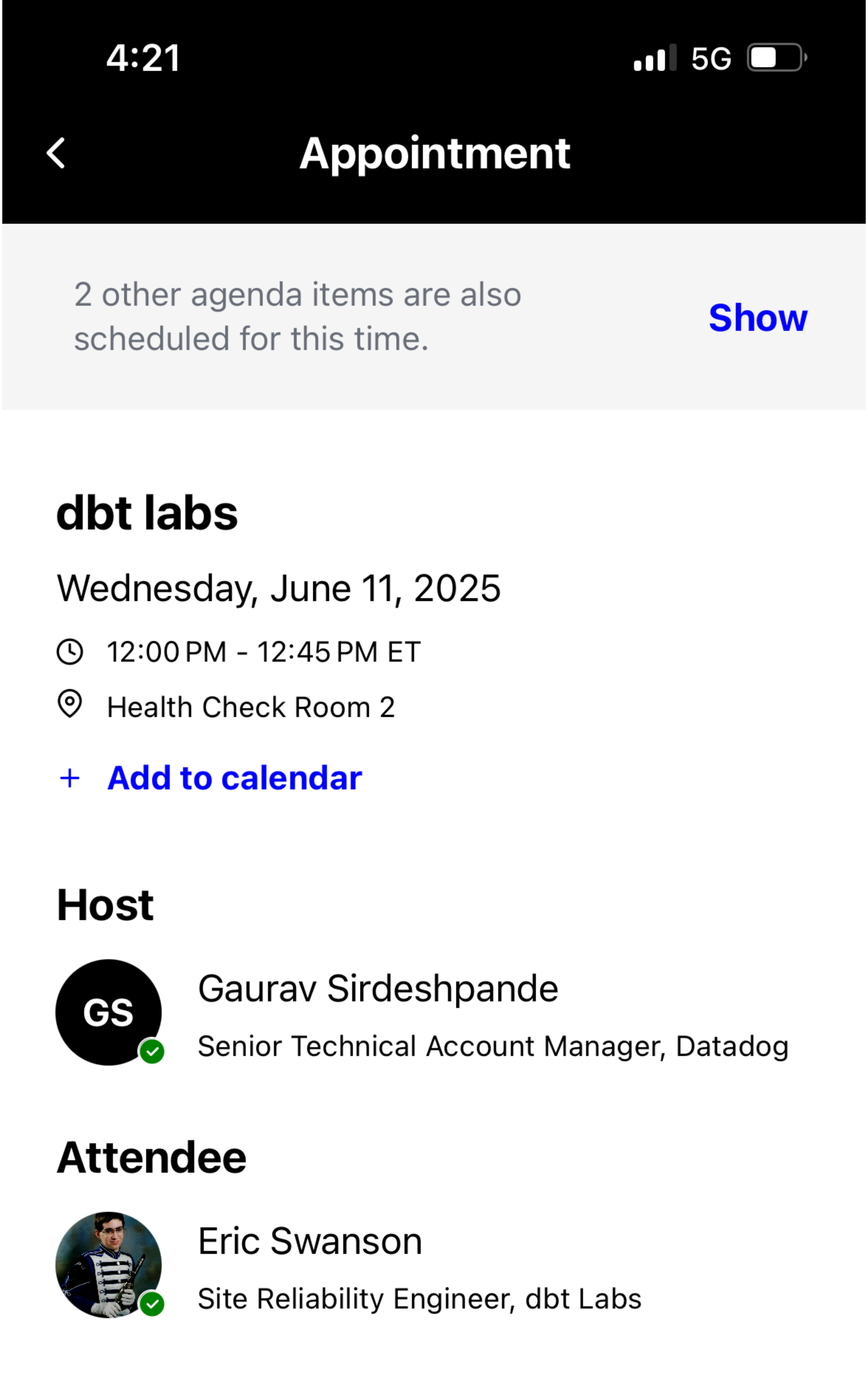
# [TAM] DASH 2025 Health Check

Datadog Support

## Welcome to your DASH Health Check!

This notebook is prepared by a Datadog Technical Account Manager (TAM) to review your current implementations and best practices.



## Appointment

2 other agenda items are also scheduled for this time.

Show

## dbt labs

Wednesday, June 11, 2025

12:00 PM - 12:45 PM ET

Health Check Room 2

+ Add to calendar

## Host

GS

Gaurav Sirdeshpande

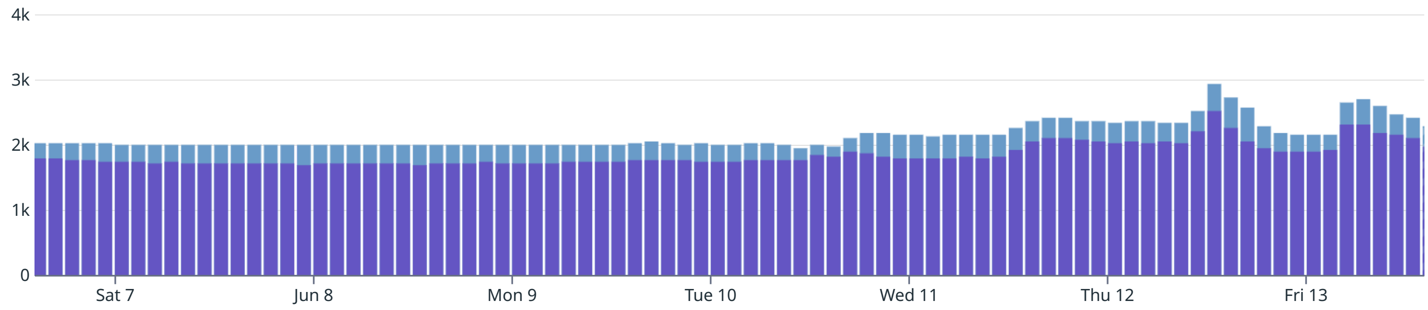
Senior Technical Account Manager, Datadog

## Attendee

Eric Swanson

Site Reliability Engineer, dbt Labs

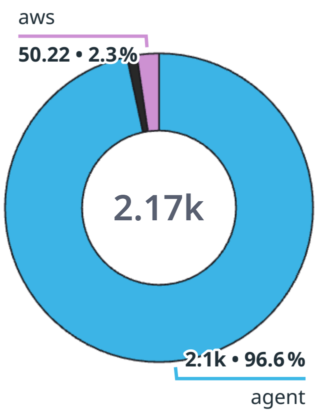
### Infra and APM Hosts



| | TAGS | METRIC | AVG | MIN | MAX | SUM | VALUE |
|---|---|---|---|---|---|---|---|
| 🟦 | * | Infra | 2.16k | 1.96k | 2.94k | 181k | 2.03k |
| 🟪 | * | APM | 1.87k | 1.72k | 2.52k | 157k | 1.79k |

[redacted]

### Infra Hosts by Type



| | HOST_TYPE | COUNT | ↓ SHARE |
|---|---|---|---|
| 🟦 | agent | 2,096.39 | 96.65 % |
| 🟪 | aws | 50.22 | 2.32 % |
| 🟨 | azure | 10.27 | 0.47 % |
| 🟩 | opentelemetry | 7.06 | 0.33 % |
| 🟥 | gcp | 5.18 | 0.24 % |

**Note:** based on EUM, add `child_org:self` to query to see results for current org.

# Infra

## Infrastructure Monitoring & Tagging  - Best Practices

Instrument cloud, on-premise, and VDI hosts with Datadog agent to get:

- Real time telemetry beyond baseline cloud metrics
- Ability to collect data from installed applications and application runtimes using OOTB integrations
- Unified monitoring experiences across multi-cloud, hybrid-cloud, and on-premise environments

Tag infrastructure with `env` to:

- Create operational segmentation that allows users to quickly differentiate performance degradation between production and sandbox environments
- Reuse OOTB dashboards from Datadog that organize signals with `env` template variables

Follow Unified Service Tagging to:

- Create a consistent nomenclature for hosts, containers, services, and applications across your ecosystem.
- Create a consistent observability attribution framework across signals

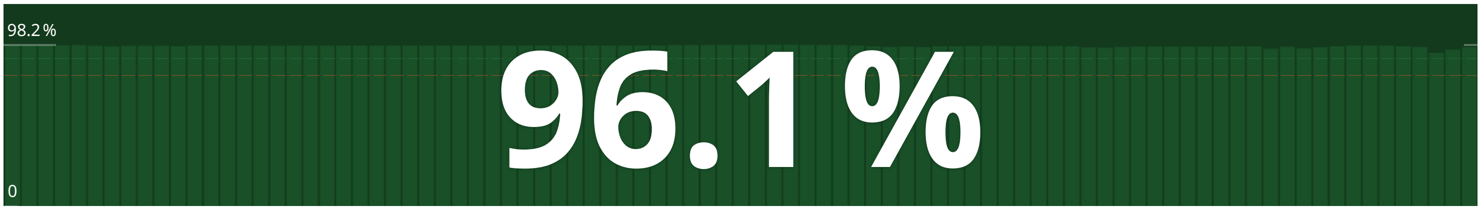
Lastly, deploying a consistent **env.value** such as `prod`, `sandbox`, `qa`, `test`, or `clone` across tech stacks allow operators to create reusable dashboards, monitors, and deployment templates across different service owners.

