



IBM ECM System Monitor Field Guides

Integration with Message Channels

August 29th, 2023
CENIT AG
Michael Wohland

Content

- Introduction 3
- Overview 3
- Disclaimer 3
- Purpose of this guide..... 4
- Requirements..... 4
- Step by Step setup 5
 - 1. Prepare the message channel to accept webhooks..... 5
 - 2. Preparing the JSON for forwarding the ESM information.....11
 - 3. Setting up the Task13
- Contact Information20

Introduction

Overview

This guide describes in detail, how IBM ECM System Monitor (ESM) can be integrated with Message Channels to forward ESM incidents.

Disclaimer

The content of this document is based on ESM Version 5.5.9.1-000. The descriptions and guidelines in this document are for informational purposes only. Up-to-dateness, content completeness, appropriateness and validity for all possible scenarios cannot be guaranteed. All information is provided on an as-is basis. The author is not liable for any errors or omissions in this document or any losses, injuries and damages arising from its use.

If you are planning to setup or configure ESM or to adjust an existing installation, it is absolutely necessary to take into account current security whitepapers, release notes and announcements from the official IBM ECM System Monitor product documentation website.

Purpose of this guide

Message Channels are more and more used to interact as a team. These channels can be filled with incident information that is collected by ESM. Every member of the team is informed and the team can react and decide what to do with this information.

Requirements

- ESM Version 5.5.9.1-000 (already installed) – any ESM Version that has the Webhook Task integrated should work
- Availability of a message channel (e.g. MS Teams or Slack)

Step by Step setup

1. Prepare the message channel to accept webhooks

Example for MS Teams:

For MS Teams there is a webpage available that describes how an incoming webhook connector can be set up for channels. The page URL is:

<https://learn.microsoft.com/en-us/microsoftteams/platform/webhooks-and-connectors/how-to/add-incoming-webhook?tabs=dotnet>

At some point of the setup you will receive a webhook URL, which you need for the integration with ESM. Note it or copy it to a document. The URL looks like this:

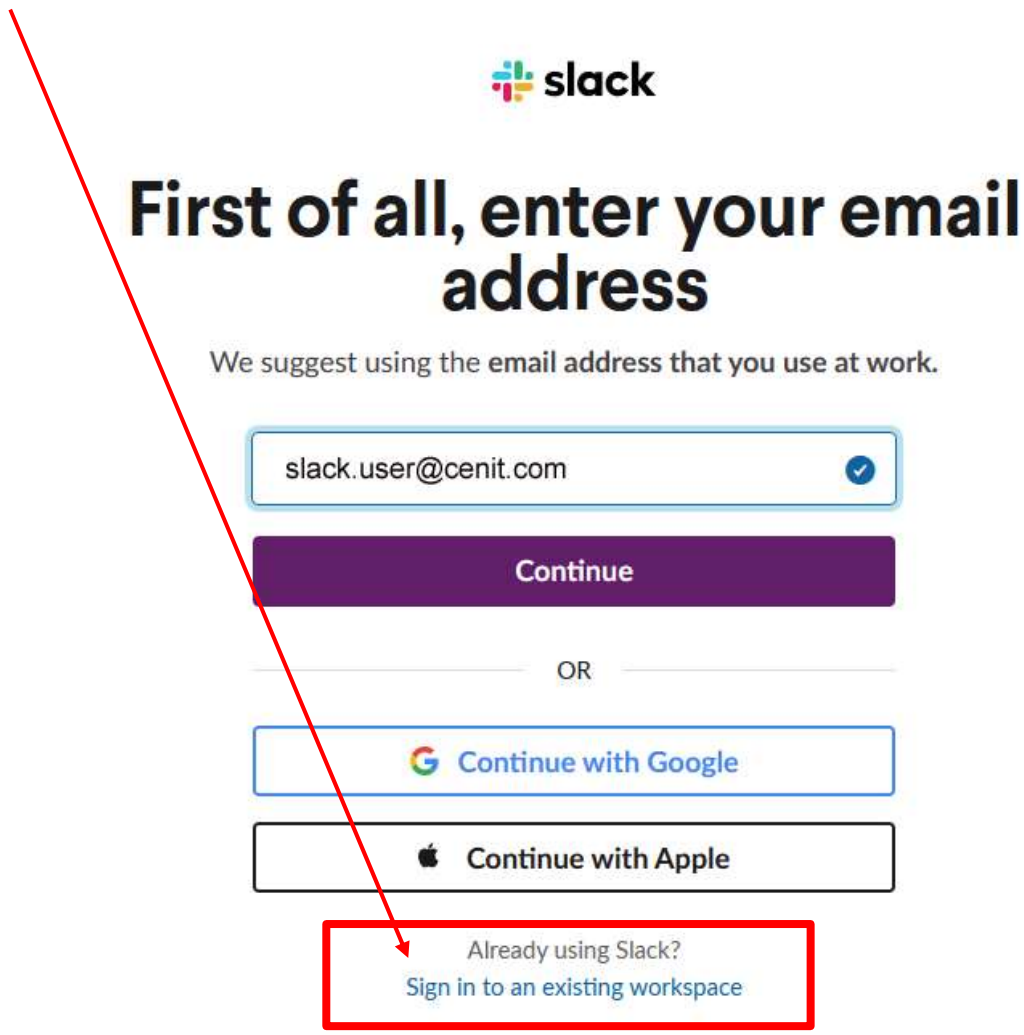
<https://cenitgroupglobal.webhook.office.com/webhookb2/baa589c7-5f5b-4a9d-89d3-7a4cbab7b71b@ce368477-a736-4353-b4ef-741ef2740d36/IncomingWebhook/343157e99ad946e5a166dd1d8ab5c8e7/a6858fdd-a65f-4093-867b-45a5958a51f1>

