upGrad ENTERPRISE

DATADOG MONITORING TOOL

19 August, 2024

upGrad ENTERPRISE

Course : Datadog

Lecture On : Monitoring Tool

Instructor : Sandeep Kumar

Introduction

- Your Name
- Total experience

Background – Development / Infrastructure

Experience on Monitoring Tools

Your expectations from this training

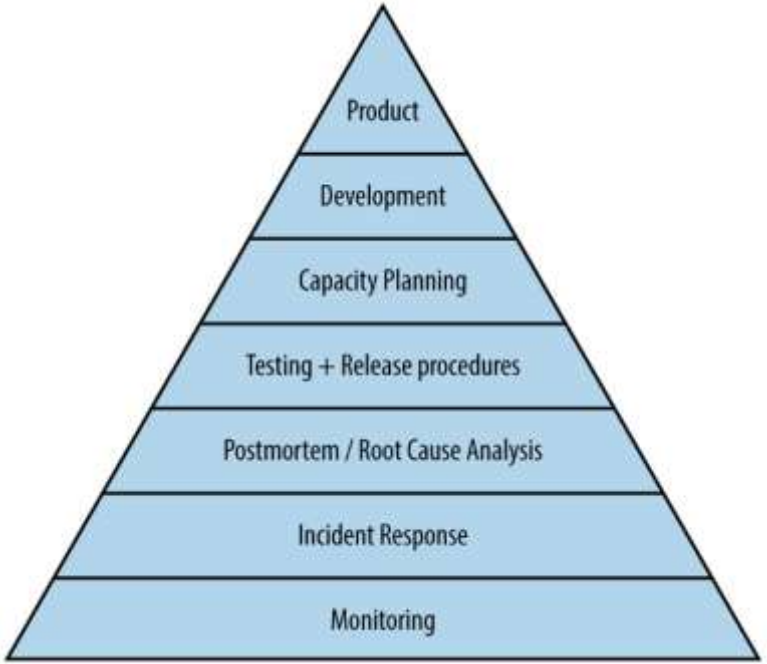
Agenda

- Overview
- Integrations
- Infrastructure
- Host Map
- Events
- Dashboards

Overview

Monitoring

Monitoring is the most basic component in their reliability pyramid and enables incident response and postmortems



Context

Observability means assembling all fragments from logs, monitoring tools and organize them in such a way which gives actionable knowledge of the whole enviroment, thus creating an insight

Monitoring tells you whether a systems is working, observability lets you ask why it isn't working

Monitoring and Observability

Once upon a time there was “Monitoring”

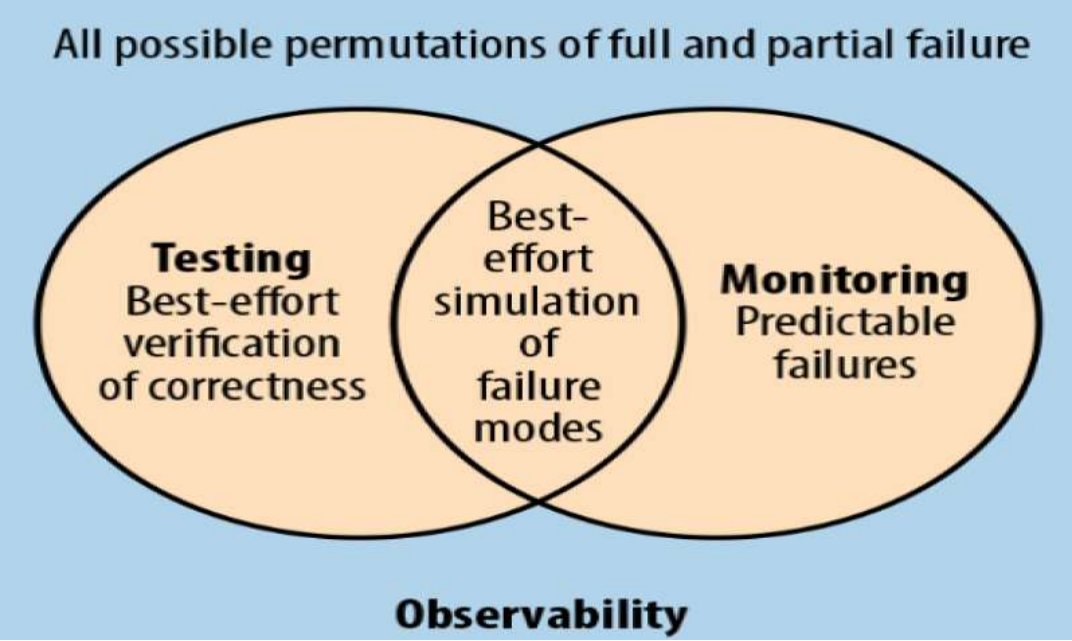
Observability is a superset of monitoring. It provides not only high-level overviews of the system’s health but also highly granular insights into the **implicit failure modes** of the system.

In addition, an observable system furnishes ample context about its **inner workings, unlocking the ability to uncover deeper, systemic issues.**

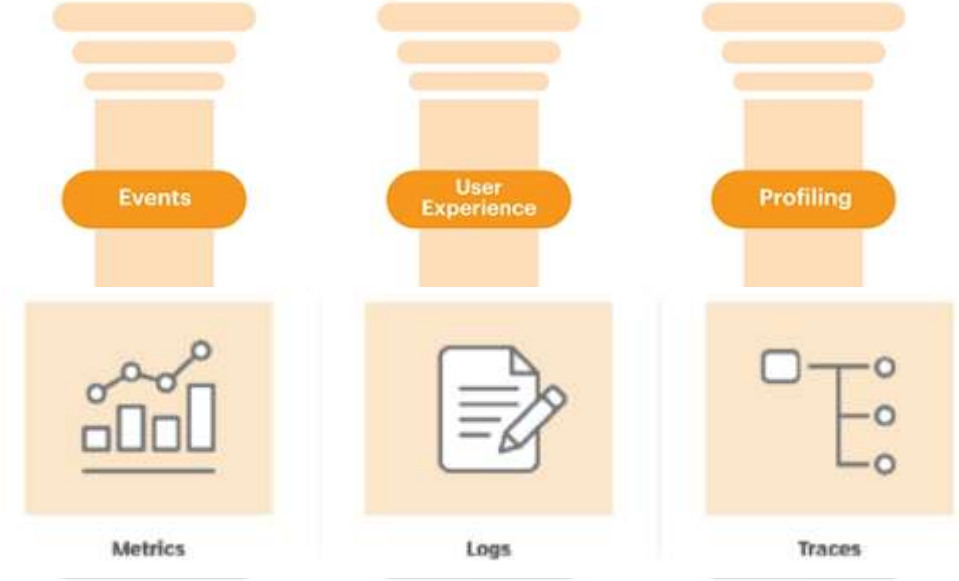
Monitoring, on the other hand, is best suited to report the **overall health of systems and to derive alerts**.

Observability

Observability



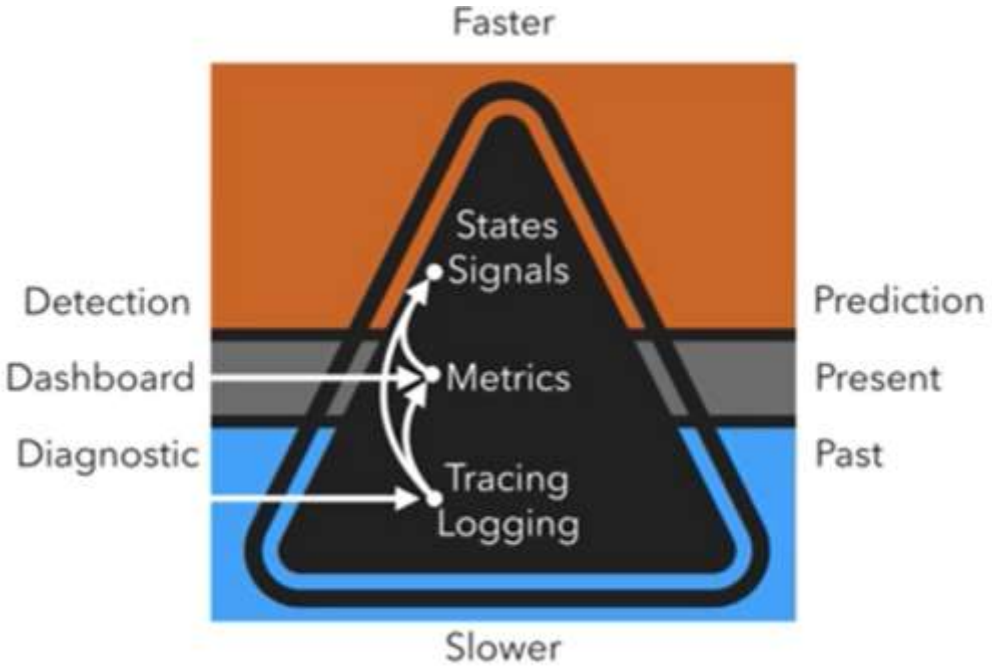
Three Pillars of Observability



How to achieve Observability

		Examples
Traces	*Relationships between events*	Application components involved during a request with an error
Metrics	*Measurement of an event*	Throughput, error rate, request rate, request duration
Logs	*Human-readable events*	System startup output, process output

The Observability Hierarchy



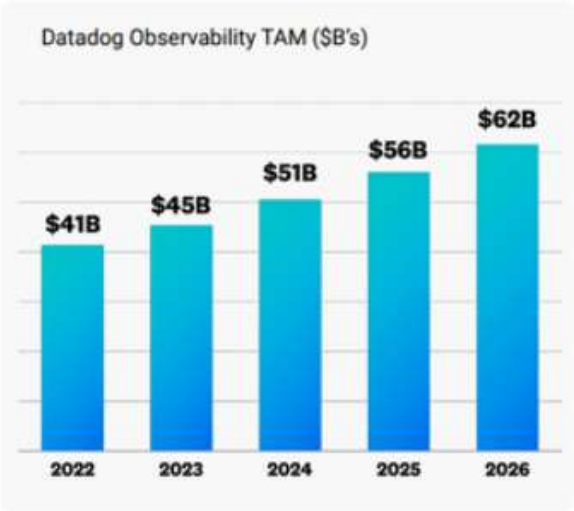
Observability

At our code, Observability is a very large opportunity

$62B

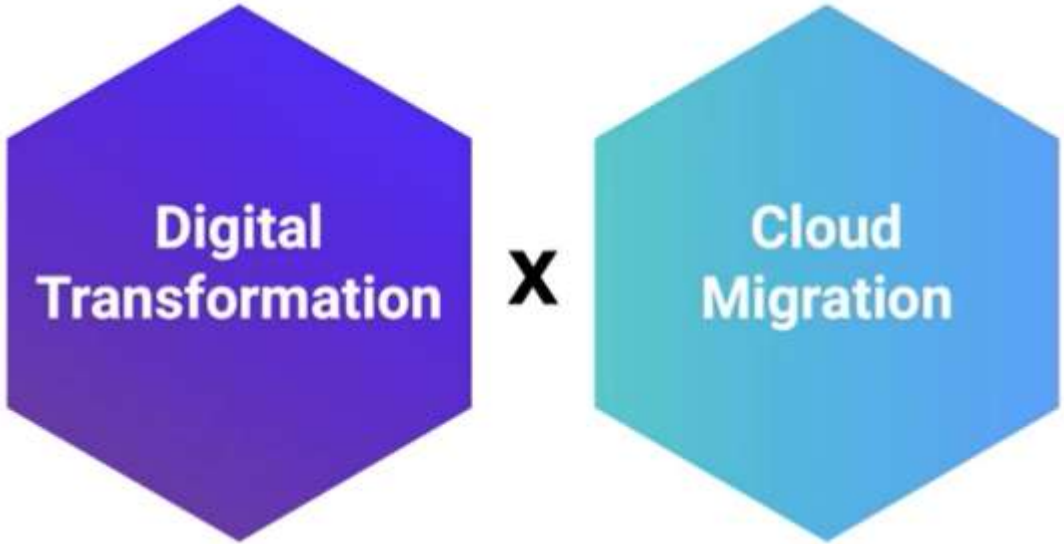
in 2026

Gartner Forecast: Enterprise Infrastructure Software, Worldwide, 2020-2026, 2Q22 Update. Published June, 2022. IT Operations Market.



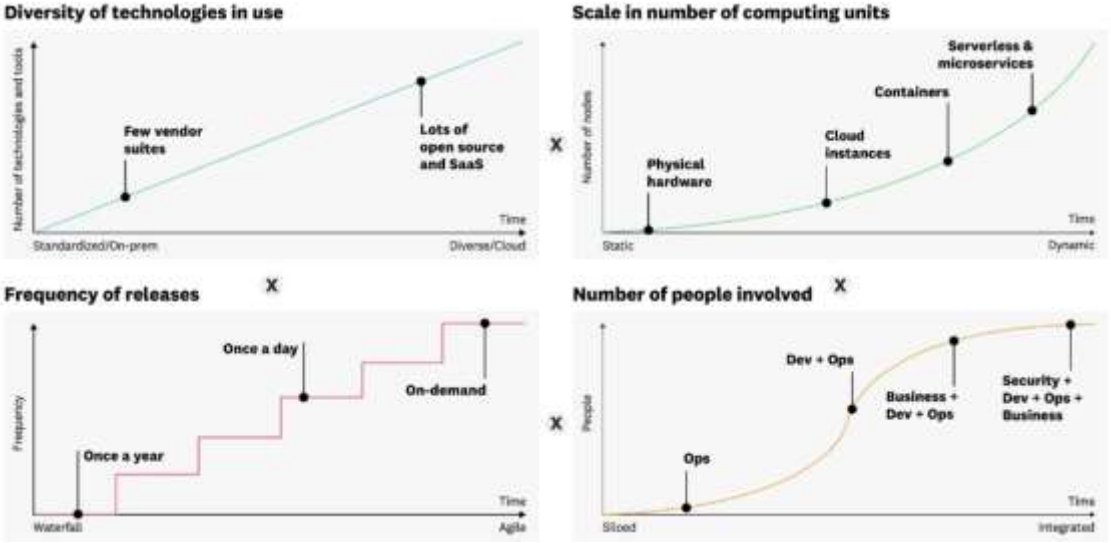
What's happening in IT today

Two broad and deep transitions



What's happening in IT today

Evolving technology paradigms create rising complexity



Telemetry

Telemetry is the collection of measurements or other data at remote points and their automatic transmission to receiving equipment for monitoring. The word is derived from the Greek roots tele, “**remote**” and metron, “**measure**”

Telemetry data types

1. Metrics

- Metrics are aggregated measurements that indicate how your service is performing. Common examples include the rate at which your service produces errors, the number of total requests your service sees within a given timeframe, and the time it takes your service to respond to a request.

- Each of these metric measurements are aggregated over some period of time, and the aggregated measurement is reported in your observability platform.

Telemetry data types

2. Events

- Events record data about things that happen in your service. For example, your service produces an error, or a user clicks a button on your web page.

- Instead of aggregating measurements over time, these record specific features of the environment at the time the event occured, like a stack trace, a user agent, or the button that the user clicked.

Telemetry data types

3. Logs

- Logs are the oldest and most basic of telemetry data types. They are arbitrary, timestamped text records, useful for debugging or understanding something specific about the system at a given time.

- For instance, you might log a message with some variable values from somewhere in your code that’s running slowly.

Telemetry data types

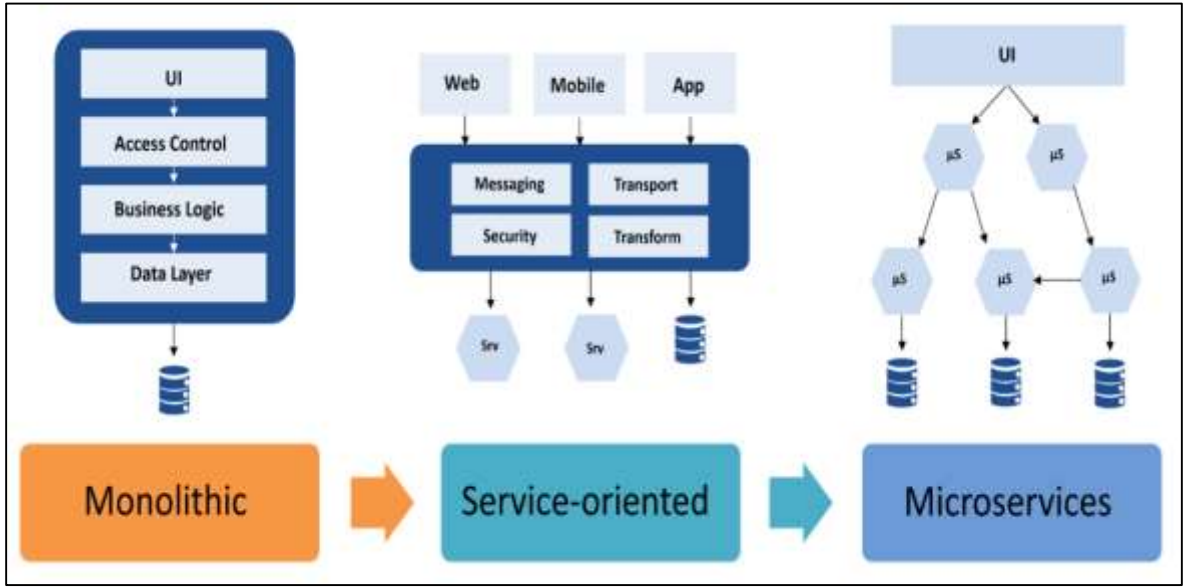
4. Traces

- Traces are the most modern telemetry data type, but also the most advanced of the bunch. A trace is a collection of hierarchically-related spans. Spans represent activities within applications.

- Each span contains timestamps for when it started and when it stopped. It contains attributes that describe features of the operation it represents, and it can even contain events that occur during the course of the operation. Traces are also effective at capturing details across network boundaries, which is important for monitoring modern computing systems.

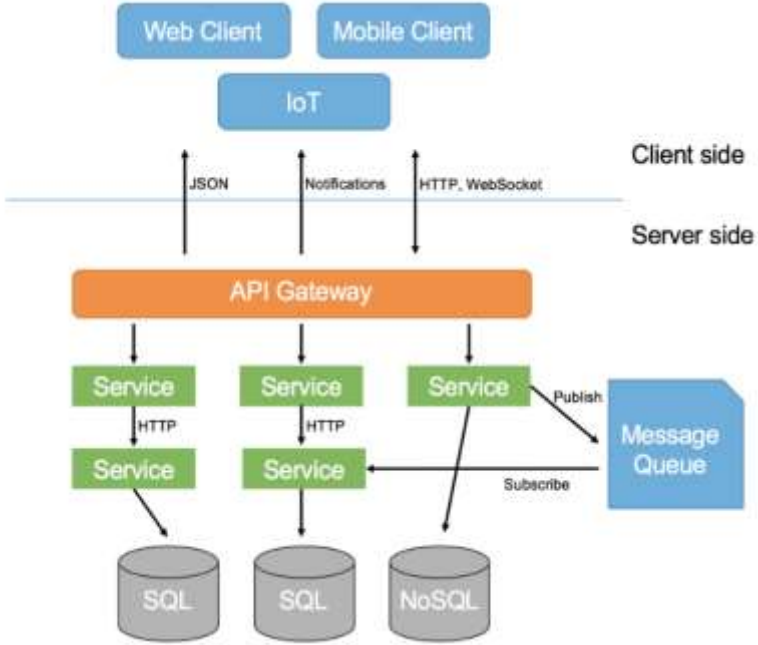
Why is Observability hard?

Evolution of Software Architectures



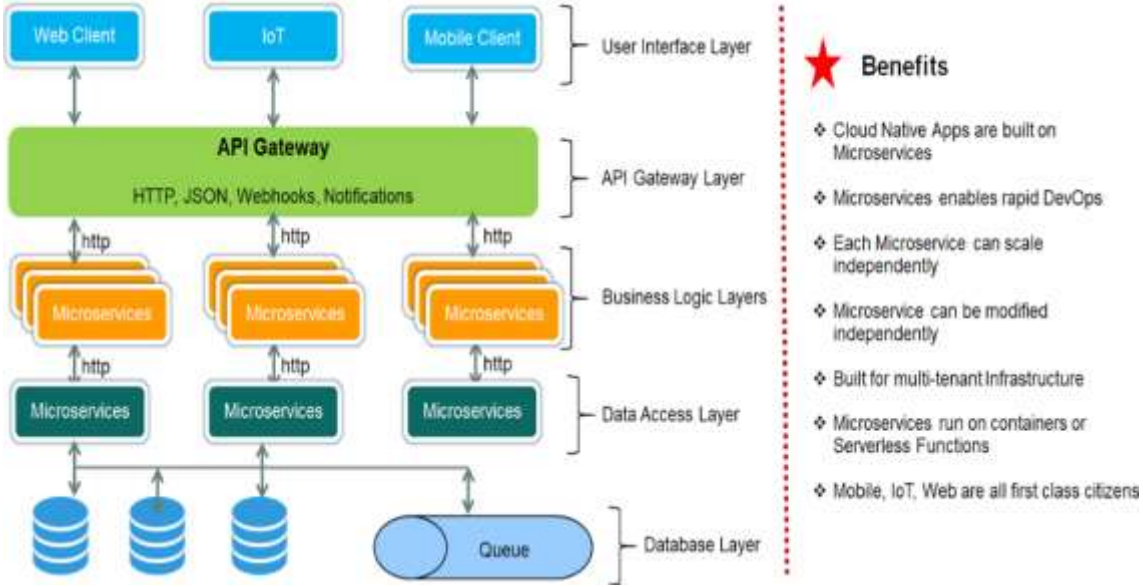
Why is Observability hard?

Service Oriented Architecture (SOA)



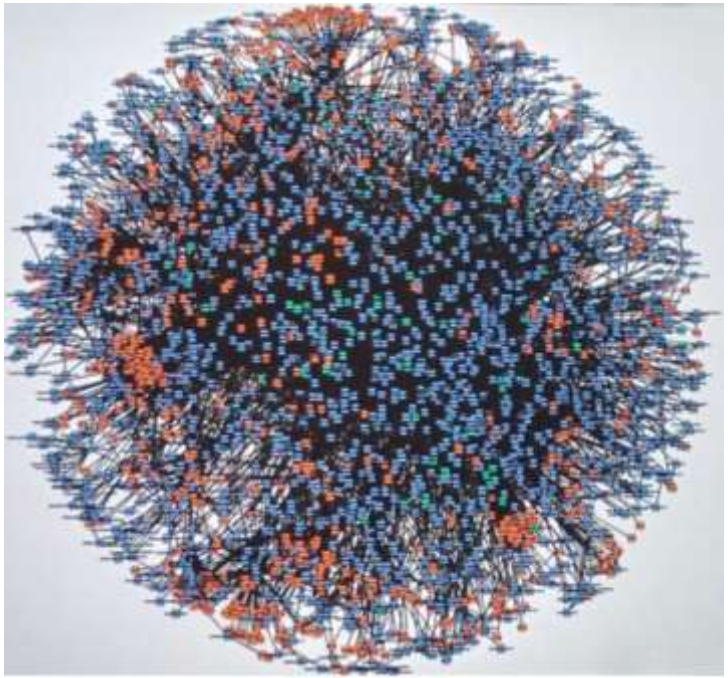
Why is Observability hard?

Software Architecture for Cloud Native Apps



Why is Observability hard?

