



UFM Telemetry Manager (UTM) Plugin

Table of contents

Overview

Deployment

UTM Deployment Scripts

UFM Plugin Mode

Standalone Mode

Manual Deployment

UTM Configuration File

GUI

TI Management

Telemetry status fields

REST API

List of Figures

Figure 0. Communications Version 1 Modificationdate 1716900218437
Api V2

Figure 1. Add Utm Plugin Version 1 Modificationdate 1716900217697
Api V2

Figure 2. Screenshot 2024 05 05 At 14.34.07 Version 1 Modificationdate
1716900217023 Api V2

Overview

Managed telemetry is a mode of high availability and improved performance of UFM Telemetry processes.

Governed by UFM Telemetry Manager (UTM) several UFM Telemetry Instances (TIs) run on one or more machines, each collecting a subset of the cluster fabric.

UTM manages the following aspects:

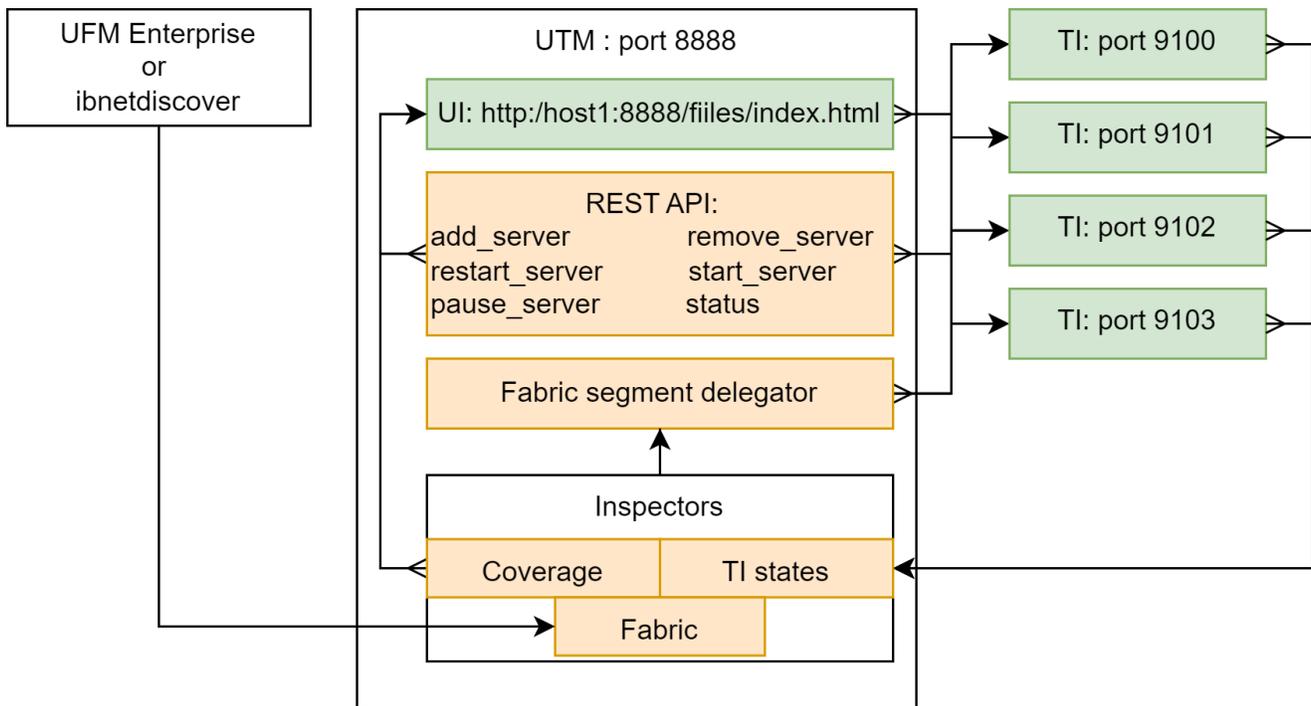
- monitoring of TI states: down, initializing, running, paused
- TI management commands: add, remove, pause, start, restart
- partitioning of fabric based on TIs health and fabric changes
- assigning fabric segments to TIs
- telemetry coverage check of a cluster

The UFM Telemetry Manager (UTM) Plugin facilitates managed telemetry in high availability mode, enhancing the performance of UFM Telemetry operations.

