opEvents - Solution Guide - System Configuration for a Production Environment

Purpose

This Solution Guide will provide a detailed, single reference for configuring opEvents for use in a production environment.

Scenario

The Opmantek Administrator is deploying opEvents 4.x in a production environment and desires to enable as much functionality as possible, beyond the default settings provided by the installer.

Prerequisites

This Solution Guide assumes the installation, setup and configuration of NMIS9.x and opEvents 4.x on a single-server. Additional Solution Guides are available for multi-tiered deployments using opHA.

System Resources and Install

Determine Starting Hardware Specifications

A minimum install of opEvents 4 requires NMIS 9 as the base and a minimum of 4vCPU and 8GBRam. However, operational system resource requirements will depend highly on the number of devices, interfaces being collected, additional syslog processing and maximum number of events / minute to bee handled. More information can be found HERE: Server Requirements

Test Login Using Default Credentials

After you have completed the install of opEvents you should test the login using the default credentials. Default Credentials (Passwords) for NMIS9 VM

Setup and Configure opEvents (Initial)

Consolidate your Browser Cookies

NMIS9 and opEvents4 use separate browser cookies to maintain your login information. This means that if you log into NMIS, you would need to login again to opEvents. You can, however configure both products to share the same browser cookie.

Configure NMIS9:

1. Open /usr/local/nmis9/conf/Config.nmis
2. Locate the 'auth_cookie_flavor' setting under 'authentication' and ensure it is set to 'omk'
3. Locate the 'auth_web_key' setting under 'authentication', this should be set to a long, random string. Make sure you COPY this as you will need it later.
4. Save and close the file.
5. If you needed to change either of these settings make sure you restart the nmis9d daemon

Configure opEvents4:

1. Open /usr/local/omk/conf/opCommon.json
2. Locate the "omkd_secrets" setting, **change this to match** the setting you copied from 'auth_web_key' earlier.
3. Save and close the file.
4. If you changed this setting make sure to restart the omkd daemon

Setup Summary Reports

Scenario: The Operations Manager has formed a Tiger Team focused on improving the reliability, performance, and security of the network. He would like a report generated daily sent to the Tiger Team and NOC Manager, and a Weekly report to the Network Architect and himself. These reports will be used to task staff with investigation and remediation.

opEvents automatically creates detailed reports on processed events on a Daily, Weekly, and Monthly basis. opEvents can be configured to email these reports upon creation.

Report Configuration is covered in detail HERE: opEvents Summary Reports

Adjust Node Summary Field List

NMIS9 shares a limited number of device fields with opEvents by default. These are defined in the 'node_summary_field_list' in the NMIS9 configuration file. You should review this setting in /usr/local/nmis9/conf/Config.nmis as you may want to append this list to include additional fields you may want to make available to opEvents for event handling or decision making.

Configure opEvents for Device Dependency Mapping

If you have opCharts installed with NMIS9 and opEvents4 you should configure opCharts for automated dependency mapping

Scenario: A firewall has been misconfigured, preventing traffic from passing through the firewall. As NMIS attempts a Ping or SNMP/WMI Collection on devices behind the firewall individual Node Down events are raised for each non-responding device. Dependency Mapping has been configured, and opCharts has mapped parent/child relationships for devices beyond the firewall, identifying common routings like server switch router firewall. These relationships are stored with each device's details in NMIS and are used when processing the Node Down event; child outages are suppressed in favor of the parent outage until the only reported outage passed to opEvents is that of the Firewall.

opCharts must be configured to map device dependency, NMIS to hold events while parent/child dependency is determined, and opEvents to process the modified event stream:

- opCharts Node Dependency Management (Root Cause Analysis)
- Leveraging NMIS Dependency in opEvents

Send an Email from an Event

Scenario: The NOC manager would like to receive an automated email for any service impacting event when the event is opened, and again when it closes or is Acknowledged.

One of the simplest responses to an event is to generate an email. You might generate an email for every event, just for events with a certain minimum Priority, only during certain times of day (or days of the week), etc. opEvents allows your Event Actions rules to be as generic or specific as you need them to be.

Before you can generate an email you will need to:

1. Define an email server to send the email through
2. Create at least one contact that emails can be sent to
3. Test and confirm the configured email server works and the configured contact receives an email

This process is well defined in this Solution Guide: opEvents - Solution Guide - Setup Email Notifications and Other Actions

Add Troubleshooting Buttons

> **Advanced Topic**
>
> This is an advanced topic and requires the administrator to be confident copy, and editing text files at the Linux command line.

It can be helpful to include troubleshooting tools into opEvents so engineering users can stay in a single window while working an issue without having to open separate command line windows or other tools. One way of doing this is by adding some simple Programmable Buttons that can be turned on/off based on various device or event criteria, or enabled all the time for every event.

The basic of configuring this feature are covered in detail HERE: opEvents Programmable Button Actions

For a starting point I usually include buttons to run a Ping, MTR, NMIS9 Polling Summary, and open a help desk ticket.

Remember that these buttons call entries located in the "script" section of /usr/local/omk/conf/EventActions.json, which can be edited interactively through the opEvents' GUI by selecting System -> Edit Event Actions from the menu.

More on Event Actions can be found HERE: Event Actions and Escalation

Add/Edit Correlation Rules

Correlation rules allow opEvents to consolidate multiple events reported within a specified window of time based on selected common fields.

Scenario: During a regional power outage several devices located within that area loose power and due to lack of battery backup or alternate power source shutdown. As NMIS attempts a Ping or SNMP/WMI Collection on these devices a Node Down event is raised for each non-responding device. Normally, opEvents would see these as individual events, creating multiple notifications. By enabling a Location based Correlation rule opEvents instead groups these Node Down events into a single event making it easier for the engineer to see the scope of the problem. This approach also has the added benefit of reducing ancillary notifications, like EMAIL, SMS/Text, and opening help desk tickets.

This solution guide provides an excellent example of configuring a Correlation based on a device's Location field: opEvents - Solution Guide - Event Consolidation Based on Location

A more in-depth discussion of event correlation rules is covered HERE: Event Correlation