+++ title = “Elasticsearch” description = “Guide for using Elasticsearch in Grafana” keywords = [“grafana”, “elasticsearch”, “guide”] aliases = [“/docs/grafana/latest/features/datasources/elasticsearch”] weight = 325 +++

Using Elasticsearch in Grafana

Grafana ships with advanced support for Elasticsearch. You can do many types of simple or complex Elasticsearch queries to visualize logs or metrics stored in Elasticsearch. You can also annotate your graphs with log events stored in Elasticsearch.

Adding the data source

1. Open the side menu by clicking the Grafana icon in the top header.
2. In the side menu under the `Dashboards` link you should find a link named `Data Sources`.
3. Click the `+ Add data source` button in the top header.
4. Select **Elasticsearch** from the **Type** dropdown.

   Note: If you’re not seeing the `Data Sources` link in your side menu it means that your current user does not have the `Admin` role for the current organization.

Name	Description
`Name`	The data source name. This is how you refer to the data source in panels and queries.
`Default`	Default data source means that it will be pre-selected for new panels.
`Url`	The HTTP protocol, IP, and port of your Elasticsearch server.
`Access`	Server (default) = URL needs to be accessible from the Grafana backend/server, Browser = URL needs to be accessible from the browser. **Note**: Browser (direct) access is deprecated and will be removed in a future release.

Access mode controls how requests to the data source will be handled. Server should be the preferred way if nothing else stated.

Server access mode (Default)

All requests will be made from the browser to Grafana backend/server which in turn will forward the requests to the data source and by that circumvent possible Cross-Origin Resource Sharing (CORS) requirements. The URL needs to be accessible from the grafana backend/server if you select this access mode.

Browser (Direct) access

> **Warning:** Browser (Direct) access is deprecated and will be removed in a future release.

All requests will be made from the browser directly to the data source and may be subject to Cross-Origin Resource Sharing (CORS) requirements. The URL needs to be accessible from the browser if you select this access mode.

If you select Browser access you must update your Elasticsearch configuration to allow other domains to access Elasticsearch from the browser. You do this by specifying these two options in your **elasticsearch.yml** config file.

```
http.cors.enabled: true
http.cors.allow-origin: "*"
```

Index settings

Elasticsearch data source details

Here you can specify a default for the `time field` and specify the name of your Elasticsearch index. You can use a time pattern for the index name or a wildcard.

Elasticsearch version

Select the version of your Elasticsearch data source from the version selection dropdown. Different query compositions and functionalities are available in the query editor for different versions. Available Elasticsearch versions are `2.x`, `5.x`, `5.6+`, `6.0+`, `7.0+`, `7.7+` and `7.10+`. Select the option that best matches your data source version.

Grafana assumes that you are running the lowest possible version for a specified range. This ensures that new features or breaking changes in a future Elasticsearch release will not affect your configuration.

For example, suppose you are running Elasticsearch `7.6.1` and you selected `7.0+`. If a new feature is made available for Elasticsearch `7.5.0` or newer releases, then a `7.5+` option will be available. However, your configuration will not be affected until you explicitly select the new `7.5+` option in your settings.

Min time interval

A lower limit for the auto group by time interval. Recommended to be set to write frequency, for example `1m` if your data is written every minute. This option can also be overridden/configured in a dashboard panel under data source options. It's important to note that this value **needs** to be formatted as a number followed by a valid time identifier, e.g. `1m` (1 minute) or `30s` (30 seconds). The following time identifiers are supported:

Identifier	Description
`y`	year
`M`	month
`w`	week
`d`	day
`h`	hour
`m`	minute
`s`	second
`ms`	millisecond

X-Pack enabled

Enables `X-Pack` specific features and options, providing the query editor with additional aggregations such as `Rate` and `Top Metrics`.

Include frozen indices When `X-Pack enabled` is active and the configured Elasticsearch version is higher than `6.6.0`, you can configure Grafana to not ignore frozen indices when performing search requests.

Logs

There are two parameters, `Message field name` and `Level field name`, that can optionally be configured from the data source settings page that determine which fields will be used for log messages and log levels when visualizing logs in [Explore]({{< relref "../explore" >}}).

