Metrics & Monitors

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.

Explore, Query, and Graph

Metrics provide an overall picture of your system. You can use them to assess the health of your environment at a glance—how quickly users are loading your website or the average memory consumption of your servers, for instance. Once you identify a problem, you can use Logs and APM to further troubleshoot.

What Data Can You Collect?

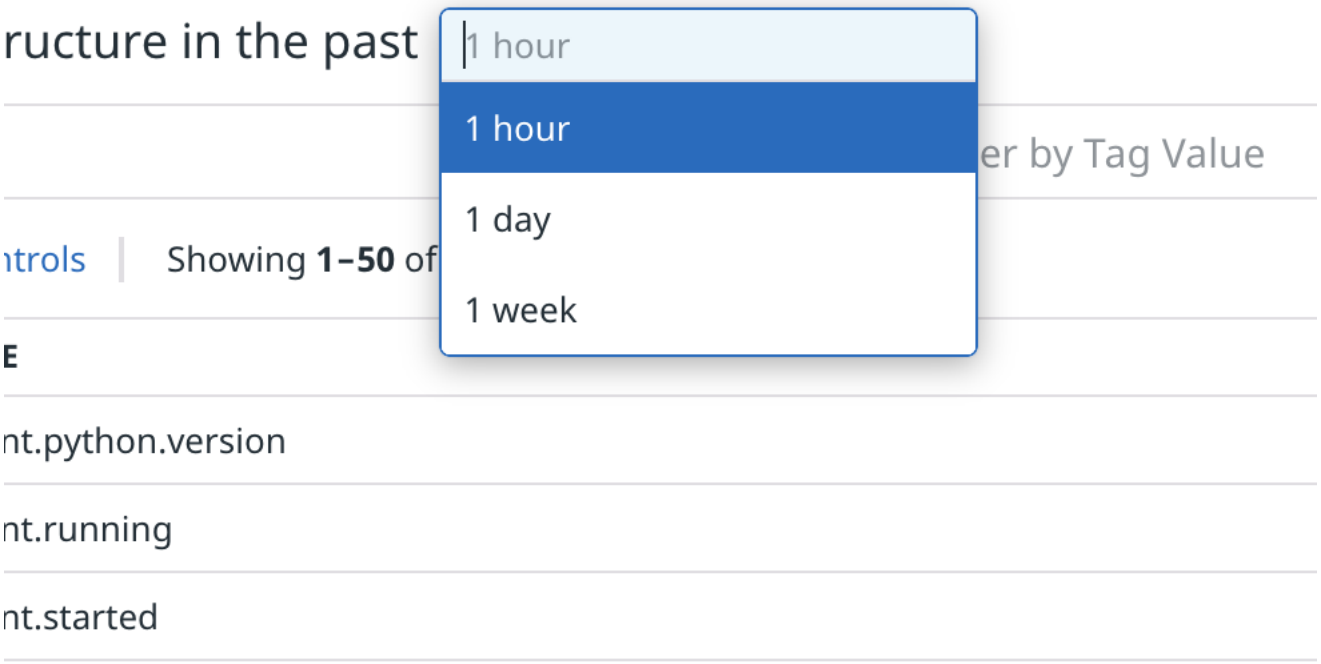
You can see all of the metrics that integrations provide by looking at their **Data Collected** tabs, or in the integration's documentation. For example,

- Look at the Postgres metrics provided in the **Data Collected** tab of the Postgres integration.
- Look at the Docker metrics provided by the Docker Integration in the Docker integration docs.

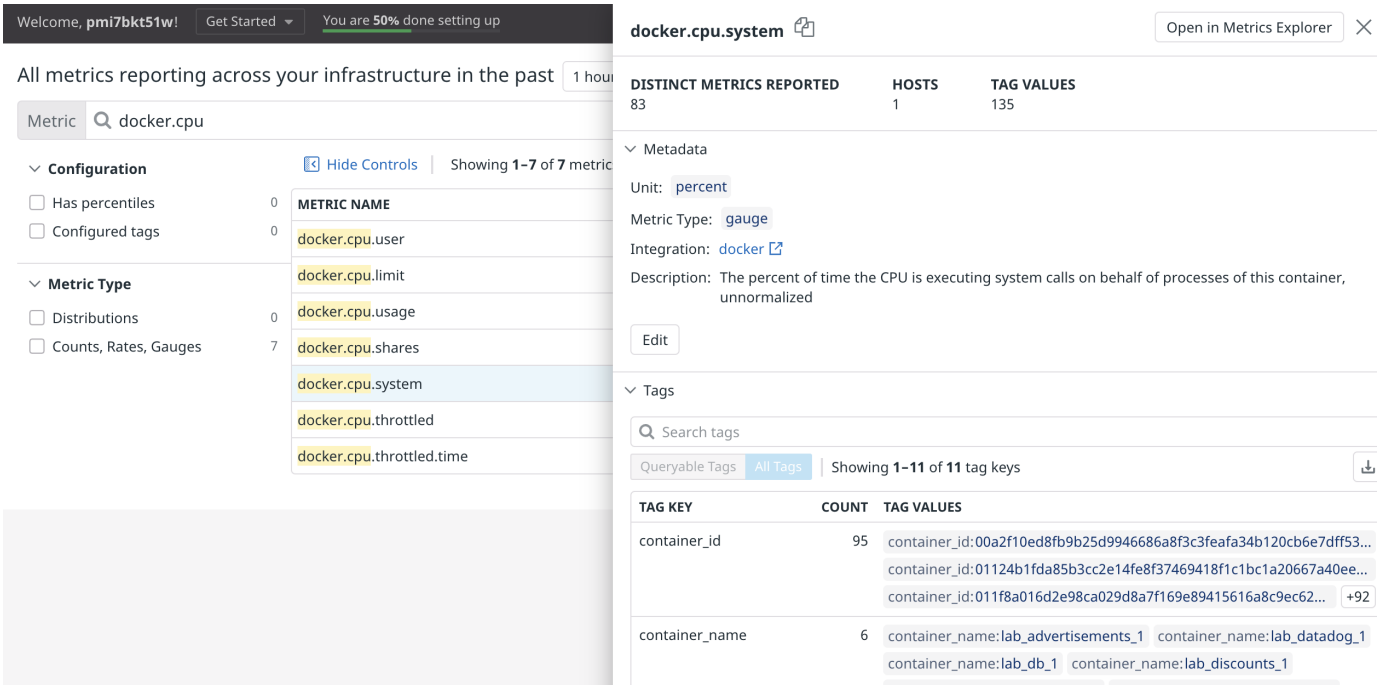
What Data Are You Collecting?

