Mechanism of SNMP Traps

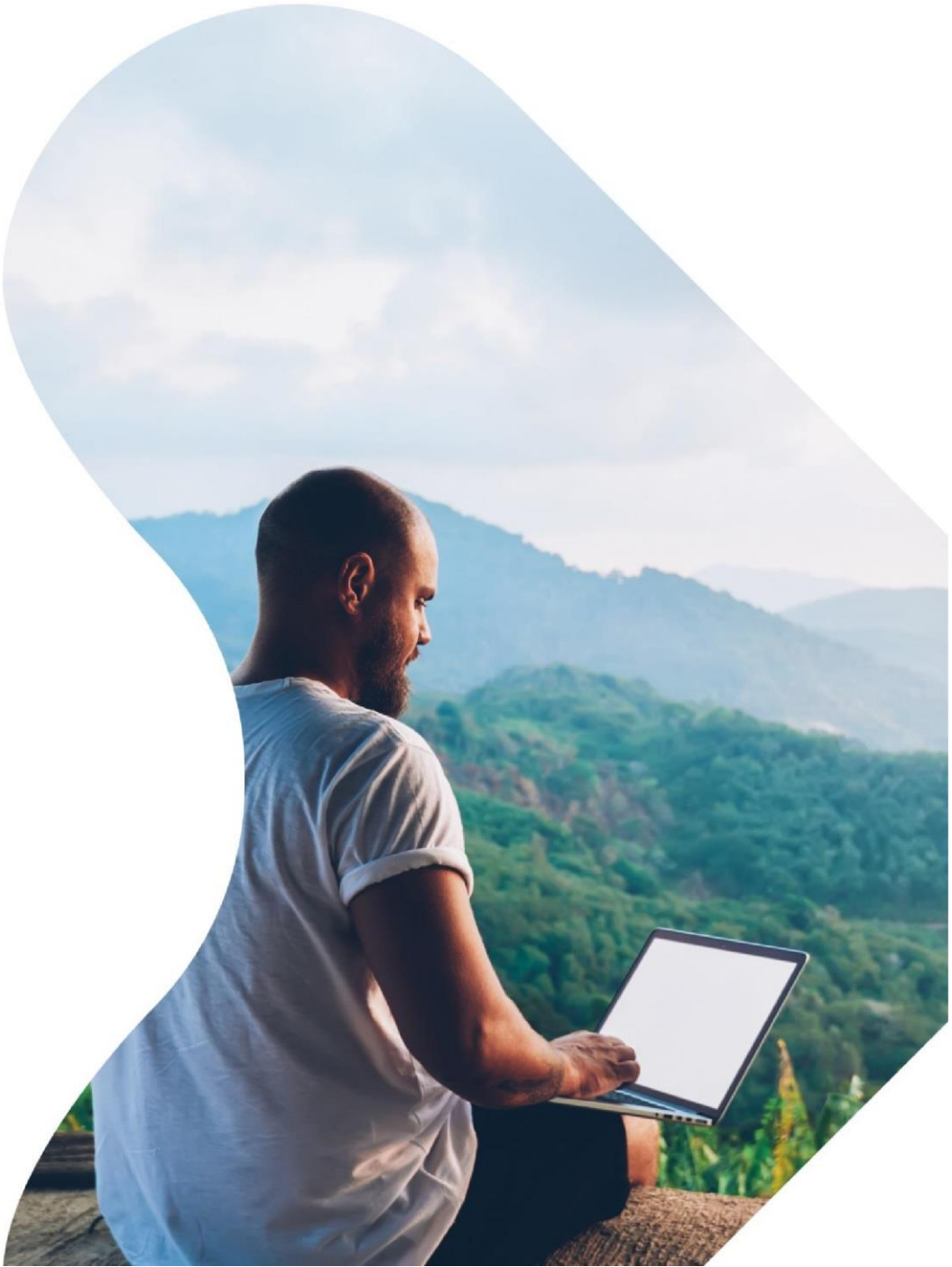


Table of Contents

CONFIDENTIALITY CLAUSE

The contents of this document are provided to "**Globe AAA"** in confidence solely for the purpose of evaluating possible business relationship.

ALL RIGHTS RESERVED

TRADEMARKS

All the brand names and other products or services mentioned in this document are identified by the trademarks or service marks of their respective owners.

DISCLAIMER

The information in this document is subject to change without notice and should not be construed as final commitment of Sterlite Technologies Ltd.