For example, if you're using a default setup of Filebeat for shipping logs to Elasticsearch the following configuration should work:

- **Message field name:** message
- **Level field name:** fields.level

Data links

Data links create a link from a specified field that can be accessed in logs view in Explore.

Each data link configuration consists of:

- **Field -** Name of the field used by the data link.
- **URL/query -** If the link is external, then enter the full link URL. If the link is internal link, then this input serves as query for the target data source. In both cases, you can interpolate the value from the field with `${__value.raw }` macro.
- **URL Label -** (Optional) Set a custom display label for the link. The link label defaults to the full external URL or name of the linked internal data source and is overridden by this setting.

- **Internal link -** Select if the link is internal or external. In case of internal link, a data source selector allows you to select the target data source. Only tracing data sources are supported.

Metric Query editor

Elasticsearch Query Editor

The Elasticsearch query editor allows you to select multiple metrics and group by multiple terms or filters. Use the plus and minus icons to the right to add/remove metrics or group by clauses. Some metrics and group by clauses haves options, click the option text to expand the row to view and edit metric or group by options.

Series naming and alias patterns

You can control the name for time series via the `Alias` input field.

Pattern	Description
`{{term fieldname}}`	replaced with value of a term group by
`{{metric}}`	replaced with metric name (ex. Average, Min, Max)
`{{field}}`	replaced with the metric field name

Pipeline metrics

Some metric aggregations are called Pipeline aggregations, for example, *Moving Average* and *Derivative*. Elasticsearch pipeline metrics require another metric to be based on. Use the eye icon next to the metric to hide metrics from appearing in the graph. This is useful for metrics you only have in the query for use in a pipeline metric.

Pipeline aggregation editor

Templating

Instead of hard-coding things like server, application and sensor name in your metric queries you can use variables in their place. Variables are shown as dropdown select boxes at the top of the dashboard. These dropdowns make it easy to change the data being displayed in your dashboard.

Check out the Templating({{< relref "../variables/_index.md" >}}) documentation for an introduction to the templating feature and the different types of template variables.

Query variable

The Elasticsearch data source supports two types of queries you can use in the *Query* field of *Query* variables. The query is written using a custom JSON string. The field should be mapped as a keyword in the Elasticsearch index mapping. If it is multi-field with both a `text` and `keyword` type, then use `"field":"fieldname.keyword"`(sometimes`fieldname.raw`) to specify the keyword field in your query.

Query	Description
`{"find": "fields", "type": "keyword"}`	Returns a list of field names with the index type `keyword`.
`{"find": "terms", "field": "hostname.keyword", "size": 1000}`	Returns a list of values for a keyword using term aggregation. Query will use current dashboard time range as time range query.
`{"find": "terms", "field": "hostname", "query": '<lucene query>'}`	Returns a list of values for a keyword field using term aggregation and a specified lucene query filter. Query will use current dashboard time range as time range for query.

There is a default size limit of 500 on terms queries. Set the size property in your query to set a custom limit. You can use other variables inside the query. Example query definition for a variable named `$host`.

```
{"find": "terms", "field": "hostname", "query": "source:$source"}
```

In the above example, we use another variable named `$source` inside the query definition. Whenever you change, via the dropdown, the current value of the `$source` variable, it will trigger an update of the `$host` variable so it now only contains hostnames filtered by in this case the `source` document property.

These queries by default return results in term order (which can then be sorted alphabetically or numerically as for any variable). To produce a list of terms sorted by doc count (a top-N values list), add an `orderBy` property of "doc_count". This automatically selects a descending sort; using "asc" with doc_count (a bottom-N list) can be done by setting `order: "asc"` but is discouraged as it "increases the error on document counts". To keep terms in the doc count order, set the variable's Sort dropdown to **Disabled**; you might alternatively still want to use e.g. **Alphabetical** to re-sort them.

```
{"find": "terms", "field": "hostname", "orderBy": "doc_count"}
```

Using variables in queries

There are two syntaxes:

- `$<varname>` Example: hostname:$hostname
- `[[varname]]` Example: hostname:[[hostname]]

Why two ways? The first syntax is easier to read and write but does not allow you to use a variable in the middle of a word. When the *Multi-value* or *Include all value* options are enabled, Grafana converts the labels from plain text to a lucene compatible condition.