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. To see what metrics have been collected from your infrastructure over a period of time, navigate to **Metrics** > **Summary**.

   Here you can see what Datadog has collected over the past hour, day, or week:



3. Start typing `docker.cpu` in the search bar to filter the list.
4. Click on `docker.cpu.system` to see the details side panel:



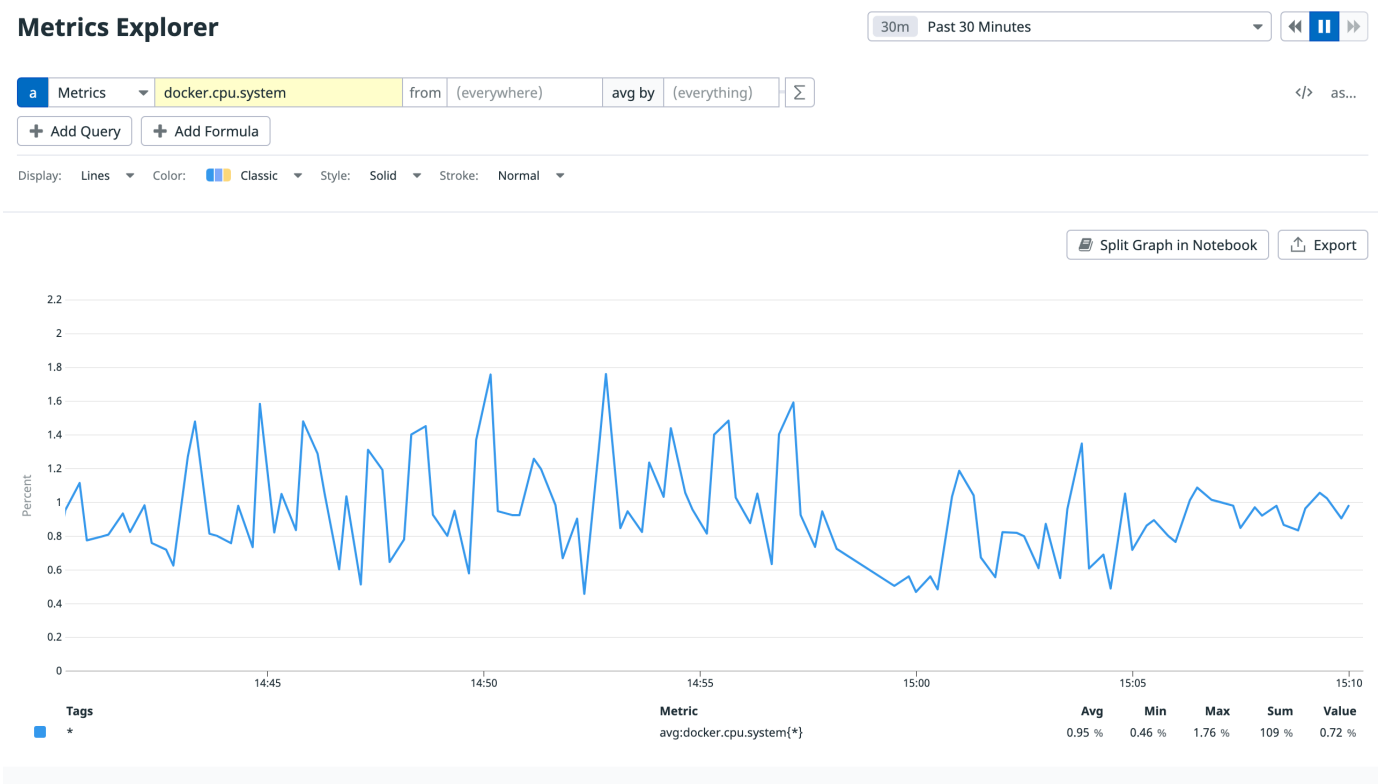
Here you see the distinct metrics reported, from which hosts, and all of the tags values that have been associated with those metrics.

Graph a Metric

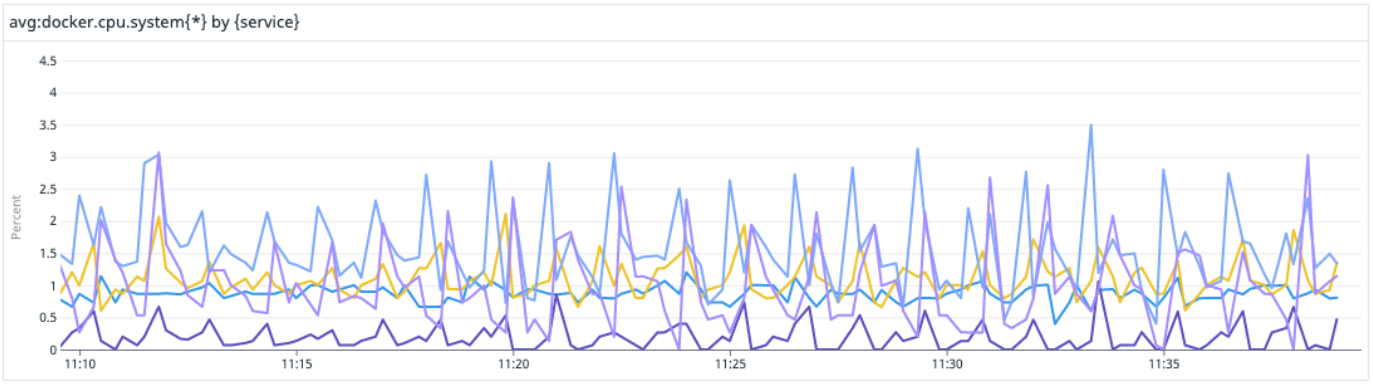
1. In the Metric details side panel, click on the **Open in Metrics Explorer** button in the upper-right corner.

