Amazon CloudWatch query editor

- Key concepts
- Choose a query editing mode
- CloudWatch Metrics query editor components
- Use Builder and Code modes
  - Builder mode
  - Code mode
- Query CloudWatch metrics
  - Metric Search queries
    - Create dynamic queries with dimension wildcards
    - Use multi-value template variables
    - Use metric math expressions
  - Query metrics across AWS monitoring accounts
  - Period macro
  - Deep-link Grafana panels to the CloudWatch console
  - Use Metric Insights syntax
  - Use Metrics Insights keywords
- Query CloudWatch Logs
  - Specify log groups to query
  - Specify Amazon CloudWatch Logs data sources to query
  - Query log groups and Amazon CloudWatch Logs data sources with OpenSearch SQL {#query-log-groups-with
  - Create queries for alerting
- Common use cases
- Cross-account observability
- Query caching

Grafana provides a query editor for the CloudWatch data source, which allows you to query, visualize, and alert on logs and metrics stored in Amazon CloudWatch. It is located on the Explore page. For general documentation on querying data sources in Grafana, refer to Query and transform data.

Key concepts

If you're new to CloudWatch, these terms are used throughout this documentation:

Term	Description
Namespace	A container for CloudWatch metrics, usually one per AWS service, such as `AWS/EC2` or `AWS/Lambda`.
Metric	A time-ordered set of data points published to CloudWatch, such as `CPUUtilization`.
Dimension	A name-value pair that identifies a specific resource for a metric, such as `InstanceId=i-1234567890abcdef0`.
Statistic	An aggregation applied to metric data over a period, such as `Average`, `Sum`, or `Maximum`.
Period	The length of time, in seconds, used to aggregate metric data points.
Metric math	Expressions that combine or transform metrics, such as computing a rate or a sum across metrics.
Metrics Insights	A SQL-like query engine for filtering and aggregating metrics in near real time.
Search expression	A CloudWatch expression that matches multiple metrics at query time, often used with wildcards.
Logs Insights QL	The CloudWatch Logs query language used to search and aggregate log data.

Choose a query editing mode

The CloudWatch data source can query data from both CloudWatch metrics and CloudWatch Logs APIs, each with its own specialized query editor.

- CloudWatch Metrics
- CloudWatch Logs

Select the API to query using the drop-down to the right of the **Region** setting.

CloudWatch Metrics query editor components

The following are the components of the CloudWatch query editor.

Setting	Description
Region	Select an AWS region if it differs from the default.
Namespace	The AWS service namespace. Examples: `AWS/EC2`, `AWS/Lambda`.
Metric name	The name of the metric you want to visualize. Example: `CPUUtilization`.
Statistic	Choose how to aggregate your data. Options are `Average`, `Maximum`, `Minimum`, `Sum`, `SampleCount`, and `IQM`. You can also enter a custom value, such as a percentile like `p99`.
Dimensions	Select dimensions from the drop-down. Examples: `InstanceId`, `FunctionName`, `latency`. You can add several dimensions to your query.
Match exact	*Optional*. When enabled, this option restricts query results to metrics that precisely match the specified dimensions and their values. All dimensions of the queried metric must be explicitly defined in the query to ensure an exact schema match. If disabled, the query will also return metrics that match the defined schema but possess additional dimensions.
ID	*Optional*. Unique identifier required by the GetMetricData API for referencing queries in math expressions. Must start with a lowercase letter and can include letters, numbers, and underscores. If not specified, Grafana generates an ID using the pattern `query[refId]` (for example, `queryA` for the first query row).
Period	The minimum time interval, in seconds, between data points. The default is `auto`. CloudWatch accepts periods of 1, 5, 10, 30, or any multiple of 60 seconds. When set to `auto` or left blank, Grafana calculates the period by dividing the time range in seconds by 2000, then rounding up to the next value among 60, 300, 900, 3600, 21600, and 86400. The available periods depend on the age of the data: because CloudWatch stores older metrics at a lower resolution under its retention policy, the shorter periods aren't available for older time ranges.

Setting	Description
Label	*Optional*. Add a customized time series legend name. The label field overrides the default metric legend name using CloudWatch dynamic labels. Time-based dynamic labels like ${MIN_MAX_TIME_RANGE} derive legend values from the current timezone in the time range picker. For the full list of label patterns and limitations, refer to CloudWatch dynamic labels.

