IBM Z OMEGAMON Data Provider
Version 1.1

Sample Elastic Kibana dashboards

IBM

2021-11-29 edition

Contents

About this document

This document presents the content of the Markdown-format readme files (`README.md`) found in some directories of the corresponding repository.

If the repository is hosted on a platform such as GitHub or Bitbucket, then the readme files are presented as HTML. However, if you receive the repository as a set of unhosted files, then this document offers the readme content in an alternative presentation format.

Sample Elastic Kibana dashboards for IBM Z OMEGAMON Data Provider

This repository demonstrates visualizing OMEGAMON attributes from IBM Z OMEGAMON Data Provider in Elastic Kibana dashboards.

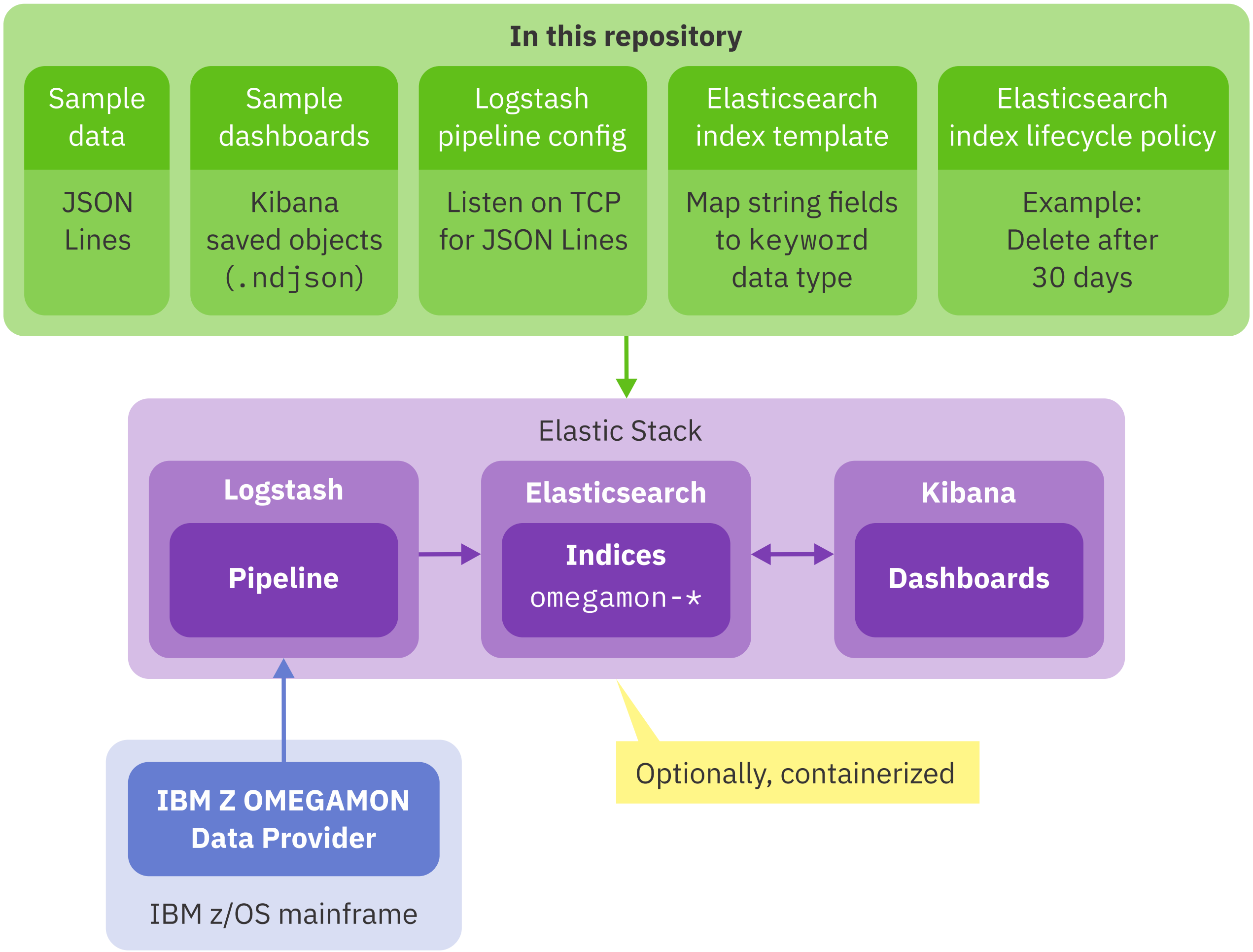
OMEGAMON includes agents that monitor z/OS systems, database products, and applications. In OMEGAMON terminology, the data that these agents collect, such as CPU usage, are known as *attributes*. In data analytics terminology, OMEGAMON attributes are *metrics*.