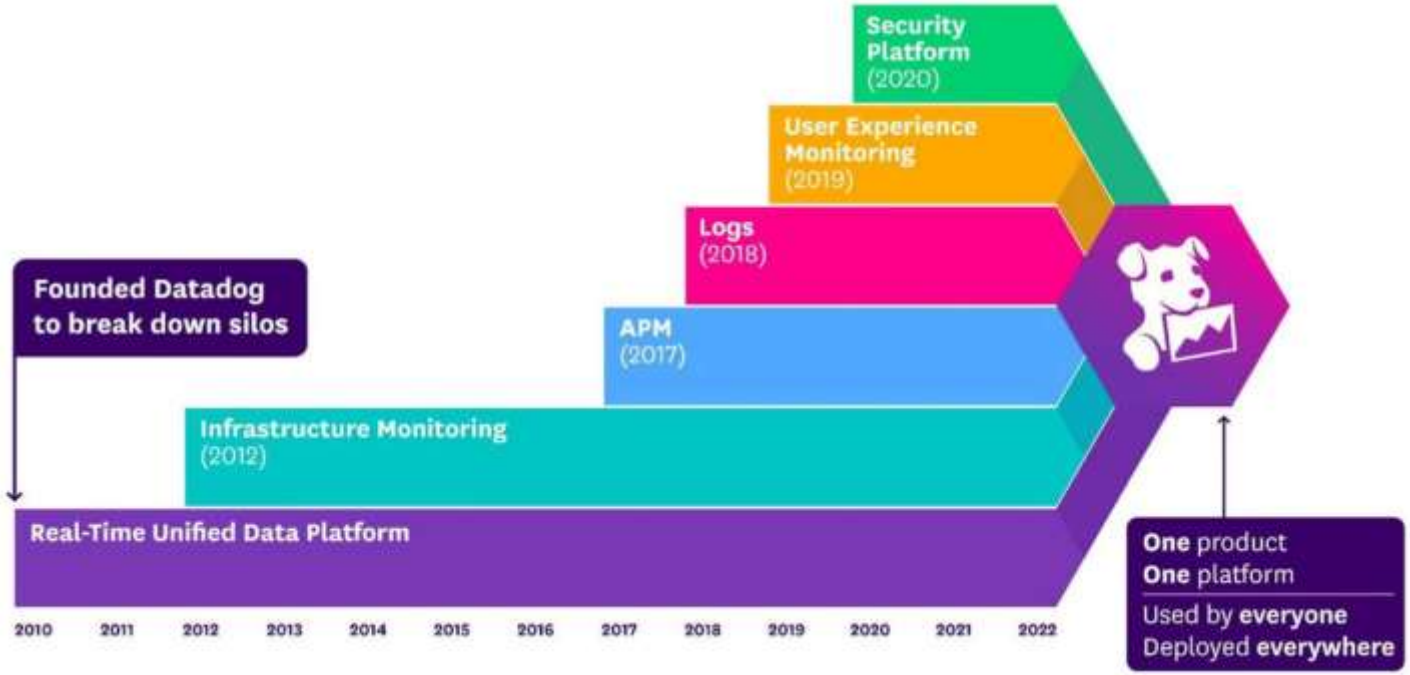
Too many Microservices



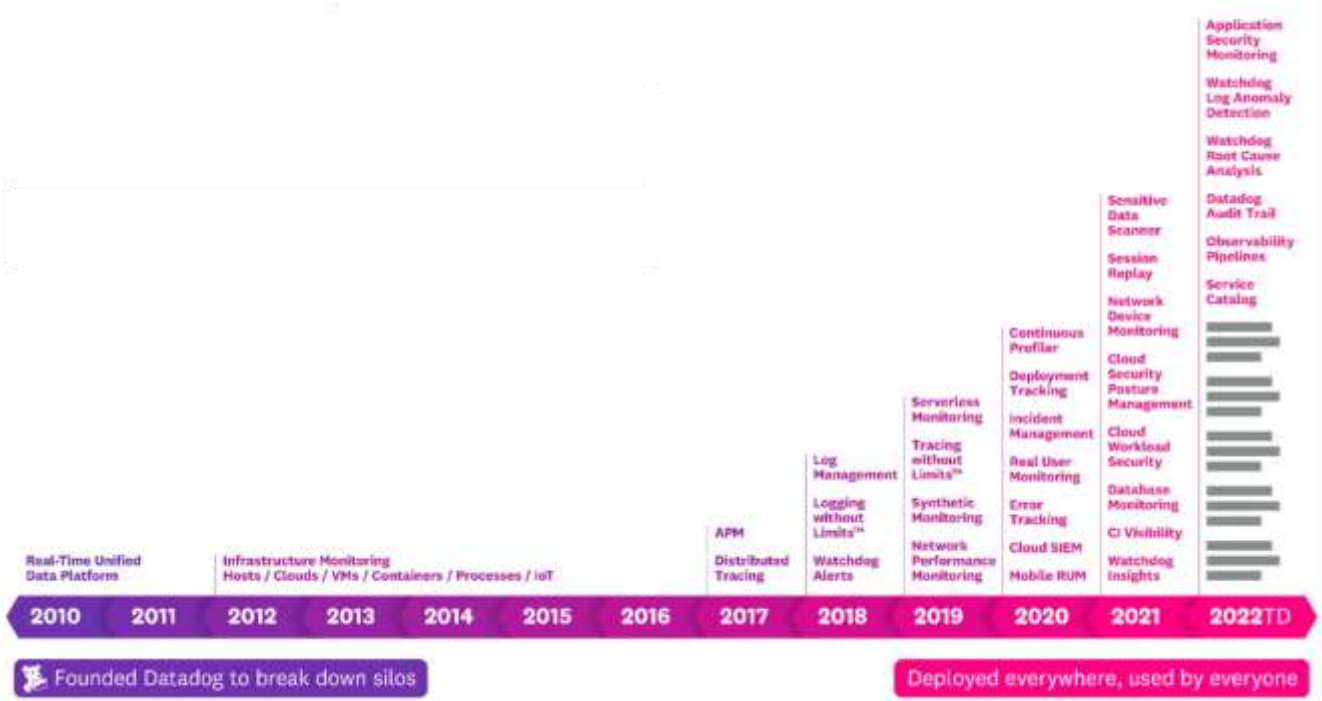
Datadog - Background

- They built Datadog to be a cloud infrastructure monitoring service, with a dashboard, alerting, and visualizations of metrics.
- Datadog is listed in Forbes' Cloud 100 and was ranked in the top ten fasted growing companies in North America in Deloitte's 2016 Fast 500 list.
- Datadog is founded in 2010 by Oliver Pomel and Alexis Le-Quoc.

Datadog – History of innovation



Datadog – History of innovation



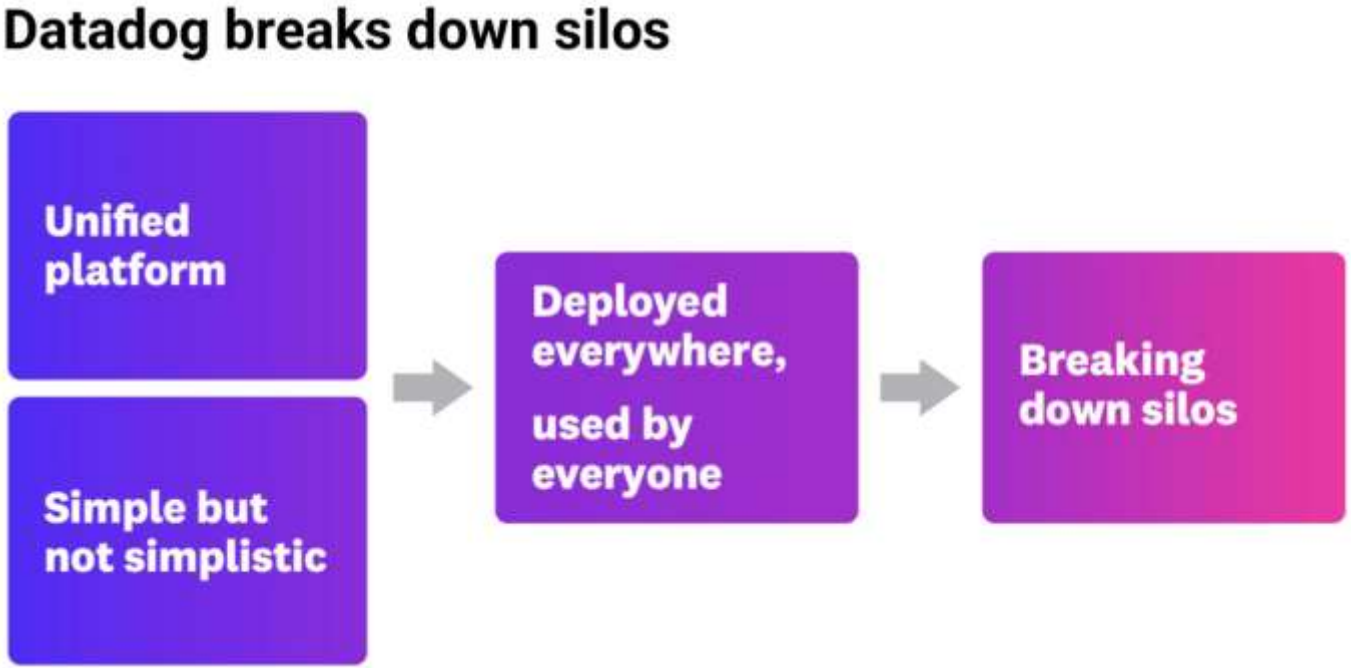
What is Datadog

- Datadog is a tool that allows you to monitor cloud infrastructure, Windows and Linux hosts, system processes, serverless functions and cloud-based applications. It can be used to visualize data, explore metrics, manage logs and perform various other tasks.
- Datadog is an observability service for cloud-scale applications, providing monitoring of servers, databases, tools and services through a SaaS-based data analytics platform.

What is Datadog

Datadog Platform

Datadog Platform



Datadog Platform

INDUSTRY RECOGNITION

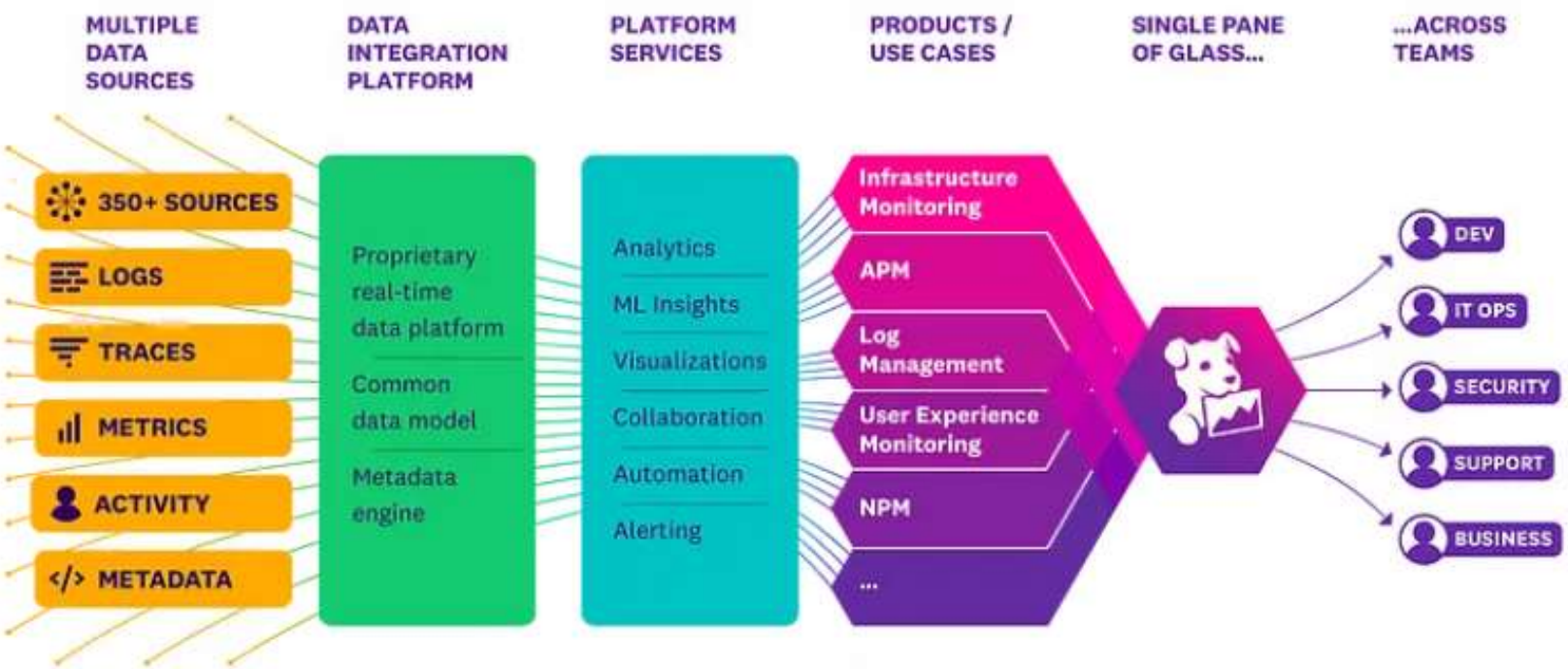
We were named a Leader in the 2022 Gartner Magic Quadrant for Application Performance Monitoring and Observability

DATADOG

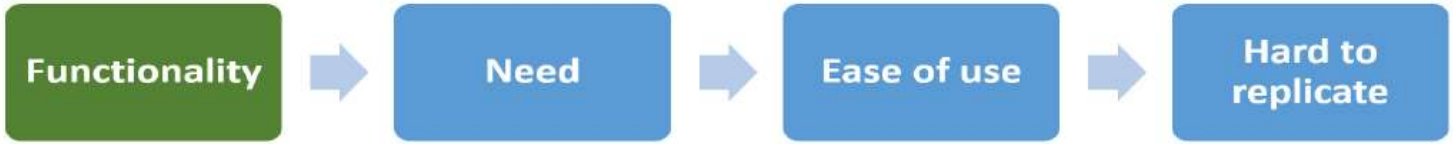
Figure 1: Magic Quadrant for Application Performance Monitoring and Observability

Source: Gartner (June 2022)

Why Datadog?


MULTIPLE DATA SOURCES
DATA INTEGRATION PLATFORM
PLATFORM SERVICES
PRODUCTS / USE CASES
SINGLE PANE OF GLASS...
...ACROSS TEAMS
350+ SOURCES
LOGS
TRACES
METRICS
ACTIVITY
METADATA
Proprietary real-time data platform
Common data model
Metadata engine
Analytics
ML Insights
Visualizations
Collaboration
Automation
Alerting
Infrastructure Monitoring
APM
Log Management
User Experience Monitoring
NPM
...
DEV
IT OPS
SECURITY
SUPPORT
BUSINESS

Why Datadog?



- **The agent** - It gathers system metrics, integrates with key software we use, and provides a standard interface to which our applications can send custom metrics.
- **Integrations-** Datadog has prebuilt integrations to pull data from almost every important service we use.
- **Events** - Through the integrations Datadog generates a consolidated event stream that we can filter and search as needed.
- **Dashboards** - Datadog lets us build dashboards that combine metrics from many different sources. We can combine and transform metrics to make them more useful. It also provides a powerful interface for interactive exploration of metrics.
- **Alerting** - Datadog has nice stream processing capabilities for generating alerts, and it can surface them in services we use like pagerduty and slack.

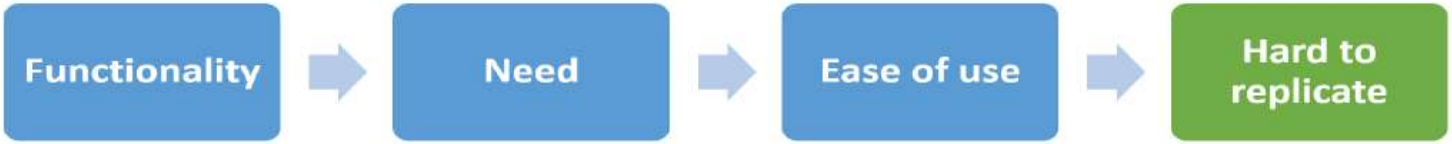
Why Datadog?

- **The agent** - We don't get nearly enough insight from cloudwatch alone, we need an on-instance tool to gather system and app metrics.
- **Integrations** - There are lots of services with operational significance, but many of them don’t provide a good way to access their data.
- **Events** - We would spend dramatically longer investigating problems if we had to look at each source of events in isolation. Many of our event sources don't even provide a way for us to view past events or to query them.
- **Dashboards** - Per-service and per-instance dashboards are important for investigating problems quickly. The consolidation of data from multiple sources is again a key feature.
- **Alerting** - We need to do anaylze trends in our metrics and alert on them.

Why Datadog?

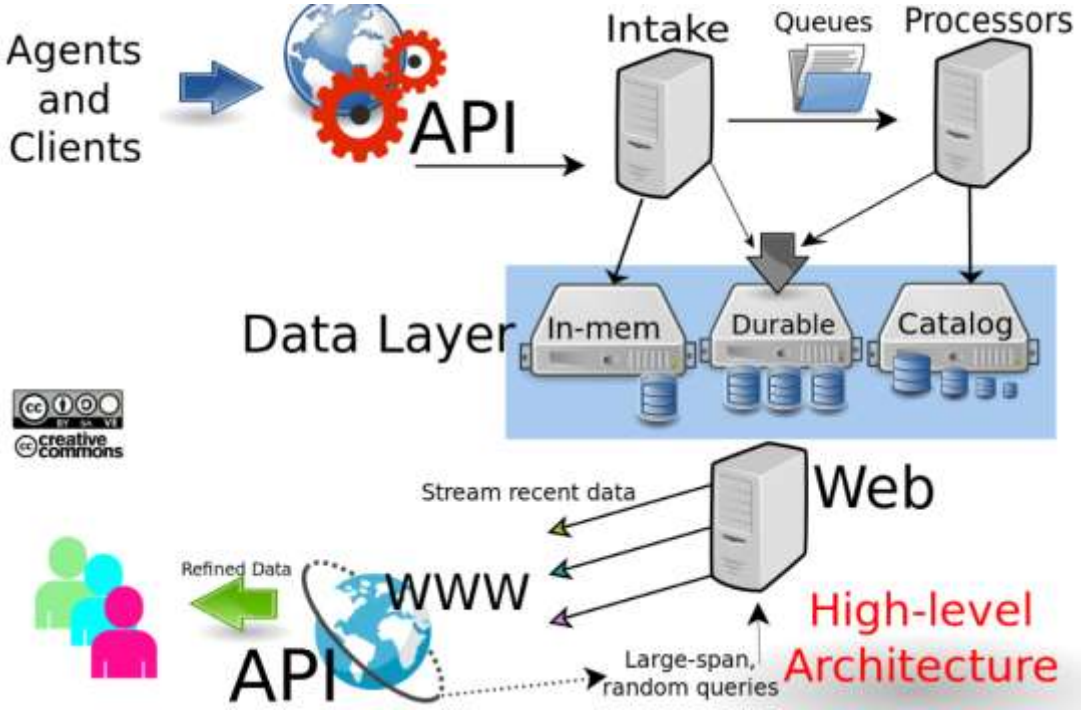
- **The agent** - The agent is deployable via a chef cookbook Datadog wrote for us. It requires minimal configuration. It knows which system and application metrics are worth gathering.
- **Integrations** - Integrating with all the data sources is literally a few clicks.
- **Events -** The interface makes searching and filtering events straightforward.
- **Dashboards** - There are prebuilt dashboards for lots of things we care about. Snazzy features like autocomplete and templating make building our own dashboards easy.
- **Alerting** - The guided steps and previewed outputs make creating alerts simple.

Why Datadog?



- Here I described a system of collectd, custom code to pull metrics from cloudwatch, custom code to pull or receive events from various sources (airbrake, cloudtrail, chef, pagerduty, jenkins, etc) influxdb, and grafana.

Datadog Architecture



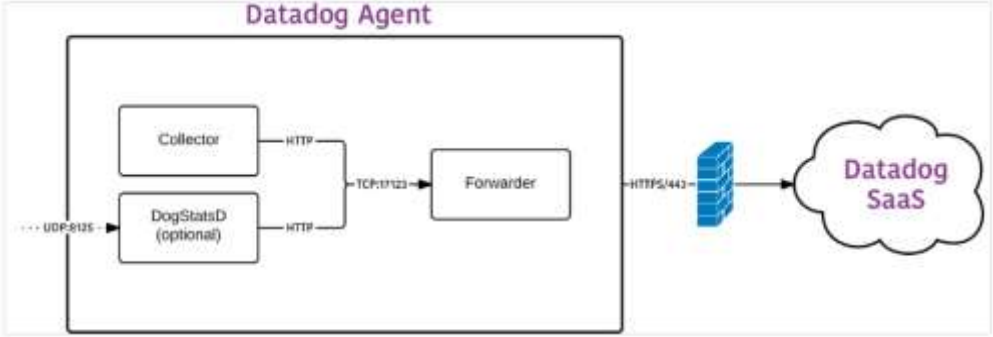
Datadog Agent

The Agent is lightweight software installed reports metrics and events from your host to Datadog (SaaS) via:

- Integrations (https://docs.datadoghq.com/integrations/)
- DogStatsD
- The API (https://docs.datadoghq.com/api/latest/)

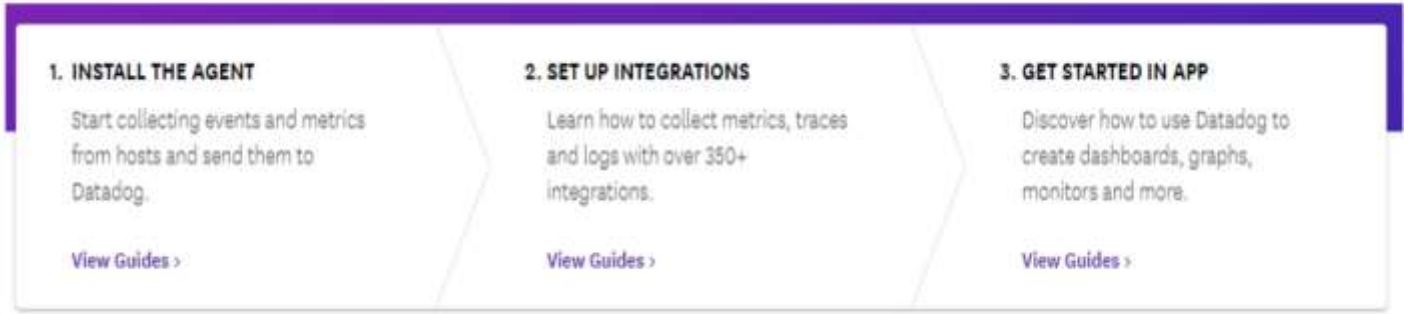
With additional steps, the agent can report live processes, logs and traces.

How Datadog Agent works?



- The **Agent collector** gathers all standard metrics, such as memory and CPU every 15 seconds
- **DogStasD:** This is a StatsD-compatible backend server that you can send custom metrics to from your applications.
- The **Agent forwarder:** The forwarder retrieves data from both DogStatsD and the collector, queues it for submission, and then sends it to Datadog.
- **SupervisorD**: The collector, DogStatsD server, and forwarder are all controlled by a single supervisor process. The supervisor is kept separate to limit the overhead of each application if you aren't running all parts. However, it is generally recommended to run all parts.

Datadog Workflow



Use cases of Datadog

1. Application Performance Monitoring (APM):
Monitor the performance of applications in real-time.
Identify and diagnose performance bottlenecks, errors, and latency issues.

2. Infrastructure Monitoring:
Collect and visualize metrics related to the health and performance of infrastructure components.
Monitor servers, containers, virtual machines, and other infrastructure elements.

3. Log Management:
Aggregate, search, and analyze logs from various sources in a centralized platform.
Correlate logs with performance metrics for efficient troubleshooting.

4. Network Monitoring:
Monitor network performance and track the flow of traffic between different components.
Identify network issues, latency, and potential bottlenecks.

Use cases of Datadog

5. Cloud Monitoring:
Monitor and optimize the performance of cloud services and resources.
Support for major cloud providers such as AWS, Azure, and Google Cloud Platform.

6. Security Monitoring and Threat Detection:
Detect and respond to security threats by monitoring abnormal behavior and identifying potential security incidents.
Integrate with security information and event management (SIEM) solutions.

7. User Experience Monitoring:
Monitor user interactions with applications to understand user experience.
Identify and resolve issues affecting user satisfaction and engagement.

8. Incident Response and Alerting:
Set up customizable alerts based on predefined thresholds and conditions.
Automate incident response workflows and collaborate on issue resolution.

Use cases of Datadog

9. DevOps and CI/CD Integration:
Integrate Datadog with CI/CD pipelines to monitor the impact of code changes on performance and reliability.
Support for popular CI/CD tools and platforms.

10. Business Analytics and Dashboards:
Create custom dashboards and reports to visualize key performance indicators and business metrics.
Publish insights with stakeholders and make data-driven decisions.

11. Container Orchestration Monitoring:
Monitor containerized environments, including orchestration platforms like Kubernetes and Docker.
Gain visibility into container health, resource usage, and performance.