Sterlite Technologies Ltd. assumes no responsibility or makes no warranties for any errors that may appear in this document and disclaims any implied warranty of whatsoever nature.

Sterlite Technologies Ltd. shall not be responsible for any liability of any nature whatsoever resulting from or arising out of use of this document.

Your Point of Contact for this document			
Name	Ami Shah	Name	Abhishek Jaiswal
Title	Associate Manager	Title	Deputy Manager
Email	ami.kothari@stl.tech	Email	abhishek.jaiswal@stl.tech
Mobile	+91-8460823008	Mobile	+91-9981453116

Revision History

Version	Description	Name	Function	Date
1.1	SNMP Trap Mechanism	Ami Shah	Associate Manager	08/08/2022

1. SNMP Introduction

The first thing you might be asking is “What is SNMP?”

SNMP stands for Simple Network Management Protocol. SNMP is an application layer protocol that uses UDP port number 161/162. SNMP is used to monitor the network, detect network faults, and sometimes even used to configure remote devices.

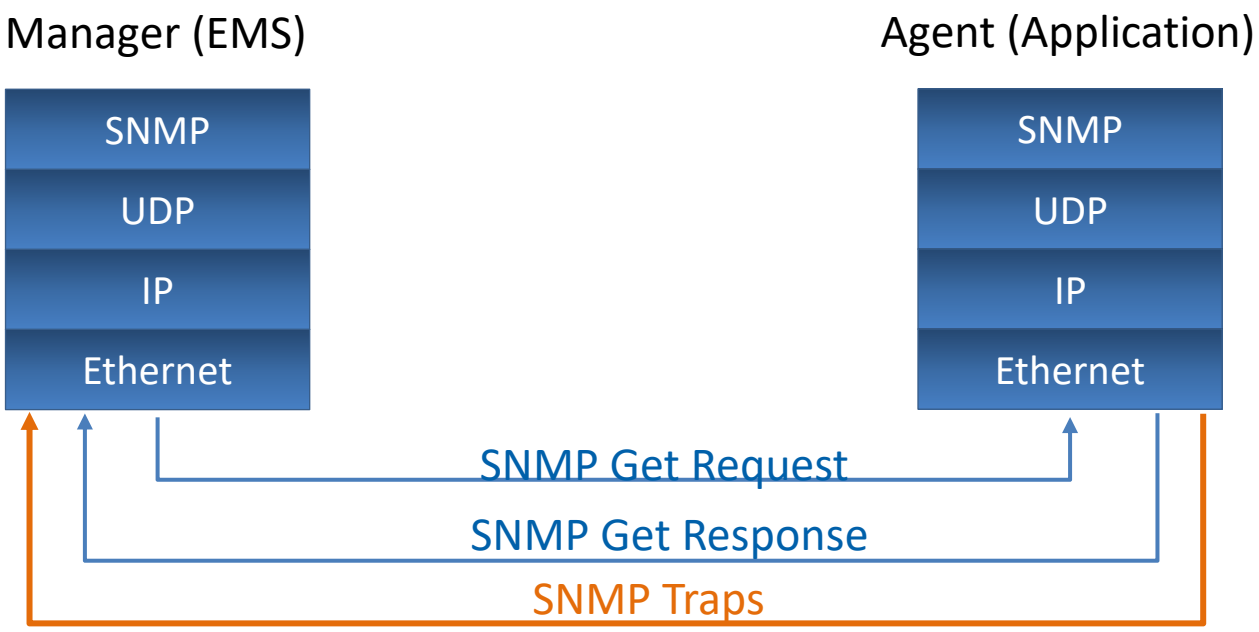
There are three parts of SNMP allowing it to function:

1) **SNMP Manager**
   It is a centralized system used to monitor network. It is also known as Network Management Station (NMS)

2) **SNMP Agent**
   It is a software management software module installed on a managed device. Managed devices can be network devices like PC, routers, switches, servers, etc.

3) **Management Information Base (MIB)**
   MIB consists of information on resources that are to be managed. This information is organized hierarchically. It consists of objects instances which are essentially variables.

Protocol Stack of SNMP



Variables of SNMP Messages

1) GetRequest
   SNMP manager sends this message to request data from the SNMP agent. It is simply used to retrieve data from SNMP agents. In response to this, the SNMP agent responds with the requested value through a response message.

2) GetNextRequest
   This message can be sent to discover what data is available on an SNMP agent. The SNMP manager can request data continuously until no more data is left. In this way, the SNMP manager can take knowledge of all the available data on SNMP agents.