Query with template variables

In the above example, we have a lucene query that filters documents based on the `hostname` property using a variable named `$hostname`. It is also using a variable in the *Terms* group by field input box. This allows you to use a variable to quickly change how the data is grouped.

Example dashboard: Elasticsearch Templated Dashboard

Annotations

Annotations({{< relref "../dashboards/annotations.md" >}}) allow you to overlay rich event information on top of graphs. You add annotation queries via the Dashboard menu / Annotations view. Grafana can query any Elasticsearch index for annotation events.

Name	Description
`Query`	You can leave the search query blank or specify a lucene query.
`Time`	The name of the time field, needs to be date field.
`Time End`	Optional name of the time end field needs to be date field. If set, then annotations will be marked as a region between time and time-end.
`Text`	Event description field.
`Tags`	Optional field name to use for event tags (can be an array or a CSV string).

Querying Logs

Querying and displaying log data from Elasticsearch is available in [Explore]({{< relref "../explore" >}}), and in the [logs panel]({{< relref "../visualizations/logs-panel.md" >}}) in dashboards. Select the Elasticsearch data source, and then optionally enter a lucene query to display your logs.

When switching from a Prometheus or Loki data source in Explore, your query is translated to an Elasticsearch log query with a correct Lucene filter.

Log Queries

Once the result is returned, the log panel shows a list of log rows and a bar chart where the x-axis shows the time and the y-axis shows the frequency/count.

Note that the fields used for log message and level is based on an optional data source configuration.

Filter Log Messages

Optionally enter a lucene query into the query field to filter the log messages. For example, using a default Filebeat setup you should be able to use `fields.level:error` to only show error log messages.

Configure the data source with provisioning

It's now possible to configure data sources using config files with Grafana's provisioning system. You can read more about how it works and all the settings you can set for data sources on the [provisioning docs page]({{< relref "../administration/provisioning/#datasources" >}})

Here are some provisioning examples for this data source.

```
apiVersion: 1

datasources:
  - name: Elastic
    type: elasticsearch
    access: proxy
    database: '[metrics-]YYYY.MM.DD'
    url: http://localhost:9200
    jsonData:
      interval: Daily
      timeField: '@timestamp'
```

or, for logs:

```
apiVersion: 1

datasources:
  - name: elasticsearch-v7-filebeat
    type: elasticsearch
    access: proxy
    database: '[filebeat-]YYYY.MM.DD'
    url: http://localhost:9200
    jsonData:
      interval: Daily
      timeField: '@timestamp'
      esVersion: '7.0.0'
```

```
    logMessageField: message
    logLevelField: fields.level
    dataLinks:
      - datasourceUid: my_jaeger_uid # Target UID needs to be known
        field: traceID
        url: '$${__value.raw}' # Careful about the double "$$" because of env var expansio
```

Amazon Elasticsearch Service

AWS users using Amazon's Elasticsearch Service can use Grafana's Elasticsearch data source to visualize Elasticsearch data. If you are using an AWS Identity and Access Management (IAM) policy to control access to your Amazon Elasticsearch Service domain, then you must use AWS Signature Version 4 (AWS SigV4) to sign all requests to that domain. For more details on AWS SigV4, refer to the AWS documentation.

AWS Signature Version 4 authentication

> **Note:** Only available in Grafana v7.3+.

In order to sign requests to your Amazon Elasticsearch Service domain, SigV4 can be enabled in the Grafana [configuration]({{< relref "../administration/configuration.md#sigv4_auth_enabled" >}}).

Once AWS SigV4 is enabled, it can be configured on the Elasticsearch data source configuration page. Refer to [Cloudwatch authentication]({{< relref "../datasources/aws-cloudwatch/aws-authentication.md" >}}) for more information about authentication options.

{{< figure src="/static/img/docs/v73/elasticsearch-sigv4-config-editor.png" max-width="500px" class="docs-image--no-shadow" caption="SigV4 configuration for AWS Elasticsearch Service" >}}