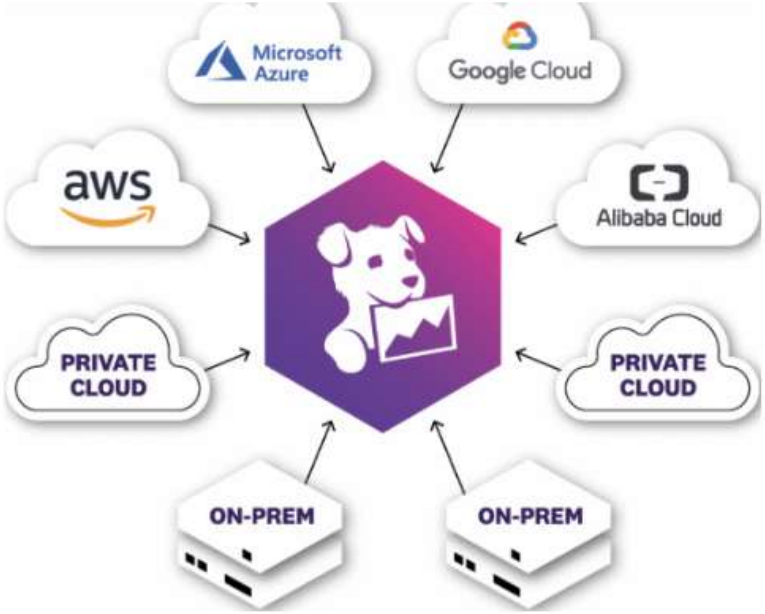
12. Serverless Monitoring:
Monitor and trace serverless functions to understand their performance and dependencies.
Support for serverless platforms like AWS Lambda.

What technologies Datadog Supports?

There are various technologies that Datadog supports such as AWS, Azure, GCP, Kubernetes, OpenShift and Pivotal platform.

- The app collects accurate system information, metrics and tags from more than 100 AWS services
- It supports more than 70 integrations with Azure services
- It collects all data from GCP through easy-to-install integrations
- It also offers a way to monitor and perform health checks on Kubernetes clusters

Cloud agnostic



Integration

Datadog - Integrations

An integration, at the highest level, is when you assemble a unified system from units that are usually considered separately.

At Datadog, you can use integrations to bring together all of the metrics and logs from your infrastructure and gain insight into the unified system as a whole – you can see pieces individually and also how individual pieces are impacting the whole.

Datadog - Integrations

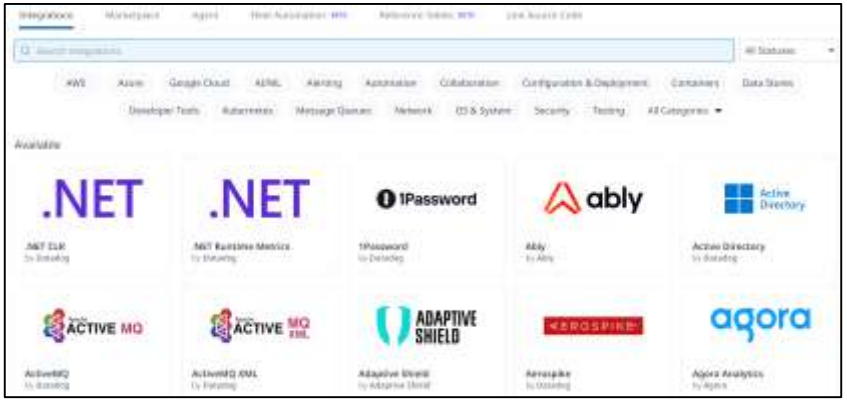
Datadog provides three main types on integrations:

Agent-based integrations are installed with the Datadog Agent and use a Python class method called check to define the metrics to collect.

Authentication (crawler) based integrations are set up in Datadog where you provide credentials for obtaining metrics with the API. These include popular integrations like Slack, AWS, Azure, and PagerDuty.

Library integrations use the Datadog API to allow you to monitor applications based on the language they are written in, like Node.js or Python.

Datadog - Integrations



- Datadog has over 600 integrations officially listed.
- Custom integrations are available through the Datadog API
- The Agent is open source
- One integrations have been configured, all data is treated the same throughout Datadog, whether it is living in a datacenter or in an online service

Datadog - Integrations

1. Apache:

The Apache check tracks requests per second, bytes served, number of worker threads, service uptime and more

Installation:

Install the Agent on your Apache Servers

Install mod_status on your Apache servers and enable ExtendedStatus

Datadog - Integrations

2. Tomcat:

This check collects Tomcat metrics, for example:

- Overall activity metrics: error count, request count, processing times etc
- Thread pool metrics: thread count, number of threads busy etc
- Servlet processing times

Installation:

- The Tomcat check is included in the Datadog Agent package. So no need to install anything else on your Tomcat servers
- This check is JMX-based so you need to enable JMX Remote on your Tomcat servers.

Infrastructure

Infrastructure Lists

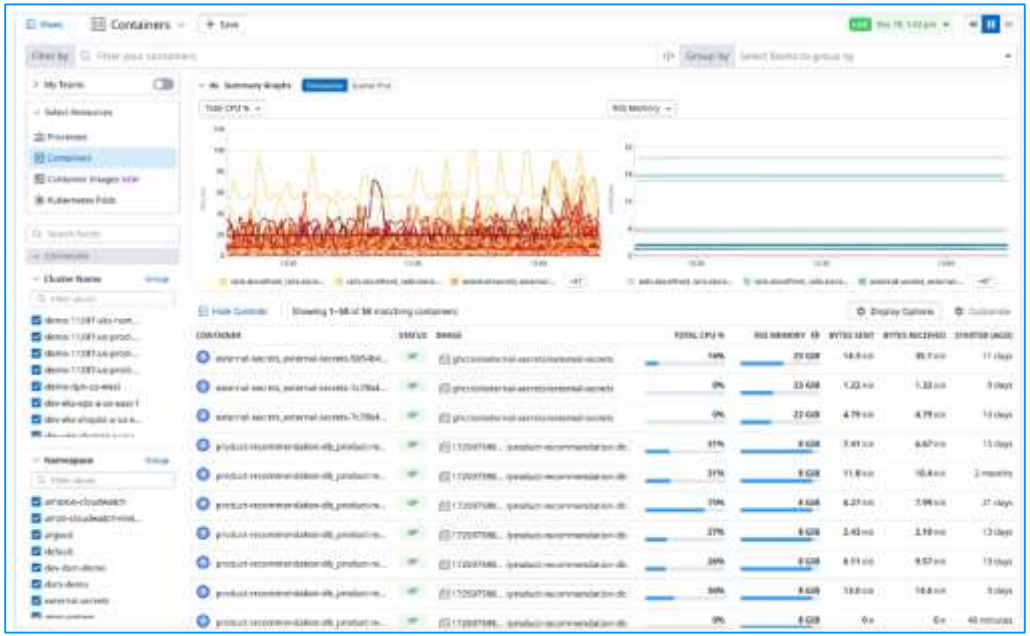
- The Infrastructure list shows all of your hosts monitored by Datadog with activity during the last two hours (default) and up to one week. Search your hosts or group them by tags. In Datadog, navigate to Infrastructure > Hosts to view the Infrastructure list.
- The following information is displayed in the infrastructure list for your hosts:
  - Hostname
  - Cloud Name
  - Instance ID
  - Status
  - CPU
  - IOWait
  - Load 15
  - Apps
  - Operating System
  - Cloud Platform
  - Datadog Agent

Containers

- The containers page provides real-time visibility into all containers across your environment.
- Taking inspiration from bedrock tools like htop, ctop and kubectl, the containers page gives you complete coverage of your container infrastructure in a continuously updated table with resource metrics at two-second resolution, faceted search and streaming container logs.

Containers

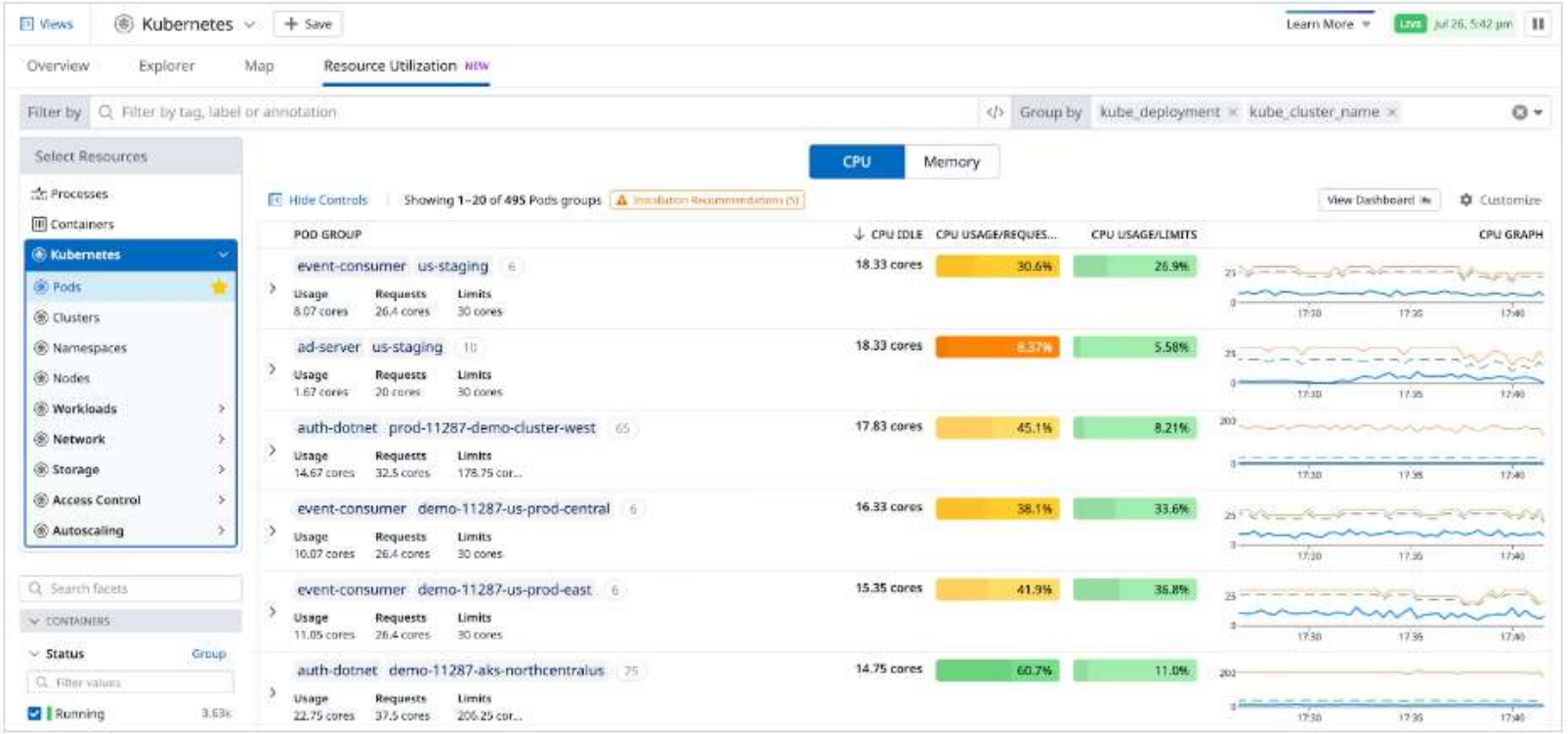
Coupled with Docker, Kubernetes, ECS, and other container technologies, plus built-in tagging of dynamic components, the Containers page provides a detailed overview of your containers' health, resource consumption, logs, and deployment in real-time:



Kubernetes

- The Datadog agent and Cluster agent can retrieve Kubernetes resources for the Orchestrator Explorer.
- This feature allows you to monitor the state of pods, deployments and other Kubernetes concepts in a specific namespace and availability zone, view resource specifications for failed pods within a deployment, correlate node activity with related logs and more.

Kubernetes


Views
Kubernetes
+ Save
Learn More
Live
Overview
Explorer
Map
Resource Utilization NEW
Filter by
Filter by tag, label or annotation
Group by
kube_deployment
kube_cluster_name
Select Resources
Processes
Containers
Kubernetes
Pods
Clusters
Namespaces
Nodes
Workloads
Network
Storage
Access Control
Autoscaling
Search facets
CONTAINERS
Status
Group
Filter values
Running
3.63k
CPU
Memory
Hide Controls
Showing 1–20 of 495 Pods groups
View Dashboard
Customize
POD GROUP
CPU IDLE
CPU USAGE/REQUES...
CPU USAGE/LIMITS
CPU GRAPH
event-consumer us-staging 6
18.33 cores
30.6%
26.9%
Usage 8.07 cores
Requests 26.4 cores
Limits 30 cores
ad-server us-staging 10
18.33 cores
8.37%
5.58%
Usage 1.67 cores
Requests 20 cores
Limits 30 cores
auth-dotnet prod-11287-demo-cluster-west 65
17.83 cores
45.1%
8.21%
Usage 14.67 cores
Requests 32.5 cores
Limits 178.75 cor...
event-consumer demo-11287-us-prod-central 6
16.33 cores
38.1%
33.6%
Usage 10.07 cores
Requests 26.4 cores
Limits 30 cores
event-consumer demo-11287-us-prod-east 6
15.35 cores
41.9%
36.8%
Usage 11.05 cores
Requests 26.4 cores
Limits 30 cores
auth-dotnet demo-11287-aks-northcentralus 75
14.75 cores
60.7%
11.0%
Usage 22.75 cores
Requests 37.5 cores
Limits 206.25 cor...

Serverless

- Datadog Serverless Monitoring provides full visibility into all of the managed services that power your serverless applications by bringing together real-time metrics, logs and traces from your serverless compute as well as related fully-managed APIs, queues, streams and data stores.

- Datadog provides solutions for monitoring AWS Lambda, Azure App Service, Azure Container Apps, and Google Cloud Run.

Processes

Datadog's Live Processes gives you real-time visibility into the processes running on your infrastructure. Use Live Processes to:

- View all of your running processes in one place
- Break down the resource consumption on your hosts and containers at the process level
- Query for processes running on a specific host, in a specific zone, or running a specific workload
- Monitor the performance of the internal and third-party software you run using system metrics at two-second granularity
- Add context to your dashboards and notebooks

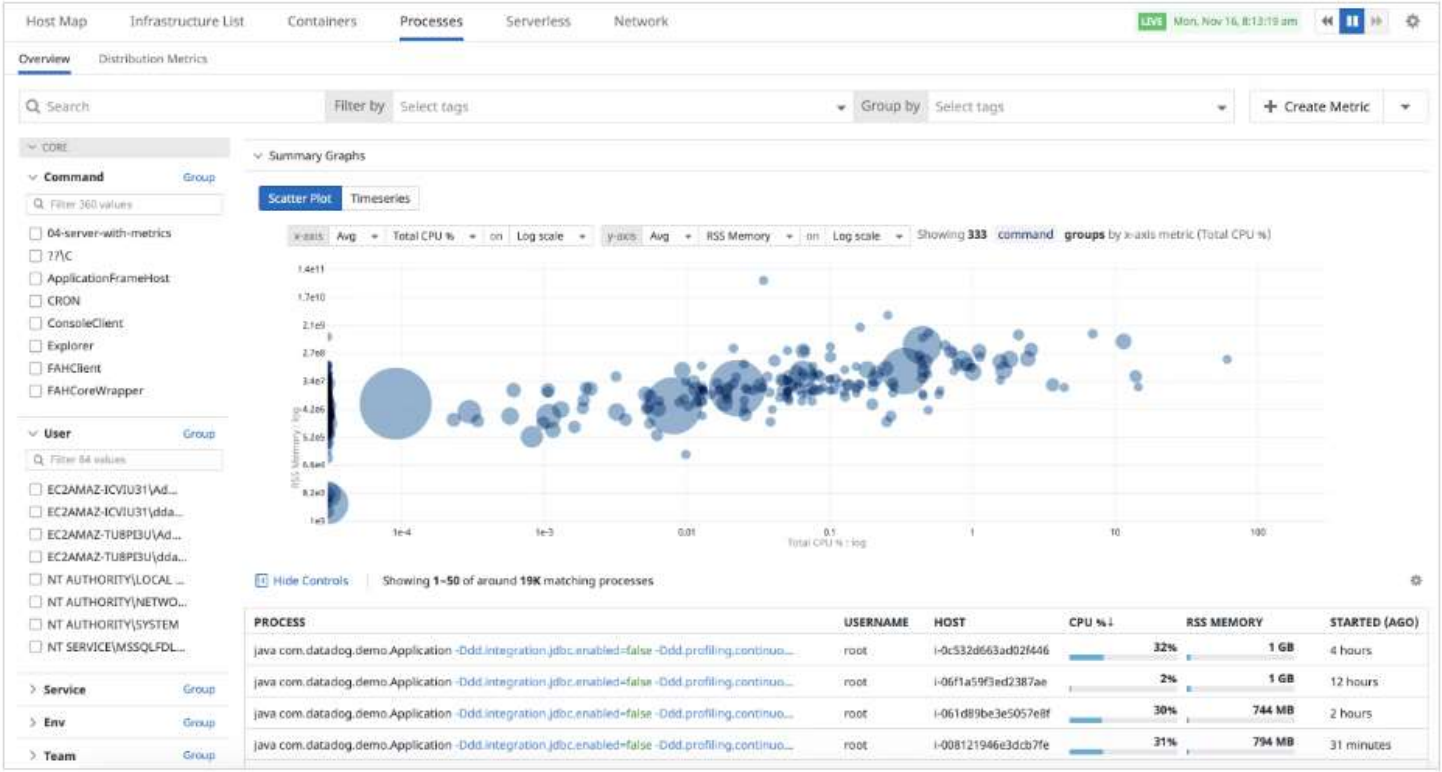
Processes

While live processes data is stored for 36 hours, you can generate global and percentile distribution metrics from your processes to monitor your resource consumption long-term. Process based metrics are stored for 15 months like any other Datadog metrics.

This can help you with:

- Debug past and ongoing infrastructure issues
- Identify trends in the resource consumption of your critical workloads
- Assess the health of your system before and after load or stress tests
- Track the effect of software deployments on the health of your underlying hosts or containers.

Processes


Host Map
Infrastructure List
Containers
Processes
Serverless
Network
Overview
Distribution Metrics
Search
Filter by
Select tags
Group by
Select tags
Create Metric
CORE
Command
Group
04-server-with-metrics
??\C
ApplicationFrameHost
CRON
ConsoleClient
Explorer
FAHClient
FAHCoreWrapper
User
Group
Service
Group
Env
Group
Team
Group
Summary Graphs
Scatter Plot
Timeseries
Total CPU %
RSS Memory
Log scale
Showing 333 command groups by x-axis metric (Total CPU %)
Total CPU % : log
Hide Controls
Showing 1~50 of around 19K matching processes
PROCESS
USERNAME
HOST
CPU %
RSS MEMORY
STARTED (AGO)
java com.datadog.demo.Application -Ddd.integration.jdbc.enabled=false -Ddd.profiling.continuo...
root
i-0c532d663ad02f446
32%
1 GB
4 hours
java com.datadog.demo.Application -Ddd.integration.jdbc.enabled=false -Ddd.profiling.continuo...
root
i-06f1a59f3ed2387ae
2%
1 GB
12 hours
java com.datadog.demo.Application -Ddd.integration.jdbc.enabled=false -Ddd.profiling.continuo...
root
i-061d89be3e5057e8f
30%
744 MB
2 hours
java com.datadog.demo.Application -Ddd.integration.jdbc.enabled=false -Ddd.profiling.continuo...
root
i-008121946e3dcb7fe
31%
794 MB
31 minutes

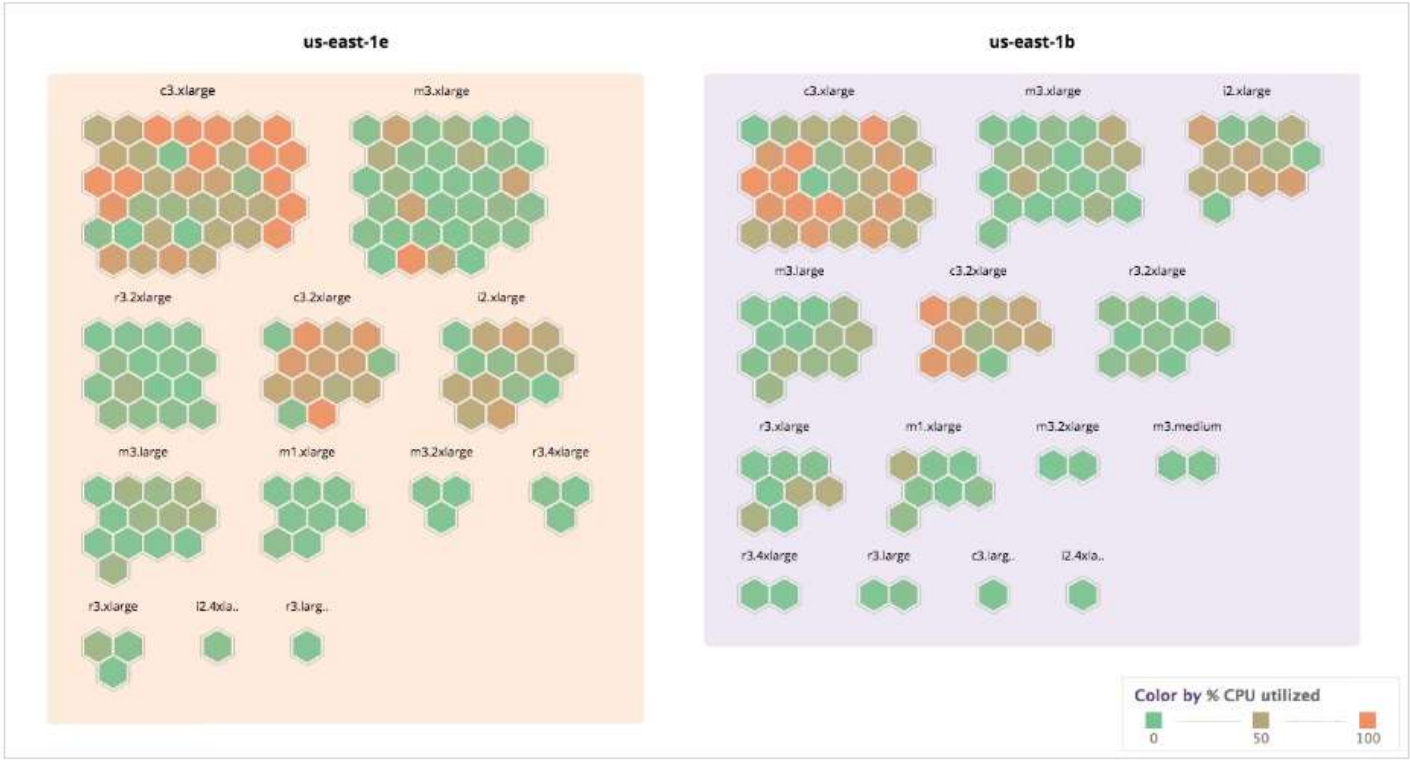
Host-Map

Host Map

The host map can be found under the infrastructure menu. It offers the ability to:

- Quickly visualize your environment
- Identify outliers
- Detect usage patterns
- Optimize resources

Host Map

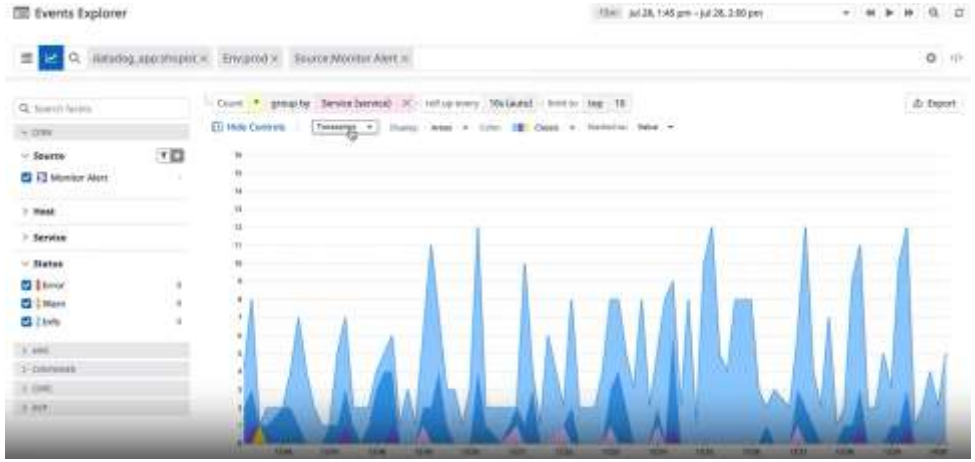

us-east-1e
c3.xlarge
m3.xlarge
r3.2xlarge
c3.2xlarge
i2.xlarge
m3.large
m1.xlarge
m3.2xlarge
r3.4xlarge
r3.xlarge
i2.4xla..
r3.larg..
us-east-1b
c3.xlarge
m3.xlarge
i2.xlarge
m3.large
c3.2xlarge
r3.2xlarge
r3.xlarge
m1.xlarge
m3.2xlarge
m3.medium
r3.4xlarge
r3.large
c3.larg..
i2.4xla..
Color by % CPU utilized
0
50
100

Event

Events

- Events are records of notable changes relevant for managing and troubleshooting IT operations such as code deployments, service health, configuration changes or monitoring alerts.
- Datadog events give you a consolidated interface to search, analyze and filter events from any source in one place.
- Without any additional setup, Datadog events automatically gather events that are collected by the Agent and installed integrations
- More than 100 Datadog integrations support event collection, including Kubernetes, Docker, Jenkins, Chef, Puppet, AWS ECS or Autoscaling, Sentry and Nagios.