**Quick links**

| Infra List | Infra Map | Container View |
|---|---|---|
| Infra list - grouped by env | Infra map - grouped by env | Containers - grouped by env |
| Infra list - hosts without env | | Container - containers without env |
| Infra list - hosts without installed agents | | |
| Infra list - agent versions | | |
| Infra list - hosts with no agent - Grouped | | |
| Prod Hosts with No agent - grouped | | |

| Resource Catalog | Fleet Automation |
|---|---|
| Resource Catalog - grouped by env | Fleet - agents grouped by env |
| Resource Catalog - resources without env | Fleet - agents without env |

## Agent Instrumentation Ratio

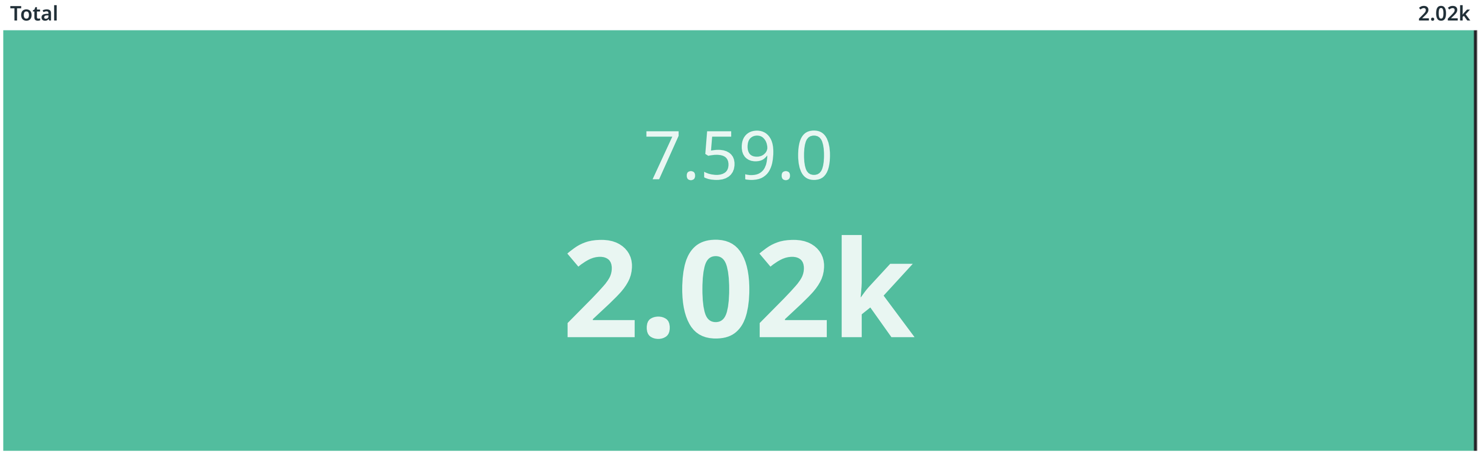


Note that agent instrumentation ratio denotes the number of Datadog agents (installed on hosts) reporting back to Datadog in real time compared to the total number of detected hosts/VMs from cloud providers.

Ideally, 100% of detected hosts are installed with Datadog agents to get the most out of VM/host monitoring. Hosts that do not need Datadog monitoring can be excluded from detection using resource exclusion filters listed below:

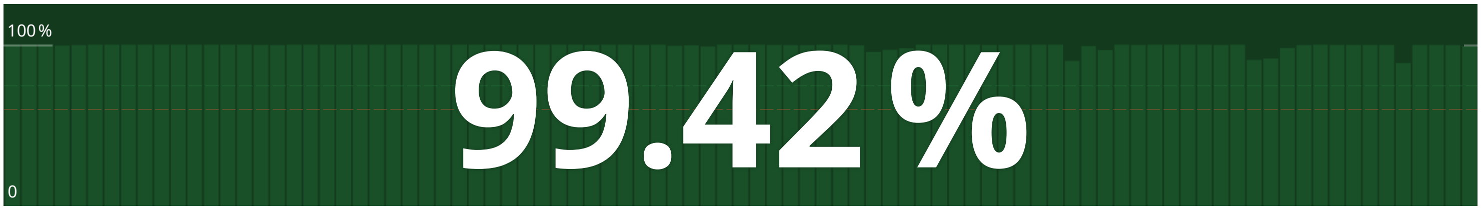
| Azure | AWS | GCP |
|---|---|---|
| Azure Native Exclusion | AWS Exclusion | GCP Exclusion |
| Azure Manual Config Exclusion | | |

## Agents Running by Version



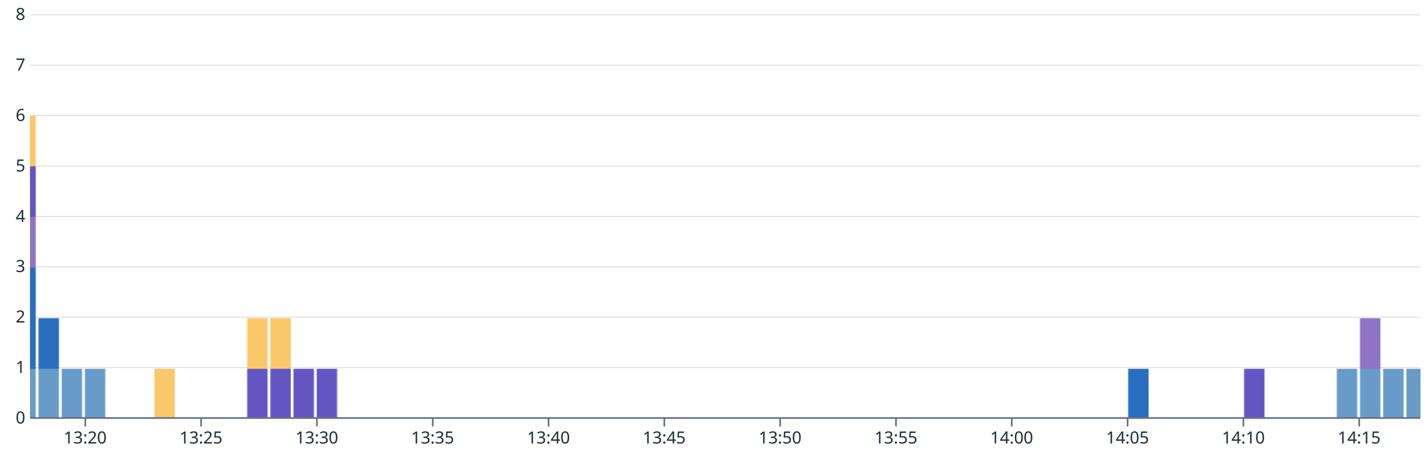
Datadog recommends you update Datadog Agent with every minor and patch release, or, at a minimum, *monthly*.

## Host Agents with "env" Tag

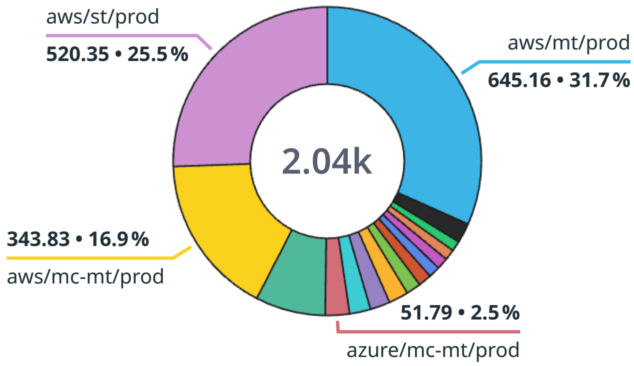


## Host Agents without "env" Tag, by component tags, timeshifted comparison

1h Past 1 Hour

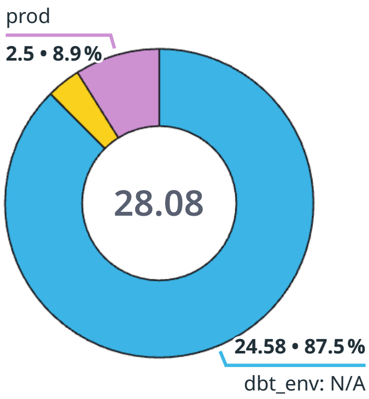


## Agent "env" Tag Values



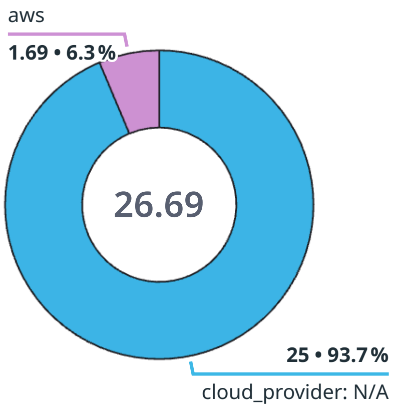
| | ENV | COUNT | ↓ SHARE |
|---|---|---|---|
| 🟦 | aws/mt/prod | 645.2 | 31.68 % |
| 🟪 | aws/st/prod | 520.4 | 25.55 % |
| 🟨 | aws/mc-mt/prod | 343.8 | 16.88 % |
| 🟩 | azure/st/prod | 149.6 | 7.35 % |
| 🟥 | azure/mc-mt/prod | 51.8 | 2.54 % |
| 🟦 | aws/mt/pr | 45.0 | 2.21 % |
| 🟪 | gcp/mc-mt/staging | 42.7 | 2.10 % |
| 🟧 | aws/mc-mt/staging | 42.0 | 2.06 % |
| 🟩 | aws/st/staging › aws/st/test | 31.6 | 1.55 % |
| 🟥 | gcp/mc-mt/prod | 28.4 | 1.39 % |

## "dbt_env" Tag Values when "env" is missing



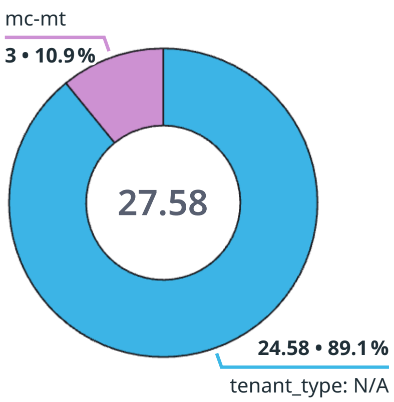
| | DBT_ENV | COUNT | ↓ SHARE |
|---|---|---|---|
| 🟦 | N/A | 24.58 | 87.53 % |
| 🟪 | prod | 2.50 | 8.90 % |
| 🟨 | staging | 1.00 | 3.56 % |