Use Builder and Code modes

Both the Metric Search and Metric Insights query types provide two editing modes. Use the **Builder** and **Code** toggle in the query editor to switch between them.

Builder mode

Builder mode lets you construct a query from drop-downs and form fields without writing any syntax:

- For **Metric Search** queries, select the namespace, metric name, statistic, and dimensions.
- For **Metric Insights** queries, select the namespace, metric name, filter, group, and order options using the Metrics Insights keywords. Grafana constructs a SQL query based on your selections.

Code mode

Code mode lets you write your query directly in a code editor:

- For **Metric Search** queries, write a metric math or `SEARCH` expression.
- For **Metric Insights** queries, write a Metrics Insights SQL query.

The code editor includes a built-in autocomplete feature that suggests keywords, aggregations, namespaces, metrics, labels, and label values. Suggestions appear after typing a space, comma, or dollar (`$`) character, or by pressing CTRL+Space.

{{< admonition type="note" >}} Template variables in the code editor can interfere with autocompletion. {{< /admonition >}}

To run the query, click **Run query** above the code editor.

Query CloudWatch metrics

You can create two types of queries in the CloudWatch query editor:

- Metric Search, which helps you retrieve and filter available metrics.
- Metric Insights, which uses the Metrics Insights feature to fetch time series data.

The query type you use depends on how you want to interact with AWS metrics. Use the drop-down in the upper middle of the query editor to select which type you want to create.

Metric Search queries

Metric search queries help you discover and filter available metrics. These queries use wildcards and filters to find metrics without needing to know exact metric names.

A valid metric query requires a specified namespace, metric name, and at least one statistic. Dimensions are optional, but if included, you must provide both a `key` and a `value`.

The `Match Exact` option controls how dimension filtering is applied to metric queries. When you enable `match exact`, the query returns only metrics whose dimensions precisely match the specified criteria.

This requires the following:

- All dimensions present on the target metric must be explicitly specified.
- Dimensions you don't want to filter by must use a wildcard (*) filter.
- The metric schema must match exactly as defined in the CloudWatch metric schema documentation.

When `Match Exact` is disabled, you can specify any subset of dimensions for filtering. The query returns metrics that:

- Match the specified namespace and metric name.
- Match all defined dimension filters.
- May contain additional dimensions beyond those specified.

This mode provides more flexible querying but may return metrics with unexpected additional dimensions.

Because the data source retrieves metrics through the CloudWatch `GetMetricData` API, a single query returns data for up to 100 metrics matching your filter criteria, which is the per-request limit that AWS enforces.

Enhance metric queries using template variables to create dynamic, reusable dashboards.

Create dynamic queries with dimension wildcards

Use the asterisk (`*`) wildcard for dimension values to create dynamic queries that automatically monitor changing sets of AWS resources.

{{< figure src="/static/img/docs/cloudwatch/cloudwatch-dimension-wildcard-8.3.0.png" max-width="500px" caption="CloudWatch dimension wildcard" >}}

The query returns the average CPU utilization for all EC2 instances in the default region. With `Match Exact` disabled and `InstanceId` using a wildcard, the query retrieves metrics for any EC2 instance regardless of additional dimensions.

Auto-scaling events add new instances to the graph without manual instance ID tracking. As with other metric search queries, AWS returns up to 100 metrics per `GetMetricData` request.

Click the **Query inspector** button and select **Meta Data** to see the search expression that's automatically built to support wildcards.

To learn more about search expressions, refer to the CloudWatch documentation. The search expression is defined by default in such a way that the queried metrics must match the defined dimension names exactly. This means that in the example, the query returns only metrics with exactly one dimension containing the name `InstanceId`.

{{< figure src="/static/img/docs/cloudwatch/cloudwatch-meta-inspector-8.3.0.png" max-width="500px" class="docs-image--right" caption="CloudWatch Meta Inspector" >}}

Disabling `Match Exact` includes metrics with additional dimensions and creates a search expression even without wildcards. Grafana searches for any metric matching at least the namespace, metric name, and all defined dimensions.

Use multi-value template variables

When defining dimension values based on multi-valued template variables, the data source uses a search expression to query for the matching metrics. This enables the use of multiple template variables in one query, and also lets you use template variables for queries that have the `Match Exact` option disabled.

Search expressions are limited to 1,024 characters, so your query might fail if you have a long list of values. We recommend using the asterisk (`*`) wildcard instead of the `All` option to query all metrics that have any value for a certain dimension name.

