

Fault Management Configuration Guide

Cloud Execution Environment

USER GUIDE

Copyright

© Ericsson AB 2016. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Trademark List

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.



Contents

1	Overview	1
1.1	Description	1
1.2	Prerequisites	1
2	REST Interface	3
2.1	Get the Existing Configuration for SNMP Trap Destinations	3
2.2	Add New SNMP Trap Destination Configuration	4
2.3	Remove an Existing SNMP Trap Destination Configuration	5
3	Watchmen CLI	7
3.1	Syntax	7
3.2	Subcommands	8
4	SNMP Trap Destination	12
4.1	Syntax	12
4.2	Examples	12





1 Overview

This User Guide (UG) describes the configuration of the Watchmen Fault Management in the Cloud Execution Environment (CEE).

The configuration tasks described in this UG are not part of the installation process. These configuration activities can be performed on a stable system.

1.1 Description

The Watchmen component in the CEE can send Simple Network Management Protocol (SNMP) traps through its northbound interface to Network Management Stations (NMSs) in order to signal the occurrence and cease of abnormal conditions. The Fault Management Northbound API operates based on the Ericsson Alarm MIB.

Note: An NMS executes applications that monitor and control managed devices. One or more NMSs can be used on any managed network.

The configuration can be performed by using either of the following interfaces:

- REST interface: See Section 2 on page 3.
- Watchmen CLI: See Section 3 on page 7.

In the commands used for the configuration through the interfaces above, the SNMP trap destination is specified by using a Net-SNMP command, `snmptrap`, as an argument of the commands. The argument contains the complete `snmptrap` command with its own arguments. Section 4 on page 12 describes the configuration of the `snmptrap` command.

1.1.1 Limitations

The following limitation applies:

- The only supported SNMP version is SNMPv2c.

1.2 Prerequisites

This section states the prerequisites that have to be fulfilled.

1.2.1 Documents

Ensure that the following document has been read:



- Documentation for the used NMS or NMSs

1.2.2

Conditions

Ensure that the following conditions are met:

- An NMS that is capable of receiving SNMP traps is available for use.
- The SNMP trap parameter settings required by the used NMS are known.



2 REST Interface

This section describes how to configure the Watchmen Fault Management through the REST interface.

The Watchmen WSGI (Web Server Gateway Interface) Server is listening on the 8052 port at the HA address of the CICs.

The required user role in Keystone is `watchmen`.

The following operations are supported:

- Get the existing configuration for SNMP trap destinations
See Section 2.1 on page 3.
- Add new SNMP trap destination configuration
See Section 2.2 on page 4.
- Remove an existing SNMP trap destination configuration
See Section 2.3 on page 5.

2.1 Get the Existing Configuration for SNMP Trap Destinations

Method: GET

URL: /v1/snmp_trap_config

Description: Returns the current configuration for SNMP trap destinations. See Section 4 on page 12 for more information.

Note: If no trap destinations are configured, an empty list is returned.

Normal response code:

200

Error codes: 401, 403, 500

Parameters: —

2.1.1 JSON Response

2.1.1.1 Syntax

```
{"snmp_trap_config":
```



```

[
  {
    config_id: <id>,
    command: "configuration for the first SNMP trap⇒
              destination",
    appendinfo: <bool>
  }
  {
    config_id: <id>,
    command: "configuration for the second SNMP trap⇒
              destination",
    appendinfo: <bool>
  }
]
}

```

See Section 4 on page 12 for more information about the SNMP trap destination configuration in the `command` row of the JSON response.

2.1.1.2 Example

```

{"snmp_trap_config":
  [
    {
      config_id: 23,
      command: "snmptrap -v 2c -c nfvm_hwm -M /usr/lib/erlang/lib/snmp-4.25/⇒
mibs:/usr/share/snmp/mibs:/usr/lib/python2.7/dist-packages/watchmen/service/⇒
mibs 129.152.57.36:162",
      appendinfo: true
    }
    {
      config_id: 34,
      command: "snmptrap -v 2c -c community003 -M /usr/lib/erlang/lib/⇒
snmp-4.25/mibs:/usr/share/snmp/mibs:/usr/lib/python2.7/dist-packages/watchmen/⇒
service/mibs 192.168.0.7",
      appendinfo: false
    }
  ]
}

```

2.2 Add New SNMP Trap Destination Configuration

Method:	POST
URL:	/v1/snmp_commands
Description:	Adds a new SNMP trap destination to the configuration. The request body must contain a trap destination configuration in JSON format. See Section 4 on page 12 for more information.

