


1h Oct 26, 11:01 pm – Oct 27, 12:01 am

[Open Notebook in Datadog](#) 

Azure Incident Response

Created by **Ryan MacLean** on Oct 24

Azure Incident Response with Datadog

Workshop Checklist

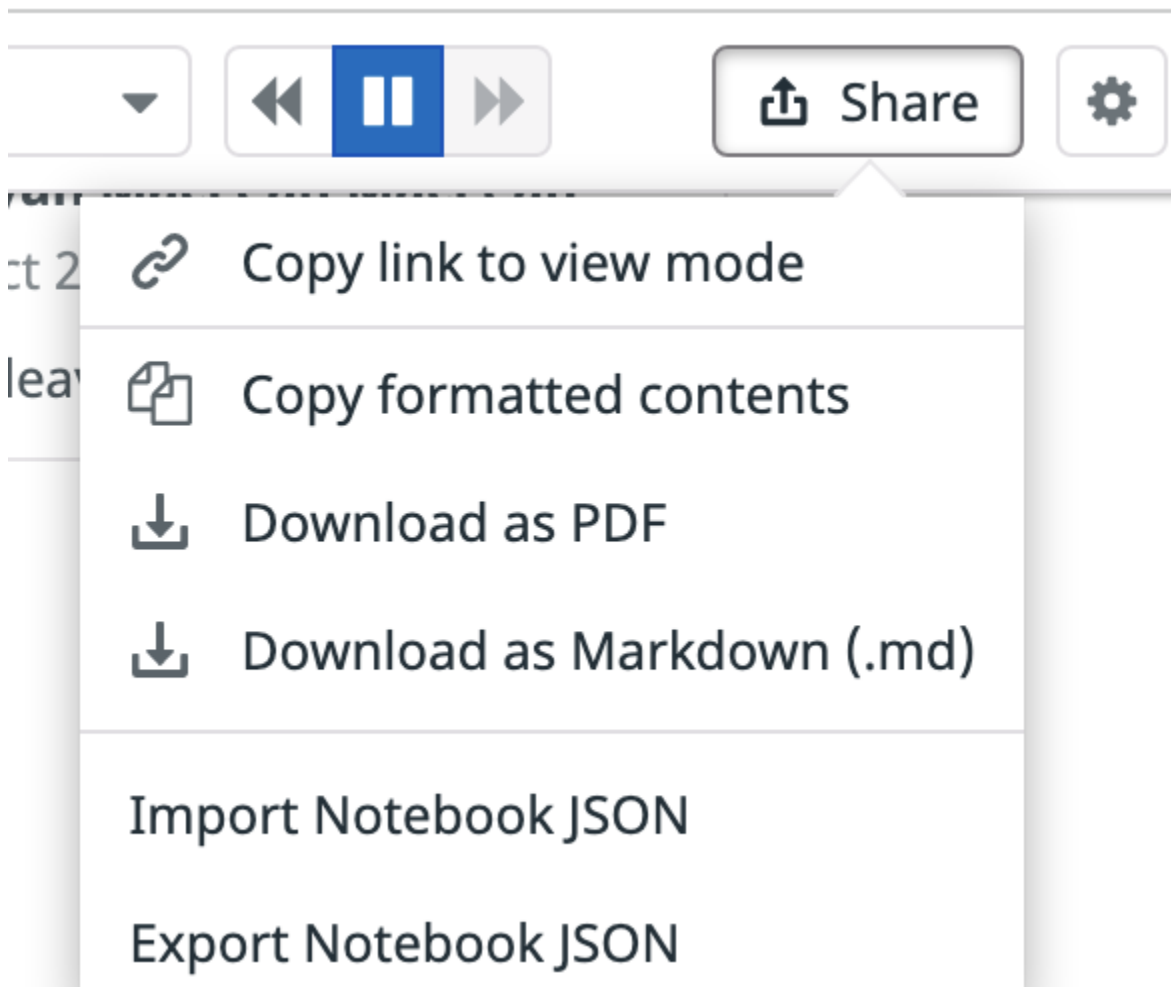
- ☒ 1. Intro
- ☐ 2. Synthetic Testing
- ☐ 3. Monitors
- ☐ 4. Creating an Incident
- ☐ 5. Updating an Incident
- ☐ 6. Resolving an Incident
- ☐ 7. Blameless Postmortem
- ☐ 8. Links and Docs
- ☐ 9. Markdown Fun!