3) GetBulkRequest
   This message is used to retrieve large data at once by the SNMP manager from the SNMP agent. It is introduced in SNMPv2c.

4) SetRequest
   It is used by the SNMP manager to set the value of an object instance on the SNMP agent.

5) Response
   It is a message sent from the agent upon a request from the manager. When sent in response to Get messages, it will contain the data requested. When sent in response to the Set message, it will contain the newly set value as confirmation that the value has been set.

6) Trap
   These are the message sent by the agent without being requested by the manager. It is sent when a fault has occurred.

7) InformRequest
   It was introduced in SNMPv2c, used to identify if the trap message has been received by the manager or not. The agents can be configured to send trap message continuously until it receives an Inform message. It is the same as a trap but adds an acknowledgement that the trap doesn't provide.

Version of SNMP

There are 3 versions of SNMP:

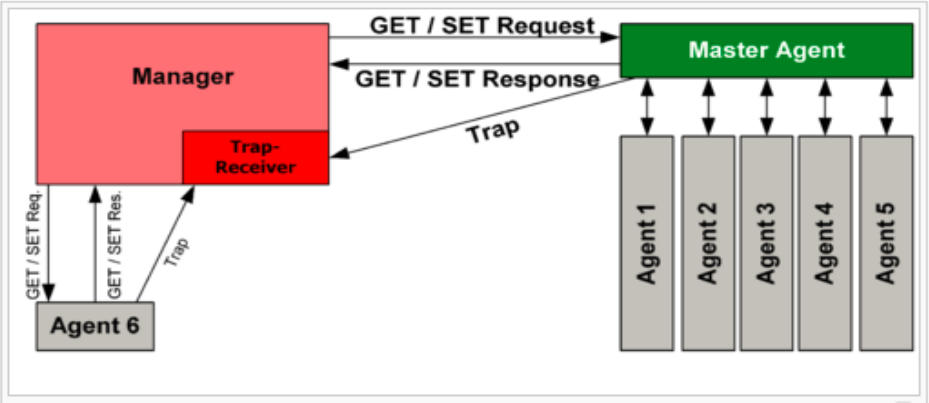
1) SNMPv1
   It uses community strings for authentication and uses UDP only.

2) SNMPv2c
   It uses community strings for authentication. It uses UDP but can be configured to use TCP.

3) SNMPv3
   It uses Hash-based MAC with MD5 or SHA for authentication and DES-56 for privacy. This version uses TCP. Therefore, the conclusion is the higher the version of SNMP, the more secure it will be.

2. SNMP Traps

An SNMP trap is any event generated and sent by the SNMP agent in a device and received by a network management system (NMS) whenever a change of state or anomaly is detected. These event messages generated by devices are received by an NMS like Site24x7, which is the trap receiver. Traps are generated instantaneously and are raw messages which an NMS has to process for network admins to understand easily.



An SNMP trap port is the port at which the manager receives the traps. This port is typically set as port number 162. However, you can change this port if necessary, and it may also differ depending on the SNMP manager you're using.

One of the complicated things about SNMP traps is they're not always effective at alerting you when major errors have occurred. For example, sometimes the device agent will send out an SNMP trap for a minor issue and miss a major problem capable of bringing your entire network down. For instance, if the device experiences a fatal issue shutting down the entire device, the SNMP agent can no longer work either (and no SNMP trap will be sent out).

SNMP traps are sent out in a particular format, showing a time, an identifier, and a value. The time shows when the error occurred. The identifier is from the MIB and is called an "OID," also known as an Object Identifier. The OID represents an element of the device being monitored, such as temperature, CPU function, or memory (or even whether the printer ink is running low). These OIDs can help you to pinpoint the problem. This information is critical when you're monitoring a large network where a single device failure can cause a cascade of issues.

3. SNMP Configuration on EMS

Enabling trap engine on EMS Server:

1. Go to /motadata/motadata/config/ by using below command :
   cd /motadata/motadata/config/

```
root@ubuntu: /motadata/motadata/config
root@ubuntu:/motadata/motadata/config# cd /motadata/motadata/config
root@ubuntu:/motadata/motadata/config# pwd
/motadata/motadata/config
root@ubuntu:/motadata/motadata/config#
```

2. vim motadata-conf.yml

```
root@ubuntu: /motadata/motadata/config
root@ubuntu:/motadata/motadata/config# vim motadata-conf.yml
```

