Log Querying and Analytics Lab

Throughout this lab, each section will be broken down into a series of steps. To navigate between sections, click each header to expand or collapse the sections.

Make sure you are logged into Datadog using the Datadog training account credentials provisioned for you. You can find that information by running `creds` in the lab terminal.

Logs Search

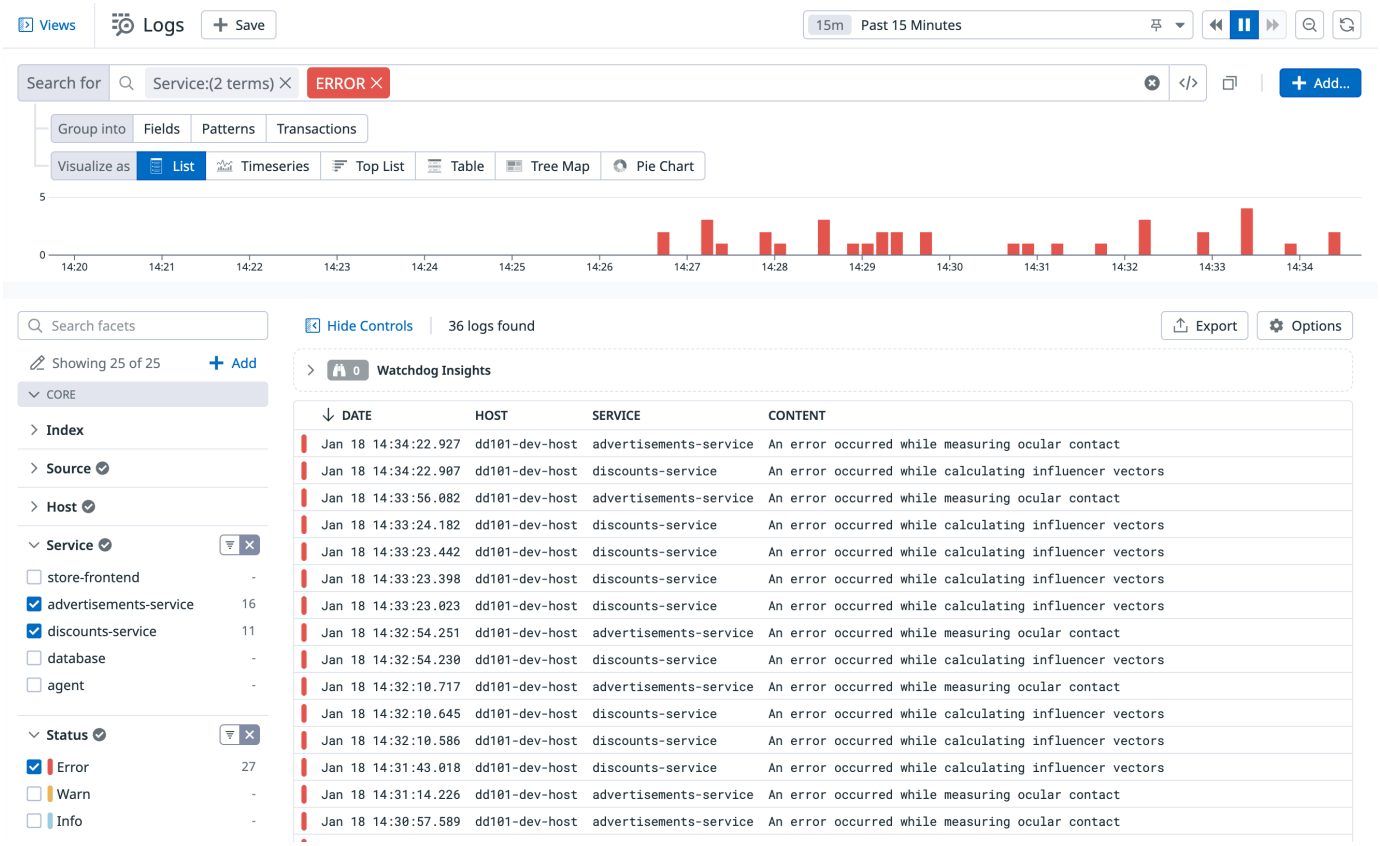
The Logs Search page displays indexed application logs matching a search query for a selected time range. A search query defines the criteria to "filter" the log entries to display. You can progressively filter the logs to home in on the specific types of log lines that you're looking for.

The Datadog Agent is collecting many log lines from all of Storedog's services and sending them to the Datadog to process.

1. Log in to Datadog using the trial credentials the lab created for you. You can run `creds` in the lab terminal whenever you need to retrieve your Datadog training account credentials.
2. Navigate to **Logs > Search**.

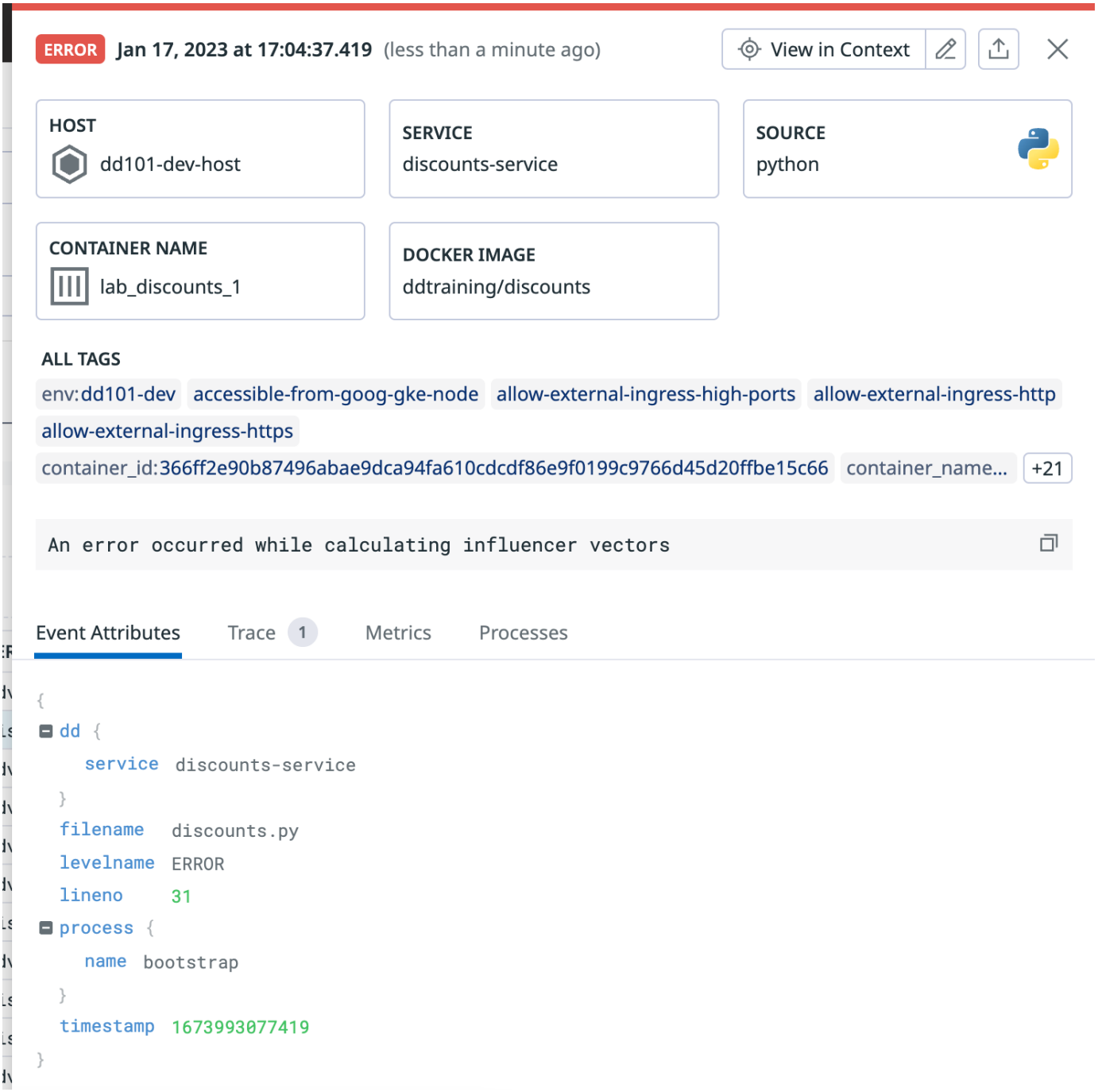
   You can build a search query using the search field at the top of the page and the Facets panel to the left of the search results.