1. Intro

In this *quick* section we'll set up a test, assign a metric as a monitor, then go through an incident.

If you're reading this as a readme on GitHub, you can download the notebook's [JSON file from](#) `json_import.json` file in the repository, and import it.

1. First [create a new notebook in Datadog](#).
2. After creating a new notebook, import it from the top-right share icon:



Note: if ever you need to export a notebook, from the same menu you can download as PDF or markdown (.md) or export the JSON file.

2. Synthetic Testing

In order to test the site we created earlier in the lab, we'll set up a synthetic monitor.

You can do that by following this [link to create a multi-step synthetic test](#).

1 Name and tag your test

Name
Give your test a name

Environment (env)
You don't have any Environment (env) tags.
Environment tags can help you organize your tests and your services.
[+ Create New Env Tag](#)

Additional Tags
Select or add related tags

2 Select locations

☒ All Locations (16)

☒ Managed Locations (16)

Americas (6)

- ☒ Canada Central (AWS)
- ☒ N. California (AWS)
- ☒ Ohio (AWS)
- ☒ Oregon (AWS)
- ☒ São Paulo (AWS)
- ☒ Virginia (Azure) **NEW**

Asia Pacific (5)

- ☒ Mumbai (AWS)
- ☒ Seoul (AWS)
- ☒ Singapore (AWS)
- ☒ Sydney (AWS)
- ☒ Tokyo (AWS)

Europe (5)

- ☒ Frankfurt (AWS)
- ☒ Ireland (AWS)
- ☒ London (AWS)
- ☒ Paris (AWS)
- ☒ Stockholm (AWS)

3 Define steps

Link multiple steps by [creating variables](#) from the request response data.
When using variables, remember that step order matters.

[Create Your First Step](#)

Create variables
Create variables to use as part of your test.
[+ Create Local Variable](#)

Documentation Variables 0

Get Started with Synthetic Monitoring
Test all layers of your applications from locations around the world to assess availability and uptime.

Documentation

Get started with Multistep API tests
Need help creating your first Multistep API test? Check out our dedicated getting started guide and launch your first Datadog Multistep API test.

Synthetic Monitoring guides
Learn how to authenticate to your endpoints, track SLOs, and troubleshoot issues without switching context using Synthetic Monitoring guides.

Blog Post

Introducing Datadog Synthetic Monitoring
An overview introducing key concepts and features to help you get started with Datadog Synthetic Monitoring.

Advanced Features

Setting up Private Locations
Set up private locations to run multistep API tests on your internal applications and from any custom locations.

Leveraging CI/CD integration
Use multistep API tests for canary releases to increase the reliability of your deployments.

In a different tab, we'll need to retrieve the URL for our app service from the [Azure App Services](#) page.

Microsoft Azure Search resources, services, and docs (G+)

Home > **to-do** App Service

Search (Cmd+/) << Browse Stop Swap Restart Delete Refresh Get publish profile Reset publish profile Share to mobile Send us your feedback

Overview

Essentials

Resource group (change) : rmac-demo

Status : Running

Location : East US

Subscription (change) :

Subscription ID :

Tags (change) : app : to-do service : to-do env : dev version : 0.1 environment : dev

URL : <https://to-do.azurewebsites.net>

Health Check : Not Configured

App Service Plan : to-do-app4 (S1: 2)

Git/Deployment username :

Git clone url :

FTP hostname :

[JSON View](#)

After running a few tests - you can export the metrics or graphs to a dashboard (new or existing) or even add one to a notebook, as seen below!