Example for Slack:

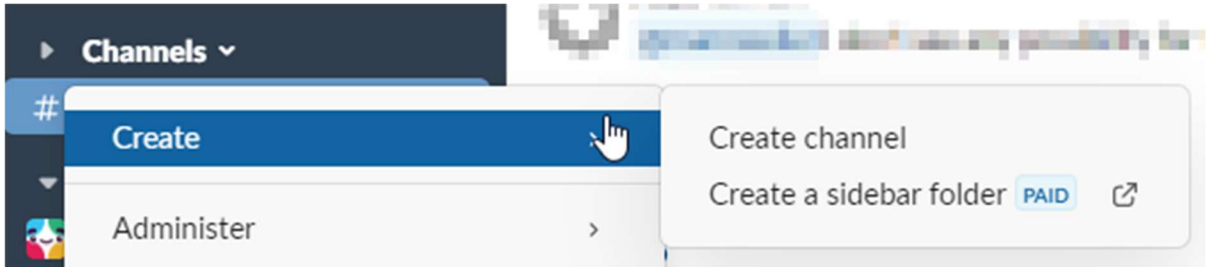
For creating a workspace in slack there is a “how to” available.

The page URL is: <https://slack.com/help/articles/206845317-Create-a-Slack-workspace>

Login to slack and open your workspace – in this case it is **ESM-CSM-Workspace**.




Once you have signed in, you can create a new channel or use an already existing one. For a new one, select the drop down at Channels and specify Create -> Create Channel:



Specify a name and create:

create channels ×

Channels are used by your team to communicate. Ideally, they should be topic-related - #marketing, for example.

 This name is already used for a channel, use another name

Name 64

esm-webhook-test

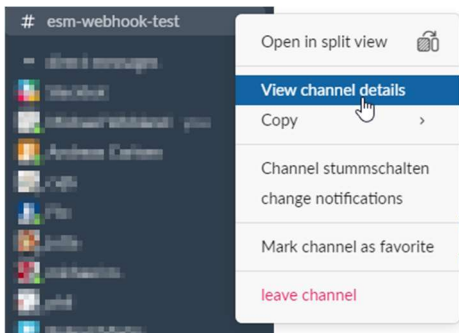
Description (optional)

What is this channel about?