You can use this repository to:

- Build a Docker image and then start a container that runs the Elastic Stack with sample Kibana dashboards and sample data preconfigured
- Configure your own Elastic Stack instance with the sample Kibana dashboards and sample data
- Visualize your own data in the sample dashboards

This repository contains:

- Elastic Stack artifacts:
  - Kibana saved objects, including dashboards and related objects such as visualizations and index patterns
  - Logstash pipeline configuration file
  - Elasticsearch index template and lifecycle policy
- Sample data in JSON Lines format
- Source files for a Docker image that configures an Elastic Stack instance with all of the above in a ready-to-use container


In this repository
Sample data
JSON Lines
Sample dashboards
Kibana saved objects (.ndjson)
Logstash pipeline config
Listen on TCP for JSON Lines
Elasticsearch index template
Map string fields to keyword data type
Elasticsearch index lifecycle policy
Example: Delete after 30 days
Elastic Stack
Logstash
Pipeline
Elasticsearch
Indices
omegamon-*
Kibana
Dashboards
Optionally, containerized
IBM Z OMEGAMON Data Provider
IBM z/OS mainframe

Getting started

To start using the sample dashboards, choose one of the following options:

- Start a Docker container that runs the Elastic Stack with the dashboards and sample data preconfigured.

  You can immediately start exploring the dashboards with the sample data. The Docker container is also preconfigured to listen for OMEGAMON attributes in JSON Lines format over TCP from IBM Z OMEGAMON Data Provider.

  See the `docker` directory.
- Configure an existing Elastic Stack instance.

  See the heading "Installing the dashboards in an existing Elastic Stack instance".

Scope and intended use of these dashboards

The sample dashboards provided in this repository are intended to be useful out-of-the-box. However, they are *not* a fully fledged solution for analyzing OMEGAMON attributes. Instead, these sample dashboards demonstrate some typical use cases for visualizing OMEGAMON attributes.

Each OMEGAMON agent monitors a set of attributes. Related attributes are organized into groups, also known as tables. Each table typically contains a dozen or more attributes. Each agent can collect many tables. In total, across all agents, there are hundreds of attribute tables.

The sample dashboards in this repository visualize data from a small subset of attribute tables collected by a single agent: IBM Z OMEGAMON Monitor for z/OS, 5.6.

The developers of these dashboards anticipate that customers will examine the dashboards, and then copy and adapt selected visualizations into bespoke dashboards to suit their own requirements.

Requirements

Depending on what you want to do:

- **You don't need OMEGAMON or a mainframe to use this repository.** You can use the dashboards with the included sample data.
  - To start a Docker container with the sample data, the only requirement is Docker.

    The Docker image was built and tested using **Docker 20.10.6 on Linux Ubuntu 20.04**.
  - To install the dashboards in your own instance of Elastic Stack with the sample data, the only requirement is Elastic Stack.

    The Elastic Stack artifacts in this repository were developed and tested with **Elastic Stack 7.14.0**.

    For compatibility with other versions of Elastic Stack, see the Elastic Stack documentation.
- To visualize your own data, you need a z/OS mainframe with IBM Z OMEGAMON Data Provider configured to forward attributes to Logstash.

  To get IBM Z OMEGAMON Data Provider, contact your IBM Software representative for OMEGAMON products.

Elastic Stack system requirements

Minimum recommended system requirements for the computer, or Docker container, running Elastic Stack:

- 16 GB RAM
- 200 GB disk space
- 4 vCPUs

Actual system requirements depend on your site-specific practices. For example, the disk space required depends on the amount of data stored in Elasticsearch, which depends on various site-specific factors, including:

- How much data you forward: Which attribute tables, and fields in those tables, you forward, and how many z/OS systems you are monitoring
- How frequently you forward data, controlled by the attribute table collection interval in OMEGAMON
- How long you keep data, controlled by an Elasticsearch index lifecycle policy

Repository contents

The following file and directory are relevant only if you are interested in Docker. Otherwise, you can ignore them:

- `Dockerfile`
  Contains instructions for building the `z-omegamon-analytics-elastic` Docker image.
- `docker/`
  Contains files required by the Docker image.