ViewsMonitors+ SaveNew Monitor

Manage MonitorsTriggered MonitorsManage Downtimes

Filter monitors

Status

☐ Triggered 1
☐ Alert 1
☐ Warn 0
☐ No Data 0
☒ OK 0

Hide ControlsShowing 1–1 of 1 result

Mute

Resolve

Delete

Edit Tags

PRIORITY	STATUS	MUTED ELAPSED	MUTED LEFT	NAME 1	TAGS
	ALERT			[Synthetics] App Service Test	check_type:api-multi check_status:l... +2

ViewsMonitors+ SaveNew Monitor

Manage MonitorsTriggered MonitorsManage Downtimes

Filter monitors

Status

☐ Triggered 1
☐ Alert 1
☐ Warn 0
☐ No Data 0
☒ OK 0

Hide ControlsShowing 1–1 of 1 result

Mute

Resolve

Delete

Edit Tags

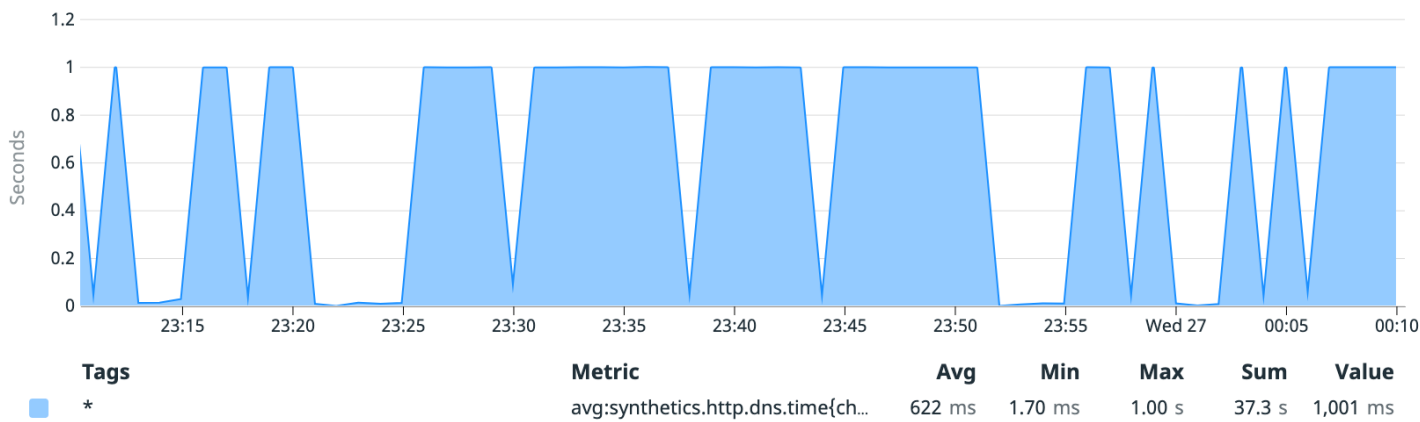
PRIORITY	STATUS	MUTED ELAPSED	MUTED LEFT	NAME 1	TAGS
	ALERT			[Synthetics] App Service Test	check_type:api-multi check_status:l... +2

You can also add to a notebook (as we've been doing in order to create this Notebook!):



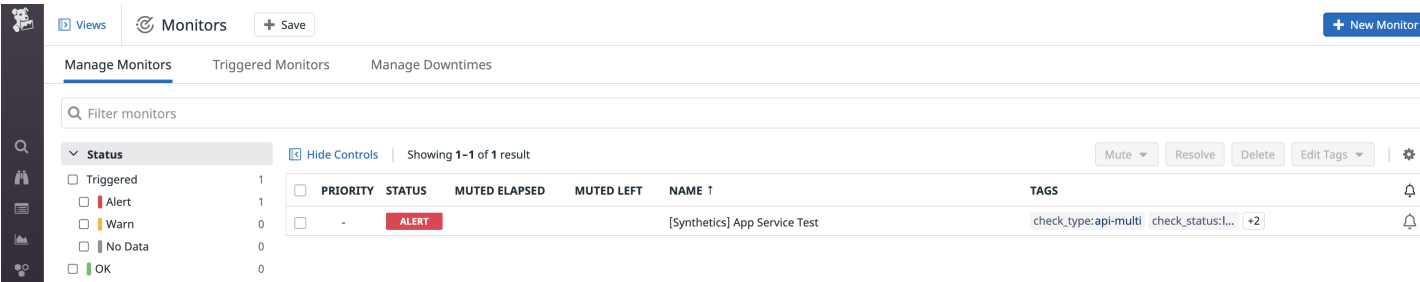
Once you see the Network Timings graph below fill up, please proceed to the next section.