   This populates the **Graph** field with the selected metric, and graphs the metric value over the timeframe selected at the top of the page.

2. Change the timeframe to `Past 30 Minutes`:



3. Filter this down to a single service. In the **from** field, enter `service:discounts-service`.
4. Select the + **Add Query** button to add a new query and this time set the **from** field to `service:advertisements-service`.
5. Remove the extra queries by clicking the X icon to the right of the query.
6. You can break out values by a tag using the **avg by** field. Remove anything from the **from** field and add `service` to the **avg by** field. This will graph each service with its own line. The resulting graph should look like the following image:

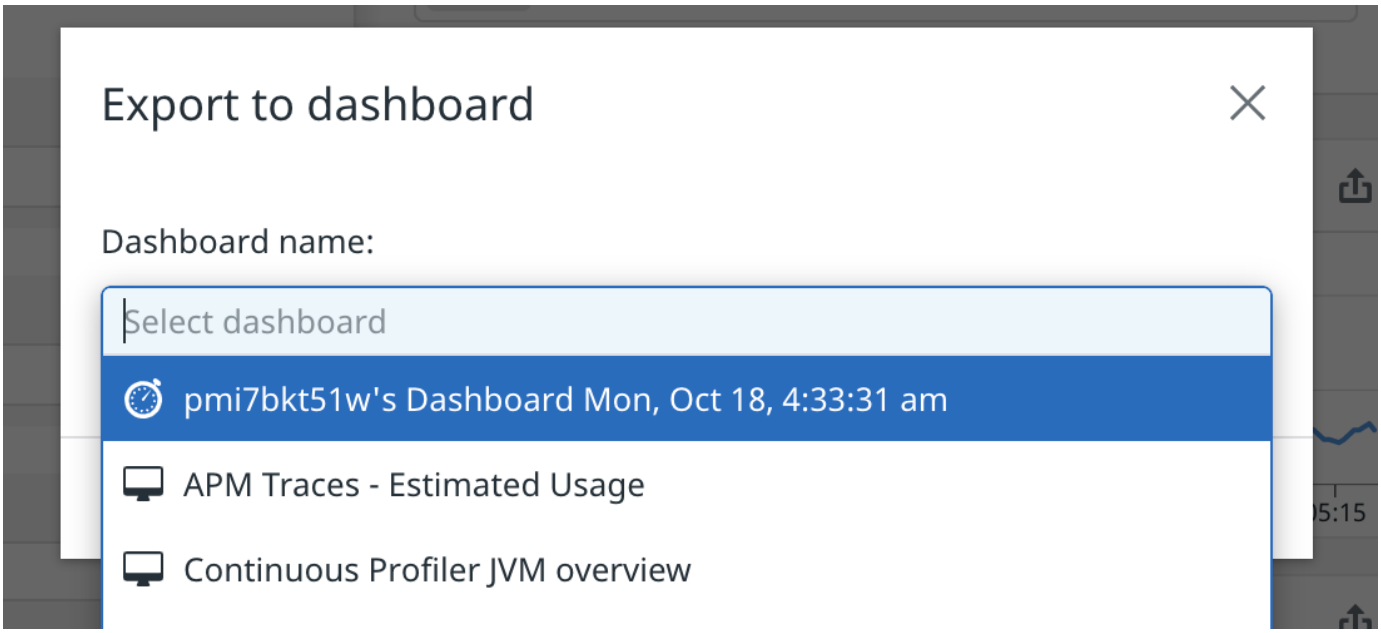


Export to a Dashboard

1. Select the **Export** button above the graph.

   In the resulting dropdown, select the **Export to Dashboard** option.