The remaining directories are relevant whether or not you use Docker:

- `data/`
  Contains sample OMEGAMON attribute data in JSON Lines format from IBM Z OMEGAMON Data Provider.
- `elasticsearch/`
  Contains an Elasticsearch index template (component, not legacy) and lifecycle policy.

- `kibana/`
  Contains Kibana saved objects, including dashboards and related objects such as index patterns.
- `logstash/`
  Contains a Logstash pipeline configuration.

The following directory is for use only by the maintainers of this repository:

- `docs/`
  Contains files required to build the `README.pdf` file, which offers an alternative presentation format to the `README.md` files in this repository. Depending on how you received the repository, this directory might not be present in your copy.

Installing the dashboards in an existing Elastic Stack instance

The following instructions assume that you have an existing Elastic Stack instance into which you want to install the Kibana dashboards that are provided by this repository. For details on installing Elastic Stack, see the documentation on the Elastic website.

The following instructions are also useful for understanding the Docker image, which automates these steps.

Import Kibana saved objects

Import the saved objects in the `kibana/export.ndjson` file into Kibana.

Either:

- In the Kibana UI:
  1. Select Management ► Stack Management ► Kibana: Saved Objects ► Import
  2. Select the following import options:
     – Check for existing objects
       - Automatically overwrite conflicts
  3. Click **Import**
- Use the Kibana import saved objects API

Tip: Rather than importing into the default space, import into a Kibana space that you have created specifically for these dashboards.

Create an Elasticsearch index lifecycle management (ILM) policy

Use the `elasticsearch/omegamon-ilm-policy.json` file as the body of an Elasticsearch create lifecycle policy API request.

Example API request:

```
curl -X PUT -H "Content-Type: application/json" -d @omegamon-ilm-policy.json "http://
localhost:9200/_ilm/policy/omegamon-ilm-policy"
```

Notes:

- The supplied policy has an active Delete phase that deletes indices after 30 days.
- Using this policy means that the sample data will be deleted 30 days after you load it into Elasticsearch. The age of the events in the sample data is not relevant. The lifecycle clock starts when an index is created, regardless of the event time stamps.
- If you *don't* configure a policy, and you keep forwarding data to this Elasticsearch instance, then you will eventually run out of disk space.
- The supplied policy is an example only. Configure a policy that matches your requirements and available disk space.

- For details on creating index lifecycle policies and associating them with indices, see the Elastic Stack ILM documentation.

Create an Elasticsearch index template

Use the `elasticsearch/omegamon-index-template.json` file as the body of an Elasticsearch create index template API request.

Example API request:

```
curl -X PUT -H "Content-Type: application/json" -d @omegamon-index-template.json "http://
localhost:9200/_index_template/omegamon"
```

The supplied index template definition sets the number of replicas for each index to 0:

```
"number_of_replicas": 0
```

This setting is appropriate only in the context of a single-node Elastic Stack instance, such as the stand-alone Docker container based on the supplied `Dockerfile`. You might choose to remove this setting, and also further customize the index template to meet your site-specific requirements.

Otherwise, the index template has only one purpose: to map incoming string fields to the `keyword` data type, rather than the default `text` data type.

Customize Kibana settings

The following customizations are recommendations only, to improve your user experience of the sample dashboards.

Avoid incomplete lists of terms in Controls dropdowns

With default Kibana settings, depending on the number of documents you have indexed, the list of terms available in a Kibana Controls dropdown might be incomplete. In the same dashboard, you might see terms in charts *that are not available in a Controls dropdown for that field*.

There are many topics in the Elastic Kibana discussion forum about this issue.

The recommended "fix" (sic, deliberately in quotes): in `$KIBANA_HOME/config/kibana.yml`, set high values for the following settings. For example:

```
kibana.autocompleteTimeout: 5000
kibana.autocompleteTerminateAfter: 10000000
```

Tip: Before editing `kibana.yml`, stop Kibana. For example, in a Linux command shell, enter `service kibana stop`. After editing, restart Kibana: enter `service kibana start`.