Events



The Event Explorer displays the most recent events generated by your infrastructure and services.

Events can include the following:

- Code deployments
- Service health changes
- Configuration changes
- Monitoring alerts

Dashboards

Dashboards

- Dashboards provide real-time insights into the performance and health of systems and applications within an organization.
- They allow users to visually analyze data, track key performance indicators (KPIs), and monitor trends efficiently.
- With dashboards, teams can identify anomalies, prioritize issues, proactively detect problems, diagnose root causes, and ensure that reliability goals are met.
- Empower your teams to make informed decisions, optimize system operations, and drive business success by providing a centralized and user-friendly interface for monitoring and analyzing critical metrics and performance indicators.

Dashboards

- Features

Configure: Overview of the configuration options for dashboards
Dashboard List: Search, view, or create dashboards and lists
Template Variable: Dynamically filter widgets in a dashboard
Datadog Clipboard
API: Manage dashboards programmatically

Dashboards

Dashboards contain graphs with real-time performance metrics.

- Synchronous mousing across all graphs in a screen-board
- Vertical bars are events. They put a metric into context.
- Click and drag on a graph to zoom in on a particular timeframe.
- As you hover over the graph, the event stream moves with you.
- Display by zone, host or total usage.
- Datadog exposes a JSON editor for the graph, allowing for arithmetic and functions to be applied to metrics.
- Share a graph snapshot that appears in the stream.
- Graphs can be embedded in a iframe. This enables you to give a third party access to a live graph without also giving access to your data or any other information.

Hands On Lab

Learning Objectives:

Do the following:

- Understand how to use the lab environment.
- Log in to Datadog with the account provisioned for you.
- Ensure you are in the correct Datadog account.
- Troubleshoot lab issues.

Hands On Lab

Learning Objectives:

Do the following:

- Explore dashboards from dashboards list
- Search and visualize log data in Logs
- Find application service details in Service Catalog
- Interpret a monitor in Manage Monitors

Datadog Agent

- Overview
- Datadog Agent
- Usage (Docker, Kubernetes, Cluster Agent)
- Log Collection
- Proxy
- Versions
- Troubleshooting

Overview

What is Datadog Agent

- The Datadog Agent is the software that runs on your hosts
- It collets events and metrics from hosts and send them to Datadog, where you can analyze your monitoring and performance data
- It can run on your local hosts (Windows, MacOS), Containerized environments (Docker, Kubernetes) and in on-premises data centers.
- You can install and configure it using configuration management tools (Chef, Puppet, Ansible)
- The Agent is able to collect 75 to 100 system level metrics every 15 to 20 seconds.
- With additional configuration, the Agent can also send live data, logs and traces from running processes to the Datadog Platform

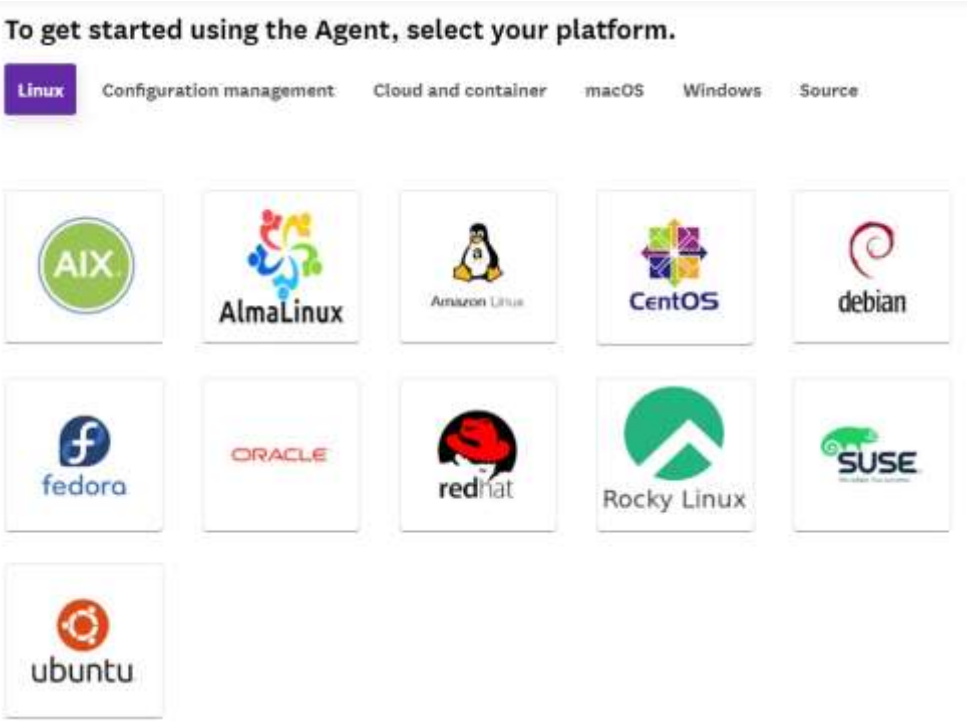
Datadog Agent

The Agent is lightweight software installed reports metrics and events from your host to Datadog (SaaS) via:

- Integrations (https://docs.datadoghq.com/integrations/)
- DogStatsD
- The API (https://docs.datadoghq.com/api/latest/)

With additional steps, the agent can report live processes, logs and traces.

Datadog Agent: Platforms



StatsD & Datadog

- Datadog is a big fan of StatsD and use it extensively internally
- The Agent forwarder sends metrics over HTTPS to Datadog
- DogStatsD is a Golang implementation Etsy's StatsD metrics aggregation daemon

StatsD is originally a simple **daemon** developed and released by Etsy to aggregate and summarize application metrics.

With StatsD, applications are to be instrumented by developers using language specific client libraries. These libraries will then communicate with the StatsD daemon using its dead-simple protocol, and the daemon will then generate aggregate metrics and relay them to virtually any graphing or monitoring backend.

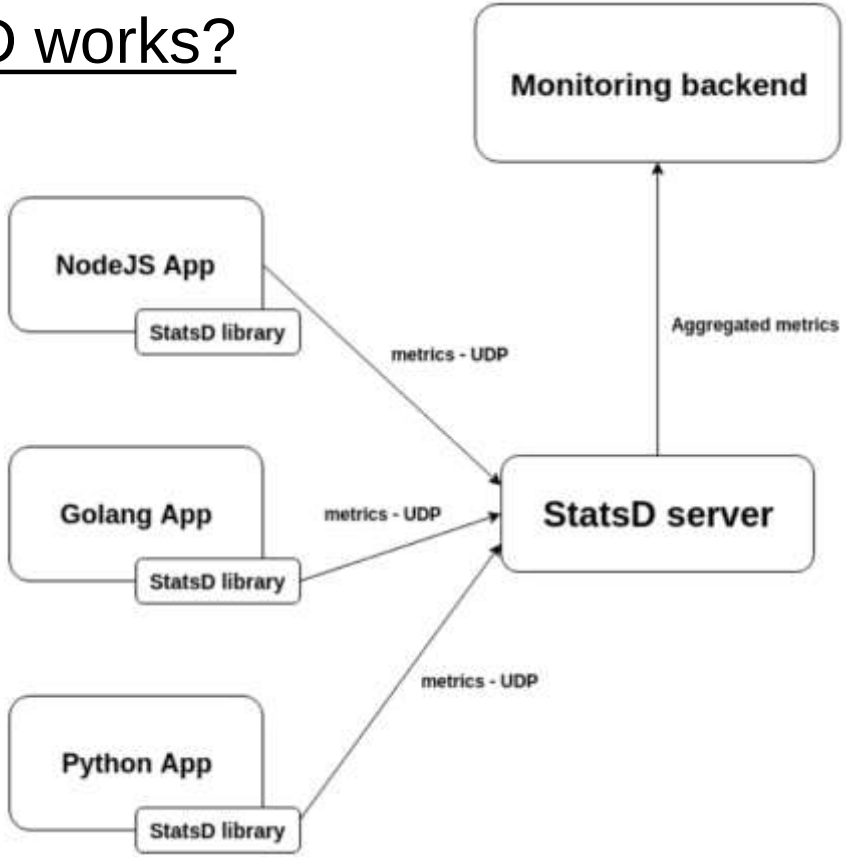
How StatsD works?

- **It all starts in your own application code**. You—the developer—instrument it with one of the many **StatsD** libraries corresponding to your app language. **StatsD** allows you to capture different types of metrics depending on your needs: today those are Gauges, Counters, Timing Summary Statistics, and Sets. This can be as simple as adding a decorator to methods you want to time, or a one-liner to track a gauge value.
- **The StatsD client library** then sends each individual call to the **StatsD** server over a UDP datagram. Since UDP is a disconnected protocol in which the recipient of a datagram doesn't send any acknowledgment to the sender, the library doesn't need to block when submitting data as it would with TCP or HTTP-based protocols. The library also doesn't buffer any data in-between calls which keeps it very simple. It does let you optionally sample the events to be sent to the server if you happen to instrument very high-throughput operations.

How StatsD works?

- **The StatsD daemon** will then listen to the UDP traffic from all application libraries, aggregate data over time and “flush”•it at the desired interval to the backend of your choice. **For example**, individual function call timings may be aggregated every 10 seconds into a set of summary metrics describing its minimum, maximum, median, 90th and 95th percentile over the 10s interval. The protocol used between the StatsD Daemon and the backend will vary depending on the backend used (most are HTTP-based).
- **The monitoring backend** will turn your metrics from a stream of numbers on the wire into usable charts and alert you when needed. Examples of backends include tools like Graphite as well as yours truly.

How StatsD works?



How StatsD works?

- Datadog is a big fan of StatsD and use it extensively internally. Customers to submit metrics from StatsD into Datadog for graphing, alerting, event correlation and team collaboration. Datadog embedded our own StatsD daemon within the Datadog agent, to make the setup as simple as possible.

- Datadog extended the StatsD protocol to support tagging, one of Datadog's killer features. This lets you add additional dimensions to your metrics, such as the application version, or type of customer a specific call related to.

- Datadog made it very easy to discover StatsD metrics in the Datadog UI. Every host will automatically advertise its metrics, so you do not have to look for them.

Datadog Agent: DogStatsD

- While StatsD accepts only **metrics**, DogStatsD accepts all three of the major Datadog data types: **metrics, events** and **service checks**.

- In v6, DogStatsD is a Golang implementation of Etsy's StatsD metric aggregation daemon. It is used to receive and roll up arbitrary metrics over UDP or Unix socket, thus allowing custom code to be instrumented without adding latency.

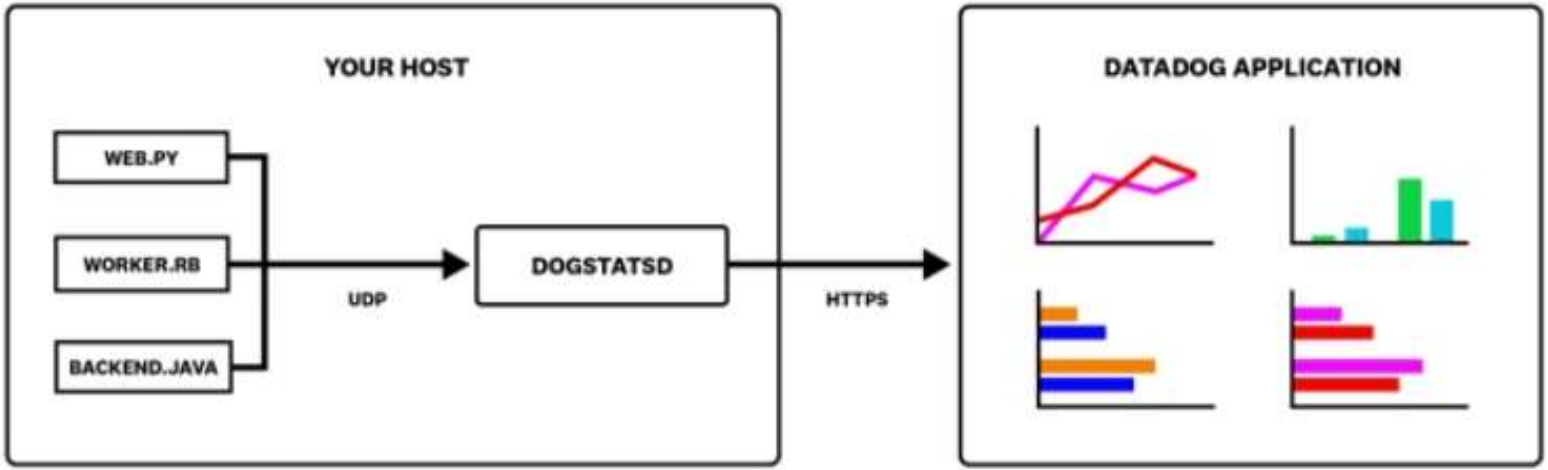
Datadog Agent: DogStatsD

- DogStatsD accepts custom metrics, events and service checks over UDP and periodically aggreagtes and forwards them to Datadog.

- Because it uses UDP, your application can send metrics to DogStatsD and resume its work without waiting for a response. If DogStatsD ever becomes unavailable, your application won’t experience an interruption.

- As it receives data, DogStatsD aggregates multiple data points for each unique metrics into a single data point over a period of time called the flush interval (10 seconds, by default)

Datadog Agent: DogStatsD



Datadog vs StatsD

- The easiest way to get your custom application into Datadog is to send them to **DogStatsD**, a metric aggregation service bundled with the Datadog Agent.

- DogStatsD implements the StatsD protocol and adds a few Datadog-specific extensions:
  - Histogram metric type
  - Service checks
  - Events
  - Tagging

- Any compliant StatsD client works with DogStatsD and the Agent, but you won't be able to use the Datadog-specific extensions.

Datadog Agent: DogStatsD

- DogStatsD is enabled by default over UDP port **8125** for Agent v6+
- By default, DogStatsD listens on UDP port **8125**
- You can also configure DogStatsD to use a Unix domain socket. To enable a custom Agent DogStatsD server UDP port.

Edit the datadog.yaml Configuration File

For Ubuntu:

Shell

```
sudo vi /etc/datadog-agent/datadog.yaml
```

Change the following lines in the datadog.yaml file

YAML

```
# DogStatsd
#
# If you don't want to enable the DogStatsd server, set this option to no
use_dogstatsd: yes
#
# Make sure your client is sending to the same UDP port
dogstatsd_port: 8125
```

Datadog Agent Collector and Forwarder

The Collector:

- The collector gathers all standard metrics every 15 seconds. Agent v6 embeds a Python 2.7 interpreter to run integrations and custom checks

The Forwarder:

- The Agent forwarder sends metrics over HTTPS to Datadog. Buffering prevents network splits from affecting metrics reporting.
- Metrics are buffered in memory until a limit in size or number of outstanding send requests are reached.
- Afterwards, the oldest metrics are discarded to keep the forwarder’s memory footprint manageable. Logs are sent over and SSL-encrypted TCP connection to Datadog

Datadog Agent UI

- You can configure the port on which the GUI runs in the **datadog.yaml** file.
- To disable the GUI, set the port's value to -1.
- For windows and macOS, the GUI is enabled by default and runs on port 5002.
- For Linux, the GUI is disabled by default.
- When the Agent is running, use the datadog-agent launch-gui command to open the GUI in your default web browser.

Datadog Agent UI: Requirements

- Cookies must be enabled in your browser. The GUI generates and saves a token in your browser which is used for authenticating all communications with the GUI server.

- To start the GUI, the user must have the required permissions. If you are able to open datadog.yaml, you are able to use the GUI

- For security reasons, the GUI can only be accessed from the local network interface (localhost/127.0.0..1), therefore you must be on the same host that the Agent is running. That is, you can’t run the Agent on a VM or a container and access it from the host machine.

Datadog Agent services in Windows

On Windows the services are listed as

SERVICE	DESCRIPTION
DataDogAgent	“Datadog Agent”
Datadog-trace-agent	“Datadog Trace Agent”
Datadog-process-agent	“Datadog Process Agent”

Datadog Agent ports in Windows and OSX

By default the Agent binds 3 ports on Linux and 4 on Windows and OSX

Port	Description
5000	Exposed runtime metrics about the Agent
5001	Used by the Agent CLI and GUI to send commands and pull information from the running Agent.
5002	Servers the GUI server on Windows and OSX
8125	Used for the DogStatsD server to receive external metrics

Datadog Agent : CLI

Some useful commands below:

DESCRIPTION	COMMAND
Start Agent as a service	`sudo service datadog-agent start`
Stop Agent running as a service	`sudo service datadog-agent stop`
Restart Agent running as a service	`sudo service datadog-agent restart`
Status of Agent service	`sudo service datadog-agent status`
Status page of running Agent	`sudo datadog-agent status`
Send flare	`sudo datadog-agent flare`
Display command usage	`sudo datadog-agent --help`
Run a check	`sudo -u dd-agent -- datadog-agent check <CHECK_NAME>`

Datadog Agent : Configuration files

Main configuration file

The location of the Agent configuration file differs depending on the operating system.

PLATFORM	COMMAND
AIX	`/etc/datadog-agent/datadog.yaml`
Linux	`/etc/datadog-agent/datadog.yaml`
macOS	`~/.datadog-agent/datadog.yaml`
Windows	`%ProgramData%\Datadog\datadog.yaml`

Datadog Agent : Configuration files

Agent configuration directory

Configuration files for Agent checks and integrations are stored in the `conf.d` directory. The location of the directory differs depending on the operating system.

PLATFORM	COMMAND
AIX	`/etc/datadog-agent/conf.d/`
Linux	`/etc/datadog-agent/conf.d/`
CentOS	`/etc/datadog-agent/conf.d/`
Debian	`/etc/datadog-agent/conf.d/`
Fedora	`/etc/datadog-agent/conf.d/`
macOS	`~/.datadog-agent/conf.d/`
RedHat	`/etc/datadog-agent/conf.d/`
Source	`/etc/datadog-agent/conf.d/`
Suse	`/etc/datadog-agent/conf.d/`
Ubuntu	`/etc/datadog-agent/conf.d/`
Windows	`%ProgramData%\Datadog\conf.d`

Datadog Agent : Configuration files

Check Configuration Files

An example for each Agent check configuration file is found in the conf.yaml.example file in the corresponding <CHECK_NAME>.d/ folder. Rename this file to conf.yaml to enable the associated check.

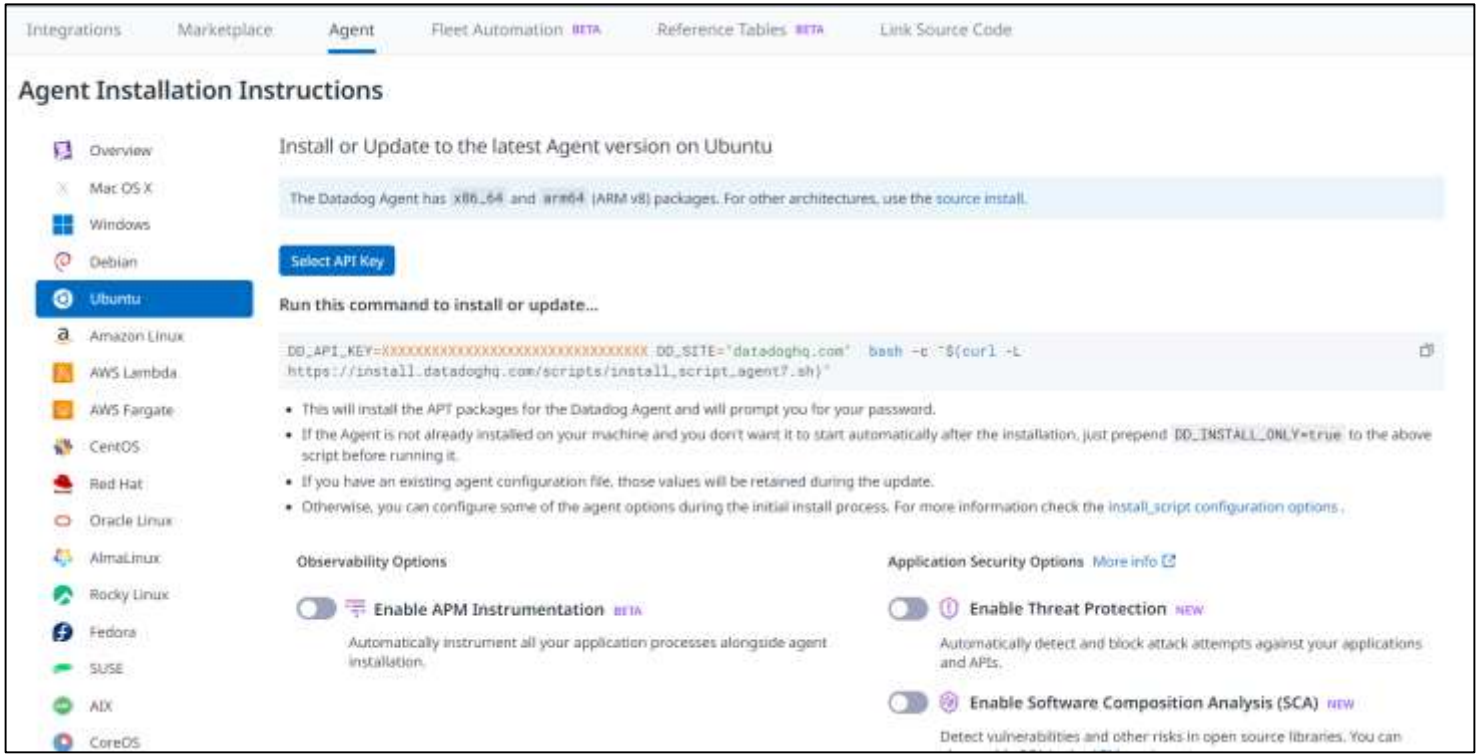
Note: The Agent loads valid YAML files contained in the folder: /etc/datadog-agent/conf.d/<CHECK_NAME>.d/. This allows complex configurations to be broken down into multiple files. For example, a configuration for the http_check might look like this:

```
/etc/datadog-agent/conf.d/http_check.d/
├── backend.yaml
└── frontend.yaml
```

How to get Datadog Agent?

- The Datadog Agent is open source and its source code is available on GitHub at https://github.com/DataDog/datadog-agent.
- The present repository contains the source code of the Datadog Agent version 7 and version 6.
- It is recommended to fully install the Agent.
- However, a standalone DogStatsD package is available for Amazon Linux, CentOS, Debian, Fedora, Red Hat, SUSE and Ubuntu. The package is used in containerized environments where DogStatsD runs as a sidecar or environments running a DogStatsD server without full Agent functionality.

How to install Datadog Agent?



Does Datadog Agent installation has overhead?

- The amount of space and resources the Agent takes up depends on the configuration and what data the Agent is configured to send.
- At the onset, you can expect around 0/08% CPU used on average with a disk space of roughly 830MB to 880MB.
- An example of the Datadog Agent resource consumption is below. Tests were made on an AWS EC2 machine c5.xlarge (4vCPU/8GB RAM) and comparable performance was seen for ARM64-based instances with similar resourcing. The vanilla Datadog-agent was running with a process check to monitor the Agent itself.