2. In the popup modal, select the dashboard you created earlier, titled by your user name and a timestamp:



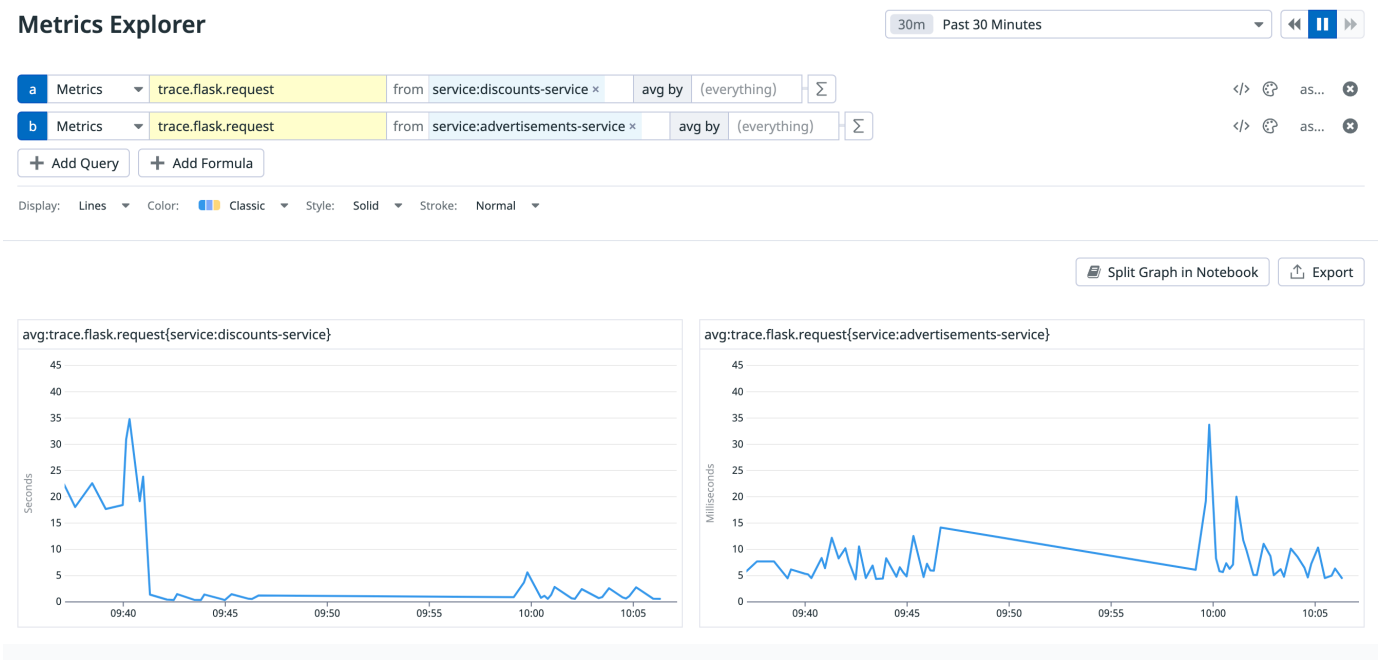
3. Click the **Export** button.

   Note: If you did not create a dashboard previously, you can export to a **New Dashboard**. In the resulting dialog, click the **Create** button.

4. Navigate to **Dashboards** > **Dashboard List** and click on your dashboard to see the graph.

Graph Another Metric

1. Return to the Metrics Explorer page and perform the same steps for the `trace.flask.request` metric from `service:discounts-service` and `service:advertisments-service`:



2. Export these to the same dashboard. If you view your dashboard, you should see the new graph.

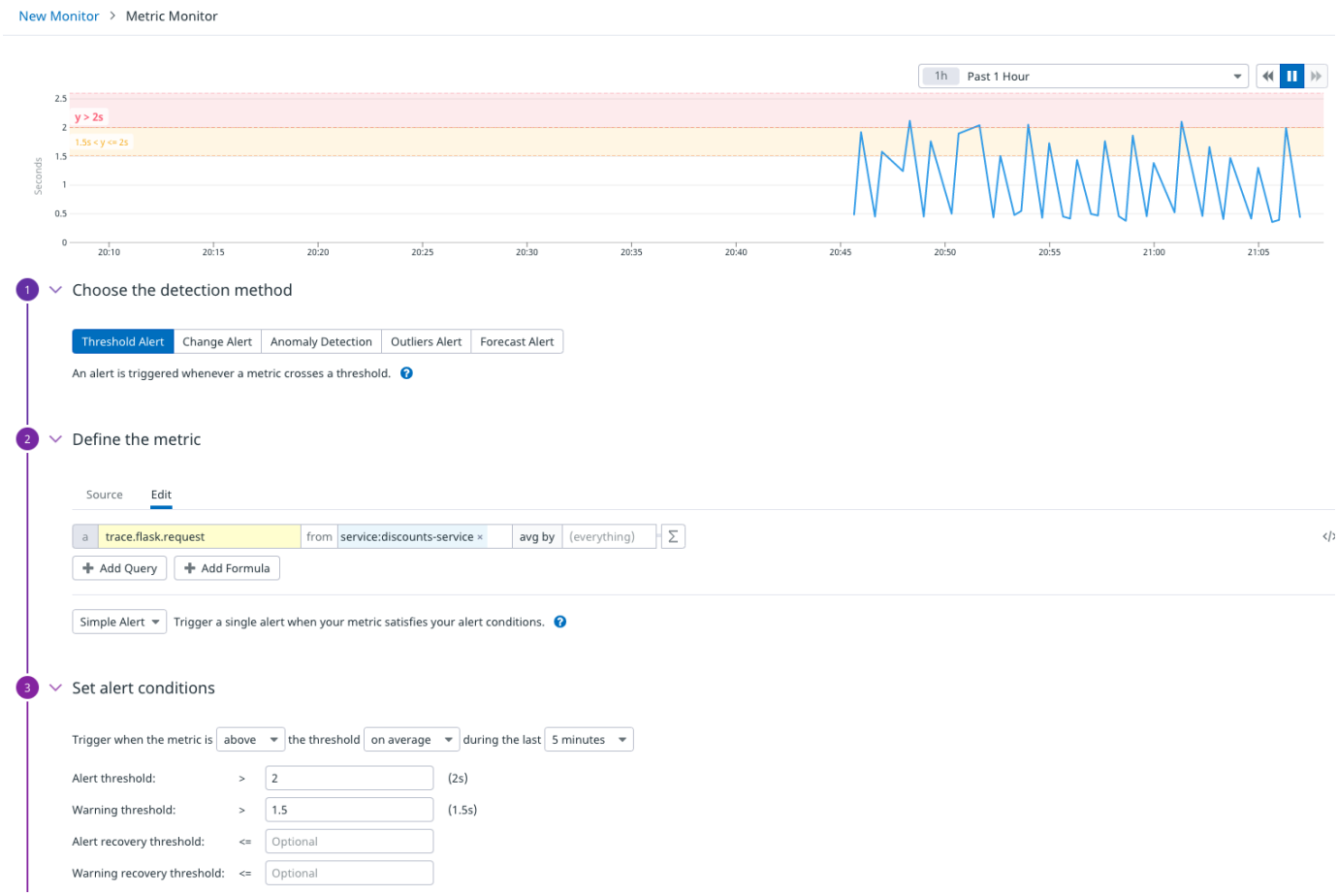
Monitor and Alert

You don't want to watch these graphs all day to see if something spikes into dangerous territory. Let a monitor do that for you.