## "cloud_provider" Tag Values when "env" is missing



| | CLOUD_PROVIDER | COUNT | ↓ SHARE |
|---|---|---|---|
| 🟦 | N/A | 25.00 | 93.7 % |
| 🟪 | aws | 1.69 | 6.3 % |

### "tenant_type" Tag Values when "env" is missing

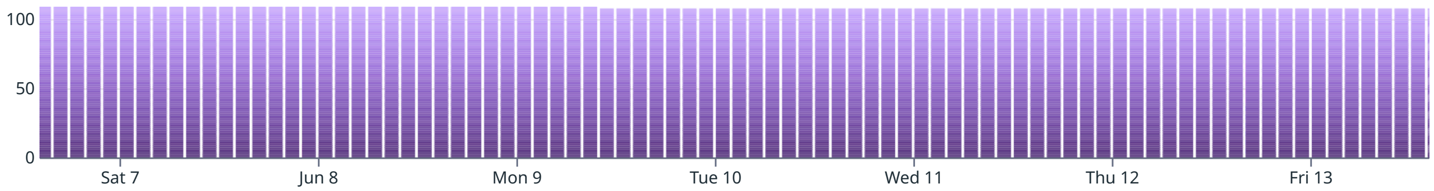


| | TENANT_TYPE | COUNT | ↓ SHARE |
|---|---|---|---|
| 🟦 | N/A | 24.58 | 89.1 % |
| 🟪 | mc-mt | 3.00 | 10.9 % |

## Monitoring Kubernetes Clusters

Deploy the Datadog Cluster Agent  (DCA) to gain insights into control-plane signals across K8s clusters. DCA monitors the control plane with Kubernetes State Core Metrics that provide insights into real-time states of workloads.

### Clusters Monitored by DCA



| | kube_cluster_name in max:datadog.cluster_agent.running{*} | AVG | MIN | MAX | SUM | VALUE |
|---|---|---|---|---|---|---|
| 🟪 | abnamroclearing-cloud | 1 | 1 | 1 | 84 | 1 |
| 🟪 | ads-cloud | 1 | 1 | 1 | 84 | 1 |
| 🟪 | aks-arrowelectronics-cloud | 1 | 1 | 1 | 84 | 1 |
| 🟪 | aks-asb-cloud | 1 | 1 | 1 | 84 | 1 |

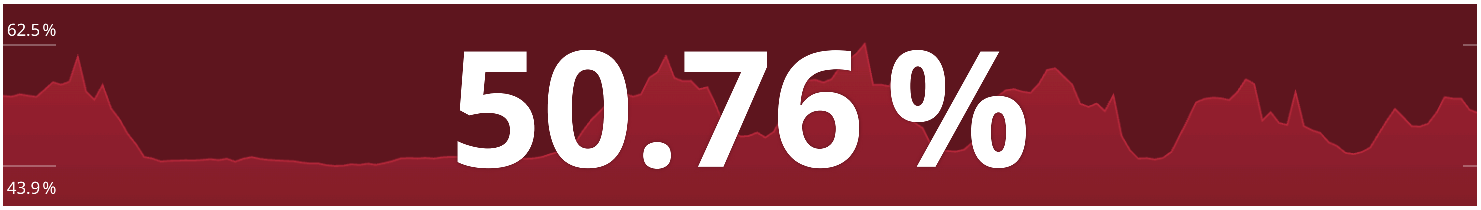
## Monitoring Kubernetes Workloads

Kubernetes workloads should be tagged similarly to allow users to quickly analyze resource utilization and application performance by `env`, `service`, and `version`.

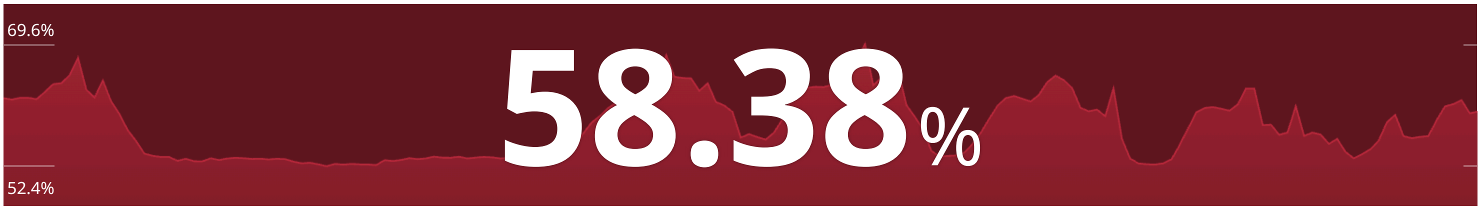
Tagging containers and workloads by version allow direct comparisons between hot-fix versions, incremental releases, canary deployments, and experimentations.

Note that for job-based clusters or workloads shared between services, version tags may not be intuitive. Consider using Datadog Jobs Monitoring instead for these scenarios.

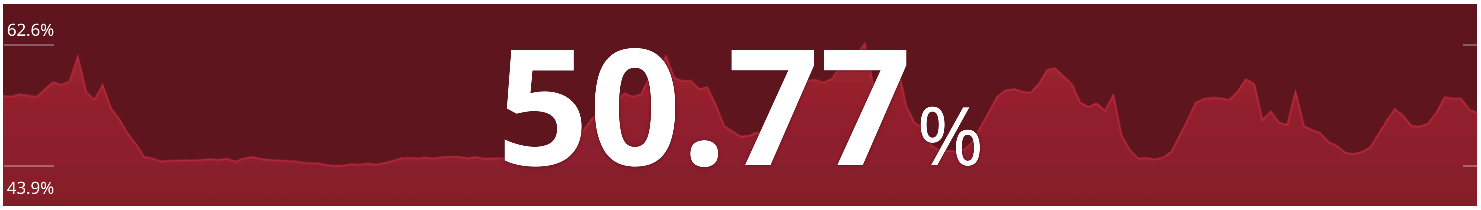
### K8s Containers with "env", "service", and "version" tags (for app-oriented clusters)