   A search query can include assigned tags, like `env` and `service`; attributes extracted from the logs, like `@http.status_code`; and text strings from log messages. Search queries built in the search field require proper search syntax.
3. In the Facets panel, under **Service**, select `advertisements-service` and `discounts-service`, and under **Status**, select `Error` to filter logs and display only error log lines from those two services:



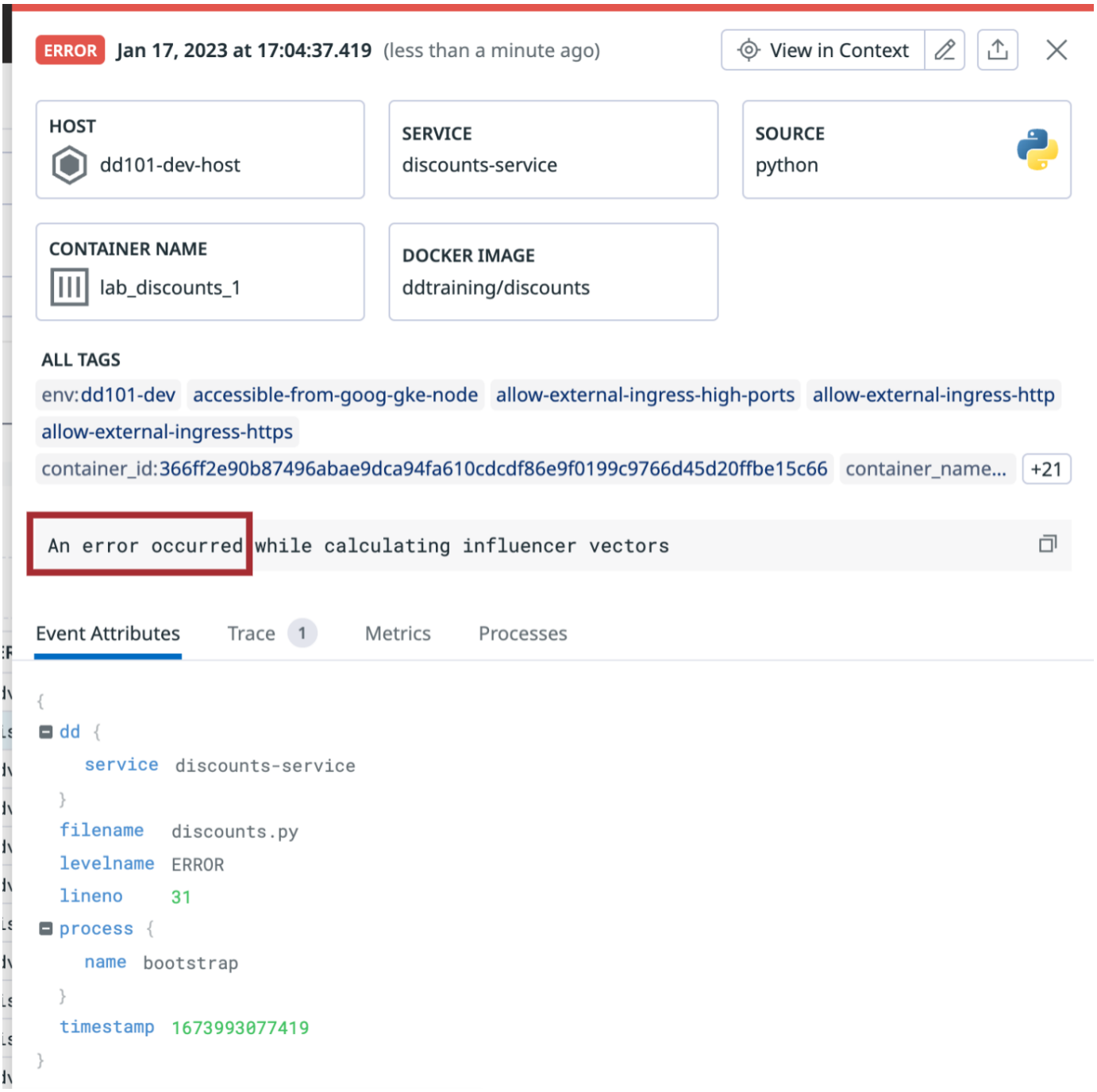
4. Find an error log that starts with `ERROR` and click it.

   The Log Side Panel opens with the log details including assigned tags, the log message, any extracted attributes, related traces, and related infrastructure metrics.