Create a monitor that will alert you when the Flask request time for the discounts service exceeds 2 seconds.

1. Navigate to Monitors > New Monitor. Note the many types of types of monitors Datadog provides. You can hover over each type to view its summary.
2. Select **Metric**.
3. For **Choose the detection method**, keep the default value of **Threshold Alert**.
4. For **Define the metric**, set **Metric** to `trace.flask.request` and **From** to `service:discounts-service`.
5. For **Set alert conditions**, set **Alert threshold** to `2`, and **Warning threshold** to `1.5`.

   Your new Metric Monitor configuration should look like this so far:



6. Under **Notify your team**, name the monitor:

   ```
   Discounts service request time
   ```

7. Paste the following message in to the message text area:

   ```
   Investigate the source. Usually good info in [APM](https://app.datadoghq.com/apm/service/discounts-servic

   Contact @incident@example.com after triage.
   ```

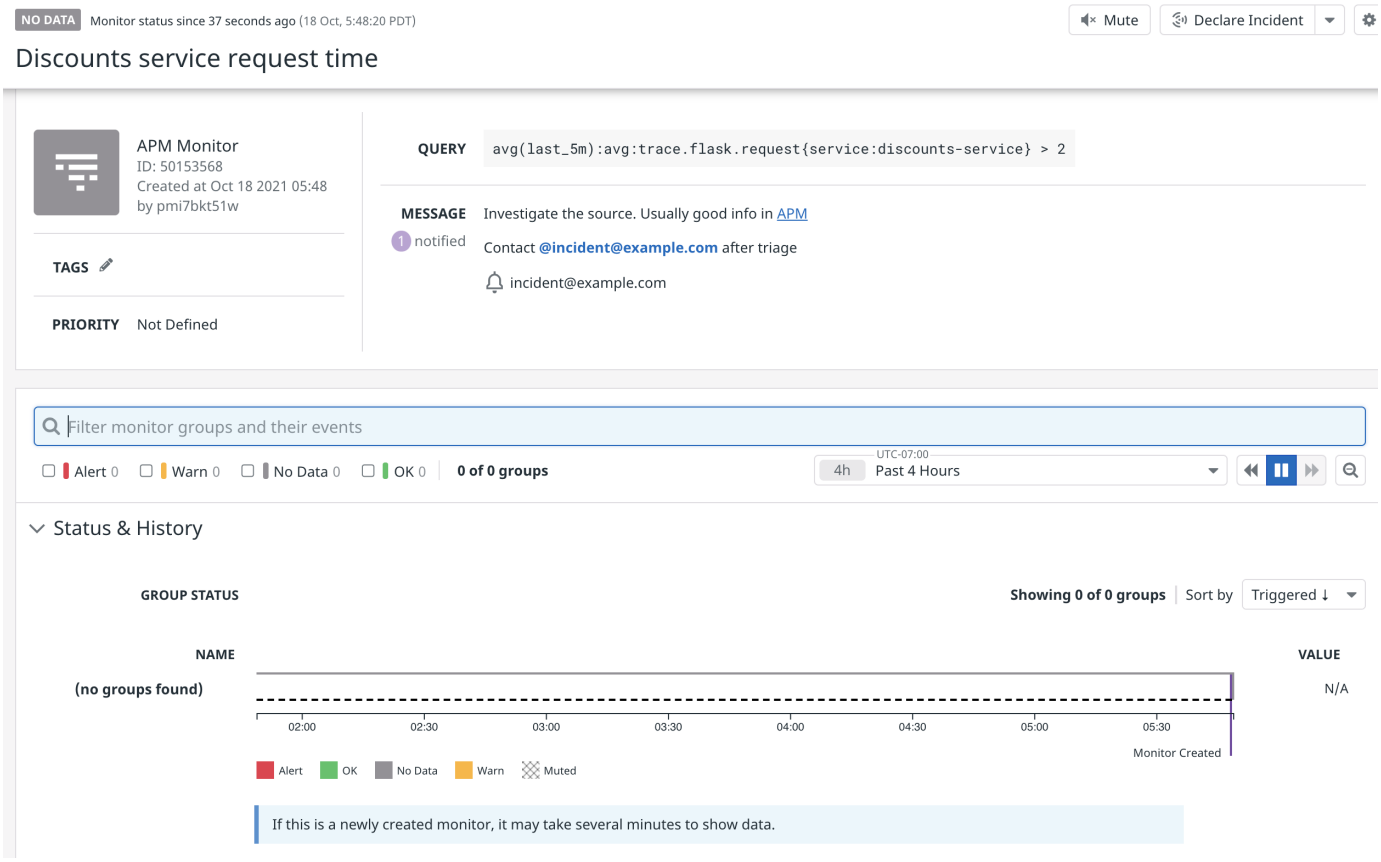
   This message contains a helpful link to the APM page for the discounts service, as well as the email address to the escalation staff member. The `@` preceding email addresses will render as a link to that user's Datadog Profile if they are part of your organization. (In this lab, the referenced user does not exist.)

   You can learn more about the special syntax features of the notification text area in the Notifications docs. These docs also cover many powerful Notifications integrations such as Slack, PagerDuty, and more.

8. Keep the rest of the default settings.

9. Click **Create**.

   Your monitor should look similar to this:



It will be a few minutes before points appear in the **EVALUATION GRAPH** in the **Status & History** section.