Datadog Agent : Start

PLATFORM	COMMAND
AIX	`startsrc -s datadog-agent`
Linux	See the Agent documentation for your OS.
Docker	Use the installation command.
Kubernetes	`kubectl create -f datadog-agent.yaml`
macOS	`launchctl start com.datadoghq.agent` *or* through the systray app
Source	`sudo service datadog-agent start`
Windows	See the Windows Agent documentation.

Datadog Agent : Stop

PLATFORM	COMMAND
AIX	`stopsrc -s datadog-agent`
Linux	See the Agent documentation for your OS.
Docker	`docker exec -it <CONTAINER_NAME> agent stop`
Kubernetes	`kubectl delete pod <AGENT POD NAME>` —note: the pod is automatically rescheduled
macOS	`launchctl stop com.datadoghq.agent` *or* through the systray app
Source	`sudo service datadog-agent stop`
Windows	See the Windows Agent documentation.

Datadog Agent : Status

PLATFORM	COMMAND
AIX	`lssrc -s datadog-agent`
Linux	See the Agent documentation for your OS.
Docker (Debian)	`sudo docker exec -it <CONTAINER_NAME> s6-svstat /var/run/s6/services/agent/`
Kubernetes	`kubectl exec -it <POD_NAME> -- s6-svstat /var/run/s6/services/agent/`
macOS	`launchctl list com.datadoghq.agent` *or* through the systray app
Source	`sudo service datadog-agent status`
Windows	See the Windows Agent documentation.
Cluster Agent (Kubernetes)	`datadog-cluster-agent status`

Usage

Agent Usage

Docker Agent for Docker, containerd and Podman

- The Datadog Docker Agent is the containerized version of the host Agent
- The Docker Agent supports Docker, containerd and Podman runtimes
- The official Docker image is available on Docker Hub, GCR and ECR-public

ECR-Public	GCR	Docker Hub
docker pull public.ecr.aws/datadog/agent	docker pull gcr.io/datadoghq/agent	docker pull datadog/agent

Agent Usage

Docker Agent for Kubernetes

Run the Datadog Agent in your Kubernetes cluster to start collecting your cluster and applications metrics, traces and logs

Installation:
You can use the following options for installing the Datadog Agent on Kubernetes

- The Datadog Operator (recommended)
- Helm
- Manual Installation

https://docs.datadoghq.com/containers/kubernetes/

Agent Usage

Docker Cluster Agent for Kubernetes

- The Datadog Cluster Agent provides a streamlined, centralized approach to collect cluster level monitoring data.
- By acting as a proxy between the API server and node-based Agents, the Cluster Agent helps to alleviate load.
- It also relays cluster level metadata to node-based Agents, allowing them to enrich the metadata of locally collected metrics.

Agent Usage

Docker Cluster Agent for Kubernetes

Using the Datadog Cluster Agent allows you to:

- Alleviate the impact of Agents on your infrastructure
- Isolate node-based Agents to their respective nodes, reducing RBAC rules to solely read metrics and metadata from the kubelet.
- Provide cluster level metadata that can only be found in the API server to the Node Agents, in order for them to enrich the metadata of the locally collected metrics.
- Enable the collection of cluster level data, such as the monitoring of services or SPOF and events
- Use Horizontal Pod Autoscaling (HPA) with custom Kubernetes metrics and external metrics

Agent Usage

Docker Cluster Agent for Kubernetes

If you're using Docker, the Datadog Cluster Agent is available on Docker Hub and GCR:

Docker Hub	GCR
hub.docker.com/r/datadog/cluster-agent	gcr.io/datadoghq/cluster-agent

Log Collection

Log Collection

Log collection requires the Datadog Agent v6.0+

Activate log collection

- Collecting logs is not enabled by default in the Datadog Agent
- To enable log collection with an Agent running on your host, change logs_enabled:false to logs_enabled:true in the Agent's main configuration file (datadog.yaml)
- HTTPS transport is the default transport used.

Log Collection

- To send logs with environment variables, configure the following: DD_LOGS_ENABLED=true
- After activating log collection, the Agent is ready to forward logs to Datadog. Next, configure the Agent on where to collect logs from.

Custom Log Collection:

- Datadog Agent v6 can collect logs and forward them to Datadog from files, the network (TCP or UDP), journald and Windows channels:
  - In the conf.d/ directory at the root of your Agent's configuration directory, create a new <CUSTOM_LOG_SOURCE>.d/ folder that is accessible by the Datadog user.
  - Create a new conf.yaml file in this new folder
  - Add a custom log collection configuration group with the parameters below
  - Restart your Agent to take into account this new configuration.
  - Run the Agent's status subcommand and look for <CUSTOM_LOG_SOURCE> under the checks section.

Proxy

Agent Proxy Configuration

If your network configuration restricted outbound traffic, proxy all Agent traffic through one or several hosts that have more permissive outbound policies.

A few options are available to send traffic to Datadog over SSL/TLS for hosts that are not directly connected to the Internet.

- Using a web proxy, such as Squid or Microsoft Web Proxy, that is already deployed to your network
- Using HAProxy (if you want to proxy more than 16-20 Agents through the same proxy)
- Using the Agents as a proxy (for **up to 16 Agents** per proxy, **only on Agent v5**)

Agent Proxy Configuration

Datadog FIPS compliance

- The Datadog Agent FIPS Proxy ensures that communication between the Datadog Agent and Datadog uses FIPS-compliant encryption.
- The Datadog Agent FIPS Proxy is a separately distributed component that you deploy on the same host as the Datadog Agent. The proxy acts as an intermediary between the Agent and Datadog intake.
- The Agent communicates with the Datadog Agent FIPS Proxy, which encrypts payloads using a FIPS 140-2 validated cryptography and relays the payloads to Datadog. The Datadog Agent and the Agent FIPS Proxy must be configured in tandem to communicate with one another.

Agent Proxy Configuration

Environment variables

Starting with Agent v6.4, you can set your proxy settings through environment variables:

- `DD_PROXY_HTTPS` : Sets a proxy server for `https` requests.
- `DD_PROXY_HTTP` : Sets a proxy server for `http` requests.
- `DD_PROXY_NO_PROXY` : Sets a list of hosts that should bypass the proxy. The list is space-separated.

Environment variables have precedence over values in the `datadog.yaml` file. If the environment variables are present with an empty value, for example: `DD_PROXY_HTTP=""` , the Agent uses the empty values instead of lower-precedence options.

On Unix hosts, a system-wide proxy might be specified using standard environment variables, such as `HTTPS_PROXY` , `HTTP_PROXY` , and `NO_PROXY` . The Agent uses these if present. Be careful, as such variables also impact every requests from integrations, including orchestrators like Docker, ECS, and Kubernetes.

The Agent uses the following values in order of precedence:

1. `DD_PROXY_HTTPS` , `DD_PROXY_HTTP` , and `DD_PROXY_NO_PROXY` environment variables
2. `HTTPS_PROXY` , `HTTP_PROXY` , and `NO_PROXY` environment variables
3. Values inside `datadog.yaml`

Do not forget to restart the Agent for the new settings to take effect.

Versions

Version

Agent v7 is the latest major version of the Datadog Agent

The only change from Agent v6 is that this version only includes support for Python3 for integrations and custom checks

Learn how to upgrade your Agent to version 7 using this link https://docs.datadoghq.com/agent/upgrade/?tab=linux

Version

From Agent 6

Linux | Windows | MacOS

Run the following Agent installation command to upgrade your Agent from version 6 to version 7:

The following command works on Amazon Linux, CentOS, Debian, Fedora, Red Hat, Ubuntu, and SUSE:

```
DD_API_KEY="<DATADOG_API_KEY>" bash -c "$(curl -L
https://install.datadoghq.com/scripts/install_script_agent7.sh)"
```

Troubleshooting

Agent Troubleshooting

If you have not yet installed the Datadog Agent, go to the dedicated Agent integration page for installation instructions. If you just installed the Agent, it may take a few moments before you start seeing metrics appear. The first place you should check for metrics is the Metrics Explorer.

If you think you might be experiencing issues, follow this checklist first:

- Is your Agent container stopping right after starting? It can be a **hostname detection issue**.
- Is your host connected to the internet or able to access it through a proxy?
- If using a proxy: is your Agent configured for this proxy?
- Is the Datadog API key set up in your datadog.yaml configuration file the API key corresponding to your Datadog platform?
- Is the site configured in your datadog.yaml configuration file matching the one from your organization?
- Is there only one Datadog Agent running on your host?
- Did you restart the Datadog Agent after editing a yaml configuration file?

If the answer to all questions above is yes, then **run the status** command for more details about your Agent and its integrations status. You can also check the **Agent logs** directly and **enable debug mode** to get more logging from the Agent.

If you're still unsure about the issue, you may reach out to the **Datadog support team** with a flare from your Agent.

APM

APM (Application Performance Monitoring)

Try to validate these questions on day to day scenarios

ASSESSMENT	YES	NO	NOT SURE
Are we measuring real end-user experiences on our website today?			
Are we meeting customer SLAs?			
Is our app delivering consistent response imes across users and geographies?			
Do we have under-performing regions? Do we have a web performance plan in place to correct those problems?			
Is our site delivering a consistent user experience, regardless of browser type?			
Do we know how our website response time and availability compares with our competition?			
Can we quantify how third-party technology and services such as ad networks and payment processes are impacting our site's performance?			
Can we monitor and troubleshoot problems for our mobile app using the same APM tool we use for our web applications?			

What is APM?

- Application performance monitoring (APM) is the practice of tracking key software application performance metrics using monitoring software and telemetry data.
- APM gives you grouped views of your application's performance trends for quick and easy diagnosis of performance problems
- Practitioners use APM to ensure system availability, optimize service performance and response times, and improve user experiences.
- Understand trends, isolate anomalies and get actionable insight for problem resolution and code optimization.
- Mobile apps, websites, and business applications are typical use cases for monitoring. However, with today's highly connected digital world, monitoring use cases expand to the services, processes, hosts, logs, networks, and end-users that access these applications — including a company's customers and employees.

Why do we need APM?

- APM tools help digital teams monitor and resolve variables affecting application performance.
- Without APM tools, resolving performance issues becomes challenging, leading to customer frustration and app abandonment.
- Customers rely heavily on apps for daily activities, especially in the work-from-home era, making app performance crucial.
- Poor app performance, such as crashes or slow load times, can harm brand reputation and reduce revenue.
- Identifying the root cause of performance problems is difficult, as they can stem from coding errors, database issues, or network and hosting problems.
- Modern applications are complex, with millions of lines of code, interconnected digital services, and containerized environments hosted across multiple clouds.

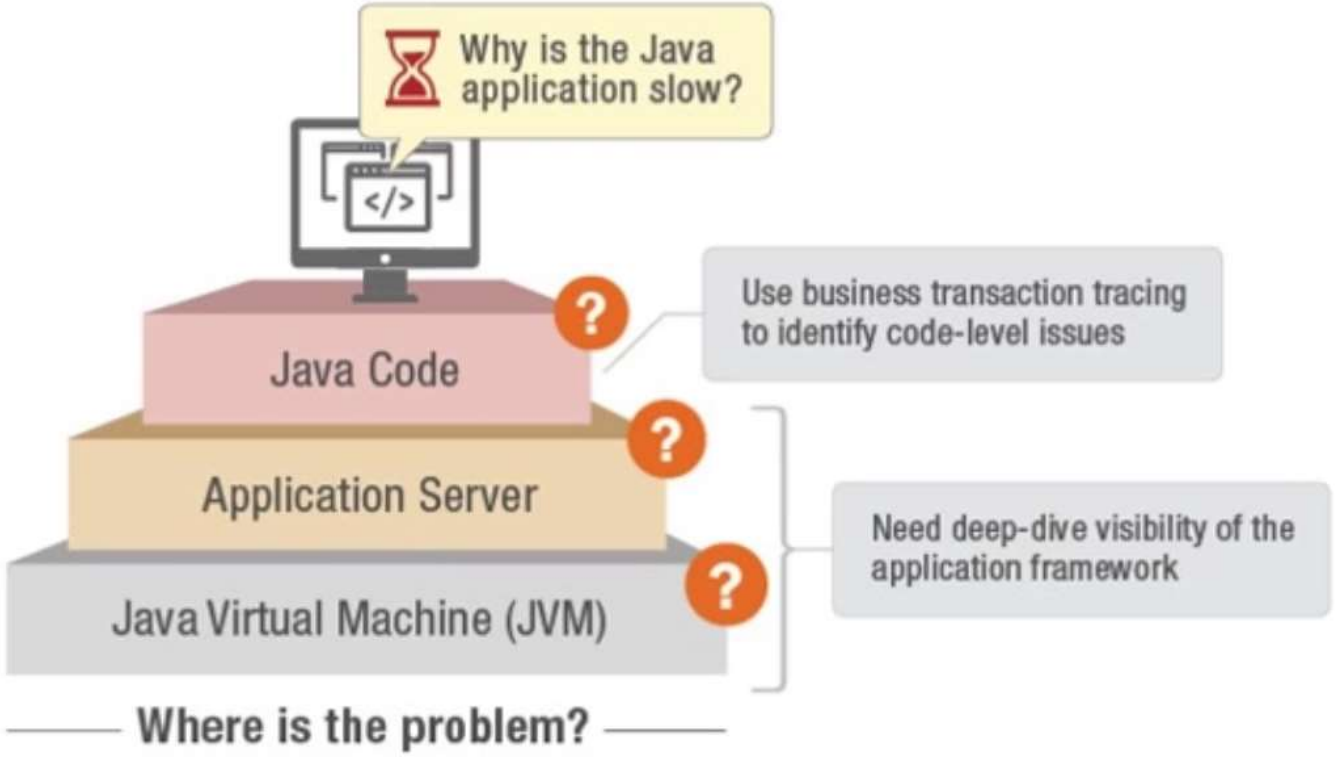
Why Performance is very critical?

- A 1 second delay in response time can reduce conversions by 7%, page views by 11% and customer satisfaction by 16%

- More than half (51%) of online consumers in the US said that site slowness is the top reason they would abandon a purchase

- When online service fails, 75% of consumers move to another channel, which can lead to millions of lost dollars.

Why APM?


Why is the Java application slow?
Java Code
Application Server
Java Virtual Machine (JVM)
Use business transaction tracing to identify code-level issues
Need deep-dive visibility of the application framework
Where is the problem?

Methodologies to obtain data

Agent-Based Method:

- **Overview**: This method involves installing the Datadog Agent or a piece of software injected within the application which collects and sends data to Datadog.
- **Capabilities**: The agent can capture detailed metrics, traces, and logs directly from your application environment, providing deep visibility into application performance.
- **Supported Environments**: Works well with various environments, including on-premises servers, cloud instances, and containers.
- **Advantages**: Provides real-time, granular data, including resource consumption and detailed traces, enabling more accurate troubleshooting.
- **Disadvantages**: The negative aspect though of having one of these agents is it will incur some overhead onto your application but the visibility you gain is priceless and the overhead is minimal.

Methodologies to obtain data

Agentless Method:

- **Overview**: In this method, Datadog integrates with third-party services and cloud providers to collect data without needing to install an agent. This is generally done by spanning ports and analyzing the packet level information.
- **Capabilities**: Relies on APIs and other integrations to gather performance data, which is then sent to Datadog for monitoring.
- **Supported Environments**: Typically used with SaaS applications or services where agent installation is not possible or practical.
- **Advantages**: Easier to implement in environments where installing an agent is challenging, and reduces overhead on the host system.
- **Disadvantages**: It get constrained in terms of the amount of data you obtain.

Methodologies of how we monitor?

Real User Monitoring: RUM is the notion that with the evolving complexity of modern day computing in order to truly identify performance you must go all the way back to the end user. This involves going all the way through delivery chain back to the data centre and having all the metrics associated with those transactions.

Business Transactions: BT's make our lives easier. The purpose of BT's is to take all the data collected and aggregate them into high level concepts. It draws many similarities to a Select Statement from SQL. For example: if the business wants to look at the revenue per item, an APM tool would ind the methods which return these values and make these metrics. The amount of data collected can be scary so BT's simplify this data and make it translatable to any dialect in the corporate world.

End Goal of APM

- Automatic Reports
- Alerts
- Warning Signals

1. When your response time of your transactions start increasing, rad flags are waved.
2. When a transaction starts failing, a developer already knows of the issue because the APM team has reported on the trending data.

APM Selection Criteria

- Platform/Language support
  - Does it support Ruby, Python, .Net?
- Developer Familiarity
- Ease of use
  - Setup -> diagnostics -> fix
- Integration Support
  - HipChat/Slack/Jira/GitHub/Chef/Puppet
- Pricing

Datadog APM

- Datadog Application Performance Monitoring (APM) provides deep visibility into your applications, enabling you to identify performance bottlenecks, troubleshoot issues, and optimize your services.

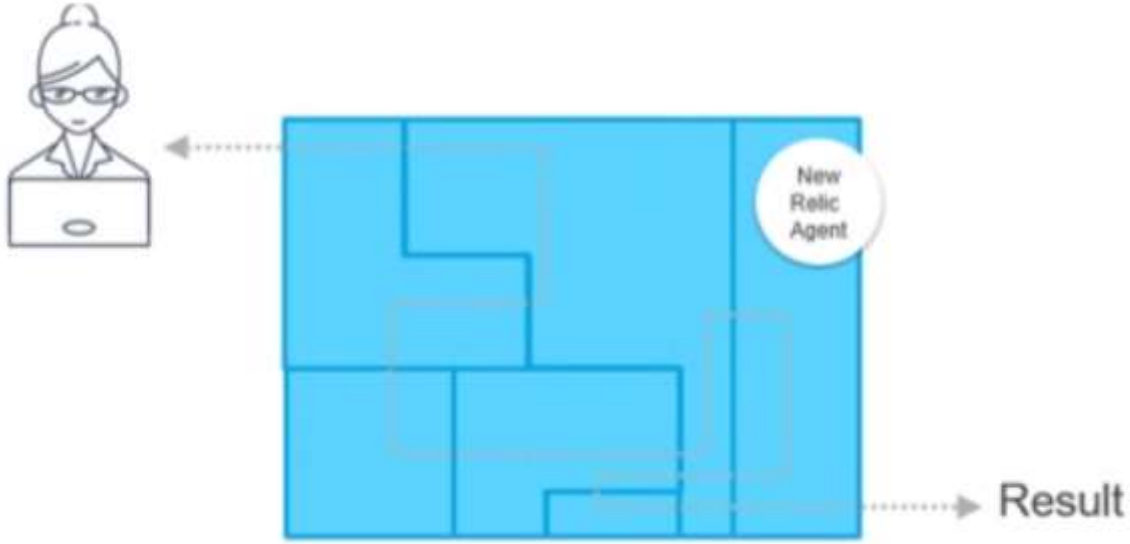
- With distributed tracing, out-of-the-box dashboards, and seamless correlation with other telemetry data, Datadog APM helps ensure the best possible performance and user experience for your applications.

Datadog APM Competitors

- Appdynamics
  - Enterprise focus, no Ruby/Python SDK support
- Compuware
  - APM, DynaTrace and Gomez
- IBM, HP, Dell, Microsoft
- Splunk, Logstash
- Systems Monitoring: Gomez, Pingdom, Nagios
- Real User Monitoring: GTMetrics, Google PageSpeed

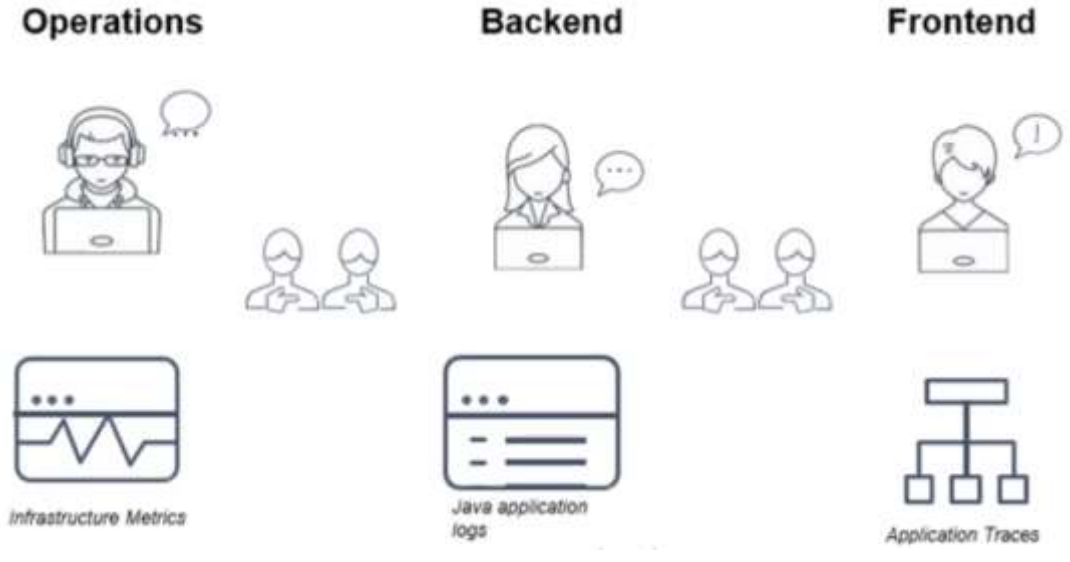
What is a transaction trace?

- **Traces**: How did an Application arrive at a result
- Tracing provides information on which segments of an application were involved in a result.



What is a transaction trace?

- **To answer operational questions, teams define their own metrics, logging and alerting practices.**
- **Important operational data becomes siloed by service, app or infrastructure type**



Hands-On Lab

Learning Objectives:

- Explain the purpose of the Datadog Agent
- Install and configure the Datadog Agent
- Install and configure common Datadog Integrations
- Configure APM and explain how to instrument applications
- Enable NPM and analyze networking data
- Configure, search and aggregate Logs
- Explore, graph and monitor metrics
- Create and layout Dashboards to track a variety of metrics in one place
- Navigate fluidly across all these products

Datadog Tagging

Datadog Tags: Common Issues

Datadog Tags

- Tags are a way of adding dimensions to Datadog telemetries so they can be filtered, aggregated, and compared in Datadog visualizations.
- Using tags enables you to observe aggregate performance across several hosts and narrow the set further based on specific elements.
- In summary, tagging is method to observe aggregate data points.

Datadog Tags

- Tagging binds different data types in Datadog, allowing for correlation and call to action between metrics, traces and logs.
- This is accomplished with reserved tag keys.

TAG KEY	ALLOWS FOR
`host`	Correlation between metrics, traces, processes, and logs.
`device`	Segregation of metrics, traces, processes, and logs by device or disk.
`source`	Span filtering and automated pipeline creation for Log Management.
`service`	Scoping of application specific data across metrics, traces, and logs.
`env`	Scoping of application specific data across metrics, traces, and logs.
`version`	Scoping of application specific data across metrics, traces, and logs.
`team`	Assign ownership to any resources

Datadog Tags

Reserved Tags

Host, Service, Env, Source, Device, Version

Technical Context

Env, Location, Region, Application, OS. Language

Business Context

Team, Owner, Project, Department, Tier, Priority

Datadog Tags

- Datadog recommends looking at containers, VMs and cloud infrastructure at the service level in aggregate.
- For example: look at CPU usage across a collection of hosts that represents a service, rather than CPU usage for server A or server B separately.

Defining Tags

Below are Datadog's tagging requirements:

1. Tags must **start with a letter** and after that may contain the characters listed below:

   - Alphanumerics
   - Underscores
   - Minuses
   - Colons
   - Periods
   - Slashes

   Other special characters are converted to underscores.

2. Tags can be **up to 200 characters** long and support Unicode letters (which includes most character sets, including languages such as Japanese).

3. Tags are converted to lowercase. Therefore, `CamelCase` tags are not recommended. Authentication (crawler) based integrations convert camel case tags to underscores, for example `TestTag` → `test_tag`.

Defining Tags

4. A tag can be in the format `value` or `<KEY>:<VALUE>`. Commonly used tag keys are `env`, `instance`, and `name`. The key always precedes the first colon of the global tag definition, for example:

TAG	KEY	VALUE
`env:staging:east`	`env`	`staging:east`
`env_staging:east`	`env_staging`	`east`

5. Tags should not originate from unbounded sources, such as epoch timestamps, user IDs, or request IDs. Doing so may infinitely increase the number of metrics for your organization and impact your billing.