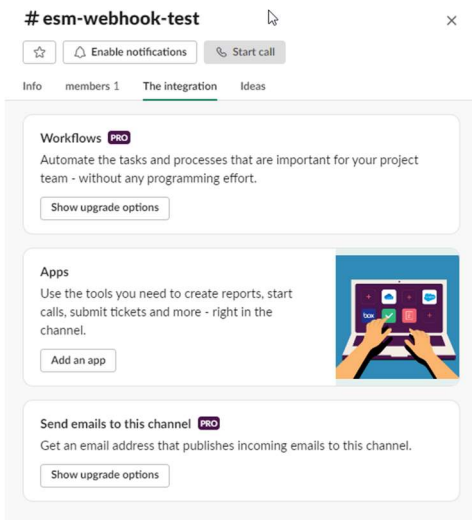
Set to "Closed".
When a channel is converted to a private channel, viewing and joining is invitation-only.

Create

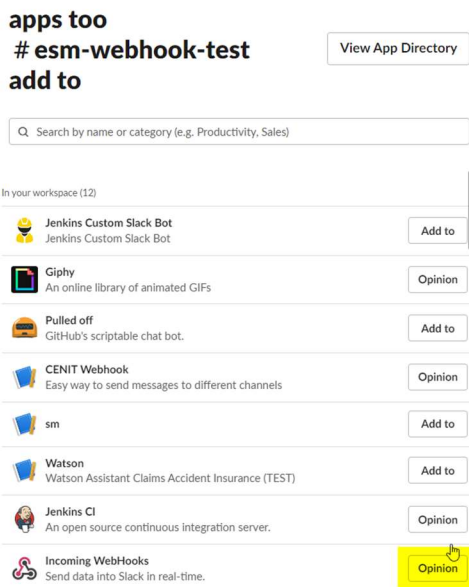
Right click on the channel and specify "View channel details":



In the next window that opens, select the integration tab and click on “Add an app”:

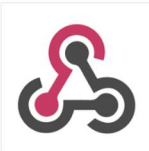


In the next window that opens, select “Opinion” next to Incoming Webhook:



Then select configuration:

Incoming webhooks



Incoming webhooks
*Please note, this is a legacy custom integration - an outdated way for teams to integrate with Slack. These integrations lack newer features and they will be deprecated and possibly removed in the future. **We do not recommend their use.** Instead, we suggest that you check out their replacement: [Slack apps](#).*


Incoming Webhooks are a simple way to post messages from external sources into Slack. They make use of normal HTTP requests with a JSON payload, which includes the message and a few other optional details described later.

Message Attachments can also be used in Incoming Webhooks to display richly-formatted messages that stand out from regular chat messages.

configuration

Add to slack:

< Browse apps



Incoming webhooks

Description configuration Security & Compliance

*Please note, this is a legacy custom integration - an outdated way for teams to integrate with Slack. These integrations lack newer features and they will be deprecated and possibly removed in the future. **We do not recommend their use.** Instead, we suggest that you check out their replacement: [Slack apps](#).*

Add to Slack


More info and support

- Get app support
- Privacy Policy
- Conditions

Incoming webhooks are a simple way to post messages from external sources into Slack. They make use of normal HTTP requests with a JSON payload, which includes the message and a few other optional details described later.

Message Attachments can also be used in Incoming Webhooks to display richly-formatted messages that stand out from regular chat messages.

Select the channel that was just created or that you want to use for forwarding the info:




Incoming webhooks

Send data into Slack in real time.

Please note, this is a legacy custom integration - an outdated way for teams to integrate with Slack. These integrations lack newer features and they will be deprecated and possibly removed in the future. **We do not recommend their use.** Instead, we suggest that you check out their replacement: [Slack apps](#).

Incoming webhooks are a simple way to post messages from external sources into Slack. They make use of normal HTTP requests with a JSON payload, which includes the message and a few other optional details described later.

[Message Attachments](#) can also be used in Incoming Webhooks to display richly-formatted messages that stand out from regular chat messages.

**New to Slack integrations?**

Check out our [getting started](#) guide to familiarize yourself with the most common integrations. You will also get a lot of tips that you should keep in mind when developing. You can also [register as a developer](#) to let us know what you're working on and to receive future updates on our APIs.