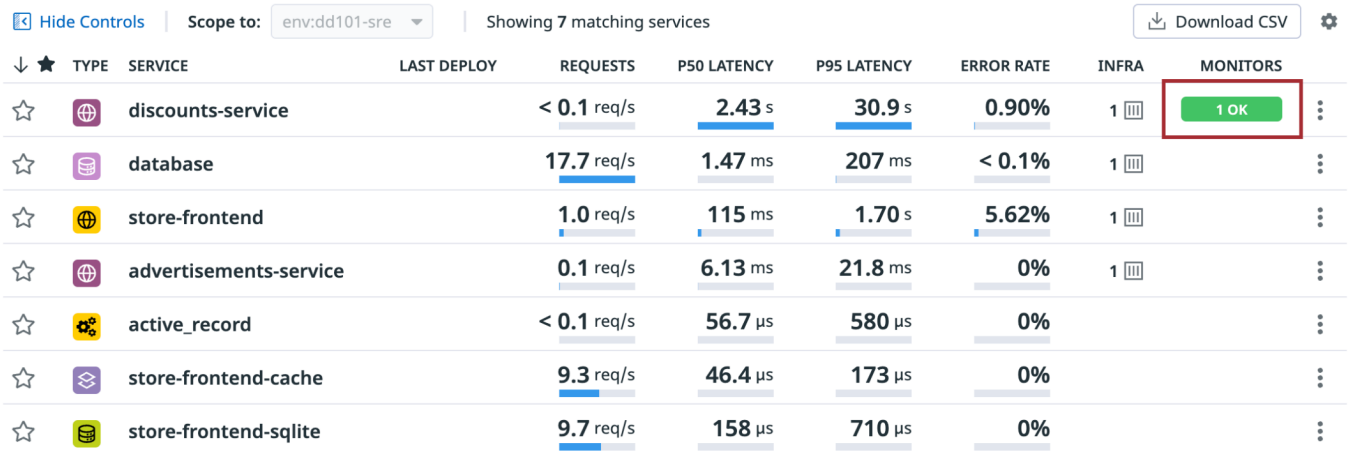
10. Scroll to the bottom of the monitor page and examine the **Events** section. The **Events** section displays all events associated with this monitor, including a summary of its creation and updates.

    Note: **Watchdog** is an algorithmic feature for APM performance and infrastructure metrics that automatically detects potential application and infrastructure issues. It can take days to weeks for Watchdog to collect enough data to detect anomalies, so you won't see anything in this tab during the short duration of this lab. You can learn more about Watchdog in the docs.

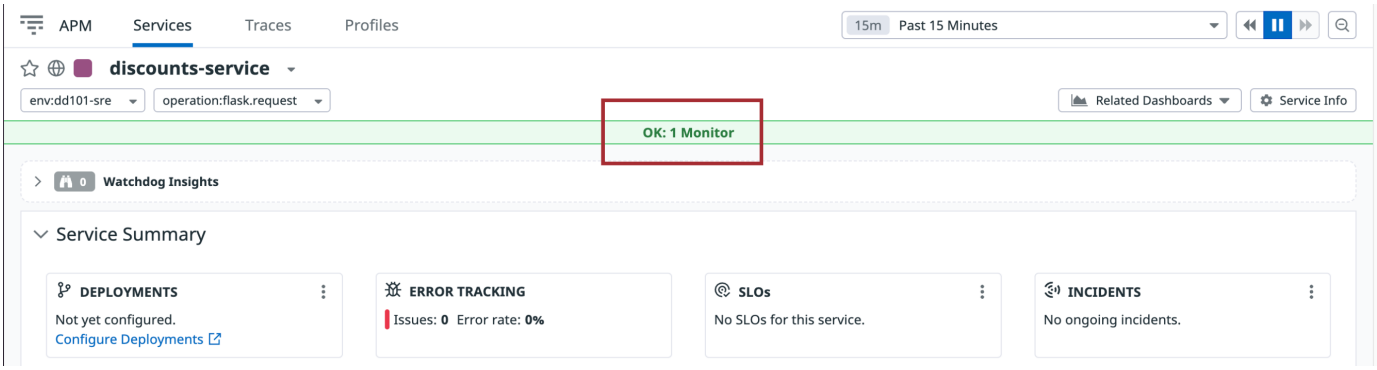
Monitors and APM

Datadog will automatically display an indication of a service's monitor in several APM pages. These indicators link to the associated monitor's status page:

