

### 3 Elasticsearch setup

Elasticsearch support is experimental!

Setup procedure considered in this section is applicable to the following Elasticsearch versions: **5.0.x** -> **6.1.x**. In case an earlier or later version of Elasticsearch is used, some functionality may not work as intended.

Zabbix has recently started to support storage of historical data by means of Elasticsearch instead of a database. Users are now given the possibility to choose the storage place for historical data between a compatible database and Elasticsearch.

If all history data is stored in Elasticsearch, trends are **not** calculated nor stored in the database. With no trends calculated and stored, the history storage period may need to be extended.

#### Configuration

To ensure proper communication between all elements involved make sure server configuration file and frontend configuration file parameters are properly configured.

#### Zabbix server and frontend

Zabbix server configuration file draft with parameters to be updated:

```
### Option: HistoryStorageURL
# History storage HTTP[S] URL.
#
# Mandatory: no
# Default:
# HistoryStorageURL=
### Option: HistoryStorageTypes
# Comma separated list of value types to be sent to the history storage.
#
# Mandatory: no
# Default:
# HistoryStorageTypes=uint,dbl,str,log,text
```

Example parameter values to fill the Zabbix server configuration file with:

```
HistoryStorageURL=http://test.elasticsearch.lan:9200
HistoryStorageTypes=str,log,text
```

This configuration forces Zabbix Server to store history values of numeric types in the corresponding database and textual history data in Elasticsearch.

Elasticsearch supports the following item types:

```
uint,dbl,str,log,text
```

Supported item type explanation:

Item value type	Database table	Elasticsearch type
Numeric (unsigned)	history_uint	uint
Numeric (float)	history	dbl
Character	history_str	str
Log	history_log	log
Text	history_text	text

Zabbix frontend configuration file (conf/zabbix.conf.php) draft with parameters to be updated:

```
// Elasticsearch url (can be string if same url is used for all types).
$HISTORY['url'] = [
    'uint' => 'http://localhost:9200',
    'text' => 'http://localhost:9200'
];
// Value types stored in Elasticsearch.
$HISTORY['types'] = ['uint', 'text'];
```

Example parameter values to fill the Zabbix frontend configuration file with:

```
$HISTORY['url'] = 'http://test.elasticsearch.lan:9200';
$HISTORY['types'] = ['str', 'text', 'log'];
```

This configuration forces to store Text, Character and Log history values in Elasticsearch.

It is also required to make \$HISTORY global in conf/zabbix.conf.php to ensure everything is working properly (see conf/zabbix.conf.php.example for how to do it):

```
// Zabbix GUI configuration file.
global $DB, $HISTORY;
```

### Installing Elasticsearch and creating mapping

Final two steps of making things work are installing Elasticsearch itself and creating mapping process.

To install Elasticsearch please refer to [Elasticsearch installation guide](#).

Mapping is a data structure in Elasticsearch (similar to a table in a database). Mapping for all history data types is available here: database/elasticsearch/elasticsearch.map.

Creating mapping is mandatory. Some functionality will be broken if mapping is not created according to the instruction.

To create mapping for text type send the following request to Elasticsearch:

```
curl -X PUT \
  http://your-elasticsearch.here:9200/text \
  -H 'content-type:application/json' \
  -d '{
```

```
"settings" : {
  "index" : {
    "number_of_replicas" : 1,
    "number_of_shards" : 5
  }
},
"mappings" : {
  "values" : {
    "properties" : {
      "itemid" : {
        "type" : "long"
      },
      "clock" : {
        "format" : "epoch_second",
        "type" : "date"
      },
      "value" : {
        "fields" : {
          "analyzed" : {
            "index" : true,
            "type" : "text",
            "analyzer" : "standard"
          }
        },
        "index" : false,
        "type" : "text"
      }
    }
  }
}
```

Similar request is required to be executed for Character and Log history values mapping creation with corresponding type correction.

To work with Elasticsearch please refer to [Requirement page](#) for additional information. [Housekeeper](#) is not deleting any data from Elasticsearch.