Post to channel

Start by choosing a channel for your incoming webhook to post messages to.

- Choose a channel...
- Michael Wohland Michael Wohland
- # generally
- Andreas Carlsen Andreas Carlsen
- cvjb Christian Brussow
- # esm webhook test
- Flo Flo
- # Freudenberg

Click on “Add inbound webhooks integration”:

Post to channel

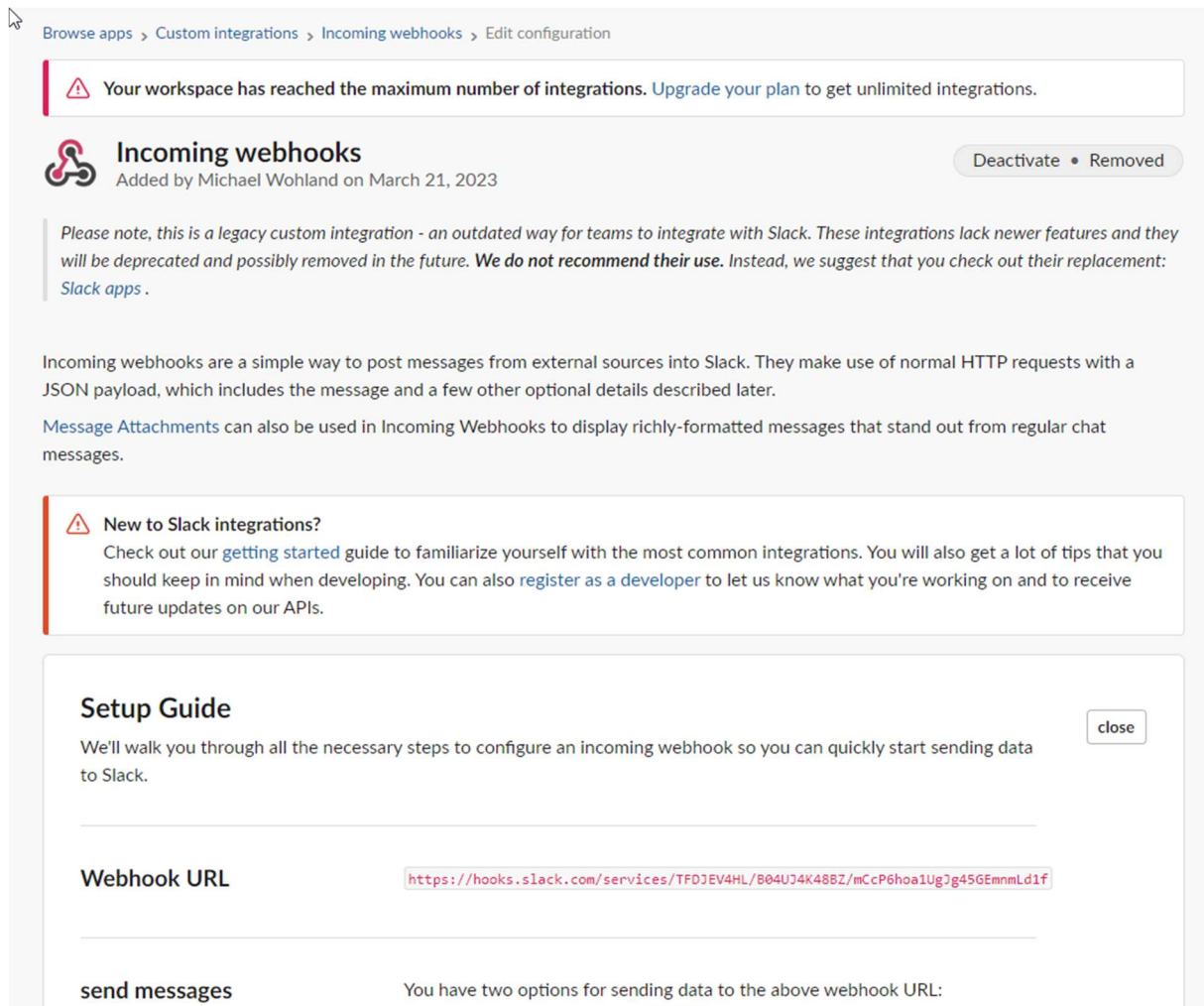
Start by choosing a channel for your incoming webhook to post messages to.