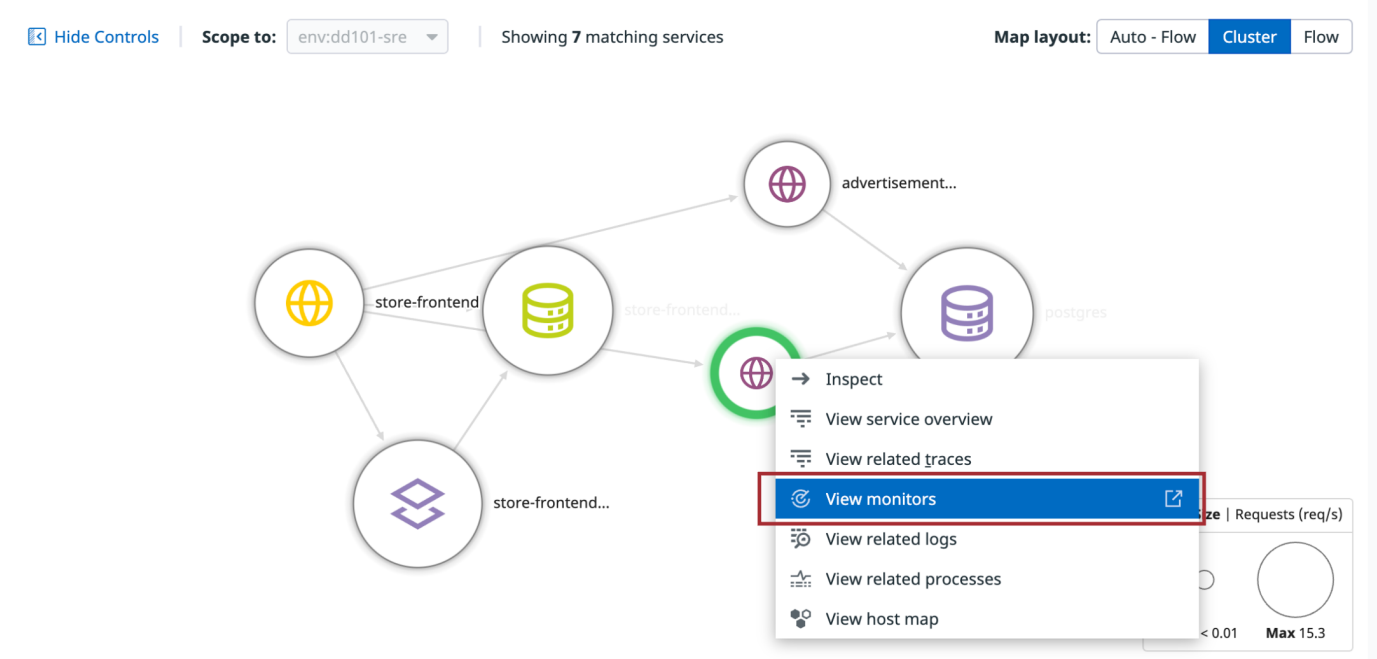
- Under the **MONITORS** column of the Service Catalog page:



- At the top of a Service details page:



- In the Service Map view:



Surfacing monitor status throughout APM pages makes it very easy to check on the status and history of the key metrics that are important to you.

Service Level Objectives (SLOs)

SLOs provide a framework for defining clear targets around application performance. They measure a target percentage of a metric over time to track how well a service is meeting its obligation to its consumers.

An example of an SLO might be, "The discounts service should be running 99% of the time for any 7-day period."

Now that you have a monitor on the discounts service request time, you can use that as the Service Level Indicator (SLI) for a new SLO.

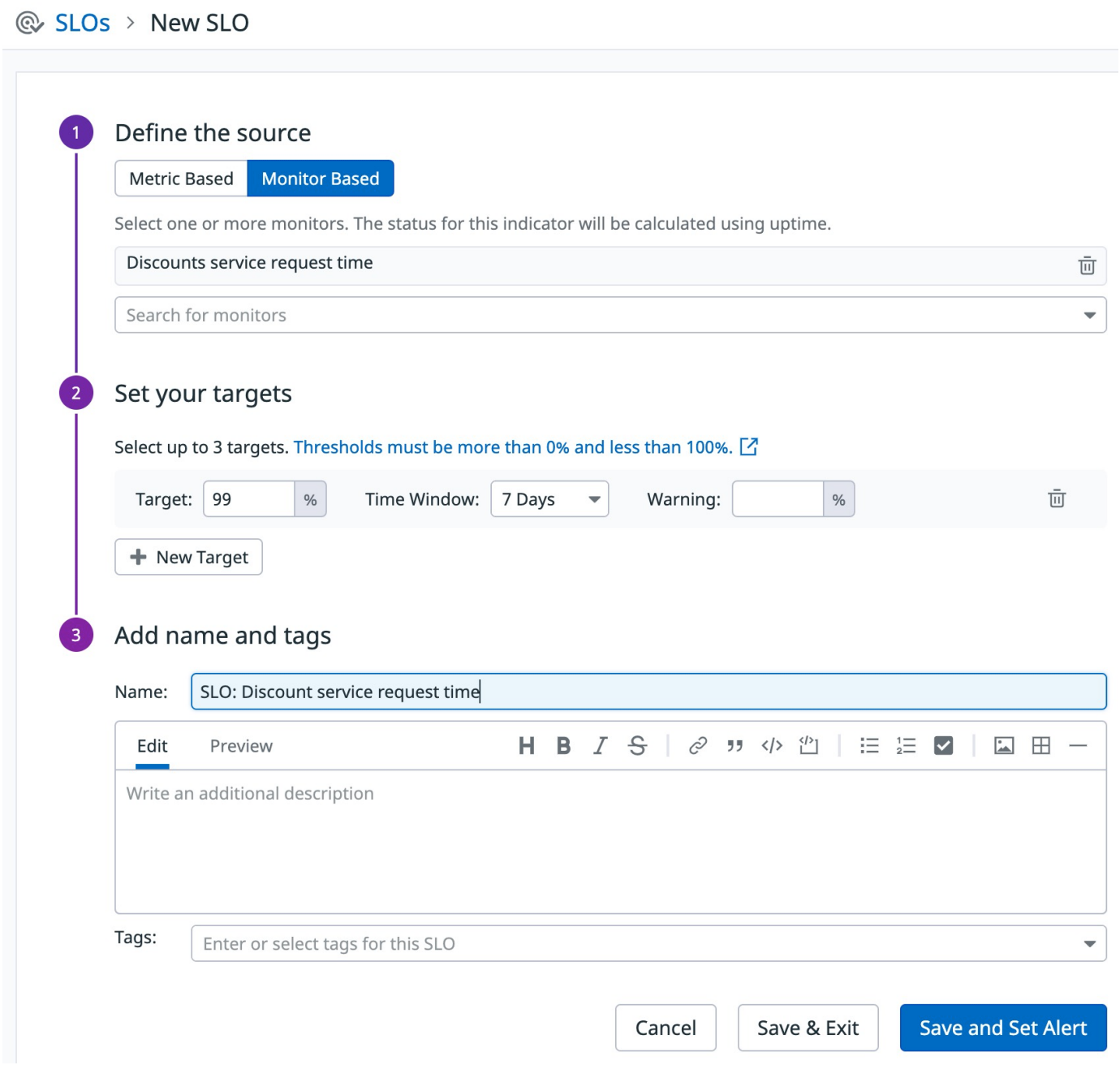
Create this SLO for the discounts service:

1. Navigate to Monitors > New SLO.
2. Under **Define the source**, select `Monitor Based`.
3. From the dropdown, select the `Discounts service request time` monitor.
4. Under **Set your targets**, click the **New Target** button.

