UFM Server Health Monitoring

Table of contents

The UFM Server Health Monitoring module is a standalone module that monitors UFM resources and processes according to the settings in the */opt/ufm/files/conf/UFMHealthConfiguration.xml* file.

For example:

- Each monitored resource or process has its own failure condition (number of retries and/or timeout), which you can configure.
- If a test fails, UFM will perform a *corrective operation*, if defined for the process, for example, to restart the process. You can change the configured corrective operation. If the corrective operation is set to "None", after the defined number of failures, the *give-up* operation is performed.
- If a test reaches the configured threshold for the number of retries, the health monitoring initiates the *give-up* operation defined for the process, for example, UFM failover or stop.
- By default, events and alarms are sent when a process fails, and they are also recorded in the internal log file.

Each process runs according to its own defined schedule, which you can change in the configuration file.

Changes to the configuration file take effect only after a UFM Server restart. (It is possible to kill and run in background the process nohup python /opt/ufm/ufmhealth/UfmHealthRunner.pyo &.)

You can also use the configuration file to improve disk space management by configuring:

- How often to purge MySQL binary log files.
- When to delete compressed UFM log files (according to free disk space).

The settings in the */opt/ufm/files/conf/UFMHealthConfiguration.xml* file are also used to generate the UFM Health Report.

The following section describes the configuration file options for UFM server monitoring.

UFM Health Configuration

The UFM health configuration file contains three sections:

- Supported Operations—This section describes all the operations that can be used in tests, and their parameters.
- Supported Tests—This section describes all the tests. Each test includes:
    - The main test operation.
    - A corrective operation, if the main operation fails.
    - A give-up operation, if the main operation continues to fail after the corrective operation and defined number of retries.

  The number of retries and timeout is also configured for each test operation.
- Test Schedule - This section lists the tests in the order in which they are performed and their configured frequency.

The following table describes the default settings in the */opt/ufm/files/conf/UFMHealthConfiguration.xml* file for each test. The tests are listed in the order in which they are performed in the default configuration file.

You might need to modify the default values depending on the size of your fabric.

For example, in a large fabric, the SM might not be responsive for *sminfo* for a long time; therefore, it is recommended to increase the values for timeout and number of retries for **SMResponseTest**.

Recommended configurations for *SMResponseTest* are:

- For a fabric with 5000 nodes:
    - Number of retries = 12
    - Frequency = 10
- For a fabric with 10000 nodes:
    - Number of retries = 12
    - Frequency = 20

Test Name / Description	Test Operation	Corrective Operation (if Test Operation fails)	No. Retries / Give-up Operation	Test Frequency
CpuUsageTest Checks total CPU utilization.	CPUTest Tests that overall CPU usage does not exceed 80% (this percentage is configurable).	None If UFM Event Burst Management is enabled, it is automatically initiated when the test operation fails	1 Retry None	1 minute
AvailableDiskSpaceTest Checks available disk space.	FreeDiskTest Tests that disk space usage for */opt/ufm* does not exceed 90% (this percentage is configurable).	CleanDisk Delete compressed UFM log files under */opt/ufm*	3 Retries None	1 hour
CheckIBFabricInterface Checks state of active fabric interface.	IBInterfaceTest Tests that active fabric interface is up.	BringUpIBFabricInterface Bring up the fabric interface	3 Retries SMOrUFMFailoverOrDoNothing	35 seconds
CheckIBFabricInterfaceStandby (HA only) Checks state of fabric interface on standby.	IBInterfaceTestOnStandby Tests that fabric interface on standby is up.	None	1 Retry None	1 minute
MemoryTest Checks total memory usage.	MemoryUsageTest Tests that memory usage does not exceed 90% (this percentage is configurable).	None	1 Retry None	1 minute
SMProcessTest	SMRunningTest	RestartProcess	1 Retry	10 second

Test Name / Description	Test Operation	Corrective Operation (if Test Operation fails)	No. Retries / Give-up Operation	Test Frequency
Checks status of the OpenSM service.	Tests that the SM process is running.	Restart the SM process	UFMFailoverOrDoNothing	s
SMResponseTest Checks responsiveness of SM (when SM process is running).	SMTest Tests SM responsiveness by sending the sminfo query to SM.	None	9 Retries UFMFailoverOrDoNothing	10 seconds
IbpmTest Checks status of the IBPM (Performance Manager) service.	ProcessIsRunningTest Tests that the IBPM service is running.	RestartProcess Restart the IBPM service	3 Retries None	1 minute
ModelMainTest Checks status of the main UFM service	ProcessIsRunningTest Tests that the UFM service is running.	RestartProcess Restart the UFM service	3 Retries UFMFailoverOrDoNothing	20 seconds
HttpdTest Checks status of the httpd service.	ProcessIsRunningTest Tests that the httpd service is running.	RestartProcess Restart the httpd service	3 Retries None	20 seconds
MySqlTest Checks status of the MySql service.	ConnectToMySql Tests that the MySql service is running.	None	1 Retry UFMFailoverOrDoNothing	20 seconds
CleanMySql Purges MySql Logs	AlwaysFailTest Fails the test in order to perform the corrective action.	PurgeMySqlLogs Purge all MySql Logs on each test	1 Retry None	24 hours