Under the governance of UFM Telemetry Manager (UTM), multiple UFM Telemetry Instances (TIs) are executed on one or more machines, with each TI responsible for collecting a specific portion of the cluster fabric.

Key functionalities managed by UTM include:

- Monitoring TI statuses: down, initializing, running, paused
- Execution of TI management commands: add, remove, pause, start, restart
- Fabric partitioning based on TI health and fabric changes
- Assigning fabric segments to TIs
- Verification of telemetry coverage across the cluster



Deployment

As a first step, get the UTM image:

```
docker pull mellanox/ufm-plugin-utm
```

The UTM plugin is designed to operate either as a UFM plugin or independently or in standalone mode.

In both setups, it is advisable to utilize UTM deployment scripts. These scripts streamline the process by enabling the deployment or cleanup of the entire setup with just a single command. This includes UTM, host TIs, and preparation of the Switch Telemetry image.

UTM Deployment Scripts

Get deployment scripts and examples by mounting the local folder UTM_DEPLOYMENT_SCRIPTS (/tmp/utm_deployment_scripts in this example) and running get_deployment_scripts.sh :

```
$ export UTM_DEPLOYMENT_SCRIPTS=/tmp/utm_deployment_scripts
```

```
$ docker run -v "$UTM_DEPLOYMENT_SCRIPTS:/deployment_scripts" --rm --name utm-  
deployment-scripts -ti mellanox/ufm-plugin-utm:latest /bin/sh  
/get_deployment_scripts.sh
```

The content of the script folder consists of:

- Examples - Contains run/stop scripts for both standalone and UFM plugin modes. Each example script is an example of actual deployment script usage.
- hostlist.txt - Specifies the hosts, ports, and HCAs for TIs to be deployed
- Scripts - Contains actual deployment scripts. Entry-point script `deploy_managed_telemetry.sh` triggers the rest two scripts, depending on input arguments.

```
$ cd $UTM_DEPLOYMENT_SCRIPTS  
$ tree  
.  
  examples  
    run_standalone.sh  
    run_with_plugin.sh  
    stop_standalone.sh  
    stop_with_plugin.sh  
  hostlist.txt  
  README.md  
  scripts  
    deploy_bringup.sh  
    deploy_managed_telemetry.sh  
    deploy_ufm_telemetry.sh
```

Note

All example/deployment scripts should run from the UTM_DEPLOYMENT_SCRIPTS folder.

Hostlist File

Please note the following:

- The `hostlist.txt` file should be set before running any script.
- The hostname and port will be used for communication and HCA for telemetry collection.
- To ensure optimal functionality, UTM only supports a single fabric for managed TIs, even if different HCAs on the same machine are connected to different fabrics.
- Both local and remote hosts are supported for TI deployments.

```
$ cat hostlist.txt
```

```
# List lines in the following format:
```

```
# host:port:hca
```

```
#
```

```
# where:
```

```
# - host is IP or hostname. Use localhost or 127.0.0.1 for local deployment
```

```
# - port to run telemetry on.
```

```
# - hca is the target host device from which telemetry collects. Run `ssh $host ibstat`
```

```
# to find the active device on the target host.
```

```
localhost:8123:mlx5_0
```

```
localhost:8124:mlx5_0
```

Main Deployment Script

For a more customizable setup beyond what the example scripts offer, users have the option to manually run `./scripts/deploy_managed_telemetry.sh`. This primary deployment script can deploy multiple TIs and optionally UTM as well.

Use `deploy_managed_telemetry.sh --help` to get help.