   Leave the default values of 99% over 7 days.
5. Under **Add name and tags**, set **Name** to the following:

   `SLO: Discount service request time`

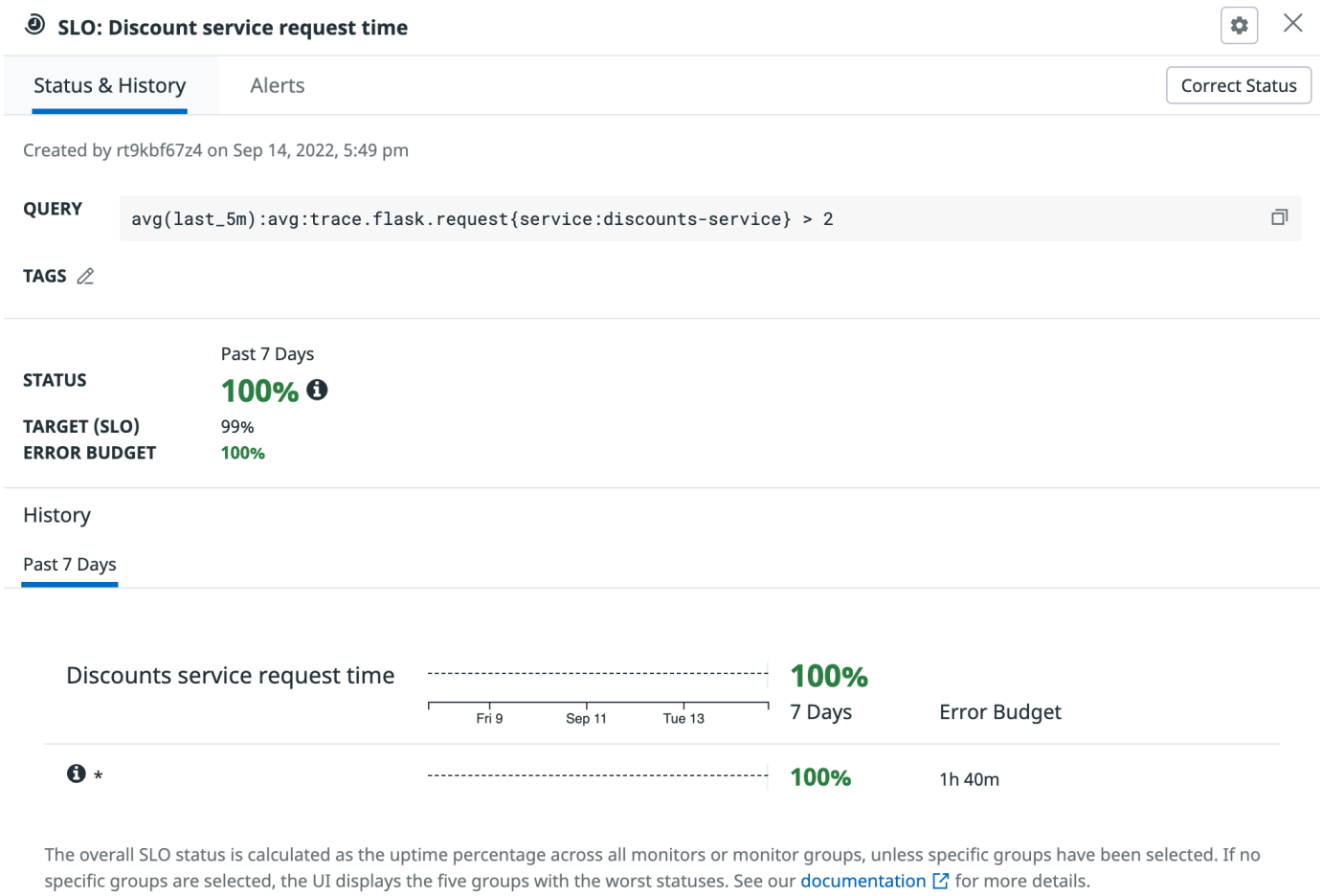
The New SLO page should look like this:



6. Click **Save and Set Alert**.
7. Notice under **Select SLO**, your SLO is selected.
8. Keep the default values for **Set alert conditions**.
9. Under **Notify Your Team**, name the monitor:

   ```
   SLO: Discounts service request time
   ```
10. Paste the following message in to the message text area:

    ```
    Discounts service request time budget depleted
    ```
11. Keep the rest of the default settings.
12. Click **Create**.
13. You should see the SLO monitor page. Click the **SLO: Discount service request time** link to open the SLO side panel in a new browser window.
14. Click the **Status & History** tab to see the SLO in action:



Because this course is coming to a close, the SLO will remain at 100% for the remainder of the course. Nice work!

To learn more about how SLOs can help you prioritize your development time, read the SLO docs.

Lab Conclusion

In this lab, you learned what metrics are available to you, how to graph them, and how to export those graphs to a monitors and dashboards. With monitors, you can set up alerts for for key metrics. With dashboards, you can display related data and graphs in one place for easy monitoring and sharing.

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.