

```
+++ title = "Jaeger" description = "Guide for using Jaeger in Grafana" keywords = ["grafana", "jaeger", "guide",  
"tracing"] aliases = ["/docs/grafana/latest/features/datasources/jaeger"] weight = 800 +++
```

Jaeger data source

Grafana ships with built-in support for Jaeger, which provides open source, end-to-end distributed tracing. Just add it as a data source and you are ready to query your traces in [Explore]({{< relref "../explore/_index.md" >}}).

Add data source

To access Jaeger settings, click the **Configuration** (gear) icon, then click **Data Sources > Jaeger**.

Name	Description
Name	The data source name in panels, queries, and Explore.
Default	The pre-selected data source for a new panel.
URL	The URL of the Jaeger instance. For example, <code>http://localhost:16686</code> .
Basic Auth	Enable basic authentication for the Jaeger data source.
User	Specify a user name for basic authentication.
Password	Specify a password for basic authentication.

Trace to logs

Note: This feature is available in Grafana 7.4+.

This is a configuration for the [trace to logs feature]({{< relref "../explore/trace-integration" >}}). Select target data source (at this moment limited to Loki and Splunk [logs] data sources) and select which tags will be used in the logs query.

- **Data source** - Target data source.
- **Tags** - The tags that will be used in the logs query. Default is `'cluster', 'hostname', 'namespace', 'pod'`.
- **Map tag names** - When enabled, allows configuring how Jaeger tag names map to logs label names. For example, map `service.name` to `service`.
- **Span start time shift** - Shift in the start time for the logs query based on the span start time. In order to extend to the past, you need to use a negative value. Use time interval units like 5s, 1m, 3h. The default is 0.
- **Span end time shift** - Shift in the end time for the logs query based on the span end time. Time units can be used here, for example, 5s, 1m, 3h. The default is 0.
- **Filter by Trace ID** - Toggle to append the trace ID to the logs query.
- **Filter by Span ID** - Toggle to append the span ID to the logs query.



Trace to logs settings

Node Graph

This is a configuration for the beta Node Graph visualization. The Node Graph is shown after the trace view is loaded and is disabled by default.

-- **Enable Node Graph** - Enables the Node Graph visualization.

Query traces

You can query and display traces from Jaeger via [Explore]({{< relref "../explore/_index.md" >}}).

{{< figure src="/static/img/docs/explore/jaeger-search-form.png" class="docs-image--no-shadow" caption="Screenshot of the Jaeger query editor" >}}

You can query by trace ID or use the search form to find traces. To query by trace ID, select the TraceID from the Query type selector and insert the ID into the text input.

{{< figure src="/static/img/docs/explore/jaeger-trace-id.png" class="docs-image--no-shadow" caption="Screenshot of the Jaeger query editor with trace ID selected" >}}

To perform a search, set the query type selector to Search, then use the following fields to find traces:

- Service - Returns a list of services.
- Operation - Field gets populated once you select a service. It then lists the operations related to the selected service. Select `All` option to query all operations.
- Tags - Use values in the [logfmt](#) format. For example `error=true db.statement="select * from User"`.
- Min Duration - Filter all traces with a duration higher than the set value. Possible values are `1.2s`, `100ms`, `500us`.
- Max Duration - Filter all traces with a duration lower than the set value. Possible values are `1.2s`, `100ms`, `500us`.
- Limit - Limits the number of traces returned.

Upload JSON trace file

You can upload a JSON file that contains a single trace to visualize it. If the file has multiple traces then the first trace is used for visualization.

{{< figure src="/static/img/docs/explore/jaeger-upload-json.png" class="docs-image--no-shadow" caption="Screenshot of the Jaeger data source in explore with upload selected" >}}

Here is an example JSON:

```
{
  "data": [
    {
      "traceID": "2ee9739529395e31",
      "spans": [
        {
          "traceID": "2ee9739529395e31",
          "spanID": "2ee9739529395e31",
          "flags": 1,
          "operationName": "CAS",
          "references": [],
          "startTime": 1616095319593196,
          "duration": 1004,
          "tags": [
            {
```

```

        "key": "sampler.type",
        "type": "string",
        "value": "const"
    }
],
"logs": [],
"processID": "p1",
"warnings": null
}
},
"processes": {
    "p1": {
        "serviceName": "loki-all",
        "tags": [
            {
                "key": "jaeger.version",
                "type": "string",
                "value": "Go-2.25.0"
            }
        ]
    }
},
"warnings": null
}
],
"total": 0,
"limit": 0,
"offset": 0,
"errors": null
}

```

Linking Trace ID from logs

You can link to Jaeger trace from logs in Loki by configuring a derived field with internal link. See the [Derived fields]({{< relref "loki.md#derived-fields" >}}) section in the [Loki data source]({{< relref "loki.md" >}}) documentation for details.

Configure the data source with provisioning

You can set up the data source via configuration files with Grafana's provisioning system. Refer to [provisioning docs page]({{< relref "../administration/provisioning/#datasources" >}}) for more information on configuring various settings.

Here is an example with basic auth and trace-to-logs field.

```

apiVersion: 1

datasources:
- name: Jaeger
  type: jaeger
  uid: jaeger-spectra

```

```
access: proxy
url: http://localhost:16686/
basicAuth: true
basicAuthUser: my_user
editable: true
isDefault: false
jsonData:
  tracesToLogs:
    # Field with internal link pointing to a logs data source in Grafana.
    # datasourceUid value must match the `datasourceUid` value of the logs data
source.
    datasourceUid: 'loki'
    tags: ['job', 'instance', 'pod', 'namespace']
    mappedTags: [{ key: 'service.name', value: 'service' }]
    mapTagNamesEnabled: false
    spanStartTimeShift: '1h'
    spanEndTimeShift: '1h'
    filterByTraceID: false
    filterBySpanID: false
secureJsonData:
  basicAuthPassword: my_password
```