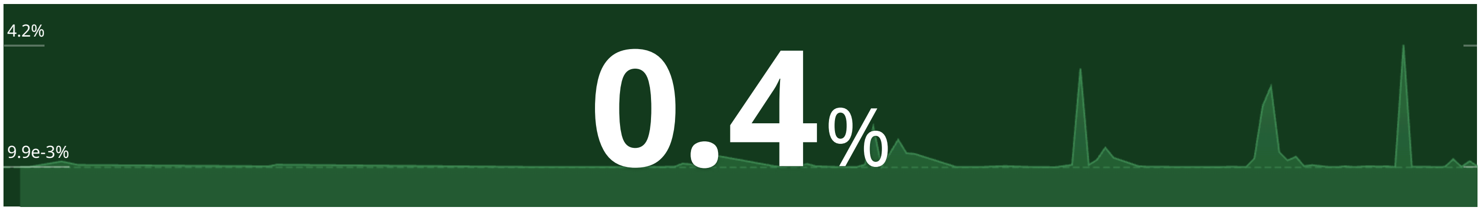
### K8s containers with "service" tags

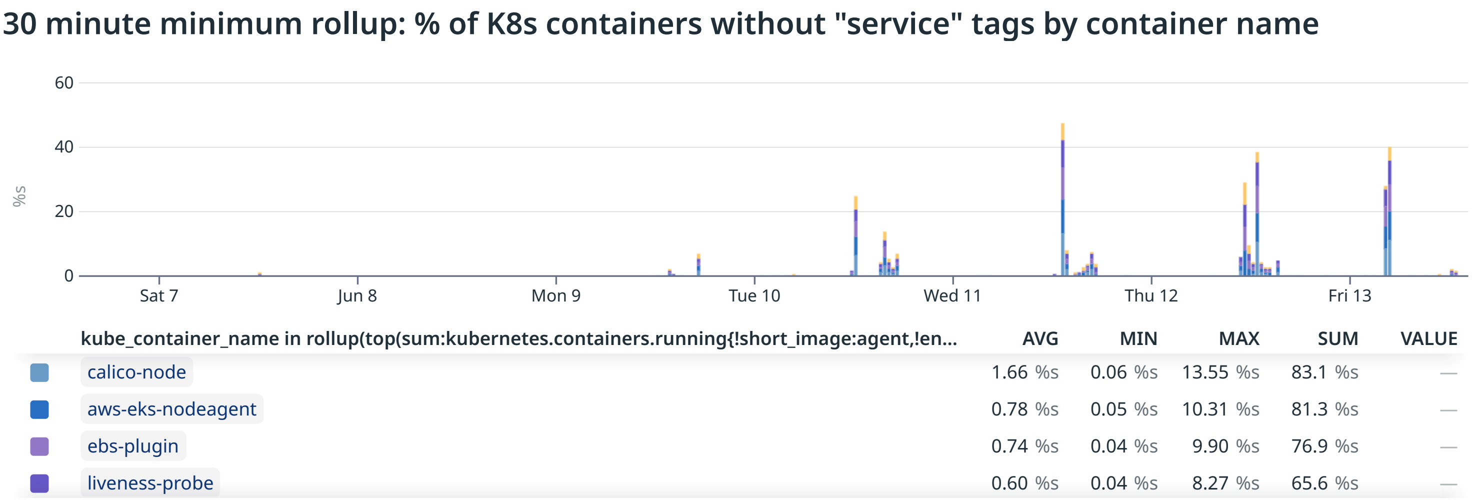


### K8s containers with "service" tags



### K8s containers without "env" tags





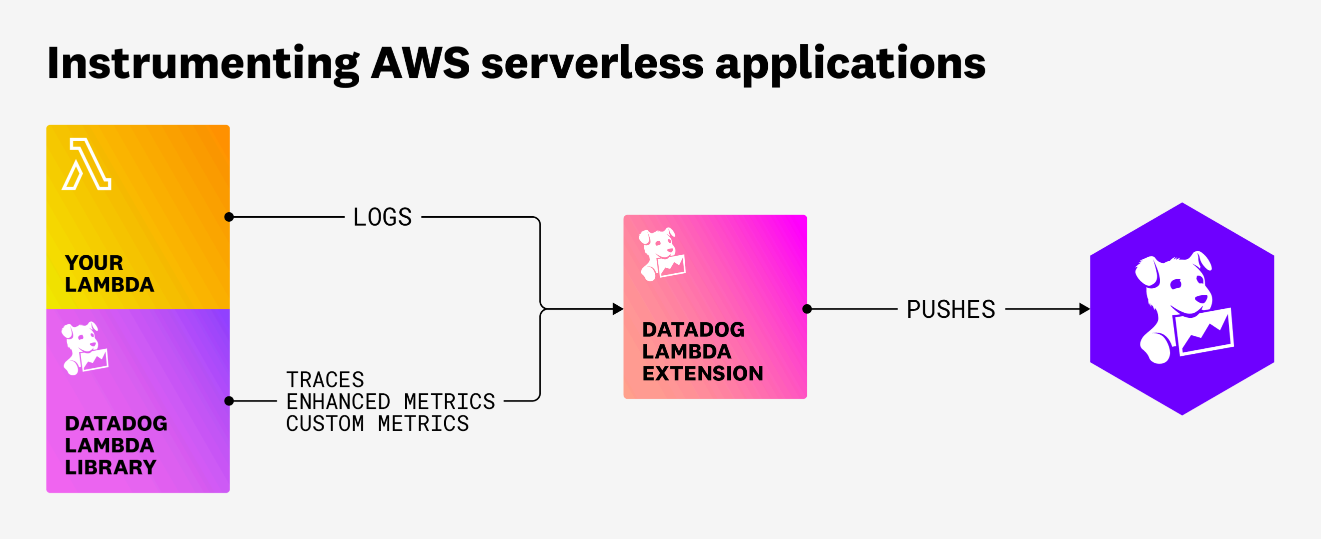
30 minute minimum rollup: % of K8s containers without "service" tags by container name

| | kube_container_name in rollup(top(sum:kubernetes.containers.running{!short_image:agent,!en... | AVG | MIN | MAX | SUM | VALUE |
|---|---|---|---|---|---|---|
| 🟦 | calico-node | 1.66 %s | 0.06 %s | 13.55 %s | 83.1 %s | — |
| 🟦 | aws-eks-nodeagent | 0.78 %s | 0.05 %s | 10.31 %s | 81.3 %s | — |
| 🟪 | ebs-plugin | 0.74 %s | 0.04 %s | 9.90 %s | 76.9 %s | — |
| 🟪 | liveness-probe | 0.60 %s | 0.04 %s | 8.27 %s | 65.6 %s | — |

# Serverless

## AWS Serverless Monitoring - Best Practices

### Overview



### Understanding billable usage

- Lambda functions incur a baseline cost per active function (including any functions detected through the AWS integration)
- Traced Lambda functions incur additional cost per traced invocation (equiv. to each new top level span)

### Instrumentation - performance monitoring

- Install the Lambda integration to get baseline performance metrics across Lambda functions
- Instrument Lambda functions with the Datadog Lambda Extension as a layer or package to gain access to enhanced metrics
- Enable Deployment Tracking to ensure deployment events can be correlated with drift or degraded performance

**Instrumentation - Tracing**

- Generate traces through the Datadog Lambda Extension
- If X-Ray usage is desired, ensure the Datadog X-Ray integration is installed and `DD_MERGE_XRAY_TRACES` is configured for the Datadog Lambda Extension

**Instrumentation - Logging**

- Configure the Datadog Lambda Extension to collect logs
- Ensure logs and traces from Lambda functions are correlated

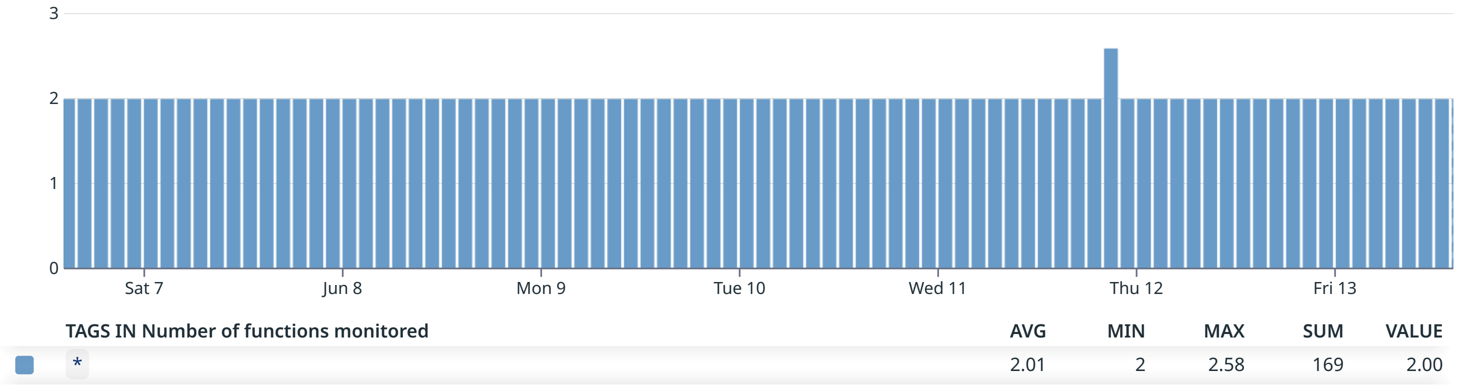
**Instrumentation - Custom Metrics**

- Generate custom metrics directly through Datadog Lambda Extension

**Analysis and monitoring**

- Enhanced lambda metrics are available once functions are instrumented. Leverage these metrics to better understand behaviors like cold starts with additional metadata resolution
- Leverage the Serverless view to understand percentage of functions instrumented. Click into each function to better understand correlated observability signals

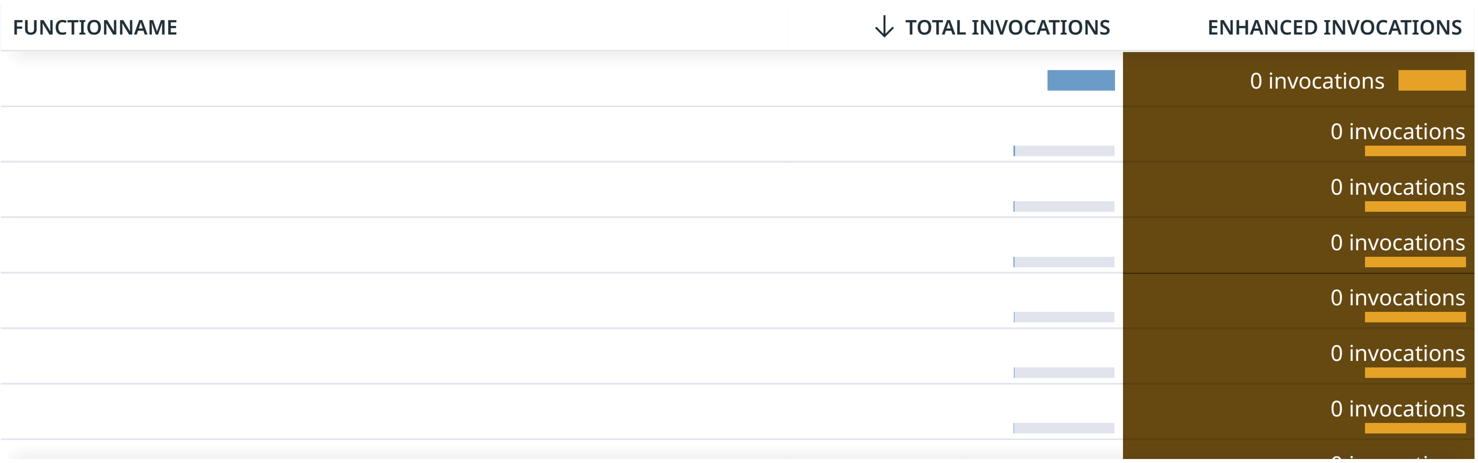
## Lambda Functions Monitored



| | TAGS IN Number of functions monitored | AVG | MIN | MAX | SUM | VALUE |
|---|---|---|---|---|---|---|
| 🟦 | * | 2.01 | 2 | 2.58 | 169 | 2.00 |

**Note:** based on EUM, add `child_org:self` to query to see results for current org.

## Serverless Invocations (Total vs. Enhanced)



| FUNCTIONNAME | ↓ TOTAL INVOCATIONS | ENHANCED INVOCATIONS |
|---|---|---|
| | | 0 invocations |
| | | 0 invocations |
| | | 0 invocations |
| | | 0 invocations |
| | | 0 invocations |
| | | 0 invocations |
| | | 0 invocations |

- Note that function names without enhanced invocations indicate that a Datadog Lambda Extension layer was never installed

**Note:** based on EUM, add `child_org:self` to query to see results for current org.

Traced Lambda Invocations

(No data)

**Note:** based on EUM, add `child_org:self` to query to see results for current org.

# APM

## Application Performance Monitoring - Best Practices

**Concepts**

- Ensure APM users have a firm understanding of trace sampling

**Instrumentation**

- Instrument applications with APM to get insights into real-time performance
- Correlate your logs and traces to ensure users can quickly navigate between traces and related logs
- Update APM instrumentation to remove legacy instrumentation