Ignore filter if field is not in index

Pinned filters in Kibana are useful to maintain consistent filtering when switching between dashboards. For example, to limit results to a particular system. However, different dashboards use different index patterns. If a visualization uses an index pattern that does not contain the field whose value is restricted by a pinned filter, then, by default, the visualization shows no results.

For example, some dashboards can be usefully filtered by the `job_name` field, so it makes sense for users to pin a filter for `job_name`. However, the data for other dashboards does not include a `job_name` field. For those dashboards, by default, a `job_name` filter causes the dashboard visualizations to display "No results found".

To ignore a filter if the field is not in the index pattern, switch on the `courier:ignoreFilterIfFieldNotInIndex` setting. In the Kibana UI:

1. Select Management ► Stack Management ► Kibana: Advanced Settings
2. Scroll to, or search for, or select, the **Search** category

3. Set **Ignore filter(s)** to **On**

Configure Logstash to listen for data

This repository contains a Logstash pipeline configuration ("config") file, `logstash/pipeline/10-omegamon-tcp-to-local-elasticsearch.conf`, that configures Logstash to listen on a TCP port for JSON Lines data from OMEGAMON Data Provider, and then forwards the data to Elasticsearch on the same computer.

How you use that config file depends on your Elastic Stack installation and your local site practices for configuring Logstash pipelines.

Single or multiple Logstash pipelines?

Before proceeding, you need to know whether your instance of Logstash is for use only with this repository or is also used for other purposes, other inputs. Specifically, you need to know whether your use of Logstash involves a *single pipeline* or *multiple pipelines*.

If you have installed a new instance of Elastic Stack as a sandbox environment for testing this repository, then you can use a single Logstash pipeline.

However, if you are using this repository with an existing instance of Elastic Stack that already has other inputs, then it is more likely that you will need to use multiple pipelines.

Single pipeline

If your instance of Logstash is for use only with this repository, then you can delete the contents of the default Logstash config directory, and then copy the supplied config file into that directory.

For example:

1. Delete the contents of the `/etc/logstash/conf.d/` directory.

   The default Logstash pipeline directory path depends on your platform.
2. Copy the `logstash/pipeline/10-omegamon-tcp-to-local-elasticsearch.conf` file to the `/etc/logstash/conf.d/` directory.

Multiple pipelines

For information about configuring multiple pipelines, see the Logstash documentation.

Refresh the Logstash config

Unless you have configured Logstash to automatically detect new pipeline configurations, stop and then restart Logstash.

For example, to stop the Logstash service on Linux, enter:

```
service logstash stop
```

Logstash can take a while to respond to that command (the signal to stop). If the response from that command ends with:

> logstash stop failed; still running.

wait for several seconds, and then enter:

```
service logstash status
```

You want to see:

> logstash is not running

Enter:

```
service logstash start
```

Test the Logstash config with a single event

Optionally, before forwarding “real” data (your own data, or the supplied sample data) to Logstash, you might wish to test the Logstash config by forwarding a single event.

The file `data/omegamon-1-line-test.jsonl` contains a single event intended for testing the Logstash config. Use a TCP forwarder to send the file to the listening TCP port.

For example, use a tool such as `socat` (used by the Docker image):

```
socat -u omegamon-1-line-test.jsonl TCP4:localhost:5046
```

or `ncat`:

```
ncat --send-only -v localhost 5046 < omegamon-1-line-test.jsonl
```

where `localhost` is the hostname of the computer running Logstash and `5046` is the port on which Logstash is listening.

Check that the event has been indexed. For example, on the Kibana Discover tab, select the `omegamon-*` index pattern, and then set a filter for the `table_name` field value `test`.

Forward the sample data to Logstash

Perform this step only if you want to use the provided sample data.

Use a TCP forwarder to send the sample data file, `data/omegamon-sample-data.jsonl`, to the listening TCP port.

