Maximo logs analysis

April 1, 2020 by kasyon

1. Introduction

Analysis of the Maximo server logs involves running two servers: **`elasticsearch`** and **`kibana`** as well as the **`maximostash`** application. The elasticsearch server contains a document database and is used to collect and index data from logs and also to answer queries. Maximostash program retrieves data from Maximo logs and sends it to the elasticsearch server. The information indexed by elasticsearch is then viewed by the kibana server, which provides an UI for querying.

2. Configuration

`Maximostash` is a tool for retrieving data from Maximo MXservers logs and WebSphere Application Server logs in order to send them to the elasticsearch. The program requires presence of a `config.properties` file with the following content:

`logs.directory` – UNC path to the folder with the logs to be analyzed (in Java backslashes are escaped), e.g .: `\\\\v00073\\c$\\IBM\\WebSphere\\AppServer\\profiles\\ctgAppSrv01\\logs\\MXServer`
it can also be a relative path to the current folder, **`.`** (dot) or NTFS symlink when run as Windows Service (UNC paths are not allowed for Windows services)

`elasticsearch.url` – elasticsearch server address, e.g. `http://localhost:9200`

`loginclusion.pattern` – RegEx pattern matching the names of logs accepted for analysis, e.g. `(.*_.*_.*\\.log.*)|(System.*\\.log.*)`

`logexclusion.pattern` – RegEx pattern matching the names of logs rejected from the analysis, e.g. `native.*\\.log.*`

`elasticsearch.index` – name of the main index labelling data from logs to be saved in the document database of elasticsearch server (the name of the main index should be the same as the Maximo server's name), e.g. `test_server` if the main index is not defined, maximostash will attempt to build it from a fragment of the logfile name (characters between the first and second underscores) or will take default value `maximo_logs`. It is important to know that the best results one achieve when logfile names will conform to the pattern `(.*_.*_.*\\.log.*)` where first substring from left is the name of server's machine (either real or virtual), second substring (after underscore) is the name of Maximo MXServer and the third substring (after second underscore) is the name of the logger (in terms of Maximo's functionality).

`filter.level` – level of filtration specified by the string: `"INFO,DEBUG,WARN,ERROR"` or the one of its substrings. It indicates which log entries should be downloaded and sent to the elasticsearch server.

`Elasticsearch` server listens port `9200` (REST) and `9300` (API) by default and does not require detailed configuration. The default settings are sufficient. Elasticsearch v 5.1.2 was used in the tests.

`Kibana` server listens by default port `5601` and requires at least one parameter to be specified in the configuration file `config\kibana.yml` :
`server.host` – IP address at which the server UI will be available. e.g. `"192.168.10.1"`

3. Installation and execution

Maximostash program can be placed in any location assuming the correct path to the log folder in `config.properties`. It is not recommended to place this program in the log folder to be analyzed due to the constant saving of the `maxstash.prefs` file, which in turn will cause unnecessary triggering of the log folder analysis even if there is no change caused by the Maximo server.

Each application should be run separately at the command prompt in the order listed below:

1. `elasticsearch\bin\elasticsearch.bat`
2. `kibana\bin\kibana.bat`
3. `java -jar maxstash` (after `config.properties` has been correctly configured).

Maximostash program is waiting for the changes in the folder given by `logs.directory` parameter in the `config.properties` file. In periods of server inactivity, if you need to analyze previously unprocessed logs, you must 'manually' initiate maximostash. This is done by performing any operation on the monitored folder with logfiles, e.g. copying any file there. As a result, after a while the maximostash program will start analyzing the logs from this folder and sending the downloaded data to the elasticsearch server.

In the case of a very large number of previously unprocessed logs of a significant size, it is recommended to run maximostash at the beginning of expected server inactivity period.

The maximostash program has two modes of operation, which it goes through automatically: the first execution, during which it processes in detail the content of all logs and it takes the most running time, and the mode of subsequent executions, during which it sends to elasticsearch only incremental changes of data in the logs. The program saves information about processed logs from the specified folder in the `maxstash.prefs` file. If the `maxstash.prefs` file is deleted, the program will assume that it has not processed any logs yet and will start processing from the beginning.

Maximostash prints to the console confirmations of sending data to the elasticsearch every 500 records. In addition, it sends messages confirming the processing of subsequent logfiles. Maximostash supports two options at the command prompt: `i` – displays information about processed logfiles, `q` – exits.

Confirmation printing can be turned off by the `quiet` parameter, which makes sense when running the program as a Windows service.