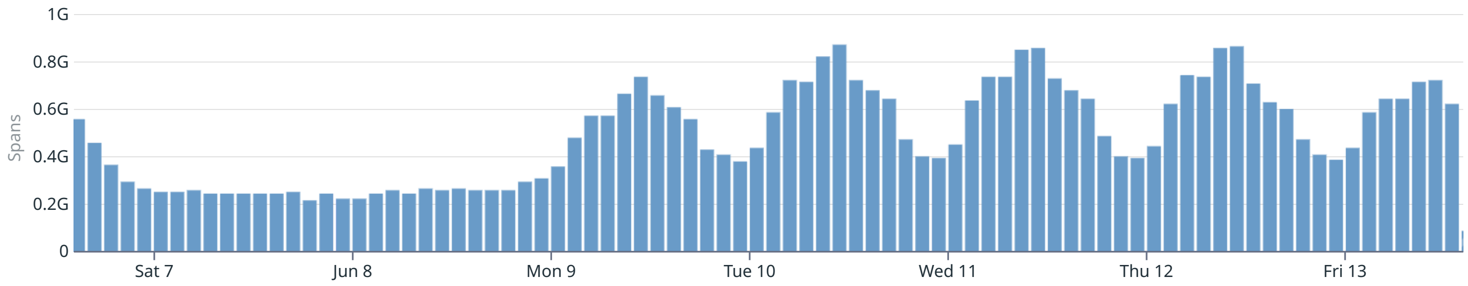
**Retention (after Datadog ingestion)**

- By default, Datadog uses intelligent retention to keep a subset of all ingested spans
- Use custom retention filters to retain an additional percentage of ingested spans  this allows you to get detailed analysis of application performance
- Use this guide to fine tune and monitor span retention

**Analysis and monitoring**

- Adding the tag `env` and `service` on monitors will ensure proper correlation with services in the Service Catalog
- Leverage trace metrics to get 100% accurate performance metrics measured at instrumentation
- Add service metadata in Service Catalog to enrich services with operational background and on-call guidance
- Use service scorecards to establish baseline governance and evaluate observability maturity

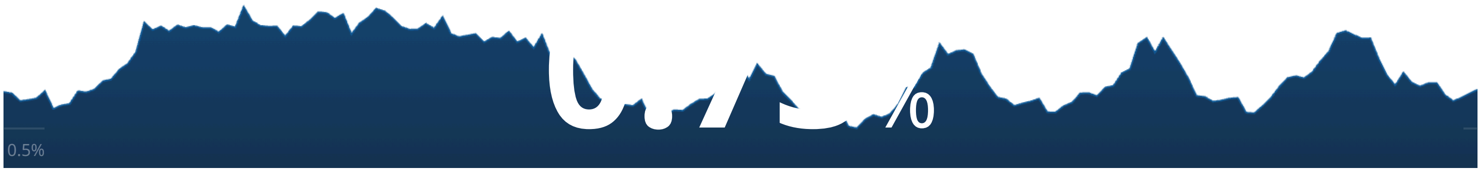
## Span Volume



| | TAGS | METRIC | AVG | MIN | MAX | SUM | VALUE |
|---|---|---|---|---|---|---|---|
| 🟦 | * | Ingested spans | 486M spans | 94.7M spans | 867M spans | 40.8G spans | 607M spans |
| 🟦 | * | Indexed spans | 3.57M spans | 793.01k spans | 5.19M spans | 299.88M spans | 3.94M spans |

**Note:** based on EUM, add `child_org:self` to query to see results for current org.

## Indexed over Ingested Spans (%)

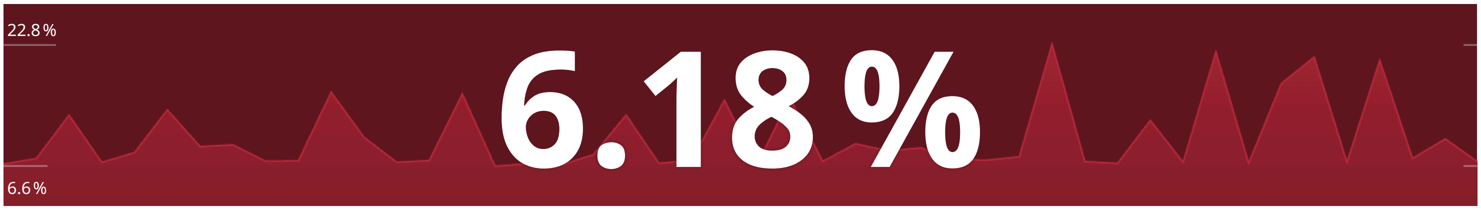


**Note:** based on EUM, add `child_org:self` to query to see results for current org.

## Service Entry Spans Tagged with "env" and "version"

| SERVICE | ↓ SPANS | % SPANS WITH ENV AND VERSION |
|---|---|---|
| | | 100 |
| | | 100 |
| | | 100 |
| | | 100 |
| | | 100 |
| | | 100 |
| | | |

## Percentage of Logs Correlated with APM Services



## APM Services with Logs Correlated

| SERVICE | SERVICE COUNT | ↓ CORRELATION FOUND |
|---|---|---|
| | | 1 |
| | | 1 |
| | | 1 |
| | | 1 |
| | | 1 |
| | | 1 |
| | | 1 |

# Logs

## Log Management — Best Practices

**Instrumentation**

- When possible, emit single line, JSON-formatted logs to ensure proper Datadog ingestion
- Ensure agent-collected logs do not exceed 256 KB, and API collected logs do not exceed 1 MB
- Leverage agent configuration to tailor log collection

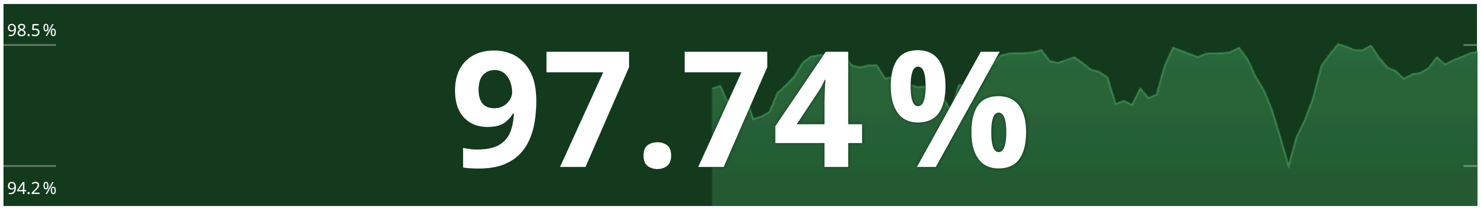
**Ingestion and Retention**

- Ensure application logs are properly parsed by pipelines, which transform and alias attributes to ensure a consistent debugging experience
- Create indexes for critical and/or error logs to support production outages
- Leverage exclusion filters in retention indexes to further refine retention and discard noise
- Leverage sampling within exclusion filters for high volume use cases
- Establish daily quotas for non-critical indexes to reduce overspend
- Use alternate destinations such as Flex Logs and Log Archives for logs that do not need immediate action or advanced visualizations/monitoring

**Analysis and Monitoring**

- Create a default retention filter with daily quota to analyze new patterns
- Analyze logs using patterns and grouping to understand ingestion/index patterns

## Ingested Logs Processed by Pipeline

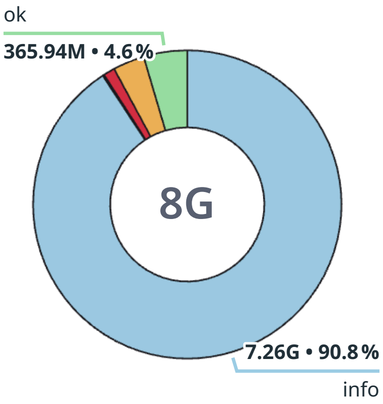


## Log Events Indexed

| DATADOG_INDEX | ↓ CURRENT MONTH | PRIOR MONTH | CHANGE IN % |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

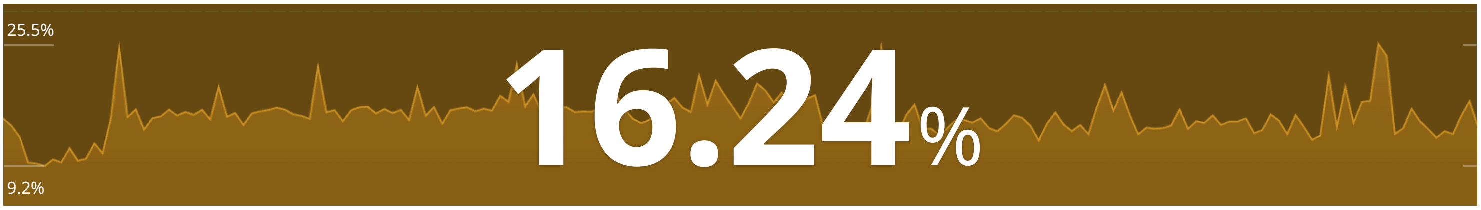
**Note:** based on EUM, add `child_org:self` to query to see results for current org.