3. Replace trap engine: No to Yes

```
#trap engine enable or not (yes/no)
trap-engine: yes
```

4. Restart EMS server and trap engine will be ON

```
root@ubuntu: /motadata/motadata/config
   Loaded: loaded (/lib/systemd/system/motadata.service; disabled; vendor preset: enabled)
   Active: active (running) since Tue 2019-09-24 15:10:42 IST; 1h 29min ago
 Main PID: 19030 (motadata)
    Tasks: 654
   Memory: 5.3G
      CPU: 12min 29.589s
   CGroup: /system.slice/motadata.service
           |-19030 /motadata/motadata/motadata &
           |-19061 redis-server *:6379
           |-19064 ./nsqlookupd -tcp-address 0.0.0.0:4160 -http-address 0.0.0.0:4161 -broadcast-
           |-19072 ./nsqd -tcp-address 0.0.0.0:4150 -lookupd-tcp-address 0.0.0.0:4160 -http-addr
           |-19126 /motadata/motadata/jdk/bin/java -server -Xms1000m -Xmx2000m -XX:+UseCompresse
           |-19242 /motadata/motadata/motadata-rpe-master
           |-19246 /motadata/motadata/motadata-metric Metric-a94d6567-afc0-4557-866d-6d8eceda6e4
           |-19259 /motadata/motadata/motadata-discovery Discovery-cc81e64e-ab32-4cbf-ae08-28593
           |-19264 /motadata/motadata/motadata-ncm NCM Collector-5bcec216-0adc-46fc-9851-fa87246
Trap Engine is started |-19268 /motadata/motadata/motadata-sla SLA Analyzer-386e9430-874e-4f5f-9f7d-affcbbda
           |-19273 /motadata/motadata/motadata-trap Trap Analyzer-009951ea-e2bc-4d11-9ec3-9d6e9f
           |-19301 python3.7 -W ignore /motadata/motadata/motadata python engine/bootstrap.py lo
           |-19322 python3.7 -W ignore /motadata/motadata/motadata python engine/bootstrap.py lo
           |-19329 python3.7 -W ignore /motadata/motadata/motadata python engine/pingbootstrap.p
           |-19347 /motadata/motadata/motadata-log Log Analyzer-36ab5ee6-bc63-4f3a-b016-3a61a240
           `-19353 /motadata/motadata/motadata-job Job-8eb7d101-f165-4b7d-9278-6d45c0e5a38e|!!|5
```

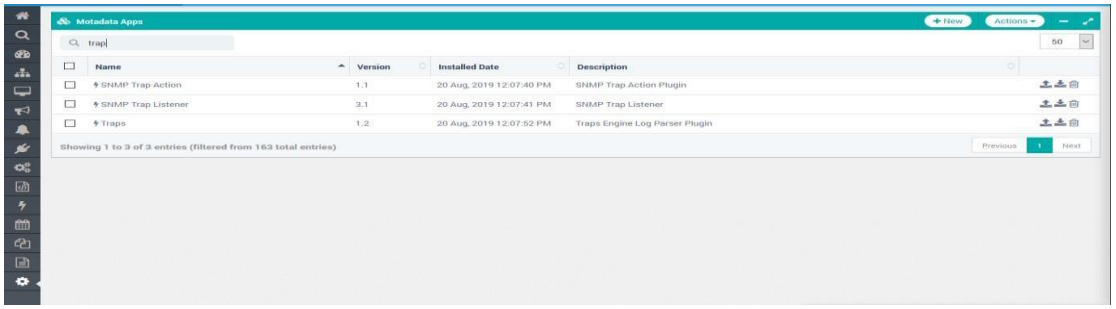
5. Tcpdump command to check the traps are reaching to EMS or not:

   E.g- tcpdump -i any src 172.16.10.1 and port 162 -- where, 162 is default trap port used in EMS