Multi-valued template variables are supported only for dimension values. Using multi-valued template variables for `Region`, `Namespace`, or `Metric Name` is not supported.

Use metric math expressions

Create new time series metrics using mathematical functions on CloudWatch metrics. This supports arithmetic operators, unary subtraction, and other functions. For available functions, refer to AWS Metric Math.

To apply arithmetic operations, assign a unique string ID to the raw metric, then reference this ID in the `Expression` field of the new metric.

{{< admonition type="note" >}} If you use the expression field to reference another query, such as `queryA * 2`, you can't create an alert rule based on that query. {{< /admonition >}}

Query metrics across AWS monitoring accounts

When you select `Builder` mode within the Metric search editor, a new Account field is displayed. Use the `Account` field to specify which of the linked monitoring accounts to target for the given query. By default, the `All` option is specified, which will target all linked accounts.

While in `Code` mode, you can specify any math expression. If the Monitoring account badge displays in the query editor header, all `SEARCH` expressions entered in this field will be cross-account by default and can query metrics from linked accounts. Note that while queries run cross-account, the autocomplete feature currently doesn't fetch cross-account resources, so you'll need to manually specify resource names when writing cross-account queries. You can limit the search to one or a set of accounts, as documented in the AWS documentation.

Period macro

If you use a CloudWatch `SEARCH` expression, consider using the `$__period_auto` macro rather than specifying a period explicitly. The `$__period_auto` macro will resolve to a CloudWatch period that is suitable for the chosen time range.

Deep-link Grafana panels to the CloudWatch console

Left-clicking a time series in the panel displays a context menu with a link to `View in CloudWatch console`. Clicking the link opens a new tab that takes you to the CloudWatch console and displays all metrics for that query. If you're not logged in to the CloudWatch console, the link forwards you

to the login page. The link provided is valid for any account but displays the expected metrics only if you're logged in to the account that corresponds to the selected data source in Grafana.

{{< figure src="/media/docs/cloudwatch/cloudwatch-deep-link-v12.1.png" caption="CloudWatch deep linking" >}}

This feature is not available for metrics based on metric math expressions.

Use Metric Insights syntax

Metric Insights uses a dialect of SQL and this query syntax:

```
SELECT FUNCTION(MetricName)
FROM Namespace | SCHEMA(...)
[ WHERE labelKey OPERATOR labelValue [AND|...]]
[ GROUP BY labelKey [, ...]]
[ ORDER BY FUNCTION() [DESC | ASC] ]
[ LIMIT number]
```

For details about the Metrics Insights syntax, refer to the AWS reference documentation.

For information about Metrics Insights limits, refer to the AWS feature documentation.

You can also augment queries by using template variables.

Use Metrics Insights keywords

This table summarizes common Metrics Insights query keywords:

Keyword	Description
`FUNCTION`	Required. Specifies the aggregate function to use, and also specifies the name of the metric to be queried. Valid values are AVG, COUNT, MAX, MIN, and SUM.
`MetricName`	Required. For example, `CPUUtilization`.
`FROM`	Required. Specifies the metric's source. You can specify either the metric namespace that contains the metric to be queried, or a SCHEMA table function. Namespace examples include `AWS/EC2`, `AWS/Lambda`.
`SCHEMA`	Optional. Narrows the query results to only the metrics that are an exact match, or to metrics that do not match.

Keyword	Description
`WHERE`	Optional. Filters the query results to only the metrics that match your specified expression. For example, `WHERE InstanceType != 'c3.4xlarge'`.
`GROUP BY`	Optional. Groups the query results into multiple time series. For example, `GROUP BY ServiceName`.
`ORDER BY`	Optional. Specifies the order in which time series are returned. Options are `ASC`, `DESC`.
`LIMIT`	Optional. Limits the number of time series returned.

Query CloudWatch Logs

The logs query editor helps you write CloudWatch Logs queries across a selected region and a set of log groups or Amazon CloudWatch Logs data sources.

You can query CloudWatch Logs using three supported query language options:

- **Logs Insights QL** - The AWS native query language specifically designed for CloudWatch Logs. It uses a SQL-like syntax with commands like `fields`, `filter`, `stats`, and `sort`. It's optimized for the CloudWatch log structure and offers built-in functions for parsing timestamps, extracting fields from JSON logs, and performing aggregations.
- **OpenSearch SQL** - OpenSearch SQL is a query language that uses a SQL-like syntax for querying data in OpenSearch. It supports standard SQL queries and is designed for users familiar with SQL.
- **OpenSearch PPL** - The OpenSearch query language is based on Elasticsearch's query DSL (Domain Specific Language). It uses a pipe-based syntax similar to Unix command-line tools or the Splunk search language, and supports complex boolean logic, range queries, wildcard matching, and full-text search capabilities.