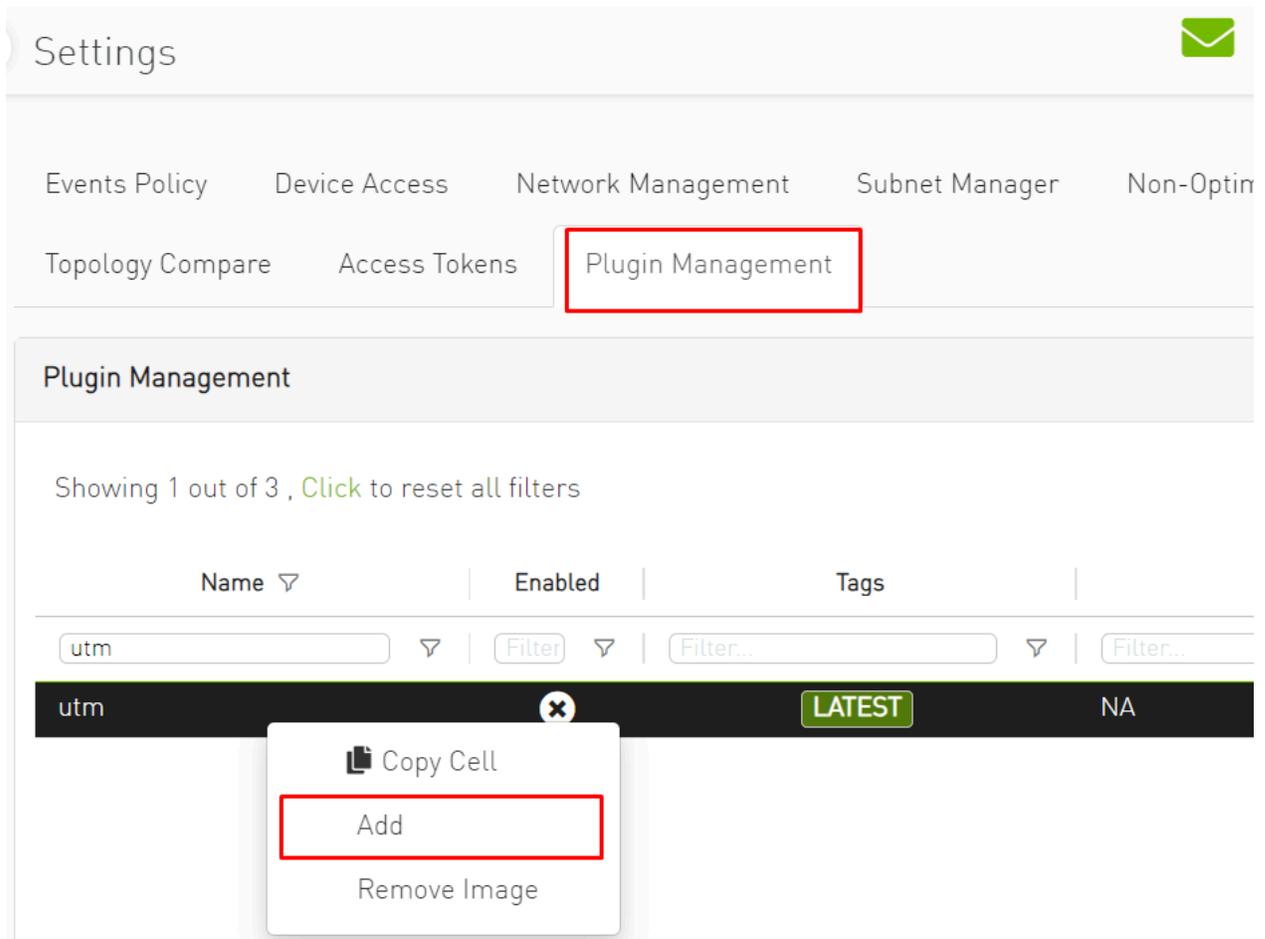
```
./deploy_managed_telemetry.sh --help
./deploy_managed_telemetry.sh options: mandatory:

mandatory:
--hostlist-file= Path to a file that lists hostname:port:hca lines
mandatory run options (use only one at the same time):
-r, --run Deploy and run managed telemetry setup
-s, --stop Stop all processes and cleanup
mandatory telemetry deployment options (use only one at the same time):
-t=, --ufmt-image= UFM telemetry docker image or tgz/tar.gz-image
or:
--bringup-package= Bringup tar.gz package
optional:
-m=, --utm-image= UTM docker image or tgz/tar.gz-image. Runs UTM only if it is set.
Configures UTM according hostlist file
--utm-as-plugin= if UTM runs as a plugin, set this flag
-d=, --data-root= Root directory for run data | Default: '/tmp/managed_telemetry/'
--switch-telem-image= Switch telemetry image (tar.gz-file or docker image). UTM will
be able to deploy it to managed switches if set
--common-data-dir= Common data folder for TIs
-h, --help Print this message
```