## Indexed Logs by Status



| | STATUS | COUNT | ↓ SHARE |
|---|---|---|---|
| 🟦 | info | 7.26G | 90.76 % |
| 🟩 | ok | 365.94M | 4.57 % |
| 🟧 | warn | 264.02M | 3.30 % |
| 🟥 | error | 96,395.7k | 1.20 % |
| ⬜ | debug | 12,939.0k | 0.16 % |
| 🟥 | alert | 200.4k | 0.00 % |
| 🟥 | critical | 16.3k | 0.00 % |
| ⬜ | notice | 8.0k | 1e-4 % |
| 🟥 | emergency | 1.4k | 1.8e-5 % |

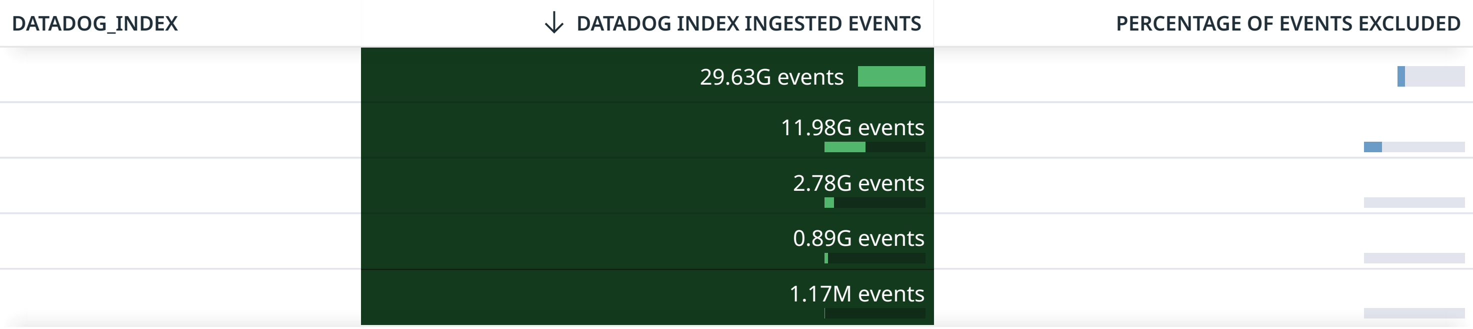
## Log Events Excluded



**Note:** based on EUM, add `child_org:self` to query to see results for current org.

## Log Events Excluded

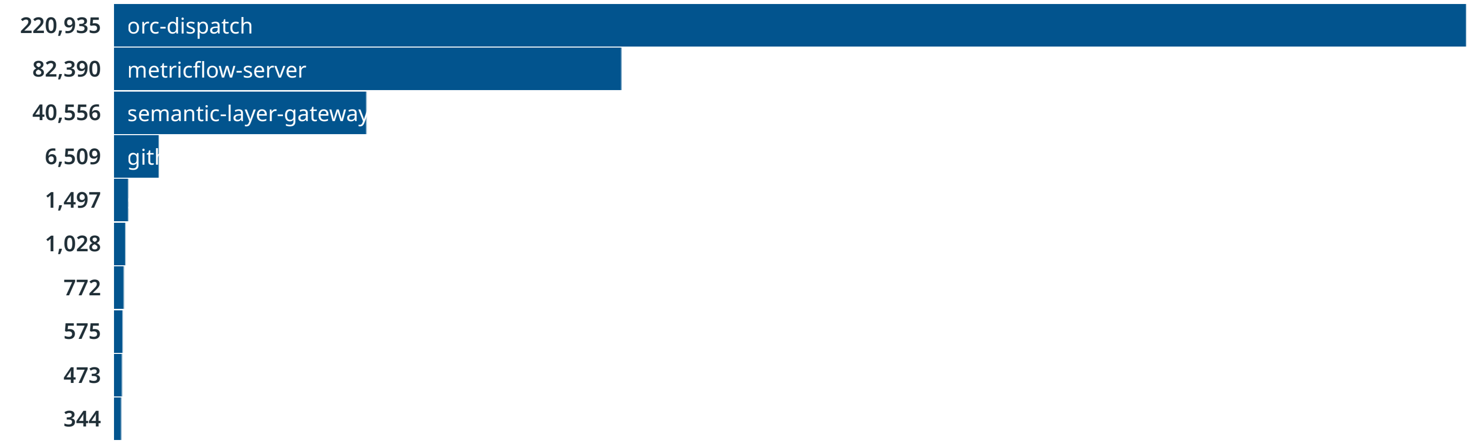
1mo May 1 - May 31



| DATADOG_INDEX | ↓ DATADOG INDEX INGESTED EVENTS | PERCENTAGE OF EVENTS EXCLUDED |
|---|---|---|
| | 29.63G events | |
| | 11.98G events | |
| | 2.78G events | |
| | 0.89G events | |
| | 1.17M events | |

**Note:** based on EUM, add `child_org:self` to query to see results for current org.

## Logs Truncated by Service



# RUM

## Real User Monitoring - Best Practices

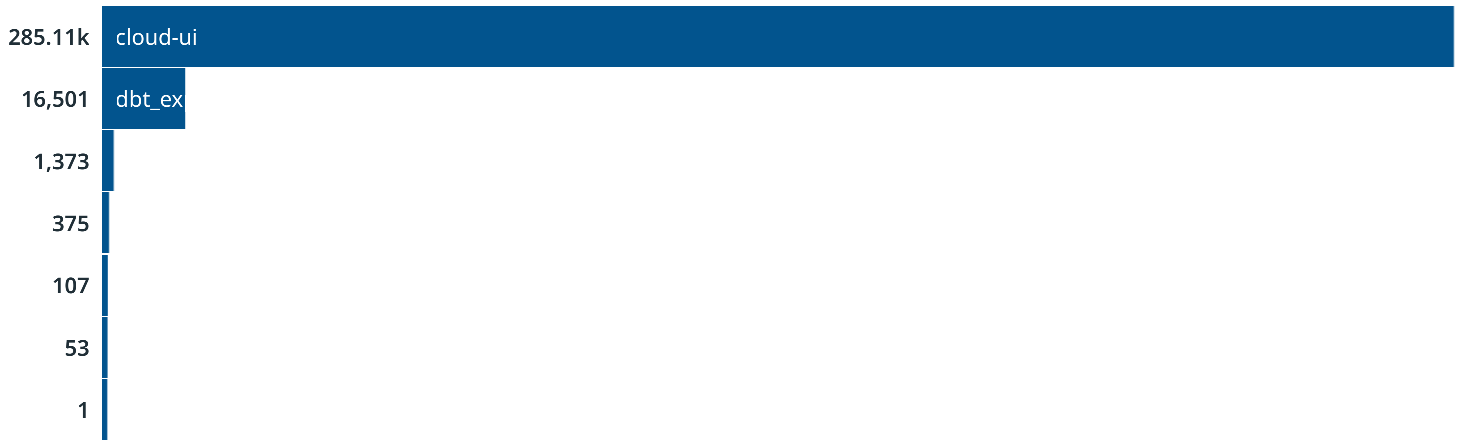
### Instrumentation

- Connecting RUM to APM allows Datadog to correlate RUM events to backend APM traces. This drastically reduce debugging lead time as problematic UI interactions quickly linked to backend errors
- Label sessions with user or equivalent metadata/attributes after authentication to efficiently analyze user behavior
- Leverage RUM Sampling to control the amount of user sessions that are being forwarded to the Datadog platform

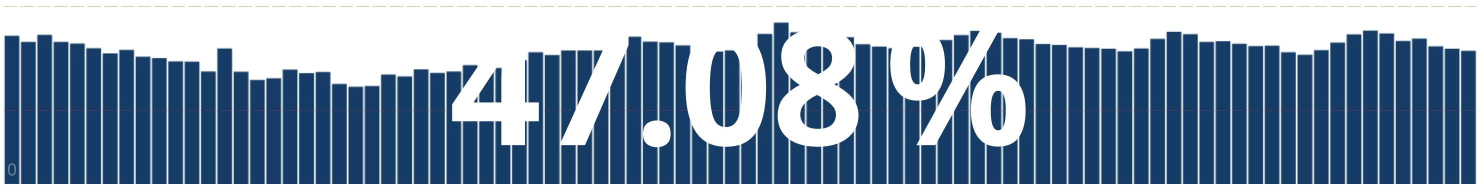
**Analysis and monitoring**

- Ensure teams use consistent attribute convention across applications instrumentation and custom attributes allow stakeholders to quickly identify end users and establish trends based on attributes
- Ensure teams understand how RUM functions and how to troubleshoot initialization
- Generate metrics from RUM sessions within the Datadog platform for analysis of user behavior beyond 30 days (default RUM retention)

