command on the master node: `/etc/init.d/ufmd status`
Connectivity	`drbdadm cstate ha_data`

Split-Brain Recovery	For automated HA solution, is it recommended to configure STONITH agents to kill (power-off) a peer node.
	Step 1:
	Manually choose a node which data modifications will be discarded.
	It is called the split-brain victim. Choose wisely; all modifications will be lost!
	When in doubt, run a backup of the victim's data before you continue.
	When running a Pacemaker cluster, you can enable maintenance mode.
	`ufm_ha_cluster enable-maintain`
	If the split-brain victim is in the Primary role, bring down all applications using this resource.
	Now, switch the victim to the Secondary role:
	`victim# ufm_ha_cluster reset standby`
	Resync starts automatically if the survivor is in a WFConnection network state.
	If the split-brain survivor is still in a Standalone connection state, reconnect it:
	`survivor#  ufm_ha_cluster reset master`
	Now the resynchronization from the survivor (SyncSource) to the victim (SyncTarget) starts immediately. There is no full sync initiated, but all modifications on the victim will be overwritten by the survivor's data, and modifications on the survivor will be applied to the victim.

UFM High-Level Architecture

The below figure illustrates the UFM high-level architecture.



FR#1

Support of Active-Standby HA approach. UFM is not designed to run with multiple instances (active-active mode). There are several constraints:

1. Single SM
2. Single SharpAM
3. Single UFM Telemetry
4. UFM is stateful and manages its internal state (cluster topology model) in RAM

FR#2

Persistent storage usage is required for the following:

1. Configuration files (UFM, SM, SharpAM, UFM Telemetry, Apache)
2. DB (SQlite) – history telemetry + configuration + app state
3. Operation history – logs, events, alarms

Solution Options

FR#1

Develop “ufm operator” examples, refer to:

- https://github.com/andreykaipov/active-standby-controller
- https://github.com/amelbakry/kubernetes-active-passive
- https://tunein.engineering/implementing-leader-election-for-kubernetes-pods-2477deef8f13
- https://github.com/mkudsi/ActiveStandbySingletonPod

FR2#

1. KVS DB (etcd), Config Maps

2. 3rd party Cache\DB with load-balancing HA built-in (Redis, MongoDB, etc)

Document Revision History

Date	Description of Changes
Aug 12, 2024	Added a note to Configure HA with SSH Trust
May 7, 2024	• Updated HA package installation link across the document • Updated Installation and Configuration
Feb 8, 2024	Updated Installation and Configuration
Dec 12, 2023	Updated the following sections: • Prerequisites • Installation and Configuration • Using File Configuration • NFS File Sharing • Using File Configuration
Nov 30, 2023	• Updated DRBD • Updated Installation and Configuration
Nov 5, 2023	• Updated the UFM HA package link across the document • Added Multi-Nodes Support
Aug 14, 2023	Updated installation command.
May 10, 2023	Updated the following sections: • Overview • Prerequisites • Installation and Configuration • Monitoring and Troubleshooting
Feb 6, 2023	First Release