Network timings (averaged)



3. Monitors

Next we'll head over to the [Monitors](#) section of Datadog in order to have a look at the automatically-created monitor from our synthetic test.



It might be red, but don't panic - we're only setting things up in our development environment 🤪

Because we set our alert to `@all`, everyone in our company would have received this alert. That could have been via [Teams](#), email, or other services you've set up in order to receive alerts or notifications.

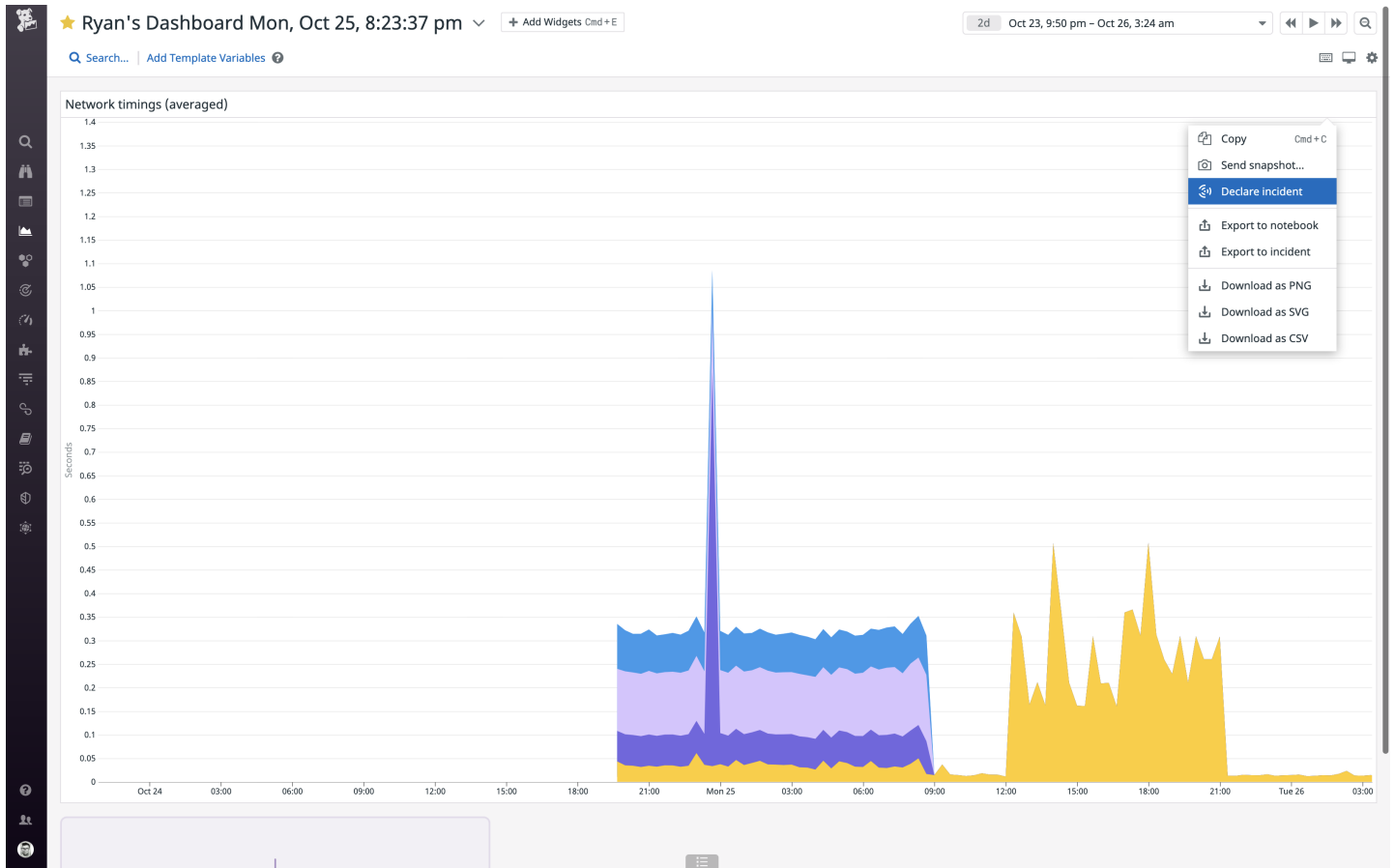
In the following live graph embedded in the Notebook note that we're able to add checkpoints for specific periods of time (the following is Synthetics Response Time by URL for the past 4 hours):