Maximostash program sends data to the elasticsearch using its REST port in the form of JSON documents with the following fields:

Field name	Description
`timestamp`	date and time of the sent event
`event`	type of incident: DEBUG, ERROR, INFO, WARN, SystemErr
`bmx`	BMX code
`logger`	a name of MAXIMO's procedure for making particular logs
`file`	name of the file containing the log
`content`	the entire substance of the event as it is recorded in the log
`serverName`	for ‘MboCount’ logger: server name
`serverIP`	for ‘MboCount’ logger: server IP address
`totalMemory`	for ‘MboCount’ logger: total amount of memory in the server
`availMemory`	for ‘MboCount’ logger: available amount of server memory

The fields shown in the table above are extracted from logfiles that comply with three formats:

- `System.*\\.log.*` logfiles (i.e. WebSphere Application Server logs)

  sample row

```
[23.07.19 09:24:31:507 CEST] 000000f6 SystemErr R at com.ibm._jsp._comp._jspService(_comp.java:656)
```

- `.*_.*_.*\\.log.*` logfiles

  sample row

```
12 Apr 2017 09:31:45:098 [ERROR] [MAXIMO] [] Failed to send messages.
```

- `.*_.*_MboCount\\.log.*` logfiles

  sample rows

```
09 Feb 2018 14:52:39:329 [INFO] BMXAA7019I - The total memory is 2450944 and the memory available is 5830528.
09 Feb 2018 14:52:39:345 [INFO] BMXAA6370I - Total number of users connected to the system: 5
09 Feb 2018 14:52:39:345 [INFO] BMXAA6369I - Server host: 192.168.10.1. Name: MXAlpha. Number of users: 5
```

Transferred data becomes visible in the web browser connected to kibana through the standard port `5601`.

4. Data visualization

To view data indexed by elasticsearch, run the kibana UI using web browser (address `http://localhost:5601`). Kibana automatically detects a field containing time data in the indexed data and displays the data for defined main index aligning them to the time axis. In the maximostash test configuration, the main index is the string `test_server` and this string should be entered in the kibana configuration tab ('Management' option, 'Index Pattern' suboption, field below the sentence `Patterns allow you to define dynamic index names ...`).

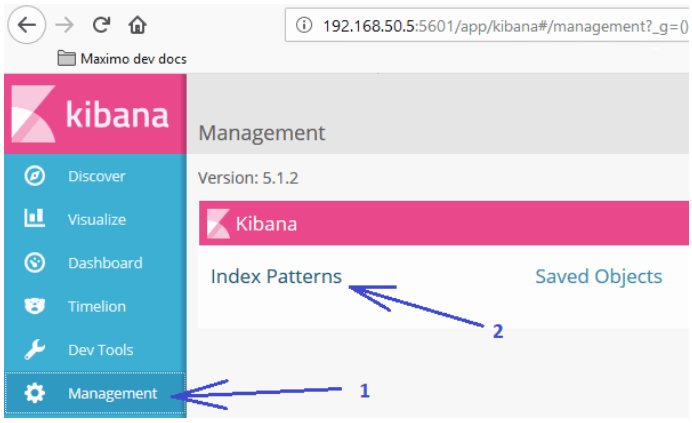

Fig. 1: Web UI of the Kibana server (Configuration tab)

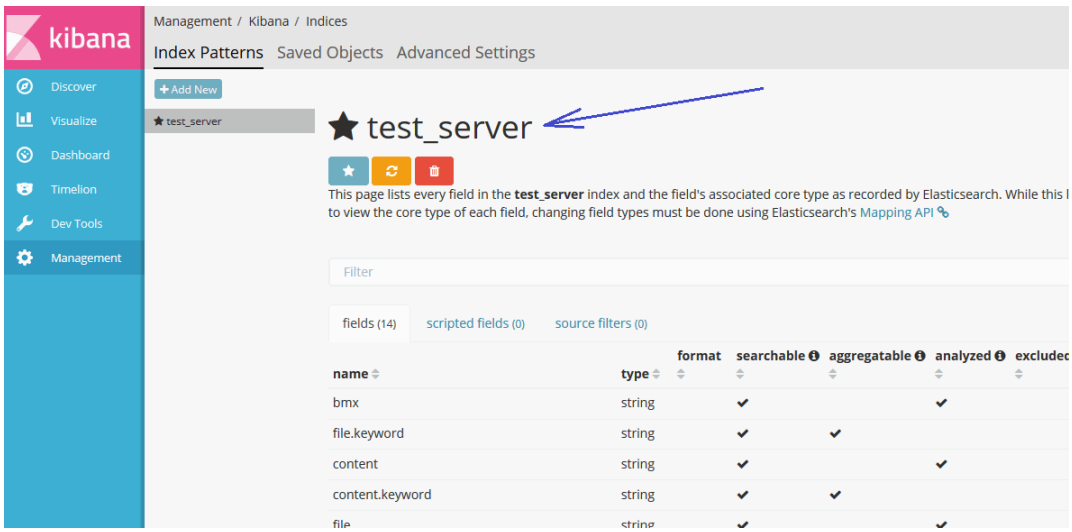

Fig. 2: Web UI of the Kibana server (name of the main index)