[or create new channel](#)

[Add inbound webhooks integration](#)

When you create an incoming webhook, you agree to the [Slack API Terms of Service](#).

The final page shows the Webhook URL along with other useful information such as possible message designs – Note the Webhook URL down:



Browse apps > Custom integrations > Incoming webhooks > Edit configuration

⚠️ Your workspace has reached the maximum number of integrations. Upgrade your plan to get unlimited integrations.

Incoming webhooks

Added by Michael Wohland on March 21, 2023 Deactivate • Removed

Please note, this is a legacy custom integration - an outdated way for teams to integrate with Slack. These integrations lack newer features and they will be deprecated and possibly removed in the future. **We do not recommend their use.** Instead, we suggest that you check out their replacement: [Slack apps](#).

Incoming webhooks are a simple way to post messages from external sources into Slack. They make use of normal HTTP requests with a JSON payload, which includes the message and a few other optional details described later.

[Message Attachments](#) can also be used in Incoming Webhooks to display richly-formatted messages that stand out from regular chat messages.

⚠️ New to Slack integrations?
Check out our [getting started](#) guide to familiarize yourself with the most common integrations. You will also get a lot of tips that you should keep in mind when developing. You can also [register as a developer](#) to let us know what you're working on and to receive future updates on our APIs.

Setup Guide close

We'll walk you through all the necessary steps to configure an incoming webhook so you can quickly start sending data to Slack.

Webhook URL `https://hooks.slack.com/services/TFDJEV4HL/B04UJ4K48BZ/mCcP6hoa1UgJg45GEmmLd1f`

send messages You have two options for sending data to the above webhook URL:

2. Preparing the JSON for forwarding the ESM information

Example for MS Teams:

For MS Teams it is possible to define a variety of JSON strings with different designs for the information that is forwarded. In this example here I am using the so called “Message Card” format. Again you can find more information for the creation of such cards in this URL from Microsoft:

<https://learn.microsoft.com/en-us/microsoftteams/platform/webhooks-and-connectors/how-to/connectors-using?tabs=cURL>

Message Card example for MS Teams:

```
{
  "@type": "MessageCard",
  "@context": "http://schema.org/extensions",
  "title": "ESM - Incident created",
  "themeColor": "0076D7",
  "summary": "ESM Webhook Test",
  "sections": [{
    "title": "$Situation.name - $Severity",
    "facts": [{
      "name": "Timestamp",
      "value": "$Sample.Timestamp"
    }, {
      "name": "Message",
      "value": "$Message"
    }, {
      "name": "Error:",
      "value": "$Error"
    }, {
      "name": "Value:",
      "value": "$Value"
    }
  ]],
  "startGroup": true,
  "markdown": true
},
"potentialAction": [{
  "@type": "OpenUri",
  "name": "Connect to ESM Server",
  "targets": [{
    "os": "default",
    "uri": "https://myESMServer:9443/"
  }]
}]
}]
}
```

As you can see in the facts section, there are name/value pairs. The name will be shown as identifier and the value is the related information from ESM. The \$Something is replaced with the actual information from the incident. This section can be enhanced as you need it.

Create a single line out of the above for later usage:

```
{ "@type": "MessageCard", "@context": "http://schema.org/extensions", "title": "ESM - Incident created", "themeColor": "0076D7", "summary": "ESM Webhook Test", "sections": [{"title": "$Situation.name - $Severity", "facts": [{"name": "Timestamp", "value": "$Sample.Timestamp"}, {"name": "Message", "value": "$Message"}, {"name": "Error:", "value": "$Error"}, {"name": "Value:", "value": "$Value"}], "startGroup": true, "markdown": true}], "potentialAction": [{"@type": "OpenUri", "name": "Connect to ESM Server", "targets": [{"os": "default", "uri": "https://myESMServer:9443/"}]}]}
```

