+++ title = "Query caching" description = "Grafana Enterprise data source query caching" keywords = ["grafana", "plugins", "query", "caching"] weight = 300 +++

Query caching

When query caching is enabled, Grafana temporarily stores the results of data source queries. When you or another user submit the exact same query again, the results will come back from the cache instead of from the data source (like Splunk or ServiceNow) itself.

Query caching works for all backend data sources, and queries sent through the data source proxy. You can enable the cache globally and configure the cache duration (also called Time to Live, or TTL).

The following cache backends are available: in-memory, Redis, and Memcached.

> ***Note:*** *Storing cached queries in-memory can increase Grafana's memory footprint. In production environments, a Redis or Memcached backend is highly recommended.*

When a panel queries a cached data source, the time until this query fetches fresh data is determined by the panel's **interval.** This means that wider panels and dashboards with shorter time ranges fetch new data more frequently than narrower panels and dashboards with longer time ranges.

Interval is visible in a panel's [query options]({{< relref "../panels/reference-query-options.md" >}}). It is calculated like this: `(max data points) / time range`. Max data points are calculated based on the width of the panel. For example, a full-width panel on a dashboard with a time range of `last 7 days` will retrieve fresh data every 10 minutes. In this example, cached data for this panel will be served for up to 10 minutes before Grafana queries the data source again and returns new data.

You can make a panel retrieve fresh data more frequently by increasing the **Max data points** setting in the panel's [query options]({{< relref "../panels/reference-query-options.md" >}}).

Query caching benefits

- Faster dashboard load times, especially for popular dashboards.
- Reduced API costs.
- Reduced likelihood that APIs will rate-limit or throttle requests.

Data sources that work with query caching

Query caching works for all Enterprise data sources, and it works for the following [built-in data sources]({{< relref "../datasources/_index.md" >}}):

- CloudWatch Metrics
- Google Cloud Monitoring
- InfluxDB
- Microsoft SQL Server
- MySQL
- Postgres
- Tempo

Some data sources, such as Elasticsearch, Prometheus, and Loki, cache queries themselves, so Grafana query caching does not improve performance.

Query caching also works for all data sources that include a backend. More specifically, caching works with data sources that extend the `DataSourceWithBackend` class in the plugins SDK.

To tell if a data source works with query caching, follow the instructions below to **Enable and Configure query caching**. If caching is enabled in Grafana but the Caching tab is not visible for the given data source, then query caching is not available for that data source.

Enable and configure query caching

You must be an Org admin or Grafana admin to enable query caching for a data source. For more information on Grafana roles and permissions, refer to [About users and permissions]({{< relref "../administration/manage-users-and-permissions/about-users-and-permissions.md" >}}).

By default, data source queries are not cached. To enable query caching for a single data source:

1. On the side menu, click Configuration > Data Sources.
2. In the data source list, click the data source that you want to turn on caching for.
3. Open the Cache tab.
4. Press the Enable button.
5. (Optional) Choose custom TTLs for the data source's queries and resources caching. If you skip this step, then Grafana uses the default TTL.

> ***Note:** If query caching is enabled and the Cache tab is not visible in a data source's settings, then query caching is not available for that data source.*

To configure global settings for query caching, refer to the [Query caching section of Enterprise Configuration]({{< relref "./enterprise-configuration.md#caching" >}}).

Disable query caching

To disable query caching for a single data source:

1. On the side menu, click Configuration > Data Sources.
2. In the data source list, click the data source that you want to turn off caching for.
3. In the Cache tab, click Disable.

To disable query caching for an entire Grafana instance, set the `enabled` flag to `false` in the [Query caching section of Enterprise Configuration]({{< relref "./enterprise-configuration.md#caching" >}}). You will no longer see the Cache tab on any data sources, and no data source queries will be cached.

Clear cache

If you experience performance issues or repeated queries become slower to execute, consider clearing your cache.

> ***Note:** This action impacts all cache-enabled data sources. If you are using Memcached, the system clears all data from the Memcached instance.*

To clear the cache:

1. Sign in to Grafana and click **Settings > Data Sources**.
2. Select a data source.
3. Click the **Cache** tab.
4. Click **Clear cache**.

Sending a request without cache

If a data source query request contains an `X-Cache-Skip` header, then Grafana skips the caching middleware, and does not search the cache for a response. This can be particularly useful when debugging data source queries using cURL.