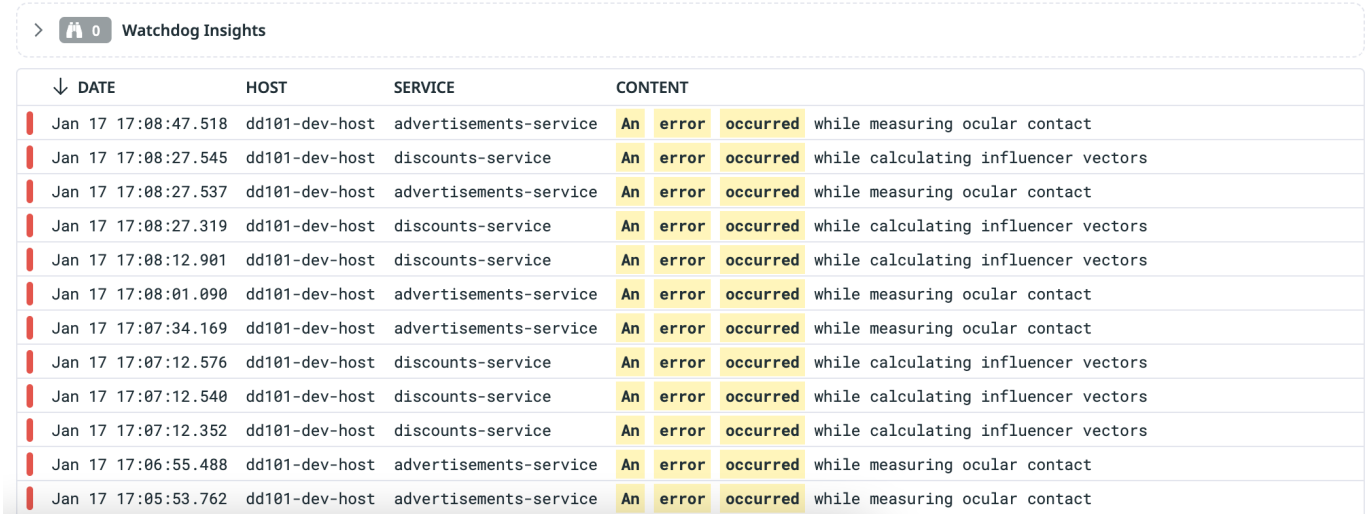


5. Get familiar with the details provided by the log side panel:
   - Click the **Trace** tab to view the associated trace.
   - Click the **Metrics** tab to view the associated infrastructure metric.
6. Notice the log message beginning with "An error occurred. . . ". Instead of using the `ERROR` status, use this string to filter error logs for these services.

   Copy the part of the log message that reads `An error occurred`, and close the log side panel.



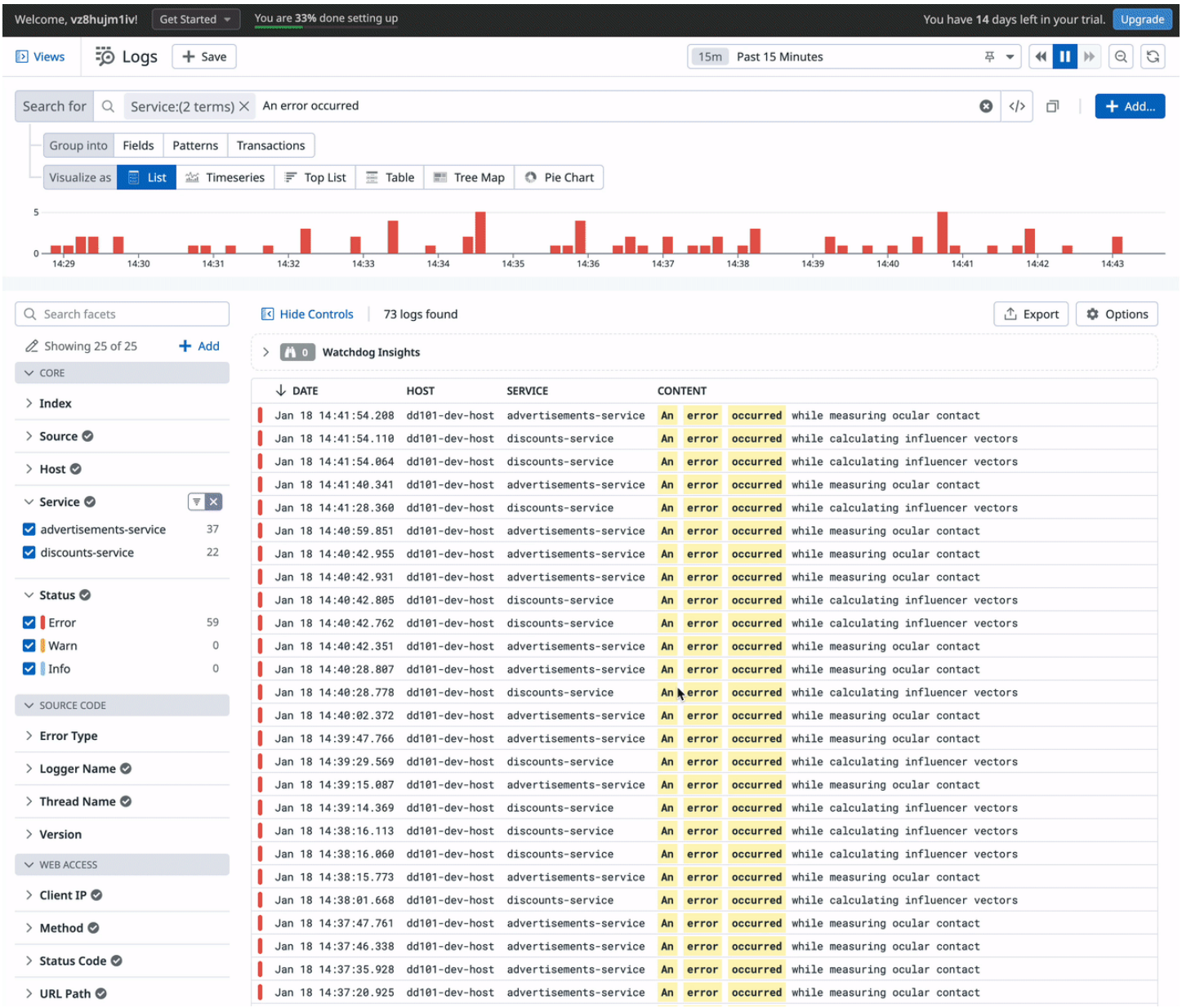
7. In the search field, delete `ERROR` from the search field.
8. Paste the log message text `An error occurred` in the search field and press Enter. Notice the same list of logs appears.



Now you know how to build a search query using the facets panel as well as the search field.

Custom Facets

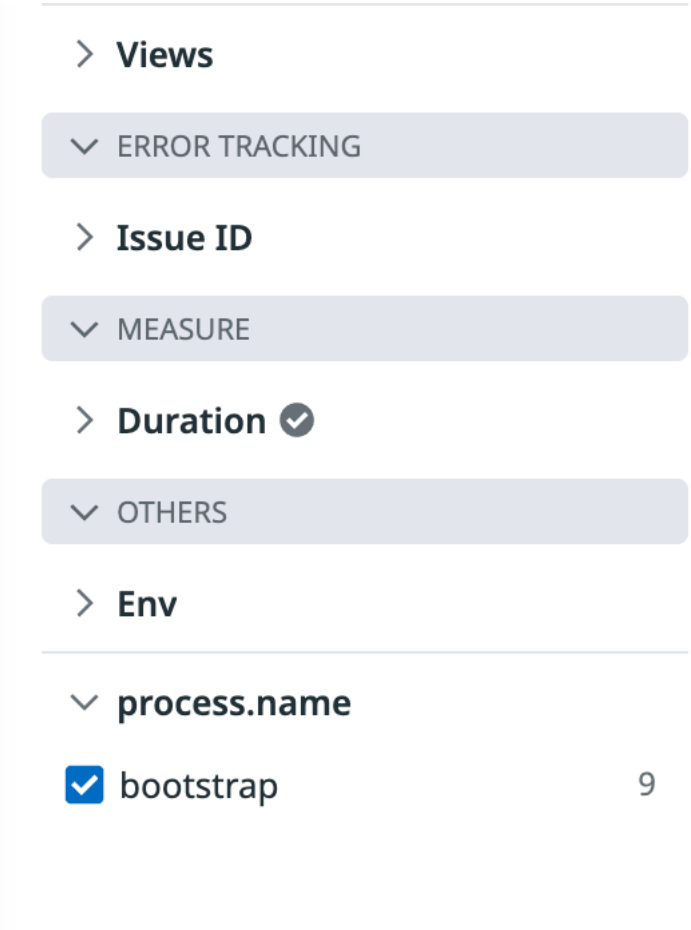
Common tags and attributes automatically appear in the Facets panel as Datadog parses logs. You can also add a tag to the Facets panel as a custom facet from the log details in the log side panel.