**Normal response code:**

200

Error codes:

401, 403, 500

Parameters:

—

2.2.1 JSON Request**2.2.1.1 Syntax**

```
{ "snmp_trap_config":
  {
    command: "SNMP trap destination configuration",
    appendinfo: <bool>
  }
}
```

See Section 4 on page 12 for more information about the configuration of the SNMP trap destination in the `command` row of the JSON request.

2.2.1.2 Example

```
{ "snmp_trap_config":
  {
    command: "snmptrap -v 2c -c nf_v_hwm -M /usr/lib/erlang/lib/snmp-4.25/⇒
mibs:/usr/share/snmp/mibs:/usr/lib/python2.7/dist-packages/watchmen/service/⇒
mibs 129.152.57.36:162",
    appendinfo: true
  }
}
```

2.3 Remove an Existing SNMP Trap Destination Configuration**Method:**

DELETE

URL:

/v1/snmp_trap_config/<id>

Description:

Removes the SNMP trap destination configuration identified by the `id` in the URL.

Normal response code:

200

Error codes:

401, 403, 500

Parameters:

—



2.3.1 Example for the URL

`/v1/snmp_trap_config/23`



3 Watchmen CLI

The Watchmen CLI can be reached from the CICs by using a terminal.

SNMP configuration through the Watchmen CLI can be performed by using the **watchmen-client** command. To execute this command, Keystone requires a user with the `watchmen` OpenStack role.

Note: Watchmen CLI supports `bash` completion.

3.1 Syntax

See the syntax of the main command below. It does not contain the syntax of subcommands, only the subcommand names. The syntax of the subcommands is included in Section 3.2 on page 8.

```
watchmen-client [-h]
watchmen-client [--help]

watchmen-client [--os-username <OS_Username>]
                [--os-password <OS_Password>]
                [--os-tenant-name <OS_Tenant_Name>]
                snmp-trap-config-list

watchmen-client [--os-username <OS_Username>]
                [--os-password <OS_Password>]
                [--os-tenant-name <OS_Tenant_Name>]
                snmp-trap-config-add

watchmen-client [--os-username <OS_Username>]
                [--os-password <OS_Password>]
                [--os-tenant-name <OS_Tenant_Name>]
                snmp-trap-config-remove
```

Note: The user must have the `watchmen` OpenStack role.

3.1.1 Optional Arguments

When the main command **watchmen-client** is used, the following optional argument can be used:

-h, --help	Displays information about the watchmen-client command.
-------------------	--

When any of the subcommands **snmp-trap-config-list**, **snmp-trap-config-add**, or **snmp-trap-config-remove** is used, the following optional attributes can be used:



--os-username	Specifies the OS username. It can be any user that has <code>watchmen</code> role.
--os-password	Specifies the OS password.
--os-tenant-name	Specifies the OS tenant name.

Note: Passing `--os-token` instead of the three arguments above is not supported.

The `--os-username`, `--os-password`, and `--os-tenant-name` must be specified in the command each time they are required by a subcommand.

Note: Watchmen CLI also supports `OS_USERNAME`, `OS_PASSWORD`, `OS_TENANT_NAME` OpenStack environment variables.

3.2 Subcommands

The following sections describe the subcommands of the `watchmen-client` command.

3.2.1 List SNMP Trap Configuration

snmp-trap-config-list

This subcommand returns the current configuration for all SNMP trap destinations.

3.2.1.1 Syntax

```
watchmen-client --os-username <username> ⇒  
--os-password <password> --os-tenant-name <tenant_name> ⇒  
snmp-trap-config-list
```

Note: The user must have the `watchmen` OpenStack role.

3.2.1.2 Required Arguments

Not applicable.

3.2.1.3 Optional Arguments

Not applicable.

3.2.1.4 Example

This example shows the command followed by the printout of the current configuration.



```
watchmen-client --os-username user004 --os-password xhT5s63 --os-tenant-name admin =>
snmp-trap-config-list
```

Config id	Command	AppendInfo
1	snmptrap -v 2c -c nf_v_hwm -M /usr/share/snmp/mibs 192.168.1.5	True
2	snmptrap -v 2c -c nf_v_hwm -M /usr/share/snmp/mibs 192.168.0.7	False

See Section 4 on page 12 for more information about the SNMP trap destination configurations in the `Command` column of the printout.

3.2.2 Add SNMP Trap Configuration

snmp-trap-config-add

This subcommand adds a new SNMP trap destination to the configuration with all required SNMP options. See Section 4 on page 12 for more information.