Example for Slack:

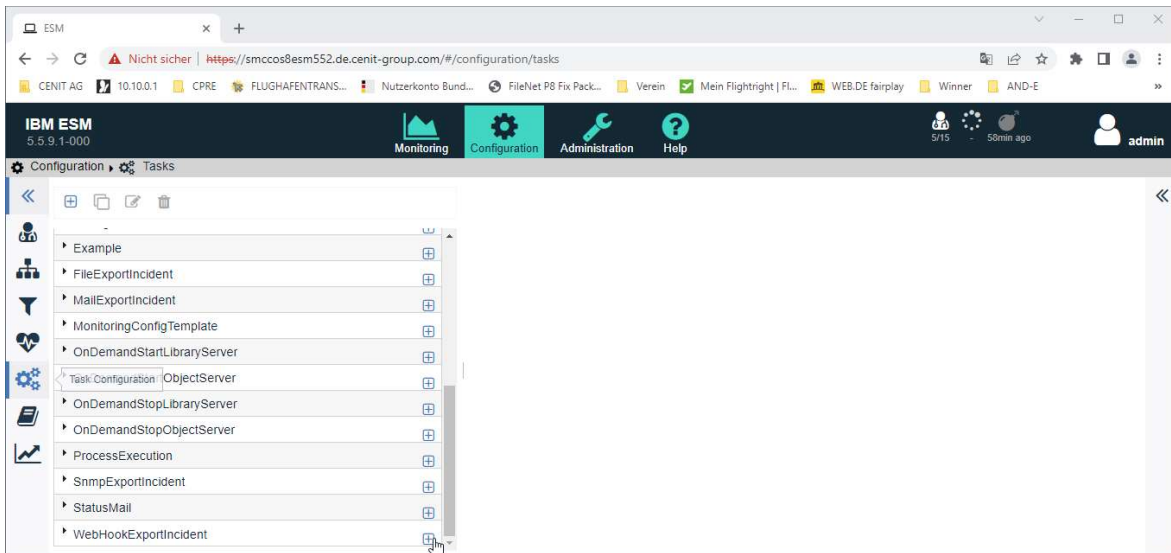
Single line example based on the message design from the final page when creating the Webhook URL above:

```
{ "text": "myservername | $Sample.Timestamp | $Severity | $Value | $Message | $Error | Classification" }
```

3. Setting up the Task

Example for MS Teams:

Open the ESM Console and switch to the configuration part. Select task in the left bar and click on the + next to WebHookExportIncident to create a new task:



The default task setup looks like this. The given parameters must be overwritten:

Task Configuration

Select the type of the task and enter its configuration data.

*Name

active

Task:

Template:

Webhook URL:

Authorization Header:

Agent

Select the location where the Task should run.

Schedule

If active when and how often should the Task be executed?

Interval Cron

Start:

End:

Interval:

First specify a name and then overwrite the template with “single line” message card JSON we created in Part 2. Also overwrite the Webhook URL with the URL you have created in Part 1. Make sure you don’t have any space in front or at the end of the entries. Do not make the task active, just save it with the button from above.



Task Configuration

Select the type of the task and enter its configuration data.

*Name:

active

Task:

Template:

Webhook URL:

Authorization Header:

Agent

Select the location where the Task should run.

Schedule

If active when and how often should the Task be executed?

Interval Cron

Start:

End:

Interval:

You can now setup this task as automated task in any situation you like.

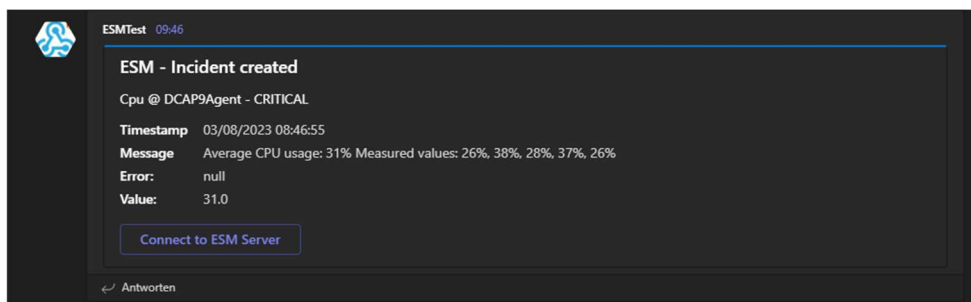


Automation

Trigger tasks automatically by occurrence of certain incidents.