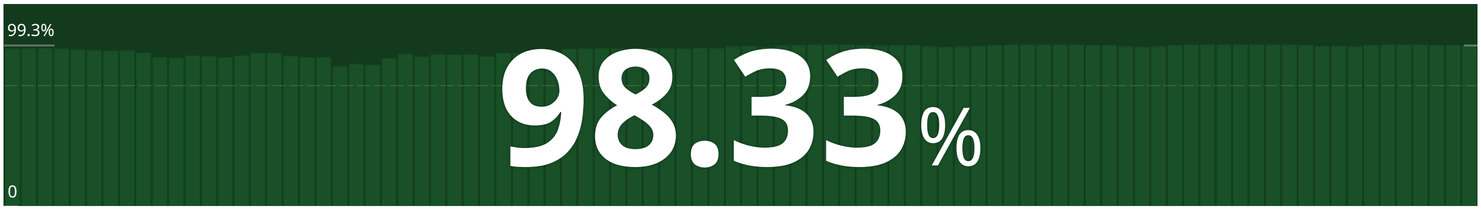
## RUM Sessions by Application



## RUM Async Request Events with APM Traces



## RUM Sessions with User IDs



*Replace User ID with RUM attribute of significance.*

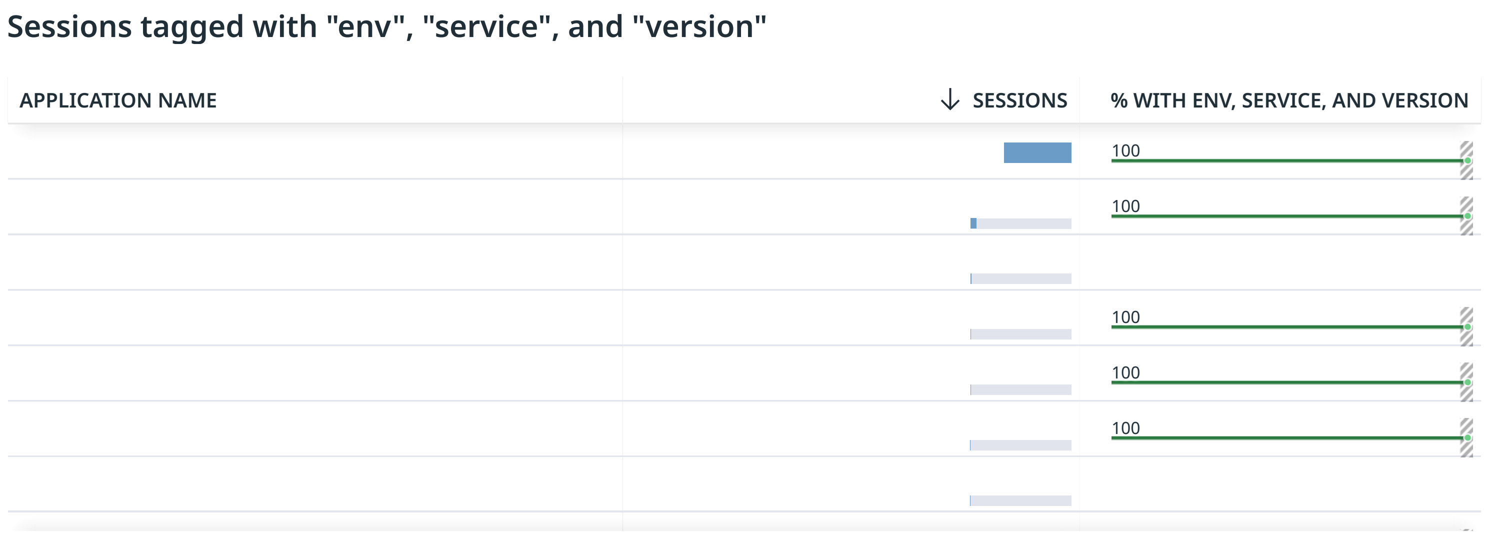
## Mobile RUM

### Enrich RUM Data

- Sessions enriched with custom attributes provide more insights into execution context and ownership. Some examples of added information are: user ID, user email, customer tier, feature flags
- Label sessions with user metadata after users authenticate in order to track user journeys and the impact of errors
- Ensure teams use consistent attribute convention across application custom attributes to allow stakeholders to quickly identify end users and establish trends based on attributes
- Add Feature Flag information to help investigate if any change you introduce is impacting your user experience or negatively affecting performance
- De-obfuscate your stack traces and setup Crash Reporting to get comprehensive crash reports and error trends
- Ensure RUM-instrumented applications follow the same convention to provide clear domain-driven analysis for users across multiple apps

### Connect Telemetry

- Connecting RUM to APM allows Datadog to correlate RUM events to backend APM traces, drastically reduce debugging lead time as problematic UI interactions quickly linked to backend errors
- Configure the `traceSampler` parameter to keep a defined percentage of the backend traces
- Connecting RUM to Logs allows Datadog to correlate RUM events to application logs, providing more context on the application activity
- Apply Unified Service Tagging on your applications to ensure accurate telemetry correlation and enabled deployment tracking



Sessions tagged with "env", "service", and "version"

| APPLICATION NAME | ↓ SESSIONS | % WITH ENV, SERVICE, AND VERSION |
|---|---|---|
| | | 100 |
| | | 100 |
| | | |
| | | 100 |
| | | 100 |
| | | 100 |
| | | |

### RUM Data Security

- Mobile RUM tracking is only run upon user consent. Review your applications compliance requirements with GDPR and similar policies and implement tracking consent as required
- The RUM client token is used to match data from the end user to a specific RUM application in Datadog. It is unencrypted and visible from the client side of an application. Because the client token is only used to send data to Datadog, there is no risk of data loss due to this token; however, Datadog recommends good client

token management to avoid other kinds of misuse, including regularly rotating the client token to ensure that it is only used by your application

### Sampling & Data Retention

- By default, data retention for production environments is 30 days for sessions, views, actions, errors, and session recordings and 15 days for resources and long tasks
- Review the guide Best Practices for RUM Sampling and set a sample rate for sessions and session replay
- Consider creating custom metrics from RUM data to retain data and analyze trends over 15 months
- RUM ensures availability of data when user devices are offline. In low-network areas, or when the device battery is too low, all RUM events are first stored on the local device in batches. They are sent as soon as the network becomes available, and the battery is high enough to ensure the RUM SDK does not impact the end user's experience. If the network is not available while your application is in the foreground, or, if an upload of data fails, the batch is kept until it can be sent successfully

# Custom Metrics

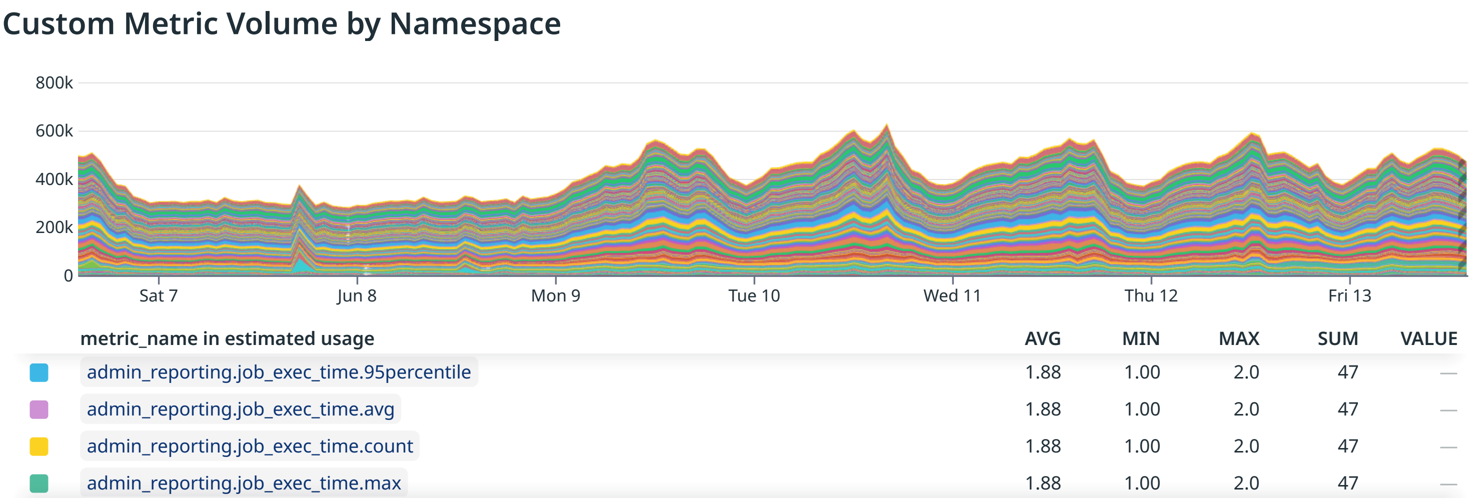
## Best Practices

- Metrics with a high cardinality compared to the other often highlights a misconfiguration
- Use Datadog's supported integrations instead of OpenMetrics or Prometheus metrics scraping to avoid unnecessary billing
- Leverage Metrics without Limits to decouple time series ingestion from indexation — allowing better control on what metrics tags are queryable and billable
- Use Billing Summary to review custom metrics costs
- Add monitors and dashboards to track estimated usage metrics for anomalies in usage
- Datadog charges based on the monthly **average** of unique custom metrics submitted to the Datadog Infrastructure service per hour