UFM Plugin Mode

1. Upload the UTM Docker image to the Docker registry on the machine running UFM Enterprise.

2. Navigate to the UFM web UI and click on "Settings" in the left panel.
3. Go to the "Plugin Management" tab.
4. Right-click on the UTM plugin row and select "Add."



5. Go to the option on the left called "Telemetry Status" to see the UTM UI page.
6. Prepare TI setup using utm_deployment_scripts example scripts:

1. Change directory:

```
cd $UTM_DEPLOYMENT_SCRIPTS
```

2. Open and configure hostlist.txt

3. Deploy and run TIs according to `hostlist.txt` and set these TIs to be monitored by UTM:

```
sudo ./examples/run_with_plugin.sh
```

4. To stop and cleanup TIs setup and unset TIs to be monitored by UTM:

```
sudo ./examples/stop_with_plugin.sh
```

Note

This script does not stop UTM plugin!

To stop the UTM plugin, go to "Plugin Management", right-click on the UTM plugin line and click on disable.

Default UFM Telemetry Monitoring

UFM Telemetry has high and low-frequency (Primary and Secondary, respectively) TIs that are running by default.

To enable meaningful monitoring:

1. Set `plugin_env_CLX_EXPORT_API_SHOW_STATISTICS=1` in the config files:

```
/opt/ufm/files/conf/telemetry_defaults/launch_ibdiagnet_config.ini  
/opt/ufm/files/conf/secondary_telemetry_defaults/launch_ibdiagnet_config.ini
```

2. Restart telemetry instances with the new config. If UFM Enterprise runs as a docker container, this command should be executed inside the container.

```
/etc/init.d/ufmd ufm_telemetry_restart
```

3. Give TIs some time to update performance metrics. The time depends on the update interval of default TIs.

Standalone Mode

In standalone mode, UTM periodically tracks fabric changes by itself and does not require UFM Enterprise.

Deploy via example scripts:

1. Change directory

```
cd $UTM_DEPLOYMENT_SCRIPTS
```

2. Open and configure `hostlist.txt`

3. Deploy and run TIs according to `hostlist.txt` and run UTM:

```
sudo ./examples/run_standalone.sh
```

4. To stop and cleanup TIs setup and UTM, run:

```
sudo ./examples/stop_standalone.sh
```

Manual Deployment

This section provides detailed instructions for manually deploying UTM and managed TIs to ensure coverage of all potential corner cases where the convenience script may not be effective.

UTM Deployment

UTM can be started with two docker run commands.

1. Set `utm_config`, `utm_data`, `utm_log`, and `utm_image` variables.

2. Initialize UTM config:

```
docker run -v $utm_config:/config \  
-v $utm_data:/data \  
--rm --name utm-init \  
--device=/dev/infiniband/ \  
$utm_image /init.sh
```

3. Run UTM

```
docker run -d --net=host \  
--security-opt seccomp=unconfined --cap-add=SYS_ADMIN \  
--device=/dev/infiniband/ \  
-v $utm_config:/config \  
-v $utm_data:/data \  
-v $utm_log:/log \  
--rm --name utm $utm_image
```

Managed/Standalone TIs Manual deployment

TI can be represented either as a UFM Telemetry docker container or as a UFM Telemetry Bring-up package.