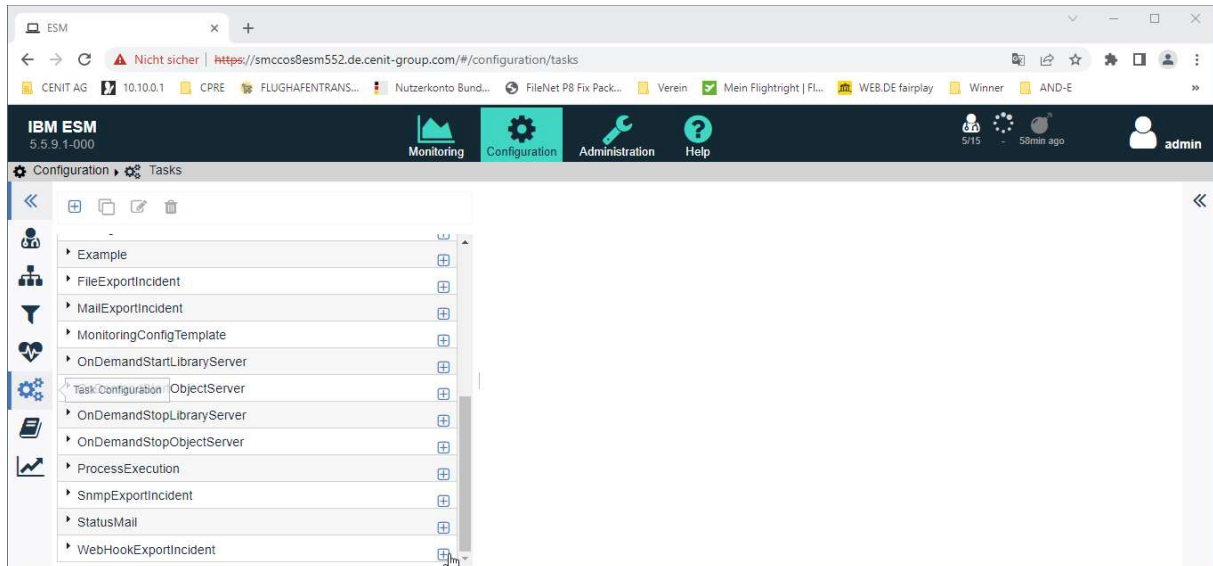
Condition	Mode	Tasks
All of: Severity >= Critical Situation == Cpu @ DCAP9Agent	First	MyWebhookTest

The output in MS Teams looks like this:



Example for Slack:

Open the ESM Console and switch to the configuration part. Select task in the left bar and click on the + next to WebHookExportIncident to create a new task:



The default task setup looks like this. The given parameters must be overwritten:



Task Configuration

Select the type of the task and enter its configuration data.

*Name

active

Task:

Template:

Webhook URL:

Authorization Header:

Agent

Select the location where the Task should run.

Schedule

If active when and how often should the Task be executed?

Interval Cron

Start:

End:

Interval:

First specify a name and then overwrite the template with “single line” JSON we created in Part 2. Also overwrite the Webhook URL with the URL you have created in Part 1. Make sure you don’t have any space in front or at the end of the entries. Do not make the task active, just save it with the button from above. The “Authorization Header” might not be needed and can be removed.

Task Configuration

Select the type of the task and enter its configuration data.

*Name:

active

Task:

Template:

Webhook URL:

Authorization Header:

Agent

Select the location where the Task should run.

Schedule

If active when and how often should the Task be executed?