1. Click one of the error logs to open the log side panel. Scroll down to view the list of **Event Attributes**.
2. Under the `process` attribute, hover over `name` and click the gear icon that appears.
3. Select **Create facet for @process.name**.

   The **Add facet** window will appear. Expand **Advanced** to view the additional fields.
4. Click **Add**. You'll see a message confirming that the facet has been successfully added.
5. Close the logs side panel.
6. Clear the search field at the top of the page and select `Past 15 Minutes` from the timeframe dropdown in the upper-right.
7. Scroll to the bottom of the Facets list. Under the **OTHERS** facet group, expand the **process.name** facet.

   You may need to wait for fresh logs to be collected and processed for the new facet to appear. Soon, you'll see the values of this attribute that was found in log entries:



8. Select `bootstrap` to filter the logs by lines where the `process.name` attribute value is `bootstrap`.

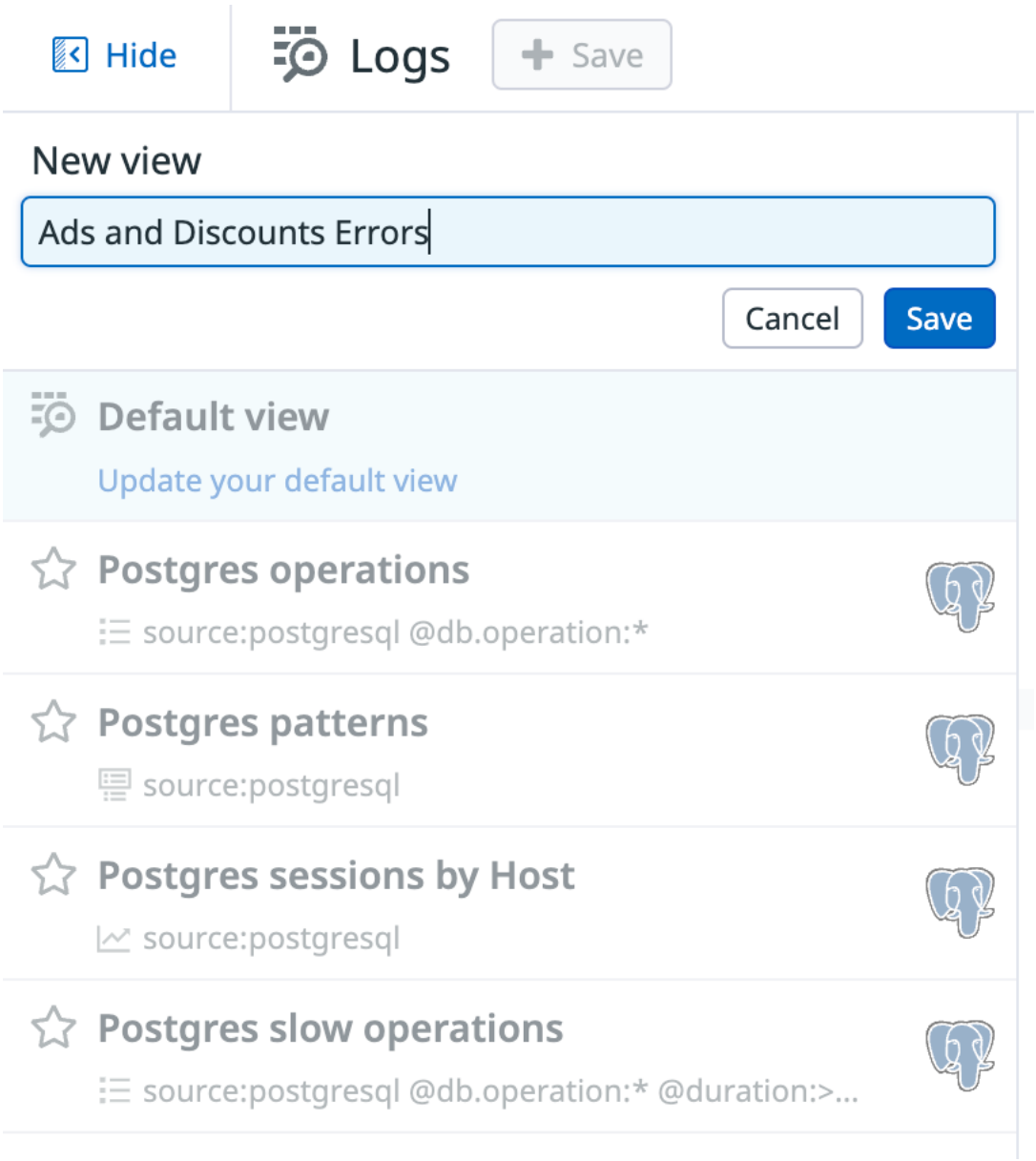
It's important to remember, though, that the tags and attributes available to you for creating search contexts depends on the tags you assign to the logs and the attributes you extract from the logs.

Saved Views

You can also save search queries as Saved Views to recall at any time:

1. In the Facets panel, under **Service**, make sure `advertisements-service` and `discounts-service` is selected, and under **Status**, `Error` is selected.
2. Click the + **Save** button above the search field to save this search query.

   The Views panel will open with a list of Saved Views. In addition to `Default view`, you will see predefined views provided by the PostgreSQL integration:



3. Under **New view**, in the **Name** field, enter `Ads and Discounts Errors` and click **Save**.

   The new view will appear in the list.

4. Click the **Default view** saved view. This will clear the search query.
5. Click the new **Ads and Discounts Errors** view you just created. You'll see the search query populate with the saved view.
6. Click **Hide** above the filtered views to close the Views panel.

Now that you've gone over the Log Search and querying, it's time to look at the different Aggregation features for analyzing logs.