6. Limitations (such as downcasing) only apply to metric tags, not log attributes or span tags.

Defining Tags

Tagging methods

Tags may be assigned using any (or all) of the following methods.

METHOD	ASSIGN TAGS
Configuration Files	Manually in your main Agent or integration configuration files.
UI	In the Datadog site.
API	When using Datadog's API.
DogStatsD	When submitting metrics with DogStatsD.

Assign Tags: Configuration File

File location

The Agent configuration file (`datadog.yaml`) is used to set host tags which apply to all metrics, traces, and logs forwarded by the Datadog Agent.

Tags for the integrations installed with the Agent are configured with YAML files located in the **conf.d** directory of the Agent install. To locate the configuration files, see Agent configuration files.

YAML format

In YAML files, use a list of strings under the `tags` key to assign a list of tags. In YAML, lists are defined with two different yet functionally equivalent forms:

```
tags: ["<KEY_1>:<VALUE_1>", "<KEY_2>:<VALUE_2>", "<KEY_3>:<VALUE_3>"]
```

or

```
tags:
    - "<KEY_1>:<VALUE_1>"
    - "<KEY_2>:<VALUE_2>"
    - "<KEY_3>:<VALUE_3>"
```

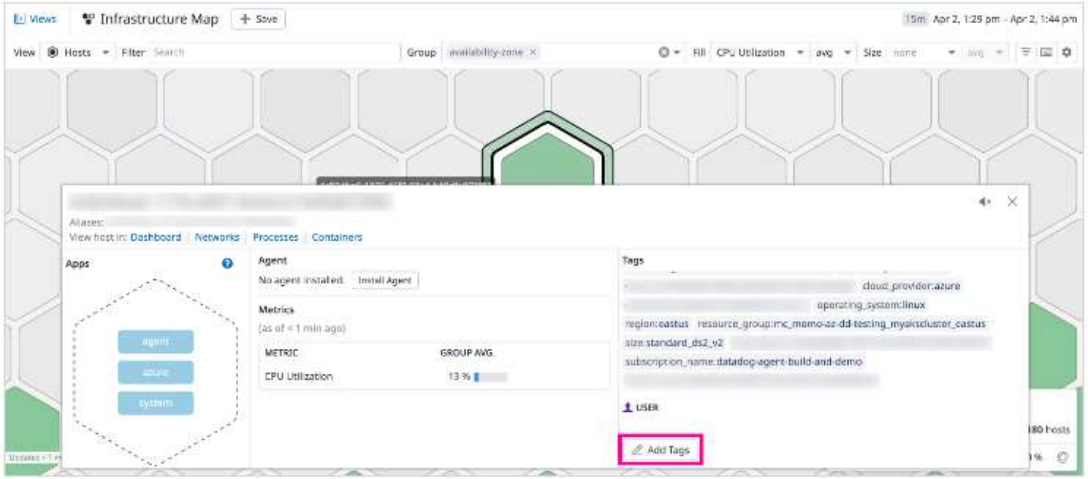
Assign Tags: Configuration File

Add these tags into the main configuration file (datadog.yaml)

```
tags:
  - environment:production
  - team:engineering
  - role:webserver
```

Assign Tags: Datadog UI

Assign host tags in the UI using the Host Map page. Click on any hexagon (host) to show the host overlay on the bottom of the page. Then, under the *User* section, click the **Add Tags** button. Enter the tags as a comma separated list, then click **Save Tags**. Changes made to host tags in the UI may take up to five minutes to apply.



Assign Tags: Datadog UI

We can assign the tags on different levels:

- Host Map
- Infrastructure List
- Monitors
- Distribution Metrics
- Integrations
- Service Level Objectives

Assign Tags: Datadog API

Tags can be assigned in various ways with the Datadog API. See the list below for links to those sections:

- Post a check run
- Post an event
- AWS Integration
- Post timeseries point
- Create or Edit a monitor
- Add or Update host tags
- Send traces
- Create or Update a Service Level Objective

Assign Tags: DogStatsD

DogStatsD implements the StatsD protocol and adds a few Datadog-specific extensions:

- Histogram metric type
- Service checks
- Events
- Tagging

Add tags to any metric, event, or service check you send to DogStatsD. For example, compare the performance of two algorithms by tagging a timer metric with the algorithm version:

```
@statsd.timed('algorithm.run_time', tags=['algorithm:one'])
def algorithm_one():
    # Do fancy things here ...

@statsd.timed('algorithm.run_time', tags=['algorithm:two'])
def algorithm_two():
    # Do fancy things (maybe faster?) here ...
```

Unified Service Tagging

Unified service tagging ties Datadog telemetry together by using three reserved tags: **env, service, and version**.
With these three tags, you can:

- Identify deployment impact with trace and container metrics filtered by version
- Navigate seamlessly across traces, metrics, and logs with consistent tags
- View service data based on environment or version in a unified fashion

Using Tags

After assigning tags, start using them to filter and group your data in your Datadog platform. Tags can be used to include or exclude data.

When including or excluding multiple tags:

- Include uses AND logic
- Exclude uses OR logic

Using Tags

Events

The Events Explorer shows the events from your environment over a specified time period. Use tags to filter the events list and focus on a subset of events. Enter tags: followed by a tag to see all the events coming from a host, integration, or service with that tag. For example, use **tags:service:coffee-house** to search for the tag **service:coffee-house**.

To search multiple tags inclusively, use parentheses and separate each tag with OR: tags:(**service:coffee-house OR host:coffeehouseprod**). To search multiple tags exclusively, separate each tag with AND: tags:(**service:coffee-house AND host:coffeehouseprod**)

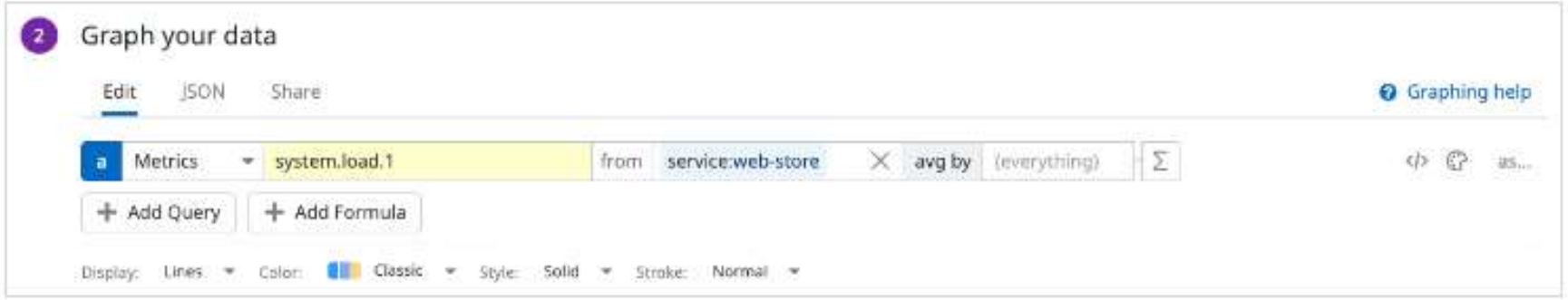
Using Tags

Dashboards:

Use tags to filter metrics to display in a dashboard graph, or to create aggregated groups of metrics to display.

To filter the metrics to display, enter the tag in the from text box.

This metric displays over all sources that have that particular tag assigned (service:web-store in the example below).



Using Tags

Dashboards:

Advanced tag value filtering is also available with Boolean filters. The following Boolean syntax is supported:

- NOT, !
- AND, ,
- OR
- key IN (tag_value1, tag_value2,...)
- key NOT IN (tag_value1, tag_value2,...)

Using Tags

Integrations:

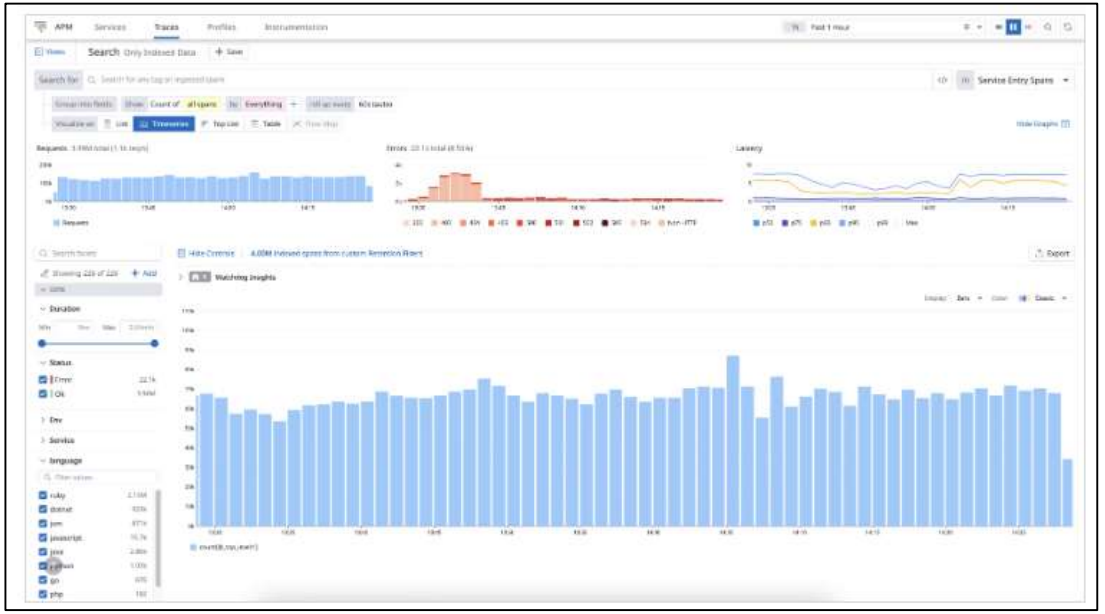
Some integrations allow you to optionally limit metrics using tags.

- The AWS integration tile has the tag filters **to hosts with tag** and **to Lambdas with tag**.
- The Azure integration tile has the tag filter **Optionally filter to VMs with tag**.
- The Google Cloud integration tile has the tag filter **to hosts with tag**.

Using Tags

APM:

In the Trace Explorer, you can filter traces with tags using the search bar or facet checkboxes. The search bar format is <KEY>:<VALUE>, for example: service:coffee-house. For advanced search, see Query Syntax.



Using Tags

Notebooks:

- When creating a Notebook graph, limit metrics by using tags in the from text box. Additionally, group metrics by using tags in the avg by text box. In the example below, metrics are limited to **service:coffee-house** and grouped by host.

- To exclude tags, use </> to edit the text then add the tag in the form !<KEY>:<VALUE>. In the example below, **service:coffeehouse** is excluded using **!service:coffeehouse**.

Using Tags

Logs:

- For Logs Search, Analytics, Patterns, and Live Tail, filter logs with tags using the search bar or facet checkboxes. The search bar format is **<KEY>:<VALUE>,** for example: **service:coffee-house**. For advanced search, see Search Logs.

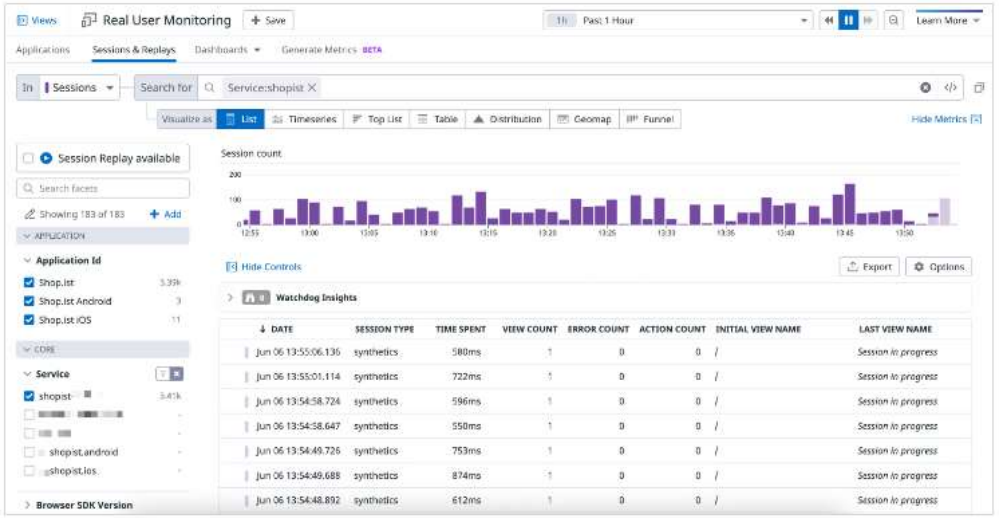
- Additionally, tags are used to filter a logs Pipeline. For example, if you only want logs from the coffee-house service to go through the pipeline, add the tag **service:coffee-house** to the filter field.

Using Tags

RUM and Session Replay:

The RUM Explorer visualizes events from your environment over a specified time period.

To filter RUM event data by tags, use the search bar or facet checkboxes. The search bar format is **<KEY>:<VALUE>,** for example: **service:shopist**. For advanced search, see Search RUM Events.

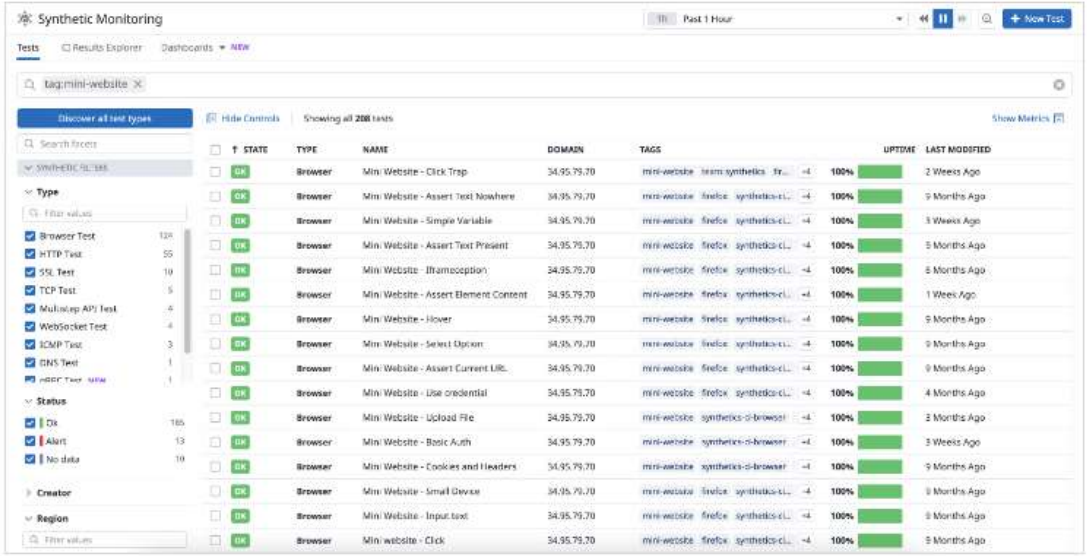


Using Tags

Synthetics:

The Synthetic Tests page lists your Synthetic tests.

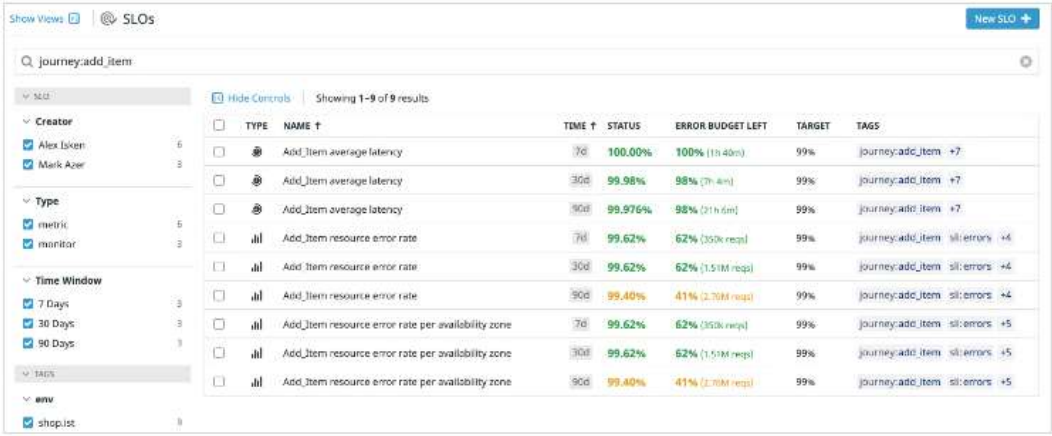
To filter tests by tags, use the search bar or facet checkboxes. The search ba format is **<KEY>:<VALUE>** For example: tag:mini-website. For advanced search, see Search and Manage Synthetic Tests.



Using Tags

Service level objectives:

To filter SLOs by assigned tags, use the search bar or facet checkboxes. The search bar format is <KEY>:<VALUE>, for example: **journey:add_item**.



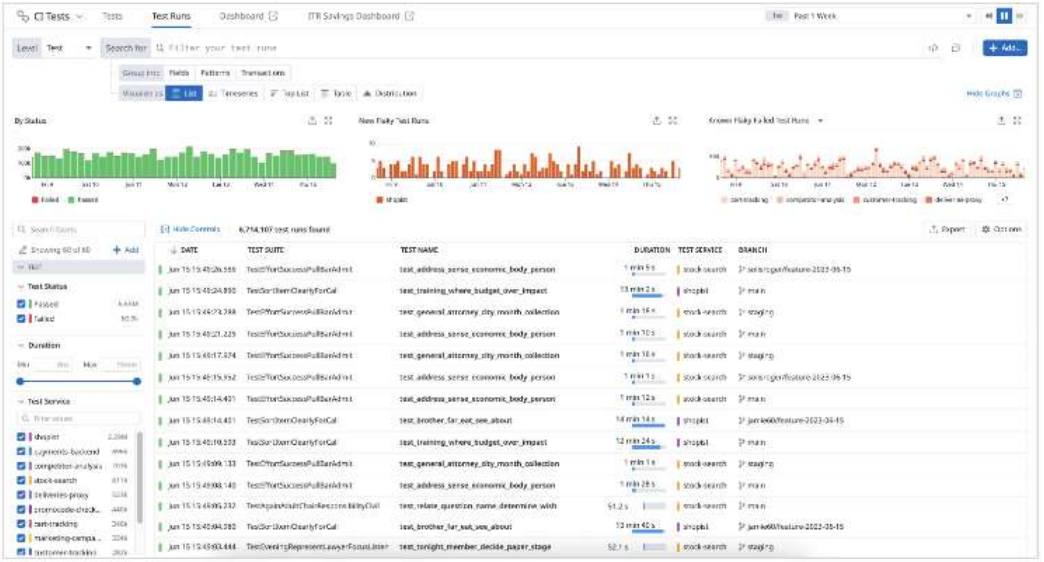
To exclude SLOs with a specific tag from your search, use -, for example: -journey:add_item.

Using Tags

CI visibility:

The CI Visibility Explorer displays your test runs run in a CI pipeline.

To filter test runs by tags, use the search bar or facet checkboxes. The search bar format is <KEY>:<VALUE>. For example: @test.status:failed.



Using Tags

Developers:
Tags can be used in various ways with the API.

See this list for links to respective sections:

- Schedule monitor downtime
- Query the event explorer
- Search hosts
- Integrations for AWS and Google Cloud
- Querying timeseries points
- Get all monitor details
- Mute a monitor
- Monitors search
- Monitors group search
- Create a Screenboard
- Create a Timeboard
- Create a SLO
- Get a SLO’s details
- Update a SLO

Using Tags : Demo

Learning Objectives:

- Assign and use tags to build container maps and dashboards in Datadog for a Kubernetes deployment
- Assign tags to app data
- Explore correlated data sing tags
- Use tags to create targeted alerts
- Search and correlate synthetic tests using tags
- Work with unified service tagging

Logs

Logs Analysis

Teams do common mistakes which might lead to system failure.

1. Avoid getting more information
2. Product or application are working fine and they skip log monitoring
3. They consider it a time consuming task.

What is log?

- Logs are the records or the activities being generated by an information system
- It is a machine data which is being generated every second.

Log Data Sources

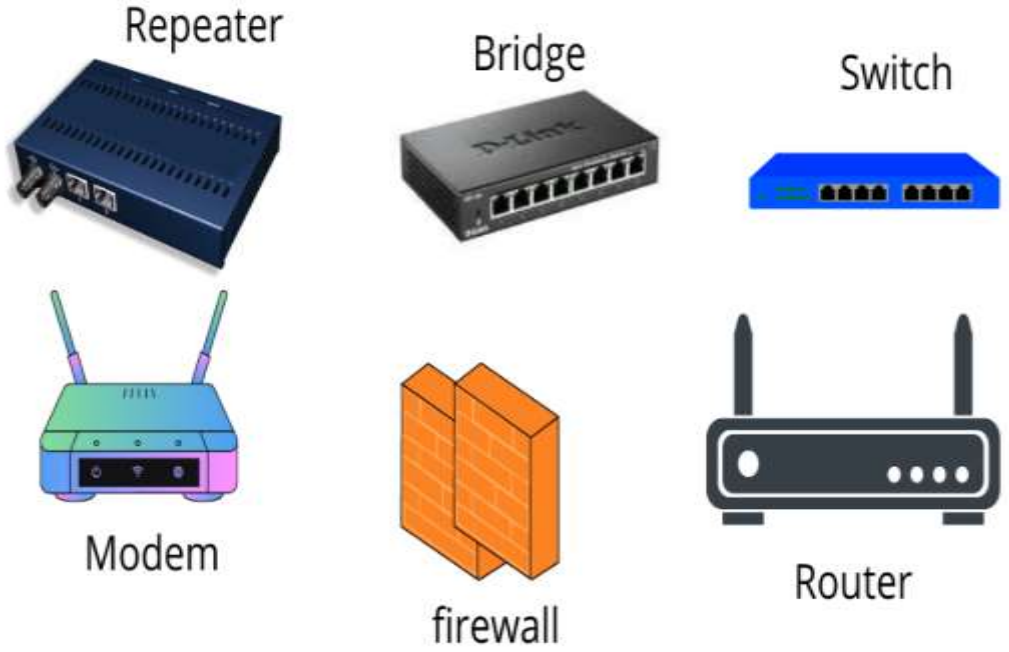
- IDS
- Proxies
- Servers
- Network infrastructure
- Firewall
- Databases
- Applications

Server Logs

- Linux/Windows
- Log files
- Access
- File system

Network Logs

- Firewall
- Warnings
- Alerts
- IP addresses



Database Logs

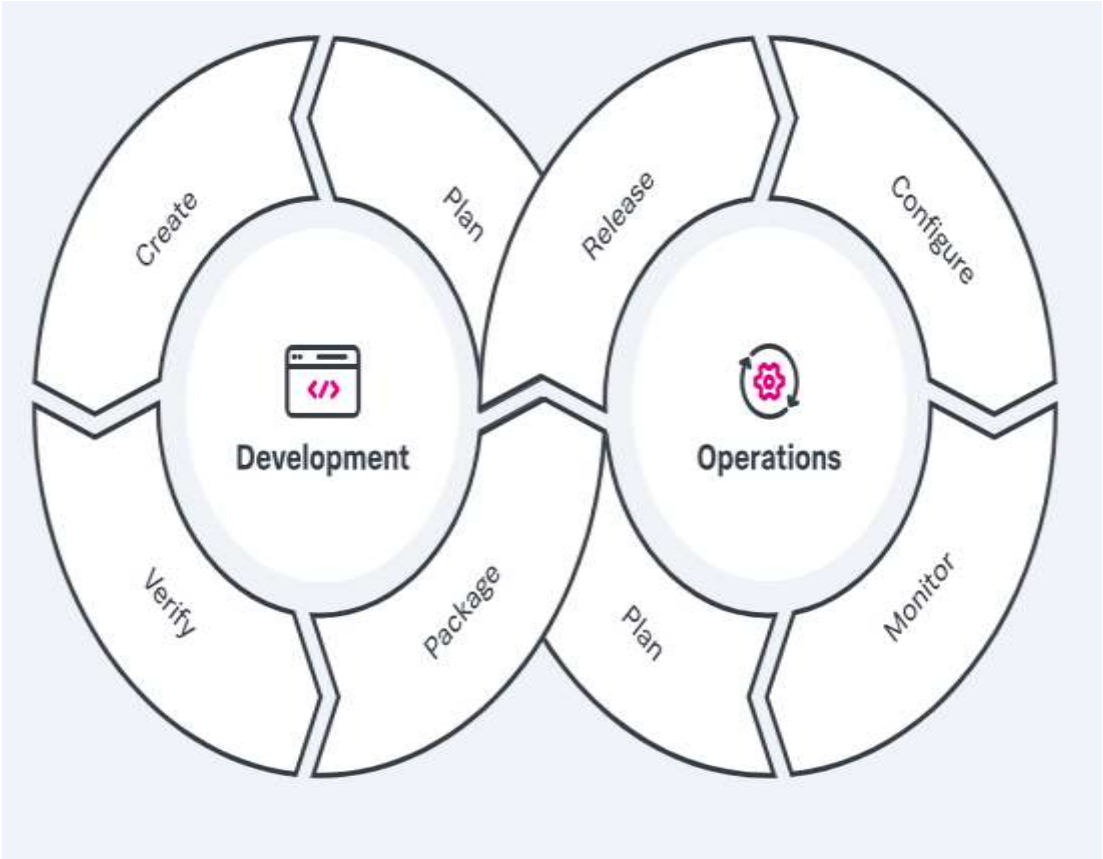
- Audit Logs
- Configurations
- Schemas
- Tables
- Queries