3.2.2.1 Syntax

```
watchmen-client --os-username <username> =>
--os-password <password> --os-tenant-name <tenant_name> =>
snmp-trap-config-add --command <command_string> =>
--enable-append-info
```

```
watchmen-client --os-username <username> =>
--os-password <password> --os-tenant-name <tenant_name> =>
snmp-trap-config-add -c <command_string> -e
```

Note: The user must have the `watchmen` OpenStack role.

See Section 4 on page 12 for more information about configuring the SNMP trap destination contained by the `command_string` in the add request.

3.2.2.2 Required Argument

--command, -c Specifies the SNMP trap destination to be added. The `command_string` contains all the required parameters. See Section 4 on page 12 for more information.



3.2.2.3 Optional Argument

--enable-append-info, -e

In case of long additional text in alarms or alerts, this argument enables the sending of `eriAlarmAppendInfo` or `eriAlarmAppendAlertInfo` traps to append further information to an existing alarm.

Note: The management system must be prepared for appending text to the additional text, and not to replace the additional text instead.

Characters of the additional text from position 256 to 510 can be appended by using this option.

3.2.2.4 Examples

```
watchmen-client --os-username username005 --os-password kl26Z2 =>
--os-tenant-name admin snmp-trap-config-add --command "snmptrap -v 2c -c =>
nf_v_hwm -M /usr/lib/erlang/lib/snmp-4.25/mibs:/usr/share/snmp/mibs:/usr/lib/=>
python2.7/dist-packages/watchmen/service/mibs 129.152.57.36:162" =>
--enable-append-info

watchmen-client --os-username username005 --os-password kl26Z2 =>
--os-tenant-name admin snmp-trap-config-add -c "snmptrap -v 2c -c nf_v_hwm -M =>
/usr/lib/erlang/lib/snmp-4.25/mibs:/usr/share/snmp/mibs:/usr/lib/python2.7/=>
dist-packages/watchmen/service/mibs 129.152.57.36:162" -e
```

3.2.3 Remove SNMP Trap Configuration

snmp-trap-config-remove

This subcommand removes the SNMP trap destination configuration specified by the `id`.

3.2.3.1 Syntax

```
watchmen-client --os-username <username>=>
--os-password <password> --os-tenant-name <tenant_name> =>
snmp-trap-config-remove --config-id <id>

watchmen-client --os-username <username>=>
--os-password <password> --os-tenant-name <tenant_name> =>
snmp-trap-config-remove -id <id>
```

Note: The user must have the `watchmen` OpenStack role.



3.2.3.2 Required Argument

--config-id, -id Specifies the identifier of the SNMP trap destination configuration to be removed.

3.2.3.3 Optional Arguments

Not applicable.

3.2.3.4 Examples

```
watchmen-client --os-username user009 --os-password 25KJ3d ⇒  
--os-tenant-name admin snmp-trap-config-remove --config-id 3
```

```
watchmen-client --os-username user009 --os-password 25KJ3d ⇒  
--os-tenant-name admin snmp-trap-config-remove -id 3
```



4 SNMP Trap Destination

In the commands used for the Fault Management configuration, the SNMP trap destination is specified by using a Net-SNMP command, `snmptrap`, as an argument of the commands. The argument contains the complete `snmptrap` command with its own arguments, and it must be configured in Watchmen for each trap destination.

This section describes the configuration of the `snmptrap` command.

For more information about the `snmptrap` command, refer to the Net-SNMP documentation.

The only supported SNMP version is SNMPv2c.

4.1 Syntax

```
snmptrap -v 2c -c <community> -M <mib_path> <destination>
```

-v 2c The SNMP version. The only supported version is v2c.

-M <mib_path> Library locations where the MIB files are stored

-c <community> The community string for the authentication

<destination> The address of the host to which the trap must be sent

The default sender (client) address is the address of one of the CICs. In case the public VIP address has to be used, then it has to be specified with the `clientaddr` attribute and the `snmptrap` command has to be issued as follows:

```
sudo /sbin/ip netns exec haproxy ⇒  
/usr/bin/snmptrap --clientaddr=<Public VIP> -v 2c ⇒  
-c <community> -M <mib_path> <destination>
```

4.2 Examples

```
snmptrap -v 2c -c nf_v_hwm -M /usr/lib/erlang/lib/snmp-4.25/mibs:/usr/share/⇒  
snmp/mibs:/usr/lib/python2.7/dist-packages/watchmen/service/⇒  
mibs 129.152.57.36:162
```

```
snmptrap -v 2c -c community003 -M /usr/lib/erlang/lib/snmp-4.25/mibs:/usr/⇒  
share/snmp/mibs:/usr/lib/python2.7/dist-packages/watchmen/service/⇒  
mibs 192.168.0.7
```