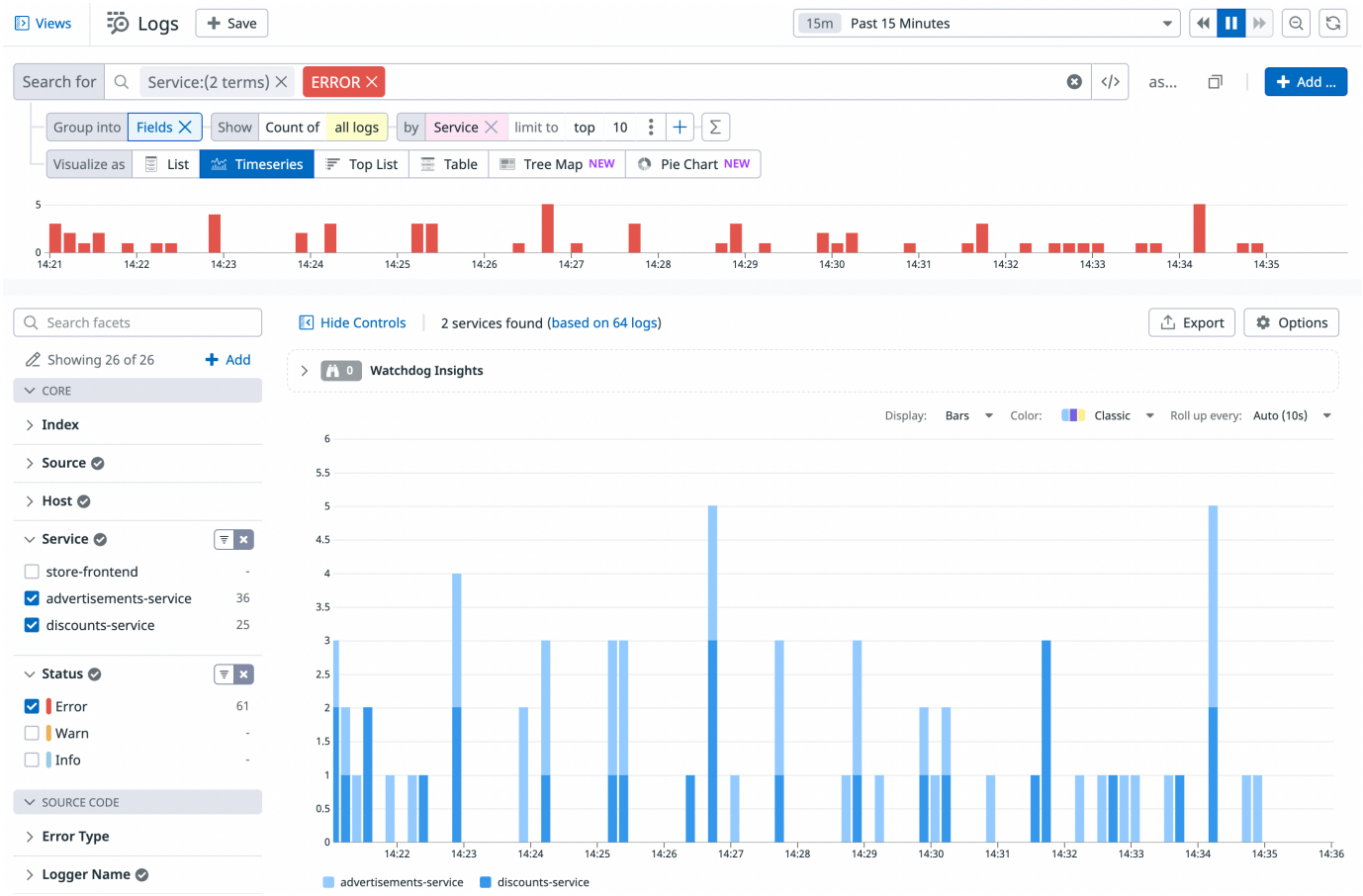
Fields Aggregation

With the Fields aggregation option, which is located right underneath the log query filter at the top of the page, all logs matching the query filter are aggregated into groups based on the value of a log facet. For these groups, you can extract counts of logs per group, unique count of coded values for a facet per group, and statistical operations on numerical values of a facet per group.

1. Navigate to **Logs** > **Search**.

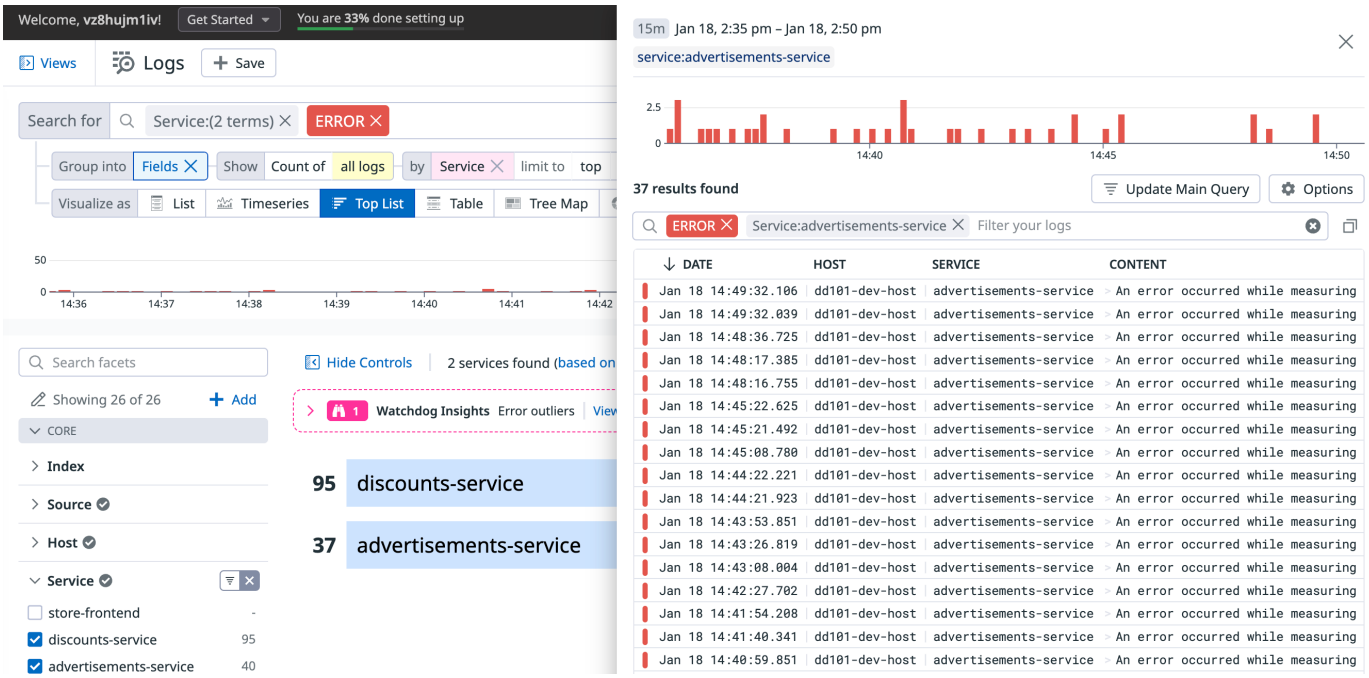
2. Open the Views panel and select the saved view you created, **Ads and Discounts Errors**.
3. Below the search field, for **Group into**, select `Fields`.