6. Netstat command to check 162 port is open or not

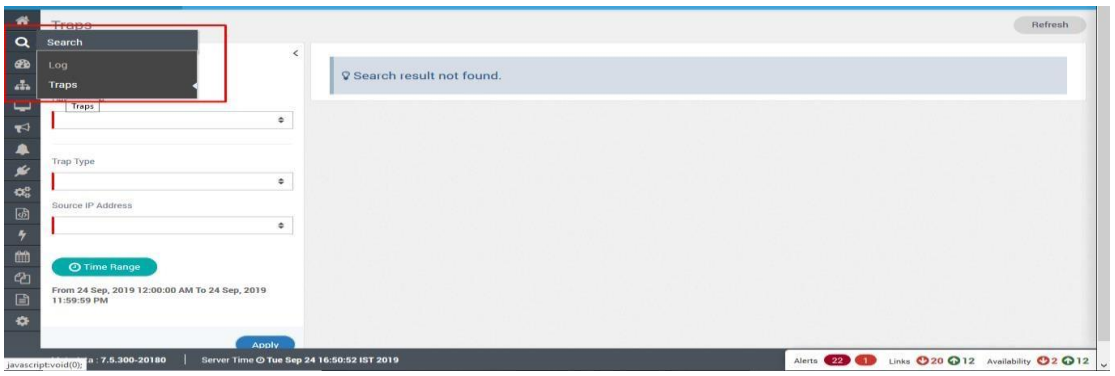
```
root@ubuntu: /motadata/motadata/config
root@ubuntu:/motadata/motadata/config# netstat -apn | grep 162
udp        0      0 0.0.0.0:162             0.0.0.0:*                           1247/snmptrapd
unix  3      [ ]         STREAM     CONNECTED     16256    1/init              /run/systemd/jour
nal/stdout
unix  3      [ ]         STREAM     CONNECTED     16255    1/init              /run/systemd/jour
nal/stdout
unix  2      [ ]         DGRAM                    19472    1162/iscsid
unix  3      [ ]         STREAM     CONNECTED     16257    1/init              /run/systemd/jour
nal/stdout
```

7. Plugins for traps :
    a. SNMP Trap Action - It will forward all the generated alerts in EMS in the form of traps to the destination device using snmp v2 protocol
    b. SNMP Trap Listener - For licensing - In newer version the trap is free.
    c. Traps : Act as forwarder as well as listener in newer version of EMS