Create a CloudWatch Logs query:

1. Select a region.
2. Select **CloudWatch Logs** from the query type drop-down.
3. Use the Logs Mode selector to choose between Logs Insights and Log Anomalies queries.

Log Anomalies

Anomaly detection uses machine-learning and pattern recognition to establish baselines of typical log content.
The Log Anomalies query editor fetches the list of anomalies detected in your CloudWatch service. In order to query log anomalies in the editor, a log anomaly detector must be created in the AWS CloudWatch console first. The log trend cell shows the number of occurrences of the pattern over

the selected query time range. The table shows 50 log anomalies at a time. If you would like to narrow down the list, you can filter anomalies by their ARN and suppressed state.

In addition to this, you can use the Logs Insights QL editor and the `anomaly` command together with the `patterns` command to define and display log anomalies in real time. See the CloudWatch Logs Insights documentation for more info.

Logs Insights

The **Query language** drop-down is available only when **Logs Mode** is set to Logs Insights.

1. Select the query language you want to use in the **Query language** drop-down.
2. Specify the log groups, CloudWatch Logs data sources, or both that you want to query.
3. Use the main input area to write your logs query. Amazon CloudWatch only supports a subset of OpenSearch SQL and PPL commands. To find out more about the syntax supported, consult Amazon CloudWatch Logs documentation

Specify log groups to query

To target log groups in your query, use one of the following methods:

- **Log group name**: Select specific log groups. This is the default behavior. You can select up to 50 log groups. Click **Select log groups** to choose the target log groups for your query. When the **Monitoring account** badge appears in the query editor header, you can search and select log groups across multiple accounts. Use the **Accounts** filter to narrow results by account. For large lists, use prefix search to narrow the selection. The selector loads log groups in pages, so you can scroll to load more results when many log groups are available.
- **Name prefix**: Select the **Name prefix** query scope and provide up to five log group name prefixes. The query will run against log groups that have names that start with the specified prefixes. Each prefix must be at least three characters and must not include `*`. This option is only available for **Logs Insights QL**.
- **All log groups**: Select the **All log groups** query scope. This queries all log groups in the selected region. This option is only available for **Logs Insights QL**.

You can also apply optional filters when specifying log groups with the **Name prefix** or **All log groups** query scopes:

- **Class**: Filter by log group class. Choose **Standard** or **Infrequent Access**.
- **Accounts**: Filter by AWS account. This option is only available when the **Monitoring account** badge appears in the query editor header. You can select up to 20 accounts.

Specify Amazon CloudWatch Logs data sources to query

Amazon CloudWatch Logs data sources categorize logs by the service or application that generates them and by the type of log. They complement log groups and let you query logs without knowing individual log group names. For details about how Amazon CloudWatch categorizes logs, refer to Data source discovery and management in the AWS documentation.

Before you use the data source selector, ensure that the IAM role or IAM user for the Grafana data source has the `logs:ListAggregateLogGroupSummaries` permission. For the complete IAM policy examples, refer to IAM policy examples.

To query logs by CloudWatch Logs data source, follow these steps:

1. Click **Select data sources**.
2. Search for a data source name or type.
3. Select up to 10 data sources.
4. Click **Apply selection**.

You can combine CloudWatch Logs data source selections with specific log groups or with the **Name prefix** and **All log groups** query scopes. When the **Monitoring account** badge appears, the selector includes data sources from the monitoring account and its linked source accounts.

The CloudWatch data source automatically adds the selected CloudWatch Logs data sources to Logs Insights QL and OpenSearch PPL queries.

{{< admonition type="note" >}} When you use **Logs Insights QL**, **OpenSearch PPL**, or **OpenSearch SQL**, you must specify a region and define the logs to query by selecting log groups, selecting CloudWatch Logs data sources, or choosing a supported log group query scope.

In **OpenSearch SQL**, you can specify log groups in multiple ways. For details, refer to the Query log groups and Amazon CloudWatch Logs data sources with OpenSearch SQL section. {{< /admonition >}}

Click **View in CloudWatch console** to interactively view, search, and analyze your log data in the CloudWatch Logs Insights console. If you're not logged in to the CloudWatch console, the link forwards you to the login page.