Synthetics Response Time by URL 4h Past 4 Hours

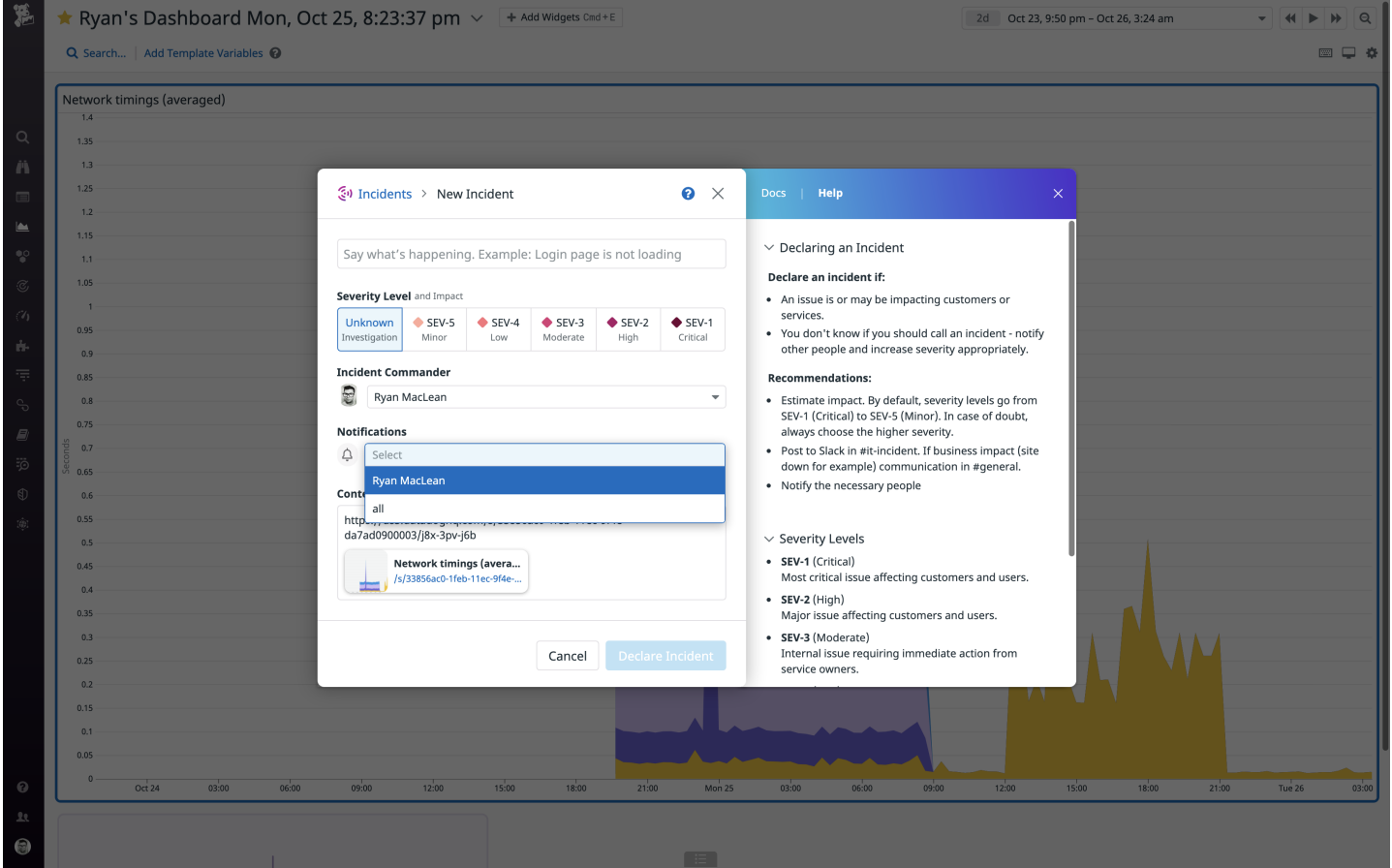
5.96 <https://datadogwebapp-475069.azurewebsites.net/>