   A graph visualization of the filtered logs will replace the Log List.
4. Group the fields to show a count of `all logs` by `service`:



5. For **Visualize as** above the graph, you'll see that **Timeseries** is selected by default, which gives you a timeline of total errors broken out by service.

   Select the other visualization options, like **Top List**, to see what they look like. Note that the visualizations are interactive. You can click on rows or graphs to view a list of the aggregated logs in a side panel:



Exporting Graphs

You can export any logs visualization to other areas of the Datadog App, such as Monitors, Dashboards, and Notebooks. You can create a custom metric from logs, or download the aggregated data as a CSV.

1. For **Visualize as**, select **Timeseries**.
2. Click the **Export** button above the graph. You can export the visualization areas of the product, such as a Logs Monitor, to a dashboard, and to generate a log-based metric.
3. Click **Export to dashboard**.
4. In the export dialog, click the **New Dashboard** link.

   A notice at the top of the screen confirms the creation of the new dashboard.
5. Under **Visualize as**, click **List** to return to the default log list view.

In the next section, you'll aggregate logs based on patterns that Datadog detects across all of your log contents.

Patterns Aggregation

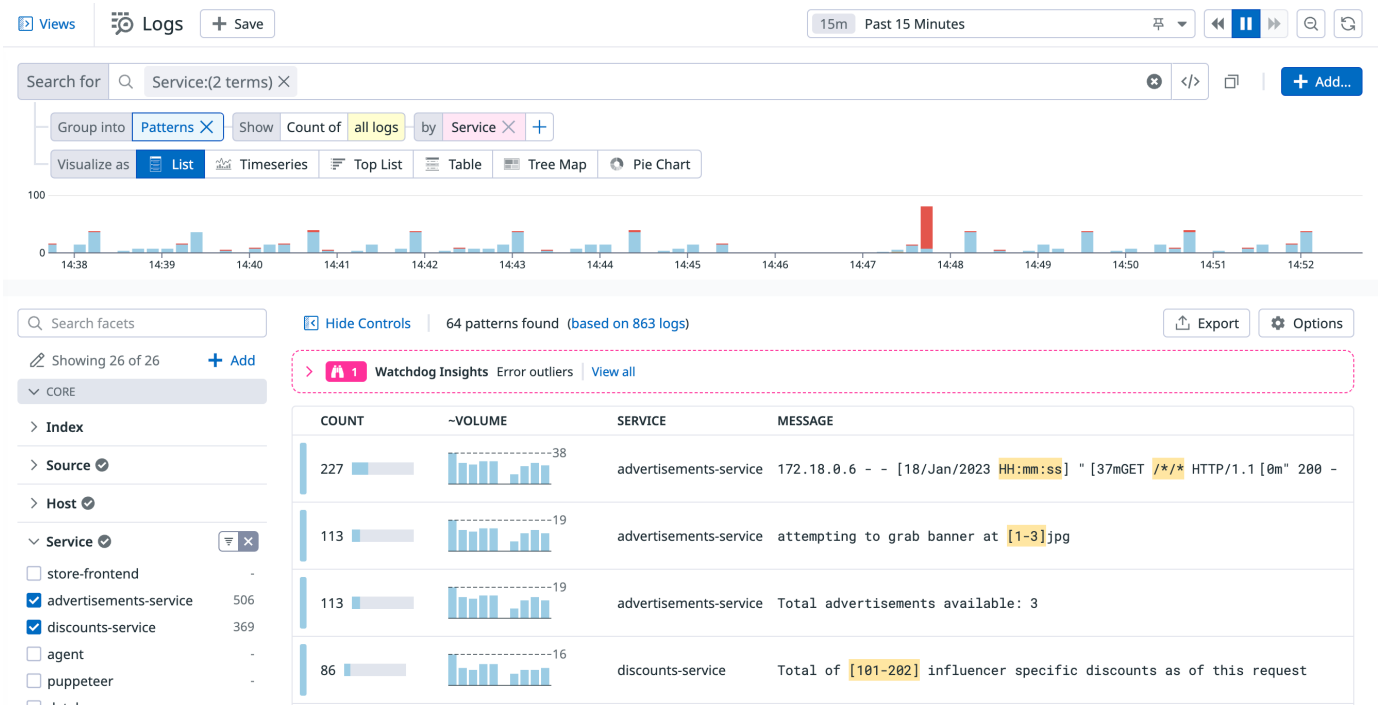
Organizing large volumes of logs from different sources can be cumbersome. However, logs from a source usually have specific patterns. The Patterns aggregation can be surfaced automatically in the Log Explorer and are listed by number of logs with a service name, log status, and log message that matches a certain pattern.

You can filter the list using search syntax and facets to focus on patterns you're interested in. And, you can click a pattern from the list to view log samples matching the pattern and even the pattern's parsing rule.

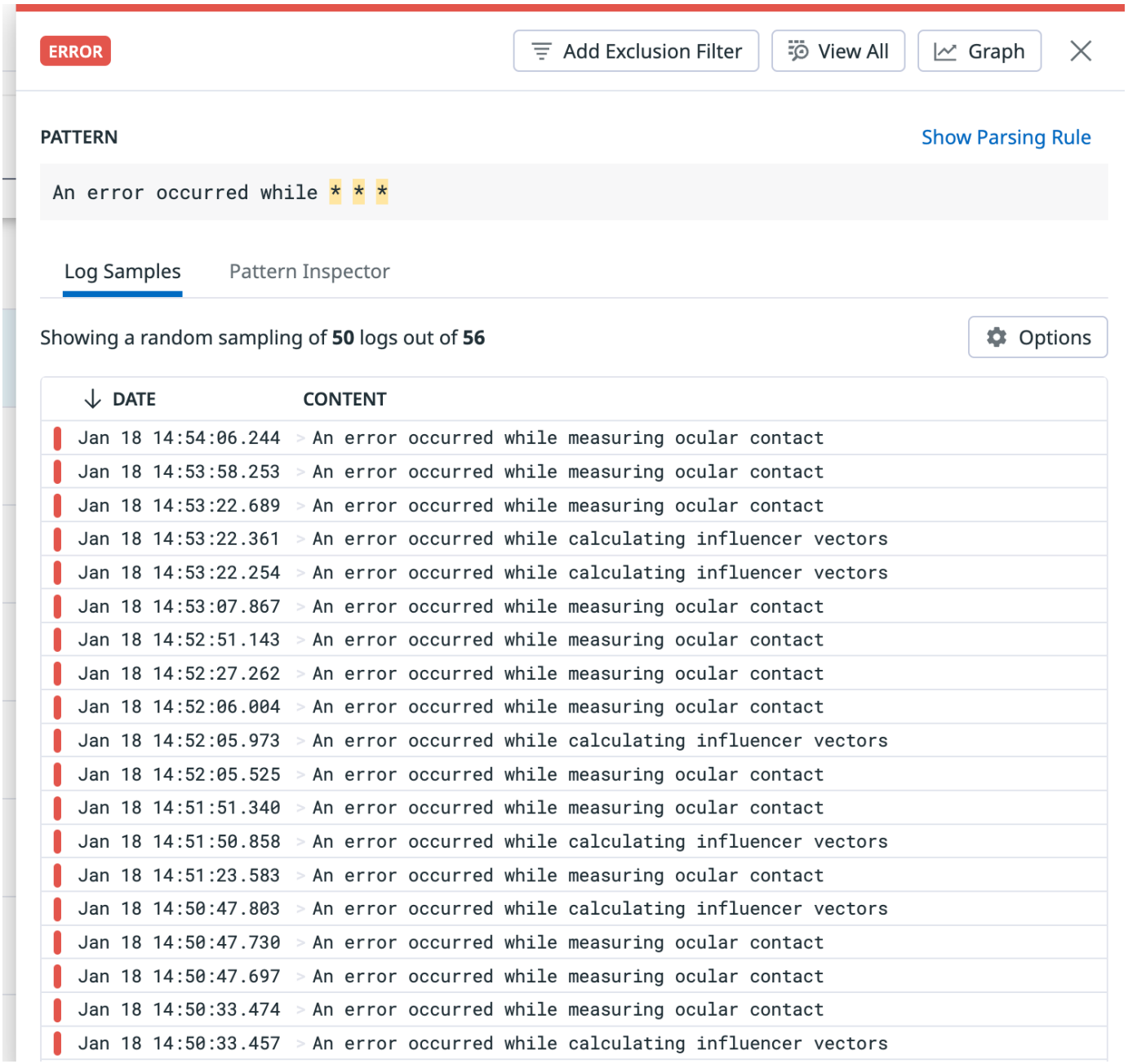
With pattern aggregation, logs that have a message with similar structures, belong to the same service and have the same status are grouped altogether. The patterns view is helpful for detecting and filtering noisy error patterns that could cause you to miss other issues.