For example:

```
socat -u omegamon-sample-data.jsonl TCP4:localhost:5046
```

Browse the Kibana dashboards

Use your web browser to go to the following Kibana URL:

```
http://localhost:5601/s/omegamon/app/dashboards#/view/d24954f0-a7e6-11eb-b38d-7b8e5ab9c939?
_g=(time:(from:'2021-10-13T12:00:01.999Z',to:'2021-10-13T13:59:55.999Z'))
```

- `localhost` assumes that you are using the Elastic Stack installed on your local computer. If that is not the case, replace `localhost` with the hostname or IP address of the computer where you have installed the dashboards.
- The `time` parameter in the URL specifies the time range of the sample data, so that you do not have to specify this range to Kibana yourself.
- The dashboard developers recommend the Chrome web browser.

Your web browser should display a "home" dashboard that is an entry point to the other sample dashboards. Click a dashboard and begin exploring the data. For details on using Kibana, see the Kibana User's Guide.

Elasticsearch index names and Kibana index patterns

The supplied Logstash config uses the following `index` option to set Elasticsearch index names:

```
index => "omegamon-%{table_name}-%{+YYYY.MM.dd}"
```

where table_name is a field in the incoming JSON data, with a value such as "ascpuutil".

The sample dashboards use corresponding table-specific index patterns, such as omegamon-ascpuutil-*.

To help you explore and experiment with the data, the supplied saved objects include an omegamon-* index pattern that searches data across all tables.

Deleting the sample data

Suppose that you have installed the sample data. Later, you decide to use the same Elastic Stack instance for your own data. You might want to delete the sample data first.

To delete all Elasticsearch indices for the sample data, send the following Elasticsearch REST API request (for example, using the Dev Tools option in Kibana):

```
DELETE /omegamon-*
```

Forwarding your own data

If you have IBM Z OMEGAMON Data Provider, you can visualize OMEGAMON attributes from your own systems in the sample dashboards.

The sample dashboards use the following attribute groups with the specified collection intervals:

table_name field value	**Attribute group**	**Collection interval (minutes)**
ascpuutil	Address Space CPU Utilization	1
km5msucap	KM5 License Manager MSU WLM Cap	5
km5wlmclpx	WLM Class Sysplex Metrics	1
km5wlmclrx	WLM Class Raw Extended Metrics	1
lpclust	LPAR Clusters	1
m5stgcdth	Common Storage Utilization History	5
m5stgdeth	KM5 Storage Details History	5
m5stgfdth	Real Storage Utilization History	5
mplxcpcsum	KM5 CPC Summary	1
mrptcls	Report Classes	1
syscpuutil	System CPU Utilization	1

To visualize your own data in the sample dashboards:

1. Create historical collections that match the attribute groups and collection intervals used by the sample dashboards (see the previous table).
2. Configure IBM Z OMEGAMON Data Provider to forward attributes from those groups to Logstash as JSON Lines over TCP.

   The following IBM Z OMEGAMON Data Provider collection configuration parameters match the specifications in the previous table:

   ```
   collections:
     - product: km5
       table: ascpuutil
       interval: 0
       destination: [pds, open]
   ```

```
  - product: km5
    table: km5msucap
    interval: 0
    destination: [pds, open]
  - product: km5
    table: km5wlmclpx
    interval: 0
    destination: [pds, open]
  - product: km5
    table: km5wlmclrx
    interval: 0
    destination: [pds, open]
  - product: km5
    table: lpclust
    interval: 0
    destination: [pds, open]
  - product: km5
    table: m5stgcdth
    interval: 0
    destination: [pds, open]
  - product: km5
    table: m5stgdeth
    interval: 0
    destination: [pds, open]
  - product: km5
    table: m5stgfdth
    interval: 0
    destination: [pds, open]
  - product: km5
    table: mplxcpcsum
    interval: 0
    destination: [pds, open]
  - product: km5
    table: mrptcls
    interval: 0
    destination: [pds, open]
  - product: km5
    table: syscpuutil
    interval: 0
    destination: [pds, open]
```

Docker image: Sample Elastic Kibana dashboards for IBM Z OMEGAMON Data Provider

The `z-omegamon-analytics-elastic` Docker image demonstrates visualizing OMEGAMON attributes from IBM Z OMEGAMON Data Provider in Elastic Kibana dashboards.

This image contains the Elastic Stack configured with:

- Kibana dashboards for analyzing OMEGAMON attributes
- Sample data for the dashboards

This image provides a quick way to try the dashboards in a self-contained “sandbox” environment, with sample data.