### Storing history data in multiple date-based indices

This section describes additional steps required to work with pipelines and ingest nodes.

To begin with, you must create templates for indices. The following example shows a request for creating uint template:

```
curl -X PUT \
  http://your-elasticsearch.here:9200/_template/uint_template \
  -H 'content-type:application/json' \
  -d '{
```

```
"template": "uint*",
"index_patterns": ["uint*"],
"settings" : {
  "index" : {
    "number_of_replicas" : 1,
    "number_of_shards" : 5
  }
},
"mappings" : {
  "values" : {
    "properties" : {
      "itemid" : {
        "type" : "long"
      },
      "clock" : {
        "format" : "epoch_second",
        "type" : "date"
      },
      "value" : {
        "type" : "long"
      }
    }
  }
}
```

To create other templates, user should change the URL (last part is the name of template), change "template" and "index\_patterns" fields to match index name and to set valid mapping that can be taken from database/elasticsearch/elasticsearch.map. For example, the following command can be used to create a template for text index:

```
curl -X PUT \
  http://your-elasticsearch.here:9200/_template/text_template \
  -H 'content-type:application/json' \
  -d '{
    "template": "text*",
    "index_patterns": ["text*"],
    "settings" : {
      "index" : {
        "number_of_replicas" : 1,
        "number_of_shards" : 5
      }
    },
    "mappings" : {
      "values" : {
        "properties" : {
          "itemid" : {
            "type" : "long"
          }
        }
      }
    }
  }'
```

```

    },
    "clock" : {
      "format" : "epoch_second",
      "type" : "date"
    },
    "value" : {
      "fields" : {
        "analyzed" : {
          "index" : true,
          "type" : "text",
          "analyzer" : "standard"
        }
      },
      "index" : false,
      "type" : "text"
    }
  }
}
}'

```

This is required to allow Elasticsearch to set valid mapping for indices created automatically. Then it is required to create the pipeline definition. Pipeline is some sort of preprocessing of data before putting data in indices. The following command can be used to create pipeline for uint index:

```

curl -X PUT \
  http://your-elasticsearch.here:9200/_ingest/pipeline/uint-pipeline \
  -H 'content-type:application/json' \
  -d '{
    "description": "daily uint index naming",
    "processors": [
      {
        "date_index_name": {
          "field": "clock",
          "date_formats": ["UNIX"],
          "index_name_prefix": "uint-",
          "date_rounding": "d"
        }
      }
    ]
  }'

```

User can change the rounding parameter ("date\_rounding") to set a specific index rotation period. To create other pipelines, user should change the URL (last part is the name of pipeline) and change "index\_name\_prefix" field to match index name.

See also [Elasticsearch documentation](#).

Additionally, storing history data in multiple date-based indices should also be enabled in the new

parameter in Zabbix server configuration:

```
### Option: HistoryStorageDateIndex
# Enable preprocessing of history values in history storage to store
values in different indices based on date.
# 0 - disable
# 1 - enable
#
# Mandatory: no
# Default:
# HistoryStorageDateIndex=0
```

## Troubleshooting

The following steps may help you troubleshoot problems with Elasticsearch setup:

1. Check if the mapping is correct (GET request to required index URL like `http://localhost:9200/uint`).
2. Check if shards are not in failed state (restart of Elasticsearch should help).
3. Check the configuration of Elasticsearch. Configuration should allow access from the Zabbix frontend host and the Zabbix server host.
4. Check Elasticsearch logs.

If you are still experiencing problems with your installation then please create a bug report with all the information from this list (mapping, error logs, configuration, version, etc.)

From: <https://www.zabbix.com/documentation/4.2/> - **Zabbix Documentation 4.2**

Permanent link: [https://www.zabbix.com/documentation/4.2/manual/appendix/install/elastic\\_search\\_setup](https://www.zabbix.com/documentation/4.2/manual/appendix/install/elastic_search_setup)

Last update: **2019/05/13 12:43**