1. Navigate to **Logs > Search**.
2. Open the Views panel and select the saved view you created, **Ads and Discounts Errors**.
3. Remove the `ERROR` status from the search field.
4. For **Group into**, select `Patterns`.

   The detected log patterns are displayed, sorted from the most common to the least common. You can see the count, the facet by which they are grouped (`service`, in this case), and the log message.

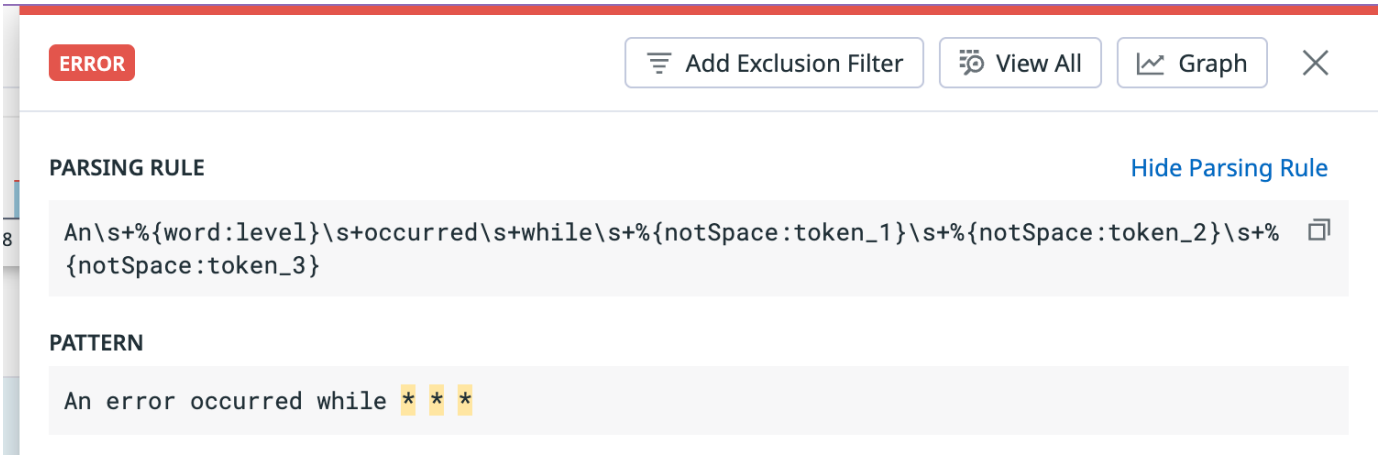


5. Group **by** to `process.name`. Here you can see the patterns grouped by the custom facet you created earlier.
6. Change **by** to `status` to see the patterns grouped by status value.
7. The log list column headings are sortable. Click the **COUNT** header twice to sort from least common to most common.
8. Click one of the patterns to open the details side panel. The pattern and a list of Log Samples are displayed.



9. In the upper-right above the Pattern, click **Show Parsing Rule** above the pattern. This is a regular expression dialect known as Grok, one of the parsers used in Datadog log pipelines.

   Parsing rules can be very powerful for creating a custom pipeline to parse semi-structured log lines into well-structured, taggable objects, in the same way that Datadog automatically parses JSON log lines.



You can take a look at the pipelines Datadog set up automatically under **Logs > Configuration > Pipelines**.

> **Note**: Creating custom pipelines is out of scope for this lab, but make a note of the Pipelines docs and check out the Going Deeper with Logs: Processing course in the Learning Center when you'd like to explore this topic in detail.

10. Close the side panel.
11. Click the X next to **Patterns** to return to the Log List.

Next, you'll learn about transaction aggregation.

Transactions Aggregation

Transaction queries allow you to aggregate related log events into a single higher-level "transaction" event using a tag or an attribute from the logs.