Query log groups and Amazon CloudWatch Logs data sources with

OpenSearch SQL {#query-log-groups-with-opensearch-sql}

When querying log groups and Amazon CloudWatch Logs data sources with OpenSearch SQL, you can use the `$__source` macro to automatically reference the log groups and CloudWatch Logs data sources selected in the query editor. This is the recommended approach as it allows you to manage both types of selection through the UI.

```
SELECT window.start, COUNT(*) AS exceptionCount
FROM `$__source`
WHERE `@message` LIKE '%Exception%'
```

The `$__source` macro expands to the proper syntax for the log groups and CloudWatch Logs data sources you've selected in the UI.

Alternatively, you can manually specify a single log group directly in the `FROM` clause:

```
SELECT window.start, COUNT(*) AS exceptionCount
FROM `log_group`
WHERE `@message` LIKE '%Exception%'
```

When querying multiple log groups you **must** use the `logGroups(logGroupIdentifier: [...])` syntax:

```
SELECT window.start, COUNT(*) AS exceptionCount
FROM `logGroups(logGroupIdentifier: ['LogGroup1', 'LogGroup2'])`
WHERE `@message` LIKE '%Exception%'
```

To reference log groups in a monitoring account, use ARNs instead of LogGroup names.

You can also write queries returning time series data by using the `stats` command. When making `stats` queries in Explore, ensure you are in Metrics Explore mode.

Create queries for alerting

Alerting requires queries that return numeric data, which CloudWatch Logs supports. For example, you can enable alerts through the use of the `stats` command.

The following is a valid query for alerting on messages that include the text "Exception":

```
filter @message like /Exception/
    | stats count(*) as exceptionCount by bin(1h)
    | sort exceptionCount desc
```

{{< admonition type="note" >}} If you receive an error like `input data must be a wide series but got ...` when trying to alert on a query, make sure that your query returns valid numeric data that can be output to a Time series panel. {{< /admonition >}}

For more information on Grafana alerts, refer to Alerting.

Common use cases

The following scenarios show how to combine the query editor's features to solve real monitoring problems.

- **Monitor an EC2 fleet:** Use a Metric Search query for `AWS/EC2` `CPUUtilization` with the `InstanceId` dimension set to a wildcard (`*`) and `Match Exact` disabled. The graph automatically includes new instances as your fleet scales.
- **Track Lambda errors and invocations:** Query the `AWS/Lambda` namespace for the `Errors` and `Invocations` metrics, then add a metric math expression such as `errors / invocations * 100` to chart an error rate.
- **Rank the busiest resources:** Use a Metrics Insights query with `ORDER BY` and `LIMIT` to return the top N resources by a metric, for example, the 10 EC2 instances with the highest average CPU utilization.
- **Investigate application errors:** Use a CloudWatch Logs query with `filter` and `stats` to count errors over time, then correlate the spikes with metric panels on the same dashboard.
- **Alert on log error rates:** Build a Logs query that returns numeric data with the `stats` command, then create an alert rule on the result. Refer to Create queries for alerting.

Cross-account observability

The CloudWatch plugin monitors and troubleshoots applications that span multiple accounts within a region. Cross-account observability enables seamless searching, visualization, and analysis of metrics and logs across account boundaries.

To enable cross-account observability, complete the following steps:

1. Go to the Amazon CloudWatch documentation and follow the instructions for enabling cross-account observability.

2. Add two API actions to the IAM policy attached to the role/user running the plugin.

Cross-account querying is available in the plugin through the **Logs**, **Metric search**, and **Metric Insights** modes. After you have configured it, you'll see a **Monitoring account** badge in the query editor header.

{{< figure src="/static/img/docs/cloudwatch/cloudwatch-monitoring-badge-9.3.0.png" max-width="1200px" caption="Monitoring account badge" >}}

Query caching

When you enable query and resource caching, Grafana temporarily stores the results of data source queries and resource requests. Query caching is available in CloudWatch Metrics in Grafana Cloud and Grafana Enterprise. It is not available in CloudWatch Logs Insights due to how query results are polled from AWS.

□□□ #2
□ □□□ □□□ 2026-10-31 15:53:47 UTC
□ □□□ □□□ 2026-10-31 15:53:50 UTC