There is an option 'Discover' in the kibana web UI. Using this option, pay attention to the first right tool on the upper bar (at the beginning with the inscription `Last 15 minutes`) with the calendar gadget select there proper Time Range corresponding to the date range of indexed recordings downloaded by maximostash and sent to elasticsearch.

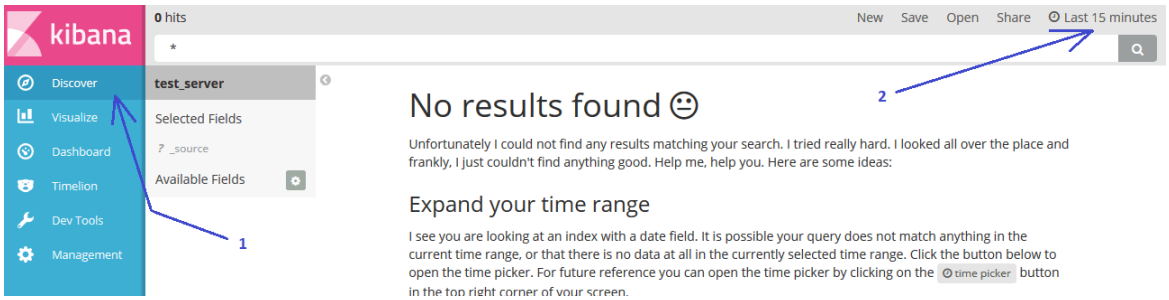

Fig. 3: Web UI of the Kibana server (Discover option)

Visualization of data indexed by elasticsearch will appear below the calendar after providing the correct date range.

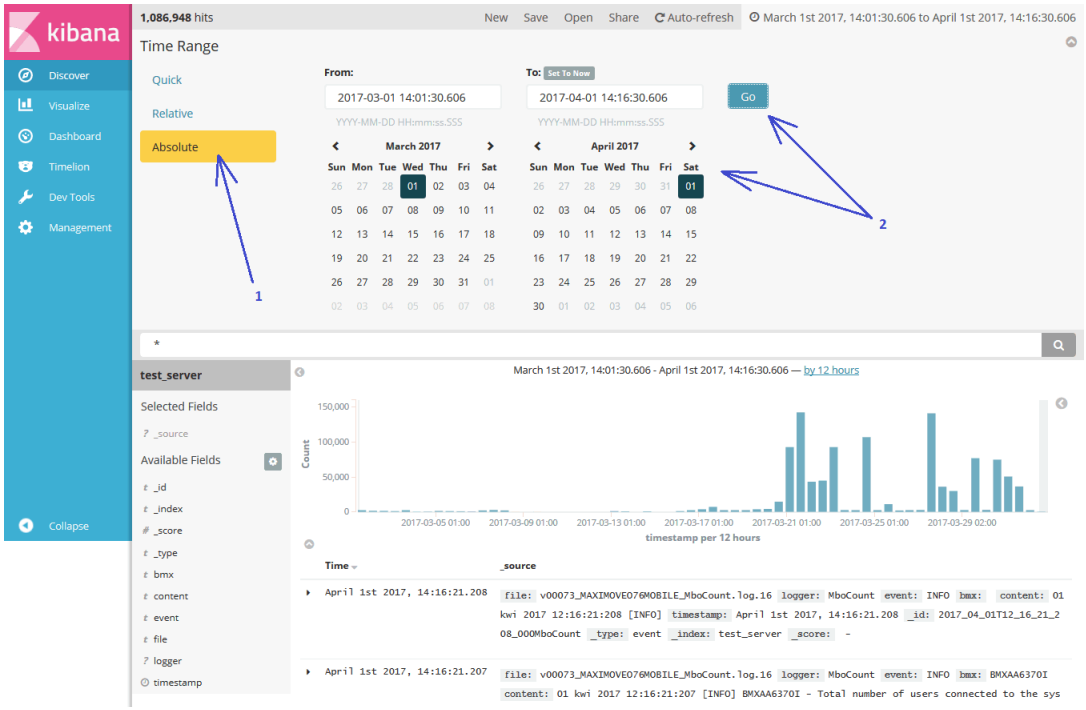

Fig. 4: Web UI of the Kibana server (data view)