Getting started

To start using the dashboards in a Docker container:

1. Get a Docker host. Either:

   - Install Docker on your personal computer, or
   - Contact your organization's software support for details of an existing Docker host

   If you install Docker, follow the installation instructions on the Docker website, including the steps to verify installation. In particular, ensure that you can successfully run the Docker “hello world” example.
2. Check that your Docker host virtual memory settings meet the Elasticsearch requirements.
3. Build the Docker image.

   Enter the following command in the root directory of this repository (containing the `Dockerfile`):

   ```
   docker build -t z-omegamon-analytics-elastic .
   ```

   Note the trailing period preceded by a space.
4. Start a container.

   For example, on your Docker host, open a command prompt and enter the following command:

   ```
   docker run -d -p 15601:5601 -p 19200:9200 -p 15046:5046 -v elastic-data:/var/lib/
   elasticsearch --name z-omegamon-analytics-elastic z-omegamon-analytics-elastic
   ```

   By default, the Docker container creates a Kibana space with the ID `omegamon`, and then imports saved objects into that space. You can set environment variables on the `docker run` command line to create a different space. For details, see “Setting environment variables to control the container startup behavior”.

The `-p` command options map ports inside the container to the following ports on your Docker host:

- Logstash listens for incoming JSON Lines data on TCP port 15046. To add your own data to the container, forward JSON Lines to this port.
- Kibana is on HTTP port 15601.
- The Elasticsearch API is on HTTP port 19200.

If these port numbers clash with existing port assignments on your Docker host, feel free to use different port numbers. For details, see the Docker command reference documentation.

The `--name` option specifies the name of the new Docker container. In this example `docker run` command, the container has the same name as the image. (The image name is the last argument on the command line.) You can choose to specify a different container name.

The `-v` option creates a named volume for the container directory that stores Elasticsearch indices, rather than creating a volume with a non-semantic generated ID. Semantic names can make volumes easier to manage. For example, to remove this volume later, rather than having to identify and refer to the corresponding volume ID, you can refer to it by name: `docker volume rm elastic-data`.

Recommendations:

- Do not share a volume between containers that are based on this image.

  If you start more than one container based on this image on the same Docker host, then use a different volume for each container. You might choose to use a volume naming convention that corresponds to container names.

- When you remove a container that is based on this image, also remove its volume.

  Unless you purposefully, deliberately want to do so (for example, you want to keep your own data that you have forwarded from your own systems), and you understand the related issues, such as the compatibility of Elasticsearch index data formats between versions, do not reuse an existing volume for a new container.

5. Wait: for the Docker image to download, for the container to start, and then for the container to initialize Elastic with the supplied dashboards and data. Depending on your connection the web, the Docker image might take several minutes to download. After that, the container might take another minute to initialize, depending on your Docker host.
6. Browse to the following Kibana URL:

   ```
   http://localhost:15601/s/omegamon/app/dashboards#/view/d24954f0-a7e6-11eb-b38d-7b8e5ab9c939?
   _g=(time:(from:'2021-10-13T12:00:01.999Z',to:'2021-10-13T13:59:55.999Z'))
   ```

   - Replace `localhost` in the URL with the name of your Docker host.
   - The `time` parameter in the URL specifies the time range of the sample data, so that you do not have to specify this range to Kibana yourself.
   - We recommend the Chrome web browser.

Your web browser should display a "home" dashboard that is an entry point to the other sample dashboards. Click a dashboard and begin exploring the data. For details on using Kibana, see the Kibana User's Guide.

When the container starts, it loads data into Elastic. Kibana allows you to view dashboards while the data is still loading, so you might see partially filled charts, such as vertical time-based bar charts with little or no data showing on the right-hand side. If this happens, wait a little longer, and then refresh the page in your browser (for example, press F5).

If your web browser does not display a list of Kibana dashboards, or you click a dashboard and the dashboard contains no data, or you experience some other problem, see “Troubleshooting”.

Upgrading from an earlier version of this repository

If you have an old Docker container based on an earlier version of this repository, and you want to upgrade to a new container, consider using the following steps (these are recommendations only; vary them depending on your needs):

1. Stop and remove the old container. Example:

   ```
   docker rm -f z-omegamon-analytics-elastic
   ```

2. Remove the volume used by the old container. Example (depending on the volume name you used for the old container):

   ```
   docker volume rm elastic-data
   ```

   Tip: To remove unused volumes without referring to specific volume names, use `docker volume prune`.