Web Logs

- Click Streams
- Location
- Browser
- Transactions
- Time

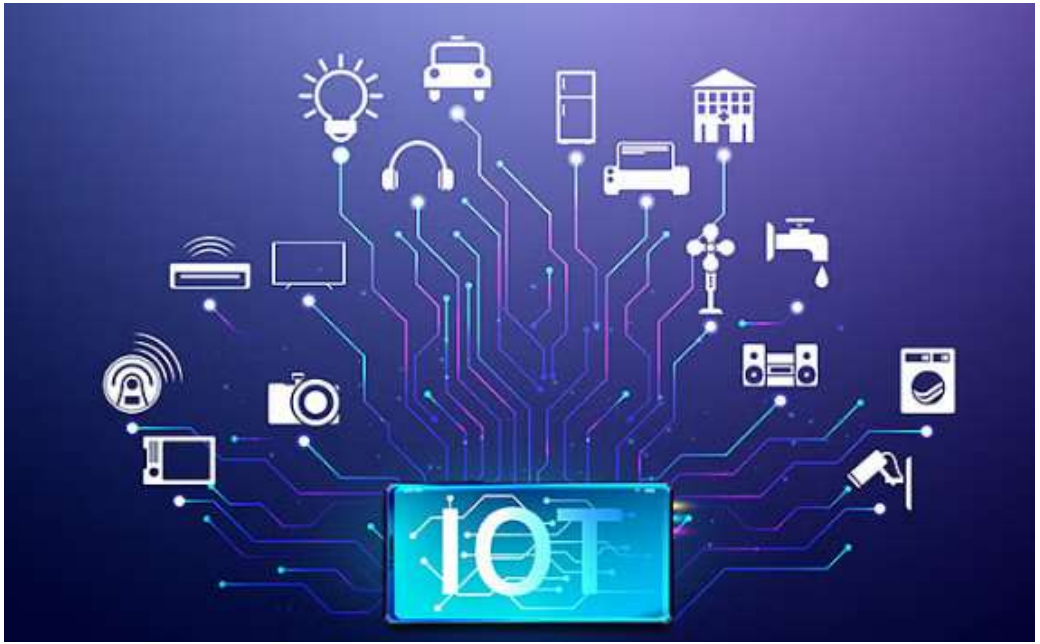
Devops Logs

- Test Logs
- Alerts
- Code Check-in
- Event Logs



IOT Logs

- GPS
- Temperature
- RFID
- Biometric
- Limitless



Why Logs are important?

Monitoring 4 Golden Signals

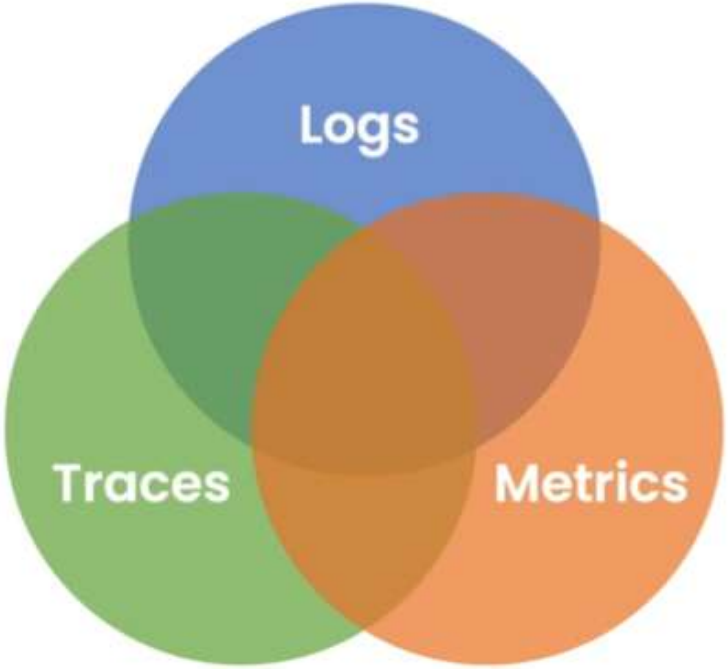
- Latency
- Traffic
- Errors
- Saturations

Why Logs are important?

Monitoring 4 Golden Signals

- **Latency** : Time it takes for a request to travel from the client to the server and back.
- **Traffic**: Number of requests a system receives over a specific period
- **Errors**: Percentage of requests resulting in erros, such as 404 Page Not Found or 500 Internal Server errors
- **Saturations**: Measures resource utilization, including CPU, memory and disk space

Why Logs are important?



Provide a chronological record of events, or transactions within a system

Quantitative measurements that offer a snapshot of a system's performance over time

Helps track the flow of requests through various services and components of a system.

Why Logs are important?

So **monitoring** is not enough to solve the issue as

- Monitoring notifies about the problem
- Provides limited information

We need more information to find the root cause so we need **logs** which has :

- Details about the problem
- Complete information

Why Logs are important?

- Resource Usage
- HTTP Errors
- Slow Queries
- Rouge Automated Robots
- Security Issues
- Debugging

Why Logs are important?

- **Troubleshooting**: Logs help identify and fix issues in systems.
- **Compliance**: Logs ensure adherence to regulatory requirements.
- **System Maintenance**: Logs support ongoing system health and maintenance.
- **Performance Monitoring**: Logs track system performance and detect bottlenecks.
- **Security Auditing**: Logs monitor access and detect unauthorized activities.
- New threat discovery
- Tracking website’s visitors
- Get more insight of network related issues
- Incident response

What to check for in Logs?

- **Error Logs**: Record of errors encountered by an application.
- **Access Logs**: Details of user access to a website or system.
- **Security Logs**: Entries related to security events like login attempts.
- **Transaction Logs**: Records of financial or data transactions.
- **Event Logs**: Documentation of specific events within a system.
- **Application Logs**: Logs generated by software applications detailing operations.
- **System Logs**: Information on the operating system's activities.
- **Audit Logs**: Records tracking user actions and system changes.
- **Performance Logs**: Metrics related to system performance like CPU or memory usage.
- **Network Logs**: Logs of network activity, including traffic patterns and connections.

Logs analysis

Monitoring Tools should follow below rules:

Pattern Detection and Recognition: Tools should be capable of filtering logs based on specific patterns, such as Apache logs, system logs, or application logs. For example, filtering out error codes from Apache access logs to focus on critical issues.

Normalization: Logs from various sources should be standardized into a consistent format. This ensures that different log elements, such as timestamps and IP addresses, are uniformly represented, making it easier to analyze. For instance, converting log timestamps from different time zones into UTC.

Logs analysis

Monitoring Tools should follow below rules:

Tagging and Classification: Monitoring tools should automatically tag and classify logs based on the relevant application and environment. For example, tagging logs from a production server as "prod" and from a testing server as "test" to differentiate their origins.

Correlation Analysis: Logs should be collected and analyzed together to identify patterns or issues that span multiple logs. For instance, correlating database error logs with application server logs to diagnose a problem affecting both systems.

Logs analysis

Monitoring Tools should follow below rules:

Artificial Ignorance: Implement machine learning processes to identify and filter out irrelevant log entries. For example, a system could learn to ignore repetitive, non-critical log entries that do not contribute to meaningful insights, focusing instead on significant anomalies or errors.

Datadog Log Management

Datadog Log Management Features

Rapid Troubleshooting:

- Datadog enables quick identification and resolution of issues by providing real-time access to logs across your infrastructure.

- For example, if a web application suddenly experiences a spike in error rates, you can instantly search and filter logs by error codes or specific user requests to pinpoint the root cause, reducing downtime.

Datadog Log Management Features

Full Observability:

- Datadog provides a unified view of your entire environment, combining logs, metrics, and traces for complete visibility into your systems.

- For instance, you can trace a user request across different services, viewing logs from the web server, application server, and database all in one place, which helps in understanding the full impact of an issue.

Datadog Log Management Features

Seamless Integration:

- Datadog integrates effortlessly with over 600 technologies and services, such as AWS, Docker, and Kubernetes.

- For example, you can automatically collect and monitor logs from your AWS Lambda functions, Docker containers, and Kubernetes pods without needing complex configurations, ensuring that all critical logs are captured.

- Use third-party log shippers such as Logstash, rsyslog or FluentD

Datadog Log Management Features

Customizable Processing:

- Datadog allows you to tailor log processing to your needs by creating custom pipelines to parse, enrich, and filter logs.

- For example, you might want to extract specific JSON fields from application logs or mask sensitive information before storage. You can set up custom rules to ensure that logs are processed according to your organization's requirements.

Datadog Log Management Features

Visualization and Alerting:

- Datadog provides powerful visualization tools and alerting capabilities, allowing you to create dashboards and set up alerts based on log data.

- For instance, you could create a dashboard to visualize error trends over time and set up an alert to notify your team when error rates exceed a certain threshold, enabling proactive issue management before they impact users.

Methods to send logs to Datadog

Datadog Agent:

- Installation: Install the Datadog Agent on your servers or containers.
- Configuration: Configure the Agent to collect logs from specific files, directories, or services (e.g., Apache, NGINX).
- Forwarding: The Agent collects logs and forwards them directly to Datadog for processing and analysis.

Methods to send logs to Datadog

APIs:

- Datadog Logs API: Send logs programmatically using the Datadog Logs API. This is useful for custom applications or services that generate logs.

- Integration: Write code that formats and sends log entries to the API endpoint, ensuring they are captured in Datadog.

Methods to send logs to Datadog

Log Shippers:

- Fluentd/Fluent Bit: Integrate Datadog with Fluentd or Fluent Bit to collect and forward logs from various sources like Docker, Kubernetes, or system logs.

- Logstash: Use Logstash to gather and filter logs, then send them to Datadog using the appropriate output plugin.

Methods to send logs to Datadog

Cloud Integrations:

- AWS CloudWatch: Set up an integration between AWS CloudWatch and Datadog to automatically forward logs from AWS services like Lambda, EC2, and S3 to Datadog.

- GCP Logging: Use Google Cloud Logging integration to forward logs from GCP services to Datadog.

- Azure Monitor: Integrate Azure Monitor logs with Datadog for seamless log collection from Azure resources.

Methods to send logs to Datadog

Container Logging:

- Docker: Configure the Datadog Agent within your Docker containers to collect logs. Alternatively, use Docker logging drivers to forward logs to Datadog.

- Kubernetes: Deploy the Datadog Agent as a DaemonSet in Kubernetes clusters to collect logs from all containers and nodes.

Methods to send logs to Datadog

File-based Logging:

- Log Files: Configure the Datadog Agent to tail log files from specified directories and send them to Datadog.

- Custom File Parsing: Set up custom log processing rules within the Datadog Agent to parse and forward specific log formats.

Methods to send logs to Datadog

Syslog:

- Syslog Forwarding: Send logs to Datadog via Syslog. Configure your system to forward syslog data to the Datadog Agent, which will then send it to Datadog.

Methods to send logs to Datadog

Third-party Integrations:

- Existing Logging Solutions: If you use other logging solutions like Splunk, ELK Stack, or Papertrail, you can integrate them with Datadog to forward logs.

Labs: Demo 1

Sends logs using the API

Region	API
US1	https://http-intake.logs.datadoghq.com/api/v2/logs
US2	https://http-intake.logs.us3.datadoghq.com/api/v2/logs
US5	https://http-intake.logs.us5.datadoghq.com/api/v2/logs
EU	https://http-intake.logs.datadoghq.eu/api/v2/logs
AP1	https://http-intake.logs.ap1.datadoghq.com/api/v2/logs
US1-FED	https://http-intake.logs.ddog-gov.com/api/v2/logs

Labs: Demo 1

Sends logs using the API

https://github.com/CloudSihmar/datadog-logs/blob/main/insert-using-curl

Labs: Demo 2

Sends logs using Datadog Agent

1. Install Datadog agent
2. Enable logs_enabled=true in datadog.yaml
3. Restart the agent
4. Create a demo log file
5. Create a directory under conf.d and a yaml file
   /etc/datadog-agent/conf.d/logs_collection.d/conf.yaml

Labs: Demo 2

Sends logs using Datadog Agent

vi /etc/datadog-agent/conf.d/logs_collection.d/conf.yaml

```
logs:
  - type: file
    path: path of the log file
    source: upgrad
    service: web
  - type: file
    path: /var/log/*.log
    source: linux
    service: system
```

Labs: Demo 3

Sends Apache logs using integration

1. Go to /etc/datadog-agent/conf.d/apache.d/conf.yaml, edit the logs section or add below information

```
logs:
  - type: file
    path: /var/log/httpd/error.log
    source: error
    service: apache
  - type: file
    path: /var/log/httpd/access.log
    source: access
    service: apache
```

Explore Log

- **Log Explorer**: Discover the Log Explorer view and how to add Facets and Measures.
- **Search**: Search through all of your logs
- **Live Trail**: See your ingested logs in real time across all the environments
- **Analytics**: Perform Log Analytics over your indexed logs
- **Patterns**: Spot Log Patterns by clustering your indexed logs together
- **Saved Views**: Use saved views to automatically configure your log explorer

Explore Log

- Log Explorer will be used for below
- Filter logs using the tags
- Aggregate and Measure (Field, Pattern, Transaction)

Search Syntax

- A query is composed of terms and operators
- We have single words or group of words to search
- We can use the operators like AND , OR, -

Operator	Usage	Description
AND	‘condition1 AND condition2’	Returns logs that match both conditions.
OR	‘condition1 OR condition2’	Returns logs that match either of the conditions.
- (NOT)	‘-condition’ or ‘condition1 -condition2’	Excludes logs that match the condition(s) after the -. Can be combined with other conditions.

Special Characters

Special Character	Usage	Example	Description
“‘’ (Quotes)	"search term"	"error occurred"	Exact phrase match. Finds logs that contain the exact phrase "error occurred".
‘(‘ and ‘)’	condition1 AND (condition2 OR condition3)	status:error AND (env:prod OR env:dev)	Groups conditions together. Useful for combining multiple conditions logically.
‘*’ (Wildcard)	search*	error*	Matches any sequence of characters. For example, error* matches error, errors, erroring, etc.
‘?’ (Single Character Wildcard)	search?	error*	Matches exactly one character. For example, error? matches errors but not error or erroring.
‘\’ (Escape)	\special_character	status\:error	Escapes special characters to be treated as literals. For example, searches for logs containing status:error literally.
‘@’	@attribute	@host:my-server	Targets specific attributes in the logs. For example, @host:my-server filters logs by the host attribute.

Numerical Values

Operator	Usage	Example	Description
`=`	`attribute:value`	`response_time=200`	Matches logs where the `response_time` attribute is exactly 200.
`!=`	`attribute!=value`	`response_time!=200`	Matches logs where the `response_time` attribute is **not** 200.
`<`	`attribute<value`	`response_time<300`	Matches logs where the `response_time` attribute is less than 300.
`<=`	`attribute<=value`	`response_time<=300`	Matches logs where the `response_time` attribute is less than or equal to 300.
`>`	`attribute>value`	`response_time>100`	Matches logs where the `response_time` attribute is greater than 100.
`>=`	`attribute>=value`	`response_time>=100`	Matches logs where the `response_time` attribute is greater than or equal to 100.
`BETWEEN`	`attribute:[min TO max]`	`response_time:[100 TO 200]`	Matches logs where the `response_time` attribute is between 100 and 200 inclusive.
`:` (Range)	`attribute:value1 TO value2`	`age:[25 TO 35]`	Matches logs where `age` is between 25 and 35 inclusive.

Facet

Facet	Description	Example Search	Explanation
`@status`	The status of an event (e.g., error, success)	`@status:error`	Returns logs where the `@status` facet is `error`.
`@host`	The host where the log was generated	`@host:my-server`	Returns logs generated by the host `my-server`.
`@service`	The name of the service producing the logs	`@service:web-app`	Returns logs from the `web-app` service.
`@env`	The environment (e.g., production, staging)	`@env:production`	Returns logs where the environment is `production`.
`@version`	The version of the application	`@version:1.2.3`	Returns logs where the application version is `1.2.3`.
`@http.status_code`	The HTTP status code for requests	`@http.status_code:500`	Returns logs where the HTTP status code is `500`.
`@user.id`	The user ID associated with the log event	`@user.id:12345`	Returns logs associated with the user ID `12345`.

Synthetic Monitoring

Synthetic Monitoring

- Synthetic monitoring is a monitoring technique that is done by isng an emulation or scripted recordings of transactions.
- Behavioral scripts are created to simulate an action or path that a customer or end-user would take on a site, application or other software

Real User Monitoring

- Real user monitoring (RUM) is a passive monitoring technology that analyzes all user interaction with a website or client interacting with a server or cloud-based application.
- Monitoring actual user interaction with a website or an application is important to operators to determine if users are being served quickly and without errors and if not, which part of a business process is failing.

Synthetic vs Real User Monitoring

Monitors

Types of Monitors

Metric Monitor

- Description: Tracks the value of a specific metric over time.
- Use Case: Alert if CPU usage exceeds 90% for more than 5 minutes.
- Example: Monitor system.cpu.user to detect high CPU usage.

Log Monitor

- Description: Alerts based on patterns in your log data.
- Use Case: Trigger an alert if the number of error logs exceeds a threshold.
- Example: Monitor logs with a pattern like @status:error and alert if occurrences surpass 10 in 5 minutes.

Trace Monitor

- Description: Tracks application traces to detect performance issues or errors.
- Use Case: Alert if the average response time of a specific service exceeds a threshold.
- Example: Monitor the latency of an API call and alert if it goes beyond 200ms.

Types of Monitors

Synthetic Monitor

- Description: Tests the availability and performance of your applications by simulating user actions.
- Use Case: Alert if a website is down or a key transaction fails.
- Example: Monitor the uptime of a webpage or the success rate of a login process.

APM Monitor

- Description: Monitors application performance data (APM) for anomalies and performance issues.
- Use Case: Detect if a particular service in your application is experiencing a high error rate.
- Example: Monitor the error rate of a service and alert if it exceeds 5%.

Network Monitor

- Description: Tracks the performance and availability of network devices and interfaces.
- Use Case: Alert if the bandwidth usage on a network interface exceeds 80%.
- Example: Monitor network interface traffic and detect high utilization.

Types of Monitors

Process Monitor

- Description: Alerts based on the status of processes running on your infrastructure.
- Use Case: Trigger an alert if a critical process stops running.
- Example: Monitor a specific process like nginx and alert if it is not running.

Host Monitor

- Description: Monitors the status and performance of individual hosts.
- Use Case: Alert if a host goes offline or experiences high memory usage.
- Example: Monitor a host’s memory usage and alert if it exceeds 90%.

Event Monitor

- Description: Alerts based on events occurring within Datadog or external systems.
- Use Case: Alert when a specific event, like a deployment, is triggered.
- Example: Monitor for deployment events and alert on failures.

Types of Monitors

Service Check Monitor

- Description: Tracks the status of custom service checks integrated with Datadog.
- Use Case: Alert if a custom service check fails or reports an error.
- Example: Monitor a custom service check for database health and alert on failures.

SLO Monitor (Service Level Objective)

- Description: Monitors compliance with defined SLOs, such as availability or error rates.
- Use Case: Alert if an SLO is not being met, such as uptime falling below 99.9%.
- Example: Monitor the uptime SLO for a service and alert if it drops below the defined threshold.

Anomaly Monitor

- Description: Detects anomalous behavior in your metrics using machine learning.
- Use Case: Alert if a metric behaves abnormally compared to historical data.
- Example: Monitor a metric like request_count and alert if it shows an unexpected spike.

Types of Monitors

Outlier Monitor

- Description: Detects outliers in a set of similar metrics, such as hosts or containers.
- Use Case: Alert if one host's CPU usage deviates significantly from others in the same group.
- Example: Monitor CPU usage across multiple hosts and detect if one is significantly higher.

Forecast Monitor

- Description: Predicts future trends based on historical data and alerts if a metric is expected to reach a critical threshold.
- Use Case: Alert if disk space is predicted to run out within the next week.
- Example: Monitor disk usage and forecast if it will exceed 90% in the next 7 days.

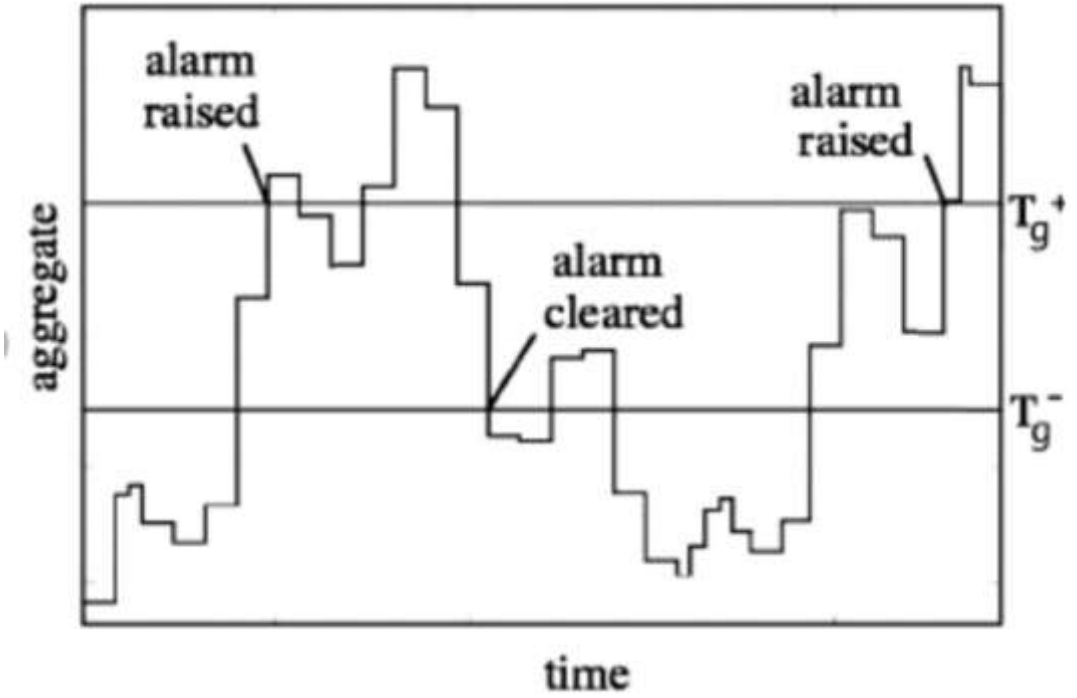
Composite Monitor

- Description: Combines multiple monitors into a single alerting condition.
- Use Case: Alert only if both CPU usage is high and memory usage is low.
- Example: Create a composite monitor that triggers when both system.cpu.user > 90% and system.mem.used < 20%.

Detection Methods

Threshold Alert

- A threshold alert compares metric values to a static threshold.
- On each alert evaluation Datadog will calculate the average/minimum/maximum/ sum over the selected period and check if it is above/below the threshold. This is the standard alert case where you know what sorts of values are unexpected.



Detection Methods

Change Alert

- A change alert evaluates the difference between a value N minutes ago and now.