### Top Custom Metrics Cardinality by Name

| METRIC_NAME | METRIC VOLUME | ↓ % TOTAL VOLUME | % PREV MONTH |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

**Note:** based on EUM, add `child_org:self` to query to see results for current org.



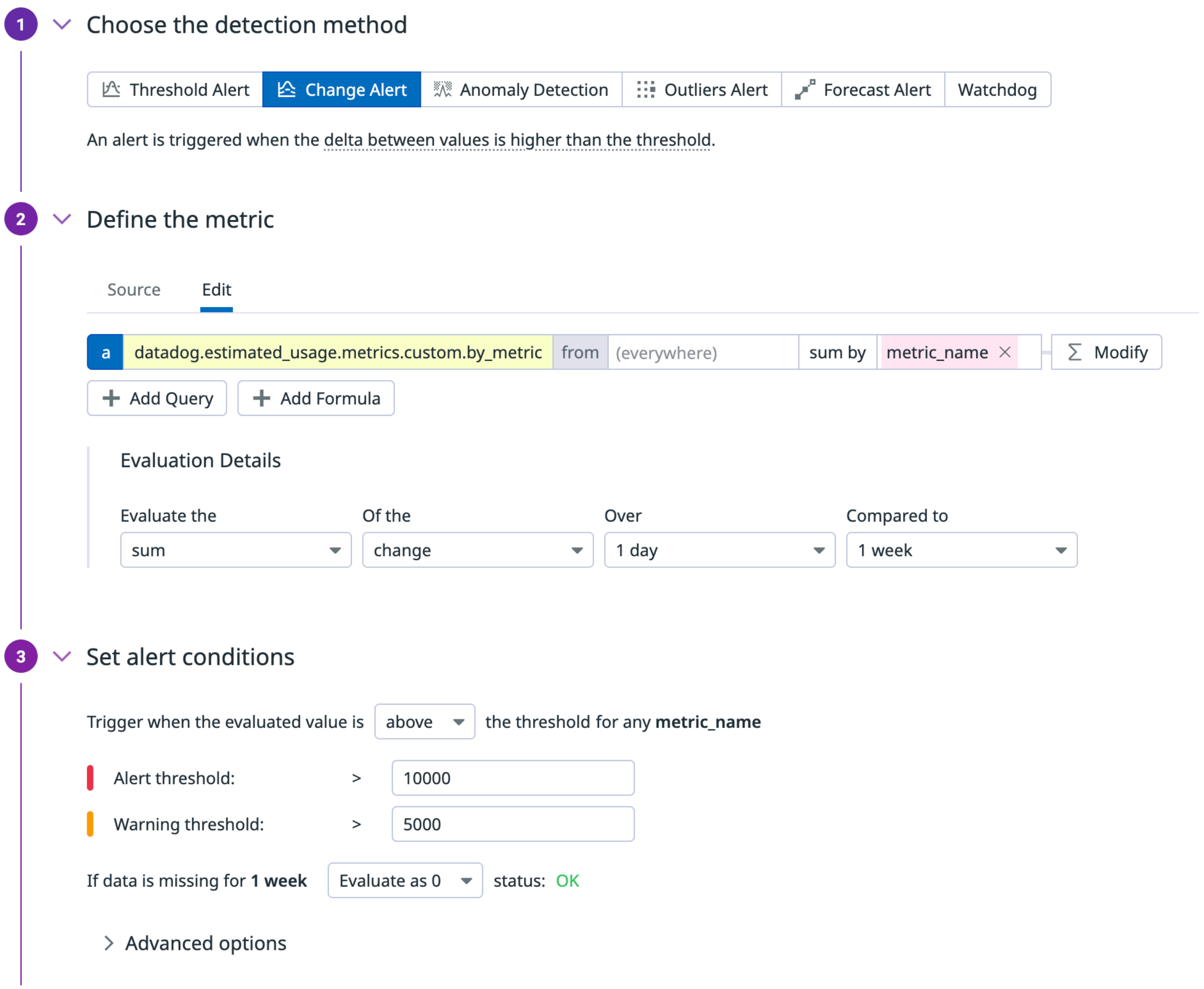
Custom Metric Volume by Namespace

| | metric_name in estimated usage | AVG | MIN | MAX | SUM | VALUE |
|---|---|---|---|---|---|---|
| 🟦 | admin_reporting.job_exec_time.95percentile | 1.88 | 1.00 | 2.0 | 47 | — |
| 🟪 | admin_reporting.job_exec_time.avg | 1.88 | 1.00 | 2.0 | 47 | — |
| 🟨 | admin_reporting.job_exec_time.count | 1.88 | 1.00 | 2.0 | 47 | — |
| 🟩 | admin_reporting.job_exec_time.max | 1.88 | 1.00 | 2.0 | 47 | — |

**Note:** based on EUM, add `child_org:self` to query to see results for current org.

## Example Monitor for Custom Metric Cardinality Spike

Below outlines an example monitor for custom metric cardinality, using a change condition:
Example JSON of monitor:



1 Choose the detection method

An alert is triggered when the delta between values is higher than the threshold.

2 Define the metric

a datadog.estimated_usage.metrics.custom.by_metric from (everywhere) sum by metric_name ×

Evaluation Details

| Evaluate the | Of the | Over | Compared to |
|---|---|---|---|
| sum | change | 1 day | 1 week |

3 Set alert conditions

Trigger when the evaluated value is above the threshold for any **metric_name**

Alert threshold: > 10000

Warning threshold: > 5000

If data is missing for **1 week** Evaluate as 0 status: OK

```
{
 "name": "Monitor cardinality increased for {{metric_name.name}}",
 "type": "query alert",
 "query": "change(sum(last_1d),last_1w):sum:datadog.estimated_usage.metrics.custom.by_metric{*} by {metric_name} > 10000",
 "message": "@example_user@example_org.com \n@slack-example_governance_channel \n\n{{#is_alert}}\nAverage cardinality has increased by over 10,000 for {{metric_name.name}} in the past day compared to the last week.\nPlease review cardinality on this dashboard link:\n\n{{/is_alert}}\n\n{{#is_warning}}\nAverage cardinality has increased by over 1,000 for {{metric_name.name}} in the past day compared to the last week.\\nPlease review cardinality on this dashboard link:\n{{/is_warning}}",
 "tags": [],
 "options": {
  "thresholds": {
   "critical": 10000,
   "warning": 5000
  },
  "notify_audit": false,
  "on_missing_data": "default",
  "include_tags": true,
  "new_group_delay": 60
 }
}
```

# Monitors

## Monitoring — Best Practices

- Leverage Dynamic Handles to alert appropriate teams
- Consider using Alert Grouping with Multi-Alert to segment recipients monitoring the same metric. This allows a single monitor to potentially replace dozens of similar monitors
- Tagging monitors with `env` and `service` ensures that monitors are correlated with the relevant APM services
- Add governance for monitor tagging with monitor tag policies
- Reduce noise floor by removing unnecessary or deprecated monitors
- Monitors muted for 15+ days should be reviewed for sanitization or removal
- Use the OOTB Monitor Quality tool to gain insights into potential improvements

### Monitors with Alerts

| MONITOR ID | EVENT NAME | ↓ ALERTS |
|---|---|---:|
| | | 32,939 |
| | | 2,886 |
| | | 1,818 |
| | | 1,806 |
| | | 1,208 |
| | | 973 |
| | | 916 |

### Alerts by Notification Recipient

| MONITOR NOTIFICATIONS | ↓ ALERTS |
|---|---:|
| | |
| | |
| | |
| | |
| | |
| | |
| | |

# Dashboards

## Dashboards — Best Practices

- Increase dashboard widget density and reuse with template variables
- Add addition event context to time series widgets with event overlays
- Leverage pre-built dashboards installed by integrations
- Work with your CSM/TAM to get a report of unused dashboards
- Leverage the Datadog Workshop for self-guided onboarding

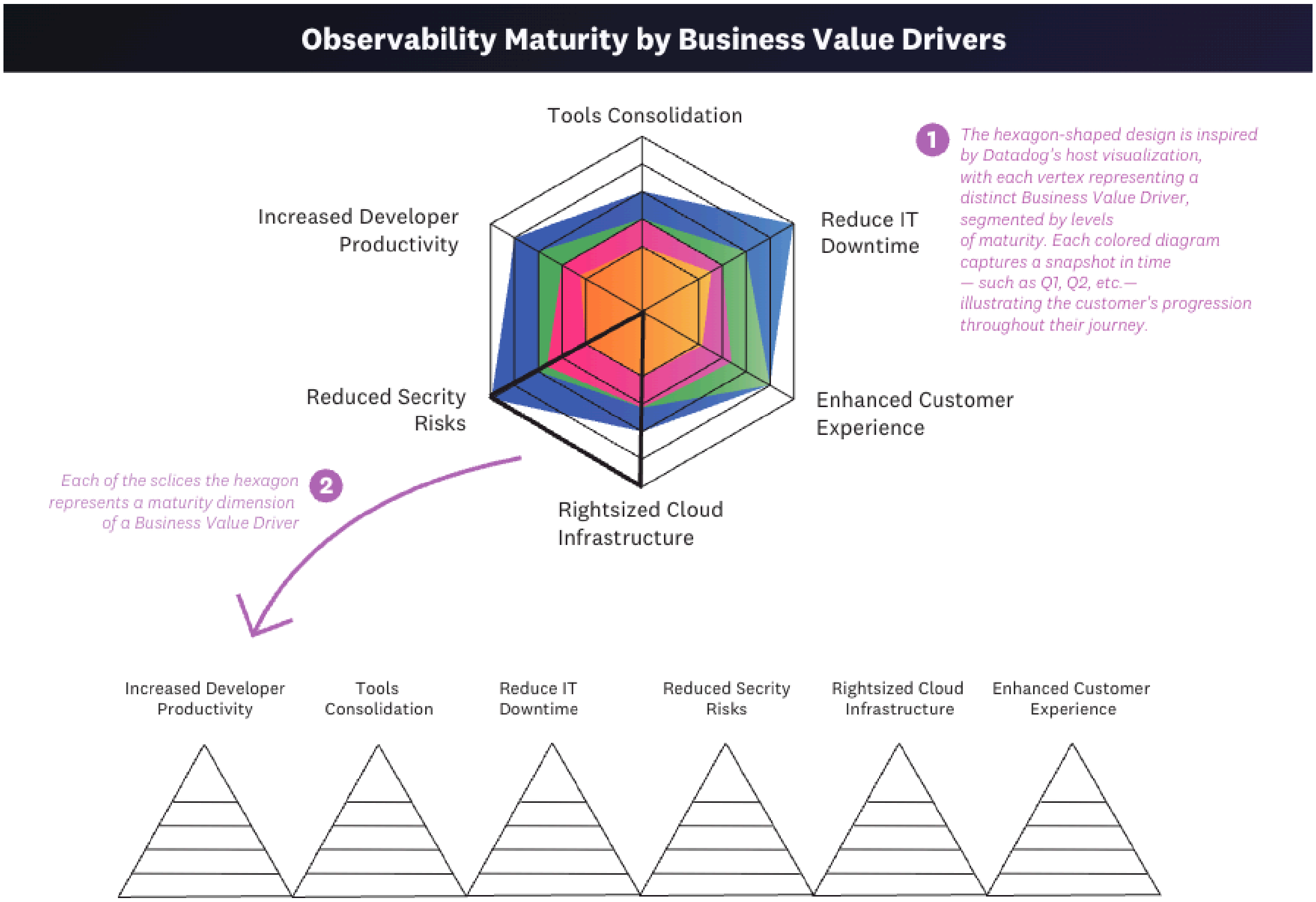
# Resources

- Datadog Foundational Enablement
- Datadog Learning Center
- Datadog Documentation
- Datadog Blog

- Datadog Services and Enablement

## Observability Maturity

Interested in collaborating with Datadog on measuring your observability maturity? Let us know!



Observability Maturity by Business Value Drivers

1 The hexagon-shaped design is inspired by Datadog's host visualization, with each vertex representing a distinct Business Value Driver, segmented by levels of maturity. Each colored diagram captures a snapshot in time — such as Q1, Q2, etc.— illustrating the customer's progression throughout their journey.

2 Each of the sclices the hexagon represents a maturity dimension of a Business Value Driver

Tools Consolidation
Increased Developer Productivity
Reduce IT Downtime
Reduced Secrity Risks
Enhanced Customer Experience
Rightsized Cloud Infrastructure
Increased Developer Productivity
Tools Consolidation
Reduce IT Downtime
Reduced Secrity Risks
Rightsized Cloud Infrastructure
Enhanced Customer Experience

Screenshot 2025-06-06 at 12.42.51 PM.png