   If you do not remove the volume, and you specify the same volume name when starting a new container, then the new container will use the same volume, with data from the old container. Reusing that volume might cause issues with the new container, for a variety of reasons, such as differences in the Elasticsearch index data format between versions.

3. Rebuild the Docker image from the updated repository, and then start a new container. For details, see "Getting started".

Troubleshooting and support

Before seeking support for this image, please ensure that you can successfully run the Docker “hello world” example described in the Docker documentation. Please do not use the support contact details provided here for general Docker issues.

If you experience a problem using these dashboards, enter the following command on your Docker host, and save the command output:

```
docker logs z-omegamon-analytics-elastic
```

then contact your local Docker expert or IBM Software Support.

Common issues

Elasticsearch error: IndexFormatTooNewException (Format version is not supported)

Explanation: Elasticsearch has encountered indices in a format version that is not supported by this version of Elasticsearch.

Probable cause: A new Docker container has been started. The new container uses the same volume as this container (the container in which this error occurred). The new container uses a more recent version of Elasticsearch. Elasticsearch in the new container has loaded data into the same volume (the same directory, the same indices) used by this container. Elasticsearch in this container does not support the index format used by the newer version.

Recommended action: Do not share volumes between containers. Stop and remove both containers. Remove the volume that has been "polluted" by the new data. Start new containers using a separate volume for each container.

Tips for new Docker users

Stopping the container

The docker run command in the “Getting started” procedure specifies a -d option that starts a container in “detached” mode rather than the default foreground mode.

In detached mode, the container continues running after you close the command shell that you used to start it, so you don't need to keep that command shell open.

To stop the container, enter the following command:

```
docker stop z-omegamon-analytics-elastic
```

Shutting down your computer stops Docker and the container. When you reboot your computer and Docker restarts, the container will *not* restart automatically: you need to restart it. To change this behavior, see the Docker documentation on restart policies.

Restarting the container

To restart the container after stopping it or after rebooting your computer, enter the following command:

```
docker restart z-omegamon-analytics-elastic
```

Other Docker commands

To list containers, both running and stopped, use the docker ps -a command.

To remove a container, use the docker rm command.

For more information about Docker commands, see the Docker documentation.

Loading data into the container

In the container, Logstash is configured to listen for JSON Lines on TCP port 5046.

Create an Elasticsearch index lifecycle management (ILM) policy

The Docker image does not configure index lifecycle policy. If you plan to forward your own data to this Elastic Stack instance, then consider creating an index lifecycle policy now to avoid running into disk capacity issues later.

As a starting point, consider creating a policy with an active Delete phase that deletes data after 30 days. Use the elasticsearch/omegamon-ilm-policy.json file as the body of an Elasticsearch create lifecycle policy API request (PUT _ilm/policy/omegamon-ilm-policy). Then add that policy name to the omegamon index template settings:

```
"index.lifecycle.name": "omegamon-ilm-policy"
```

For details on creating index lifecycle policies and associating them with indices, see the Elastic Stack ILM documentation.

Setting environment variables to control the container startup behavior

When you start a container for the first time, you can set environment variables to control the container startup behavior.

These variables are in addition to the environment variables provided by the base sebp/elk image.

Controlling the loading of sample data and Kibana saved objects

To control loading of sample data and Kibana saved objects into the container, set the following environment variables. (The saved objects define the sample dashboards.)

- INSTALL_SAMPLES
  Default value: 1, which loads the sample data and Kibana saved objects (dashboards, visualizations, saved searches, and index patterns).

  To disable all loading, set to 0 and omit all other variables.
- INSTALL_SAMPLE_DATA
  Defaults to the value of INSTALL_SAMPLES.

  To skip loading the sample data, set to 0.

  To load the sample data, set to 1.
- INSTALL_SAMPLE_OBJECTS
  Defaults to the value of INSTALL_SAMPLES.

  To skip loading Kibana objects, set to 0.

  To load Kibana objects, set to 1.

If you load Kibana saved objects but not the sample data, then you will need to load your own data.