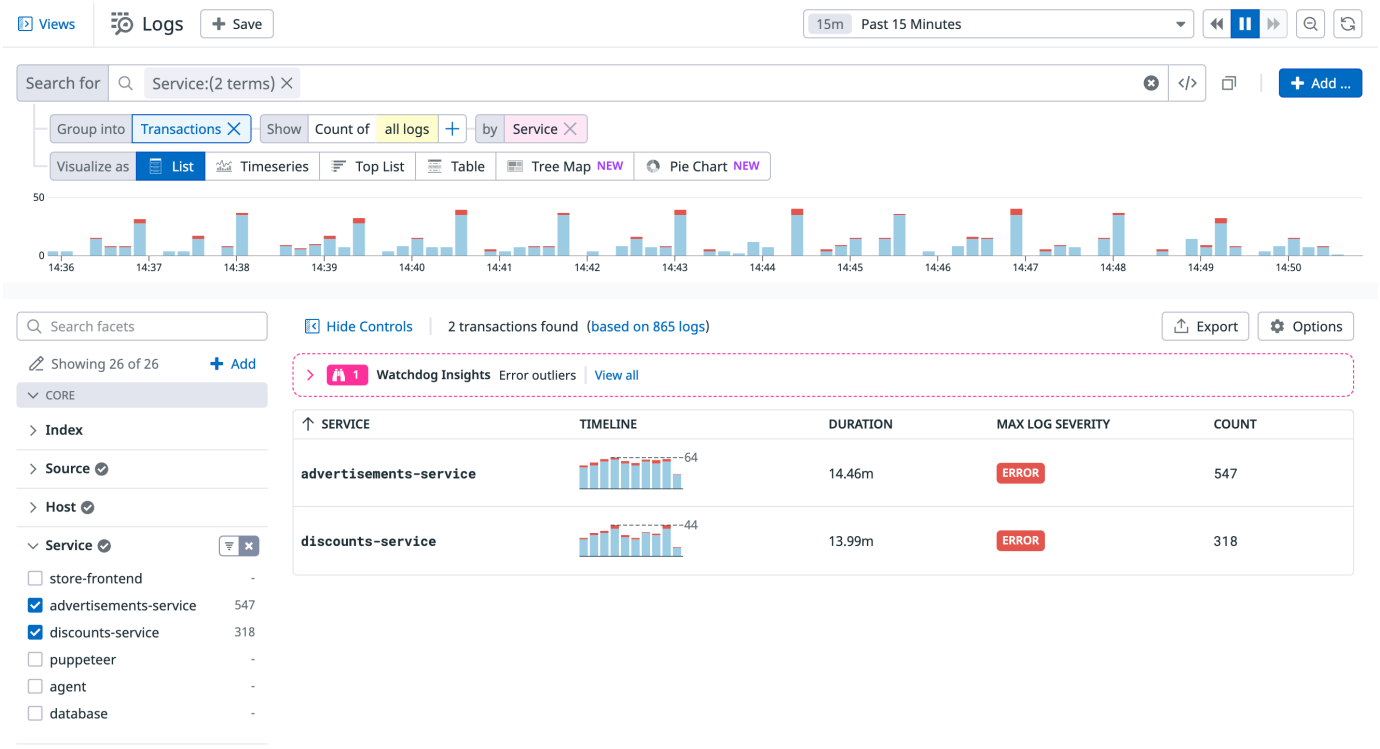
For each transaction event, the tag/attribute value and corresponding transaction duration, max severity, and event count are displayed.

With transaction queries, you can visualize complex interconnected systems through log events, identify transaction bottlenecks by comparing durations, event counts, and custom measures, and reduce mean time to detect (MTTD) by isolating transactions with errors or high latencies.

Transactions aggregate indexed logs according to instances of a sequence of events, such as a user session or a request processed across multiple micro-services.

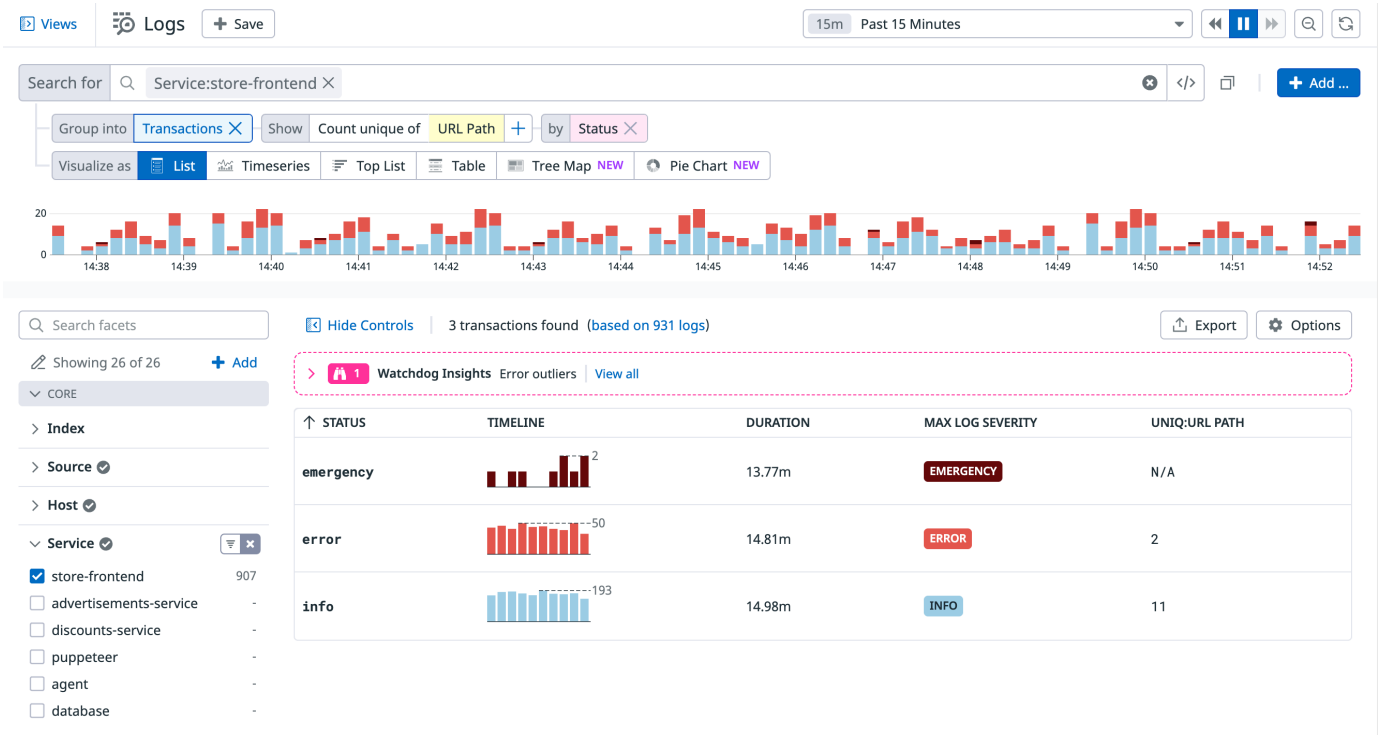
1. Navigate to **Logs > Search**.
2. Open the Views panel and select the saved view you created, **Ads and Discounts Errors**.
3. Remove the `ERROR` status from the search field.
4. For **Group into**, select `Transactions` and group the fields to show a count of `all logs` by `service`:

   The Transactions list will display each service along with the the total count of logs, the total duration of the transactions, and the maximum severity of the logs. This gives you the ability to see at-a-glance which services are performing better when it comes to communicating with other services and which ones you should pay more attention to.



5. Clear the search field above the Log List and filter by `service:store-frontend`.
6. For **Group into**, select `Transactions` and group the fields to show a count of `URL path` by `status`:

   The Transactions list will display each status along with the total count of logs:



7. Click the X next to the **Group into** field to return to the Log List.

Lab Conclusion

Congratulations! You completed the Log Querying and Analytics lab. You now know how to search logs, aggregate them by fields, patterns, and transactions.

For more information on Datadog Log Management, head over to our Introduction to Log Management Course on the Learning Center.

When you're done, enter the following command in the terminal:

```
finish
```

Click the **Check** button in the lower right corner of the lab and wait for the lab to close down before moving on to the next lesson.