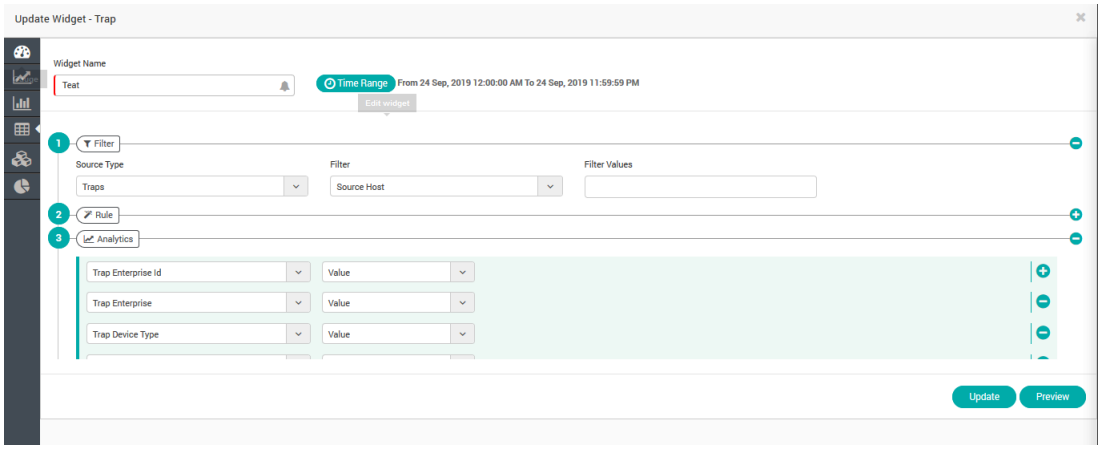


4. GUI - TRAP Module

1. Go to Search > Traps

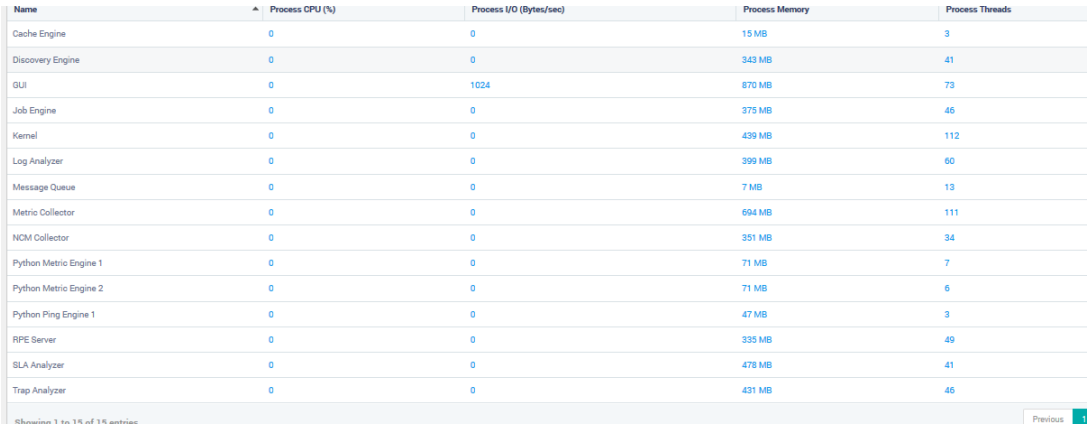


2. Widget creation example for Traps:



3. Trap engine utilization from GUI

   Go to Motadata Support then Core process statistics



Name	Process CPU (%)	Process I/O (Bytes/sec)	Process Memory	Process Threads
Cache Engine	0	0	15 MB	3
Discovery Engine	0	0	343 MB	41
GUI	0	1024	870 MB	73
Job Engine	0	0	375 MB	46
Kernel	0	0	439 MB	112
Log Analyzer	0	0	399 MB	60
Message Queue	0	0	7 MB	13
Metric Collector	0	0	694 MB	111
NCM Collector	0	0	351 MB	34
Python Metric Engine 1	0	0	71 MB	7
Python Metric Engine 2	0	0	71 MB	6
Python Ping Engine 1	0	0	47 MB	3
RPE Server	0	0	335 MB	49
SLA Analyzer	0	0	478 MB	41
Trap Analyzer	0	0	431 MB	46

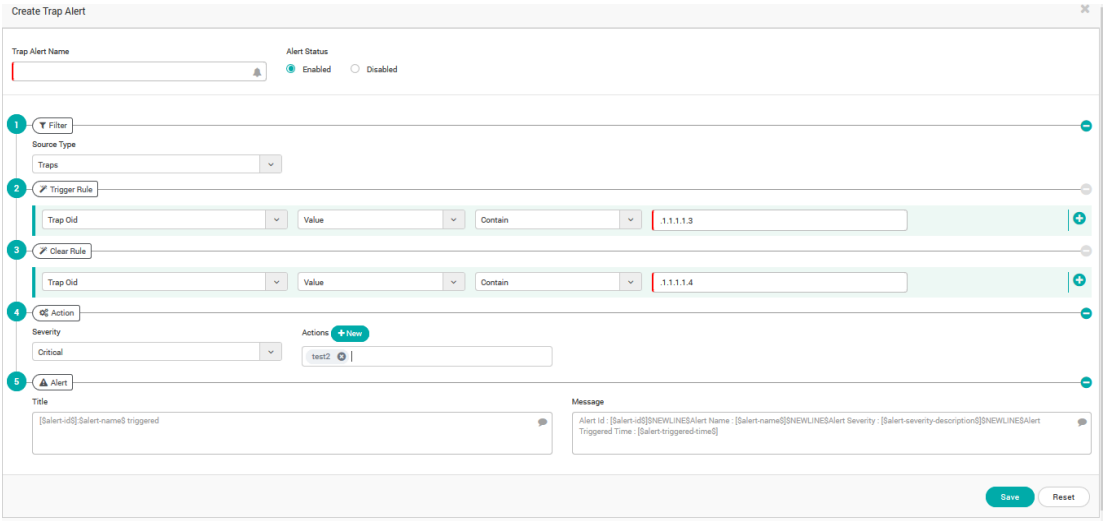
4. Trap Alert Creation example :

Note : There is difference OID's used for incident happened and when it was clear.

F.g - Hard disk failure:- .1.1.1.1.1.3

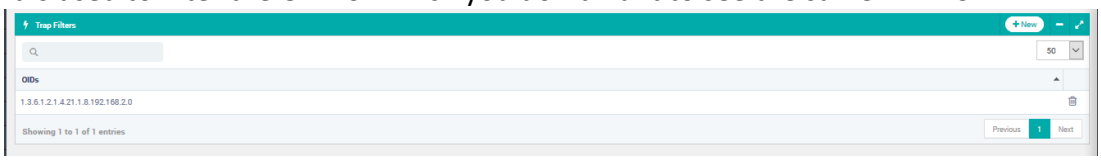
Hard disk fixed : - .1.1.1.1.1.4

So while creating alert, use different OID's.



5. Trap Filer from Admin panel

It is used to filter the OID for which you don't want to see the same in EMS.



6. Trap Forwarder is used to forward the traps from NMS to third party tool and you need to enter the IP and port onwhich you want to forward the Trap