To run the docker container in managed mode, `launch_ibdiagnet_config.ini` should have the following flags enabled:

```
plugin_env_CLX_EXPORT_API_SHOW_STATISTICS=1  
plugin_env_UFM_TELEMETRY_MANAGED_MODE=1
```

To run UFM Telemetry with Distributed Telemetry, enable its receiver and specify HCA to work on:

```
plugin_env_CLX_EXPORT_API_RUN_DT_RECEIVER=1
plugin_env_CLX_EXPORT_API_DT_RECEIVER_HCA=$HCA
```

To run bringup in managed mode, create `enable_managed.ini` file with the same flags and use `custom_config` option of `collection_start`:

```
collection_start custom_config=./enable_managed.ini
```

UTM Configuration File

The UTM configuration file `utm_config.ini` is placed under the configuration folder (which is referred to as `UTM_CONFIG` later on this document).

In the case of UFM plugin mode, `UTM_CONFIG= /opt/ufm/files/conf/plugins/utm/`.

In the case of standalone mode, the default value is `UTM_CONFIG =/tmp/managed_telemetry/utm/config` and can be changed via `--data-root` argument of deployment script.

When changes are made to the configuration file, UTM initiates a restart of its main process to implement the updated configuration.

Users may wish to adjust timeout and update rate configurations based on their specific setups. However, it is important to note that the remaining configurations are tailored to enable UTM to function as a UFM plugin and should not be modified.

Distributed Telemetry

To enable distributed telemetry set `dt_enable=1` in the corresponding section.

Note

Distributed Telemetry requires Switch Telemetry docker image tagged as `switch-telemetry:{version}` and placed under `$(UTM_CONFIG)/telem_files/` as `switch-telemetry_{version}.tar.gz` UTM scans this

file at its start. Example deployment scripts handle it for both UFM plugin and standalone modes.

For more details refer to [NVIDIA UFM Telemetry Documentation](#) Distributed Telemetry - Switch Telemetry Agent

GUI

To access the GUI within the UFM web UI, navigate to the Telemetry status section in the left panel.

The UI is accessible whether it is running as a part of UFM Enterprise or standalone via the endpoint: `http://127.0.0.1:8888/files/index.html`.

The GUI comprises several zones:

- The top pane displays general information and provides options to add a server name/IP and port for monitoring. Users can set the GUI refresh interval in the top right corner.
- The middle panes showcase TI groups, with the default group being basic. Unmanaged (standalone) TIs can be monitored and are placed in the "Unmanaged" group.
- Each group pane presents monitoring information for each TI.
- The bottom pane exhibits system events. Utilize the bottom right menu to navigate through the events history.

Telemetry Status

UFM Telemetry Manager Up Time: 2 mins, 4 secs
Fabric Source: http://127.0.0.1:8000 Refresh interval: 1 second

[Add Server](#)

Group name: unmanaged

URL ↑	Mode	DT receiver	Status	Uptime	Collected host/switch ports	Configured host/switch ports	Enabled/Discovered ports	Iteration time (sec)	Export time (sec)	Port counters time (sec)	Ports/sec
http://127.0.0.1:9001	standalone/host	0	Running	2 hours, 28 mins, 21 secs	0-35/-		30/173	0.000723	1e-05	0.000391	
http://127.0.0.1:9002	standalone/host	0	Running	2 days, 21 hours, 11 mins, 45 secs	173/-		30/173	0.38699	0.001255	0.037979	4555.15

Group name: default
Number of Discovered Ports: 30
Status Message: Sampled 29/30 ports. Cluster is sampled at max rate of 280.5 Hz.