4. Creating an Incident

From the dashboard we created by exporting the Synthetics metric, we'll declare an incident.

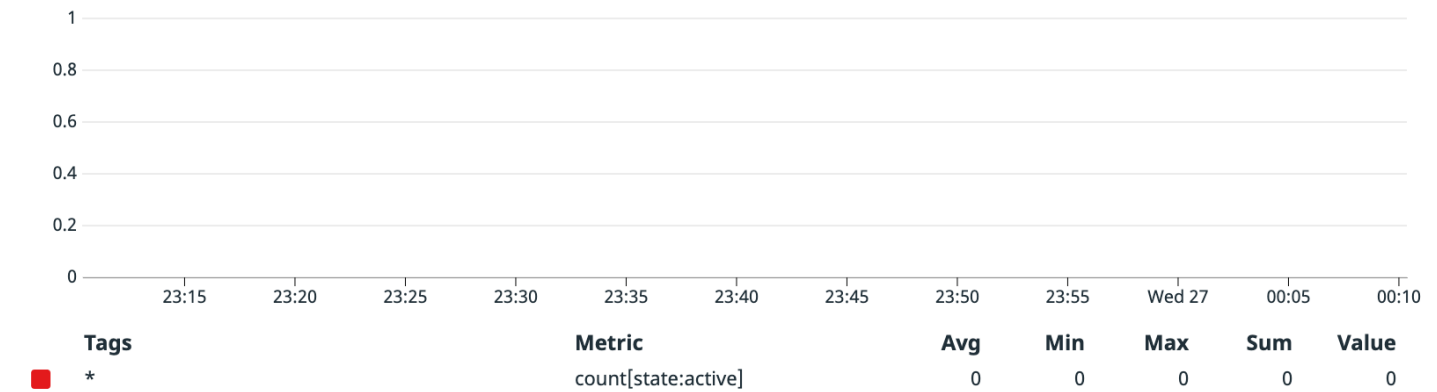


In the incident declaration, you can set a title/summary, the severity level, pick an audience for notifications as well as context and signals (ours will be pre-filled as we created the incident from a graph).



Once the incident has been created, it will appear in the following graph of Active Incidents:

Active Incidents



5. Updating the Incident

Throughout the incident lifecycle, we'll want to update the status in order to keep team team and stakeholders up-to-date on the progress.

Note that you can also link to both live chat as well as video chat - say for example you've set up a new Teams channel programmatically in order to deal with the incident, but also a live Teams video meeting muster point (or "war room"). Both can be added as links so that others can join and get updated with one click from the header on the incident's page. *ADD IMAGE HERE*

Next we'll go over adding an update, as well as sending out a notification from within the Incident Response section of Datadog.