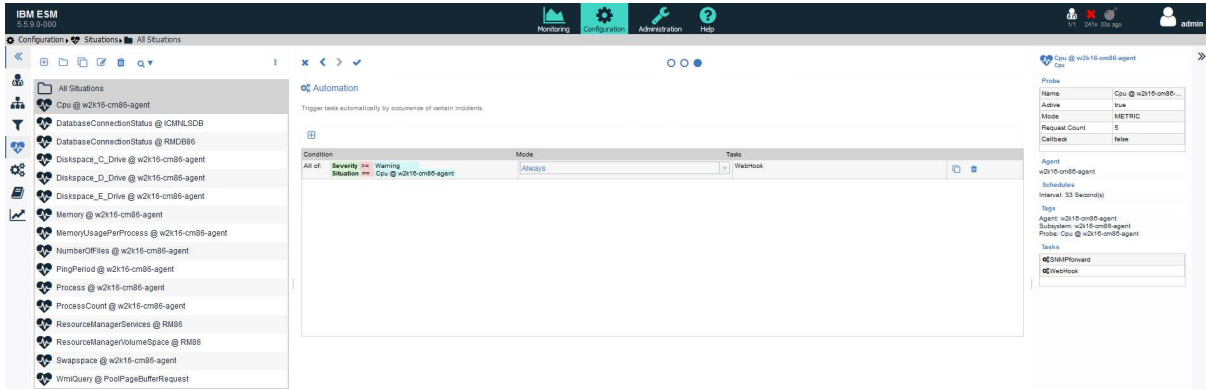
Interval Cron

Start:

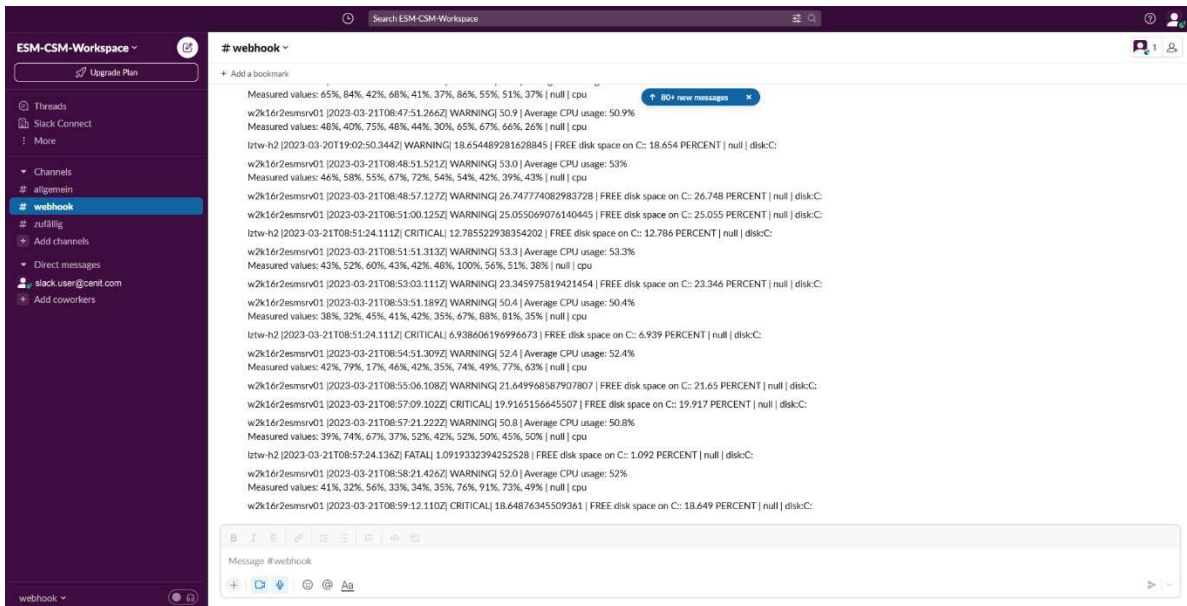
End:

Interval:

You can now setup this task as automated task in any situation you like.



Example of ESM-CSM-Workspace (configured and working)



Contact Information

If you have any questions, please contact us at ECM.SystemMonitor@cenit.com.

CENIT AG

Phone: +49 711 7825 30

Email: ECM.SystemMonitor@cenit.com