URL ↑	Mode	DT receiver	Status	Uptime	Collected host/switch ports	Configured host/switch ports	Enabled/Discovered ports	Iteration time (sec)	Export time (sec)	Port counters time (sec)	Ports/sec
http://localhost:8123	managed/host	1	Running	50 secs	9/-	9/-	30/173	0.003565	8.2e-05	0.002276	3954.31
http://localhost:8124	managed/host	1	Running	50 secs	20/-	20/-	30/173	0.005695	9.7e-05	0.004665	4287.25

Events

Severity	Time	Source	Group	Description
info	2024-05-05 11:33:18.655627	http://localhost:8124	default	Updated Telemetry Instance with 20 ports
info	2024-05-05 11:33:18.652747	http://localhost:8123	default	Updated Telemetry Instance with 9 ports
info	2024-05-05 11:33:18.257047	http://localhost:8124	default	Host TI config updated. New config_hash=1dc7e0a8aedf520c0940219c45200549c30b6dbd690856b12a7e5aa8fed0788
info	2024-05-05 11:33:18.256252	http://localhost:8124	default	Host TI qp updated. New qp_info={qp_lid: 7, qp_port: 1, qp_num: 5776-5777, total_gps: 2, timestamp: 1714908793139835}
info	2024-05-05 11:33:18.249813	http://localhost:8123	default	Host TI config updated. New config_hash=8723213085d6a6e1b913b798471eb1e6ae5863bb67e905b8fda33f01b147e952

Viewing 1-5 of 35 ⏪ ⏩

TI Management

In managed mode, UTM has the capability to dispatch commands to TIs. By right-clicking on the TI line, users can:

- Pause a currently running TI. This action redistributes fabric sharding among the active TIs.
- Resume a paused TI.
- Exclude a TI from monitoring. Although the TI remains on the machine, it enters a paused state and is removed from its group. It's important to note that empty TI groups are automatically removed.

Telemetry status fields

The table below lists each column of a Telemetry Group panel:

Field Name	Description
URL	TI URL in format http://{hostname}:{port} OR http://{IP}:{port}
Mode	standalone or managed / platform

Field Name	Description
DT receiver	With or without a Distributed Telemetry receiver. If 0, cannot receive DT data from a switch TI
Status	Down, Running, Initializing, Paused, or Restarting
Uptime	TI uptime in human-readable format
Collected host/switch ports	Ports collected from the host/switch. By default data that did not change from the last sample is not being re-exported. Such data is shown in the host part as +num_old_ports. In the screenshot above, first TI of the "unmanaged" group sampled 0 new data samples and found 35 old ones. Nothing is being sampled from Distributed Telemetry, because this TI runs without DT receiver. The resulting format is: 0+35/-
Configured host/switch ports	Ports configured to be sampled from a host and corresponding switches in total. For more details refer to Distributed Telemetry documentation .
Enabled/discovered ports	Enabled and discovered ports of the Fabric.
Iteration time	Total iteration time of UFM Telemetry data collection
Export time	Export time in the last iteration of UFM Telemetry data collection. Included to Iteration time
Port counters time	Time spent only on port counters telemetry collection. Included to Iteration time
Ports/sec	Speed of new port counters data collection during the last iteration of UFM Telemetry.

REST API

All the GUI features including TI management and monitoring can be accessed via REST API.

To access the REST API command list in UFM plugin mode via UFM proxy:

```
curl -s -k https://{UFM_HOST_IP}/ufmRest/plugin/utm/help -u {user}:{pass}
```

By default , UTM runs on port 8888. To access the command list in standalone mode directly list use:

```
curl -s http://127.0.0.1:8888/help
```

- Get the status of the monitored TIs in JSON format:

```
curl http://127.0.0.1:8888/status
```

- Add TI <http://127.0.0.1:8123> to the my_group monitoring group:

```
curl 'http://127.0.0.1:8888/add_server?url=http://127.0.0.1:8123&group=my_group'
```

- Add TI <http://127.0.0.1:8123> to default monitoring group:

```
curl http://127.0.0.1:8888/add_server?url=http://127.0.0.1:8123
```

- Remove TI from monitoring (running TI will be paused):

```
curl http://127.0.0.1:8888/remove_server?url=http://127.0.0.1:8123
```

- Pause running TI:

```
curl http://127.0.0.1:8888/pause_server?url=http://127.0.0.1:8123
```

- Resume paused TI:

```
curl http://127.0.0.1:8888/start_server?url=http://127.0.0.1:8123
```

© Copyright 2024, NVIDIA. PDF Generated on 06/06/2024