To do so, first add an update from the [Incident Response timeline](#).

Once it has been updated, on the top-right of the Incident Response page, we'll send out a notification: *ADD IMAGE HERE*

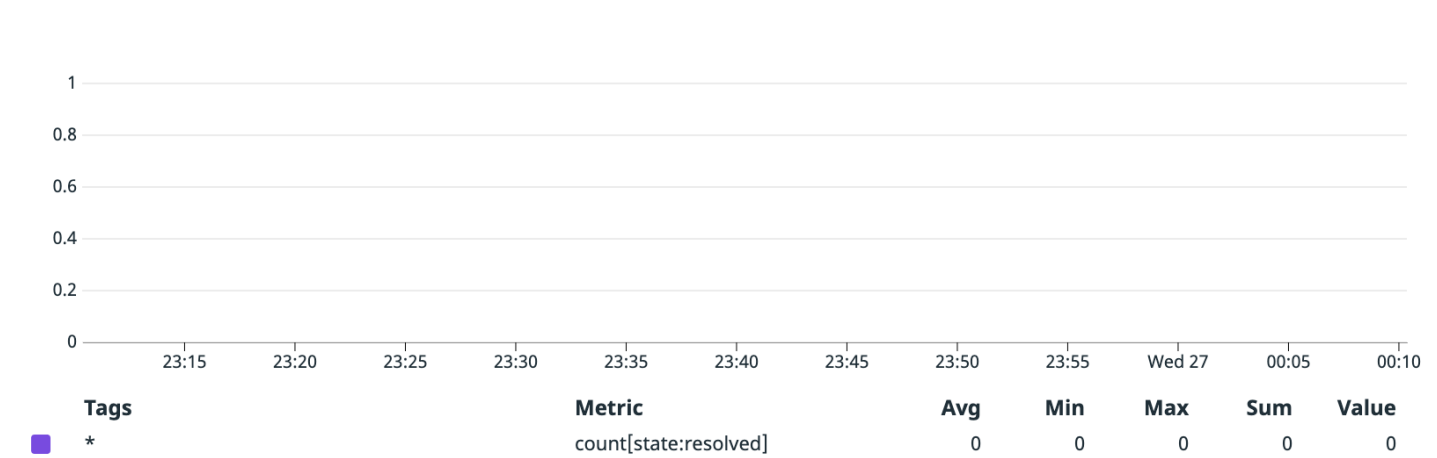
Next we'll add a task to the incident. This is like a to-do list for the team. You can assign tasks to team members as well as add a deadline, if required.

After adding a task to the incident, the [Datadog Events query should show an entry for the Incident update](#).

6. Resolving the Incident

Once we've addressed the causes of the incident, for example via a subsequent deployment, we can then resolve it, via the status on the top-left of the Incident Resolution page.

Resolved Incidents



7. Blameless Postmortem

Once the incident has been resolved, you would normally start the blameless postmortem process.

This means collecting a timeline of events, things that were tried, any dashboards related to the incident, etc. Since we collected these as we went along, when the postmortem is created in Datadog, it will collect all of these for you, and collect it in a notebook. Once the postmortem notebook has been created, you can then export it as markdown, JSON and/or PDF.

[Link to first postmortem](#)

8. Markdown Fun!

While working with Datadog notebooks, sometimes having a cheat-sheet handy can be helpful for those unfamiliar with Markdown, as well as to serve as a quick reminder to you while on-call.

Some Handy Markdown

Title

Sub-title

Sub-sub-title

Note

Emphasis

Use either `_` or `*`:

bold / **bold**

italics / *italics*

Code Snippets

Single line / inline `code`

Multiline

```
for i in {1..100};
do echo "hi from Datadog!";
done
```

Links

[link](#)

Tables

AZURE SERVICE	MONITOR
App Service	Throughput
VM	Uptime

Checklists

- ☒ Checklist Item
- ☐ Unchecked

Bullets

- Bullet
- List

Numbered Lists

1. This is the first item
2. This is the second!