You can set these environment variables using the -e option of the docker run command.

For example, if you want to use the sample dashboards, but you want to omit the sample data because you plan to forward your own logs to the container, set INSTALL_SAMPLE_OBJECTS=1 and INSTALL_SAMPLE_DATA=0:

```
docker run -d -p 15601:5601 -p 19200:9200 -p 15046:5046 -v elastic-data:/var/lib/elasticsearch
 -e INSTALL_SAMPLE_DATA=0 -e INSTALL_SAMPLE_OBJECTS=1 --name z-omegamon-analytics-elastic z-
omegamon-analytics-elastic
```

Controlling the Kibana space into which the saved objects are imported

To control which Kibana space the saved objects are imported into, set the following environment variables.

The Docker image creates a Kibana space, and then imports saved objects into that space. You can override the default space ID, description, and initials.

- KIBANA_SPACE_ID
  Default value: omegamon
- KIBANA_SPACE_NAME
  Default value: OMEGAMON analytics
- KIBANA_SPACE_INITIALS
  Default value: OM

Extending the Docker image

The Docker image defined in this repository is based on the sebp/elk image from Docker Hub.

This repository includes a shell script, docker/elk-post-hooks.sh, that is run by the "post-hooks" feature of the base image.

The sebp/elk base image provides numerous configuration features. Before extending the Docker image defined in this repository, read the sebp/elk documentation.

OMEGAMON attributes in JSON

The `data` directory contains two JSON Lines files of OMEGAMON attributes:

- `omegamon-1-line-test.jsonl`
  A single line that you can use to test that the Logstash config is working.
- `omegamon-sample-data.jsonl`
  Sample data from various attribute tables for use with the sample dashboards.

Kibana artifacts

The kibana directory contains the following files for use with Kibana:

- `export.ndsjon`
  Saved objects exported from Kibana. These objects define sample dashboards for analyzing OMEGAMON attributes.
- `omegamon-space.json`
  The body of a Kibana create space API request.

About the sample dashboards

The dashboards were developed on a screen with a resolution of 1920 1080 pixels.

The "home" dashboard offers an entry point to the dashboards. The home dashboard is, essentially, an expanded version of the "menu bar" that appears at the top of most of the other dashboards.

Most of the dashboards are for analyzing attributes. However, the **Data inventory** dashboard is for analyzing the amount and types of attributes that have been ingested.

Navigating between dashboards

To switch between dashboards, click a link in the row of links at the top of each dashboard.

Some dashboards offer drill-down links to other dashboards.

The developers of these dashboards look forward to future versions of Kibana offering better support for navigating between dashboards. For details, see Kibana issue #99740, "Custom nested (cascading, flyout) menus of links in the Kibana sidebar".

Logstash configuration

Logstash configuration for the sample Kibana dashboards consists of a single pipeline configuration (`.conf`) file supplied in the `/logstash/pipeline` directory.

Pipeline configuration

Input

The input section configures Logstash to listen for JSON Lines over TCP.

The config assumes unsecure TCP: no Transport Layer Security (SSL/TLS).

Alternative input: Apache Kafka

If you have configured OMEGAMON to publish attributes in JSON format to Apache Kafka, then you can use the Kafka input plugin for Logstash to subscribe to that topic (or, depending on your configuration: topics, plural).

Here is a rudimentary Logstash input for Kafka:

```
kafka {
   id => "omegamon_kafka_input"
   bootstrap_servers => "kafkaserver.example.com:9092"
   topics => ["omegamon_json"]
   codec => json
}
```

Filter

The filter section consists of a `date` option that uses the `write_time` field to set the `@timestamp` field.

The sample Kibana dashboards use the `@timestamp` field as the event time stamp.

Output

The output section forwards data to the Elasticsearch instance that is running on the same computer as Logstash (`localhost`).

`manage_template` does not yet support component templates

As of September 2021, the `manage_template` option of the Elasticsearch output plugin for Logstash supports only *legacy* index templates, not the newer *component* index templates. Until that support is introduced, set `manage_template` to false and use the Elasticsearch API to create a component index template. See the example "create index template" request body in the `/elasticsearch` directory.

For more information, see issue #958, "Support for new index templates", in GitHub repository `logstash-plugins/logstash-output-elasticsearch`.