Test Name / Description	Test Operation	Corrective Operation (if Test Operation fails)	No. Retries / Give-up Operation	Test Frequency
UFMServerVersionTest Checks UFM software version and build.	UfmVersionTest Returns UFM software version information.	None	1 Retry None	24 hours
UFMServerLicenseTest Checks UFM License information.	UfmLicenseTest Returns UFM License information.	None	1 Retry None	24 hours
UFMServerHAConfigurationTest (HA only) Checks the configuration on master and standby.	UfmHAConfigurationTest Returns information about the master and standby UFM servers.	None	1 Retry None	24 hours
UFMMemoryTest Checks available UFM memory.	UfmMemoryUsageTest Tests that UFM memory usage does not exceed 80% (this percentage is configurable).	None	1 Retry None	1 minute
UFMCpuUsageTest Checks UFM CPU utilization.	CPUTest Tests that UFM CPU usage does not exceed 60% (this percentage is configurable).	None	1 Retry None	1 minute
CheckDrbdTcpConnectionPerformanceTest (HA only) Checks the tcp connection	TcpConnectionPerformanceTest Tests that bandwidth is	None	2 Retry None	10 minute

Test Name / Description	Test Operation	Corrective Operation (if Test Operation fails)	No. Retries / Give-up Operation	Test Frequency
between master and standby	greater than 100 Mb/sec and latency is less than 70 usec (configurable).			

Note

The Supported Operations section of the configuration file includes additional optional operations that can be used as corrective operations or give-up operations.

UFM Core Files Tracking

To receive a notification every time OpenSM or ibpm creates a core dump, please refer to the list of all current core dumps of OpenSM and ibpm in the UFM health report.

To receive core dump notifications, do the following:

1. Set the core_dumps_directory field in the gv.cfg file to point to the location where all core dumps are created (by default, this location is set to /tmp).

   ```
   core_dumps_directory = /tmp
   ```

2. Set the naming convention for the core dump file. The name must include the directory configured in the step above.

   The convention we recommend is:

```
echo "/tmp/%t.core.%e.%p.%h" > /proc/sys/kernel/core_pattern
```

3. Make sure core dumps directory setting is persistent between reboots. Add the kernel.core_pattern parameter with the desired file name format to the /etc/systctl.conf file. Example:

```
kernel.core_pattern=/tmp/%t.core.%e.%p.%h
```

4. Configure the core file size to be unlimited.

```
ulimit -c unlimited
```

5. (Only on UFM HA master) Update the UFM configuration file gv.cfg to enable core dump tracking.

```
track_core_dumps = yes
```

Example of Health Configuration

The default configuration for the overall memory test in the *opt/ufm/files/conf/UFMHealthConfiguration.xml* file is:

```
<Test Name="MemoryTest" NumOfRetriesBeforeGiveup="3" RetryTimeoutInSeconds="10">
  <TestOperation Name="MemoryUsageTest">
    <Parameters>
      <Parameter Name="ThresholdInPercents" Value="90"/>
    </Parameters>
  </TestOperation>
  <CorrectiveOperation Name="None"/>
  <GiveupOperation Name="None"/>
```

```
</Test>
```

This configuration tests the available memory. If memory usage exceeds 90%, the test is repeated up to 3 times at 10 second intervals, or until memory usage drops to below 90%. No corrective action is taken and no action is taken after 3 retries.

To test with a usage threshold of 80%, and to initiate UFM failover or stop UFM after three retries, change the configuration to:

```
<Test Name="MemoryTest" NumOfRetriesBeforeGiveup="3" RetryTimeoutInSeconds="10">
   <TestOperation Name="MemoryUsageTest">
      <Parameters>
         <Parameter Name="ThresholdInPercents" Value="80"/>
      </Parameters>
   </TestOperation>
   <CorrectiveOperation Name="None"/>
   <GiveupOperation Name="UFMFailoverOrStop"/>
</Test>
```

Event Burst Management

UFM event burst management can lower the overall CPU usage following an event burst by suppressing events. Event burst management is configured in the *gv.cfg* configuration file.

When the overall CPU usage exceeds the threshold configured by the CpuUsageTest in the */opt/ufm/files/conf/UFMHealthConfiguration.xml* file, a High CPU Utilization event occurs.

This event initiates the UFM event burst management, which:

- Suppresses events. The default level of suppression enables critical events only.
- If, after a specified period of time (30 seconds, by default), no further High CPU Utilization event occurs, the UFM server enables all events.

To modify Event burst management configuration, change the following parameters in the gv. cfg file:

```
# The events' level in case events are suppressed (the possible levels are disable_all_events,
enable_critical_events, and enable_all_events)
# The entire feature can be turned off using the level "enable_all_events"
suppress_events_level = enable_critical_events
# The amount of time in seconds which events are suppressed
suppress_events_timeout = 30
```