- On each alert evaluation Datadog will calculate the raw difference (not absolute value) between the series now and N minutes ago then compute the average/minimum/maximum/sum over the selected period. An alert is triggered when this computed series crosses the threshold.

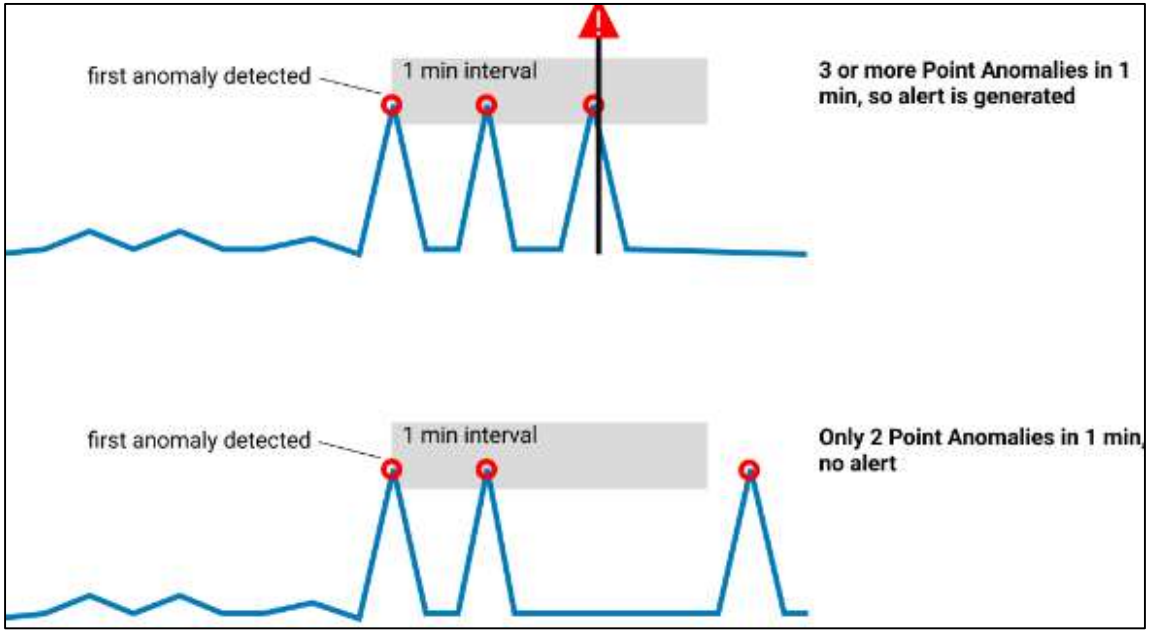
Detection Methods

Anomaly Alert

- An anomaly alert uses past behavior to detect when a metric is behaving abnormally.
- Anomaly alerts calculate an expected range of values for a series based on the past. Some of the anomaly algorithms use the time-of-day and day-of-week to determine the expected range, thus capturing abnormalities that could not be detected by a simple threshold alert (e.g. the series is unusually high for 5AM even though it would be considered normal at 10 AM).
- On each alert evaluation, Datadog will calculate the percentage of the series that falls above/below/outside of the expected range. An alert is triggered when this percentage crosses the configured threshold.

Detection Methods

Anomaly Alert

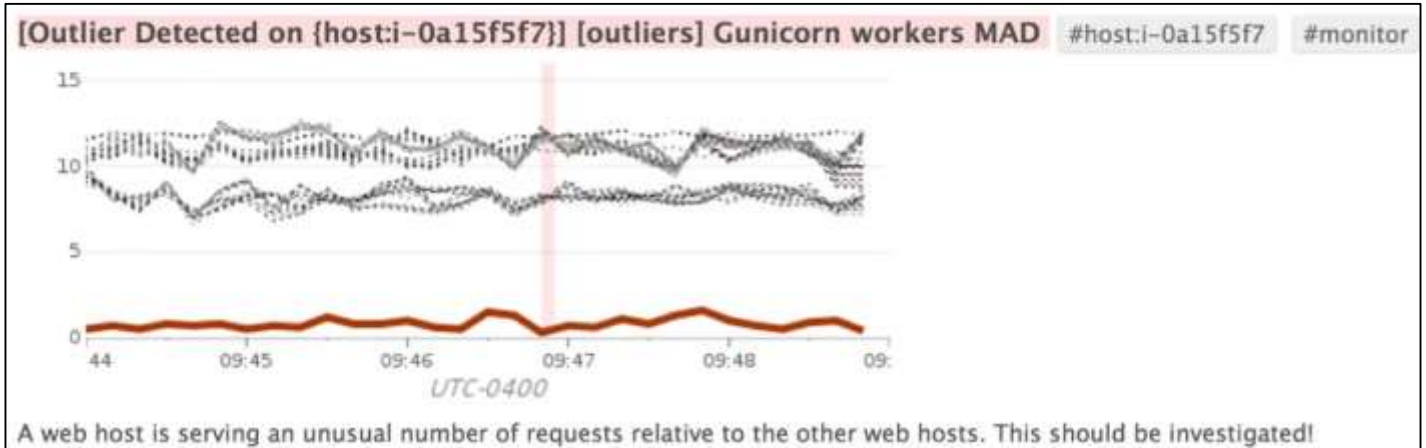


Detection Methods

Outlier Alert

- Outlier monitors detect when a member of a group (e.g., hosts, availability zones, partitions) is behaving unusually compared to the rest.

- On each alert evaluation, Datadog will check whether or not all groups are clustered together, exhibiting the same behavior. An alert is triggered whenever at least one group diverges from the rest of the groups.



Detection Methods

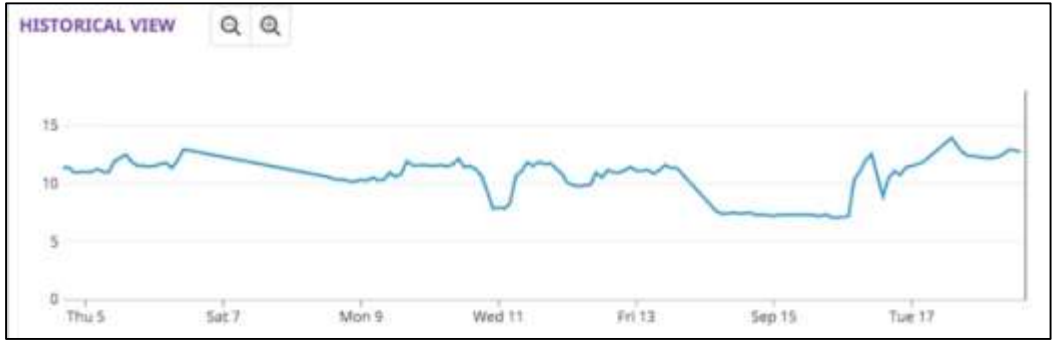
Forecast Alert

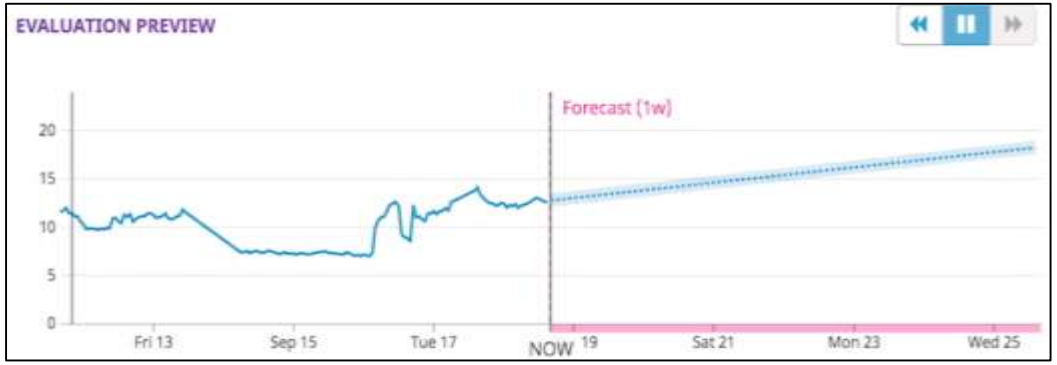
- A forecast alert predicts the future behavior of a metric and compares it to a static threshold.

- On each alert evaluation a forecast alert will predict the future values of the metric along with the expected deviation bounds. An alert is triggered when any part of the bounds crosses the configured threshold.

Detection Methods

Forecast Alert



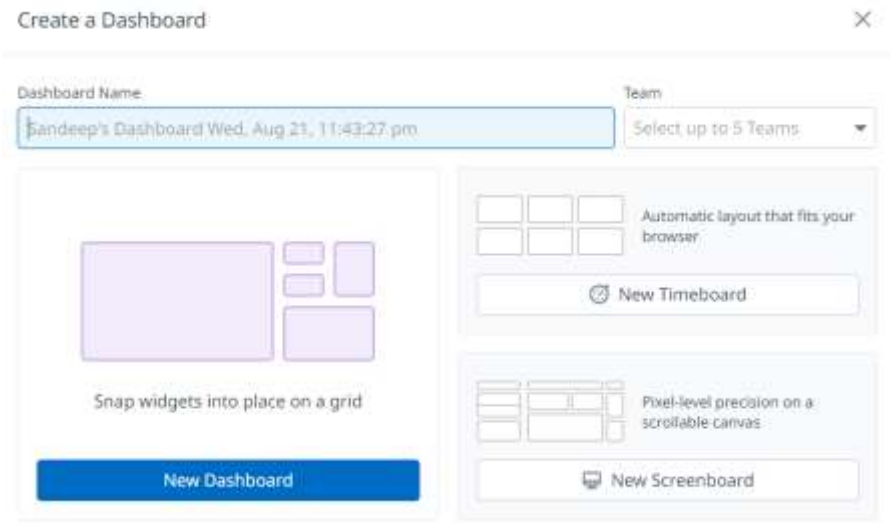


Dashboards

Dashboards

A dashboard is Datadog's tool for visually tracking, analyzing and displaying key performance metrics which enable you to monitor the health of your infrastructure.

To create a dashboard, click +New Dashboard on the Dashboard List page or New Dashboard from the navigation menu. Enter a dashboard name and choose a layout option.



Dashboards

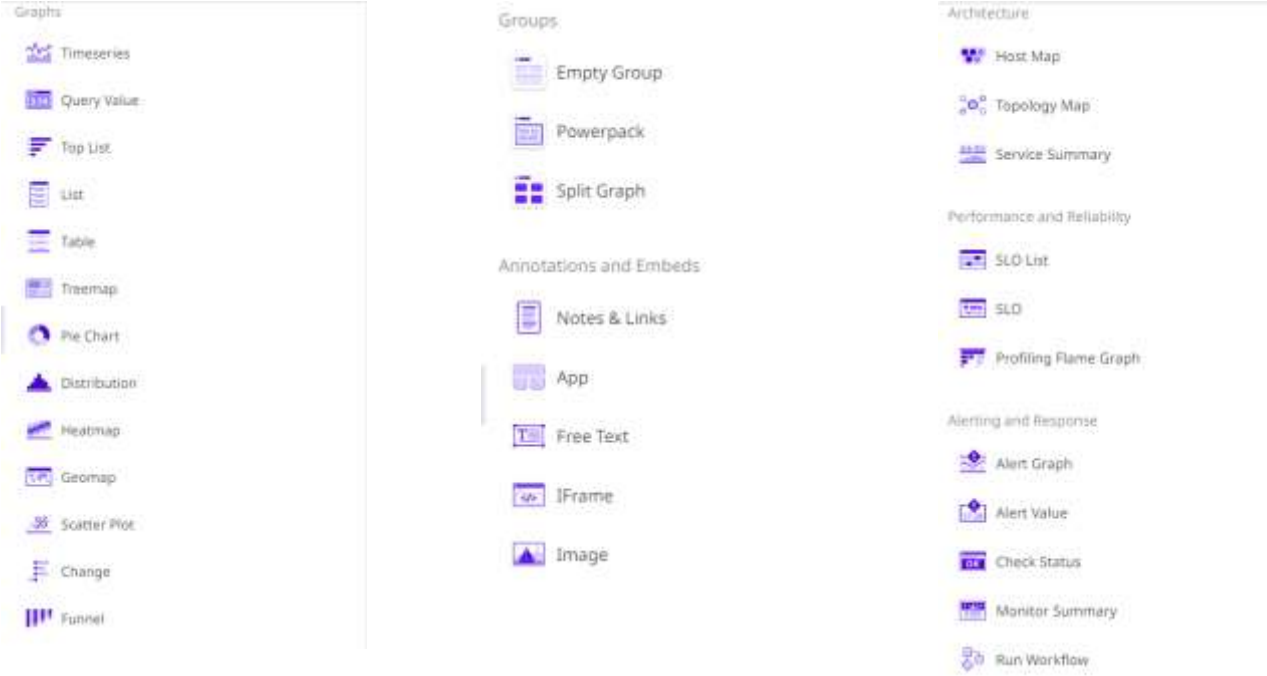
Dashboards: Dashboards are on a grid-based layout, which can include a variety of objects such as images, graphs and logs. They are commonly used as status boards or storytelling views which update in real-time and can represent fixed points in the past. They have a maximum width of 12 gird squares and also work well for debugging.

Timeboards:Timeboards have automatic layouts and represent a single point in time- either fixed or real-time- across the entire dashboard. They are commonly used for troubleshooting, correlation and general data exploration.

Screenboards: Screenboards are dashboards with free layouts which can include a variety of objects such as images, graphs and logs. They are commonly used as status boards or storytelling views that update in real-time or represent fixed points in the past.

Dashboards- Widgets

Widgets are building blocks for your dashboards. They allow you to visualize and correlate your data across your infrastructure.



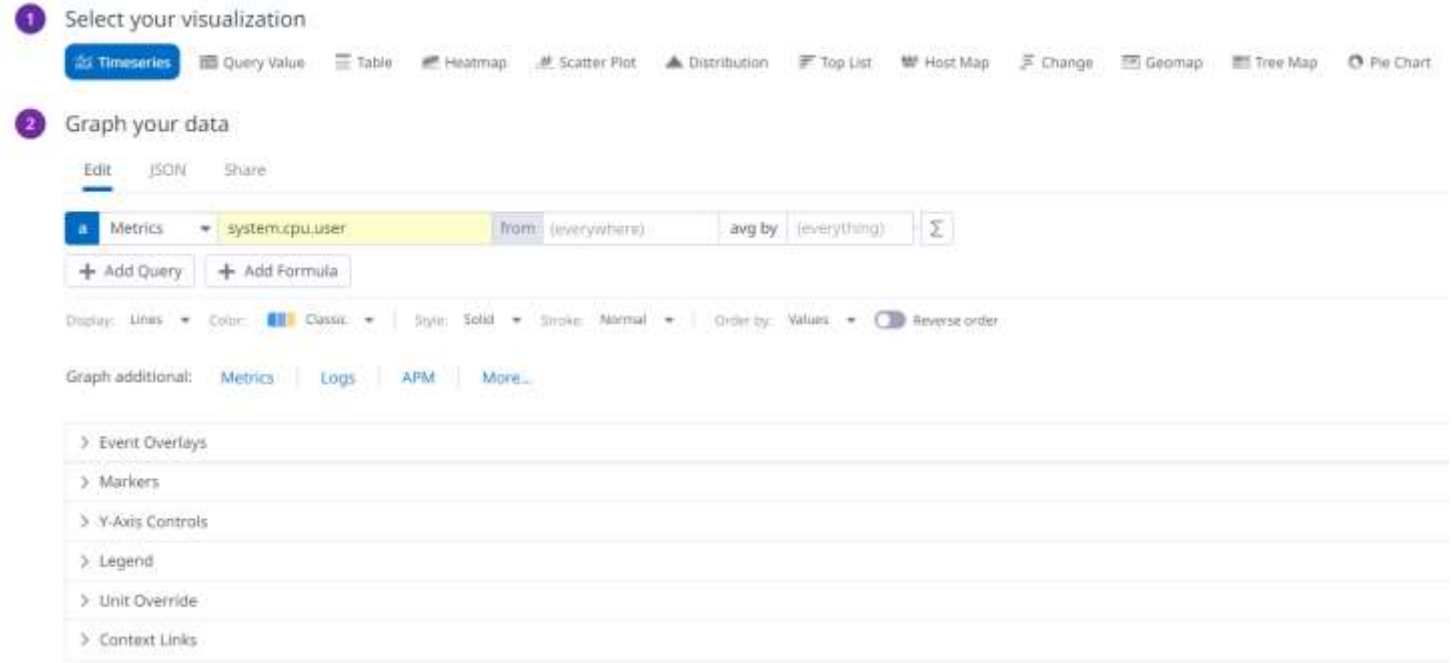
Dashboards Querying

Whether you are using metrics, logs, traces, monitors, notebooks etc, all graphs in Datadog have the same basic functionality.
You can query using the graph editor on the dashboards or notebooks pages, or you can use Quick Graphs available on any page.

Graphing Editor:
On widgets, open the graphing editor by clicking on the pencil icon in the upper right corner, The graphing editor has the following tabs:

Share: Embed the graph on any external web page
JSON: A more flexible editor which requires knowledge of the graph definition language.
Edit: The default UI tab for graphing options.

Dashboards Querying



Dashboards - Functions

Functions can modify how the results of a metric query are returned for visualizations. Most functions are applied after the results of the metric query are returned, but functions can also change the parameters before the query is made.

For example, the Rollup function changes the time aggregation of a query before the results are returned. Alternatively, arithmetic functions apply changes to the returned results of the metric query.

Dashboards – Template Variables

Template variables allow you to dynamically filter one or more widgets in a dashboard. You can build saved views from your template variable selections to organize and navigate your visualizations through the dropdown selections.

A template variable is defined by:

- **Tag or Attribute:**
  Tag: If you follow the recommended tagging format (<KEY>:<VALUE>), the Tag is the <KEY>.
- **Attribute**: Use a facet or measure as the template variable.
- **Name**: A unique name for the template variable that appears in queries on the dashboard. Template variables are automatically named after the selected tag or attribute.
- **Default Value**: The tag or attribute value that appears automatically when the dashboard is loaded. Defaults to *.
- **Available Values**: The tag or attribute values available for selection in the dropdown menu. Defaults to (all). The list of available values always includes *, which queries all values of the tag or attribute.

Network Monitoring

Network Monitoring

- Datadog Network Performance Monitoring (NPM) gives you visibility into your network traffic between services, containers, availability zones, and any other tag in Datadog.

- Connection data at the IP, port, and PID levels is aggregated into application-layer dependencies between meaningful client and server endpoints, which can be analyzed and visualized through a customizable network page and network map.

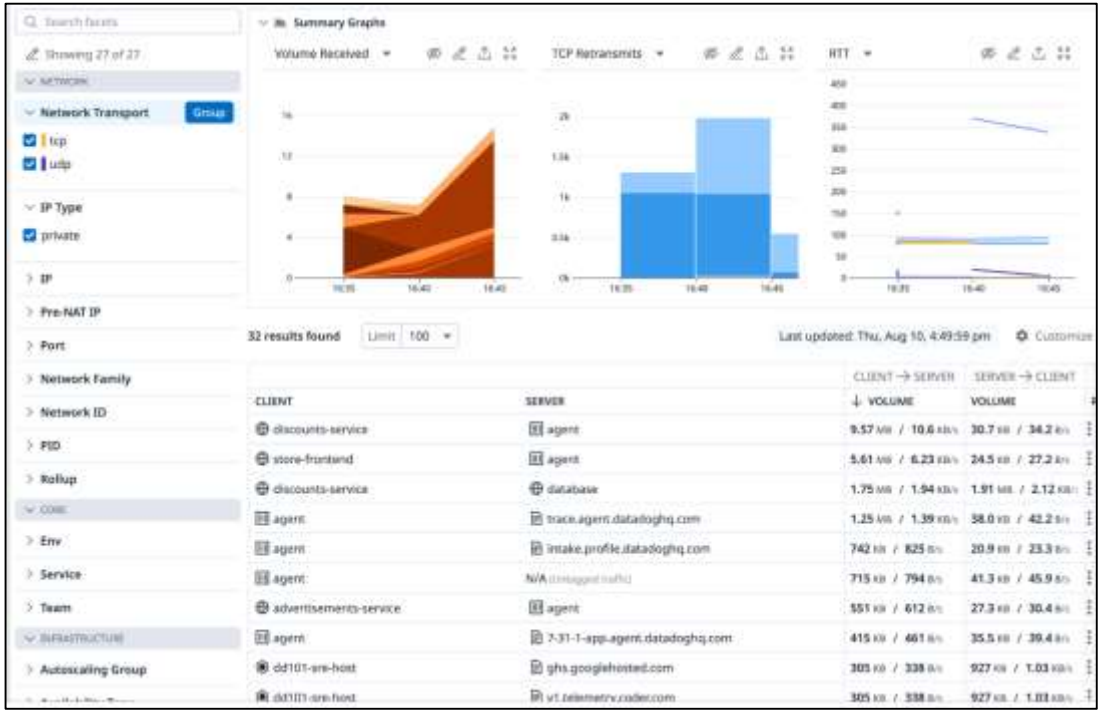
- NPM makes it simple to monitor complex networks with built in support for Linux and Windows OS as well as containerized environments that are orchestrated and instrumented with Istio service mesh.

You can pin point:

- Pinpoint unexpected or latent service dependencies.
- Optimize costly cross-regional or multi-cloud communication.
- Identify outages of cloud provider regions and third-party tools.
- Troubleshoot faulty service discovery with DNS server metrics.

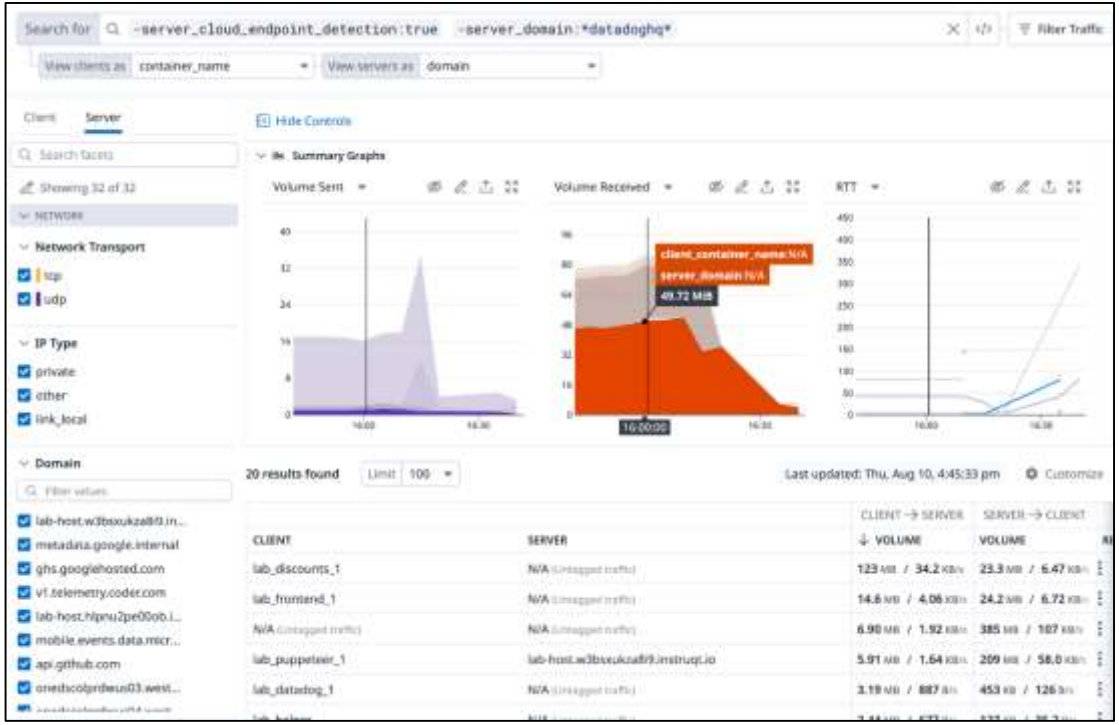
Network Monitoring

- Datadog's Network Performance Monitoring (NPM) provides multi-cloud visibility into network flows in granular detail, while also enabling you to aggregate and monitor that data using any tag available in Datadog. So you can query and aggregate connection metrics between any two objects—from services to availability zones, or from Kubernetes pods to security groups—to provide immediate insight into performance and dependencies.

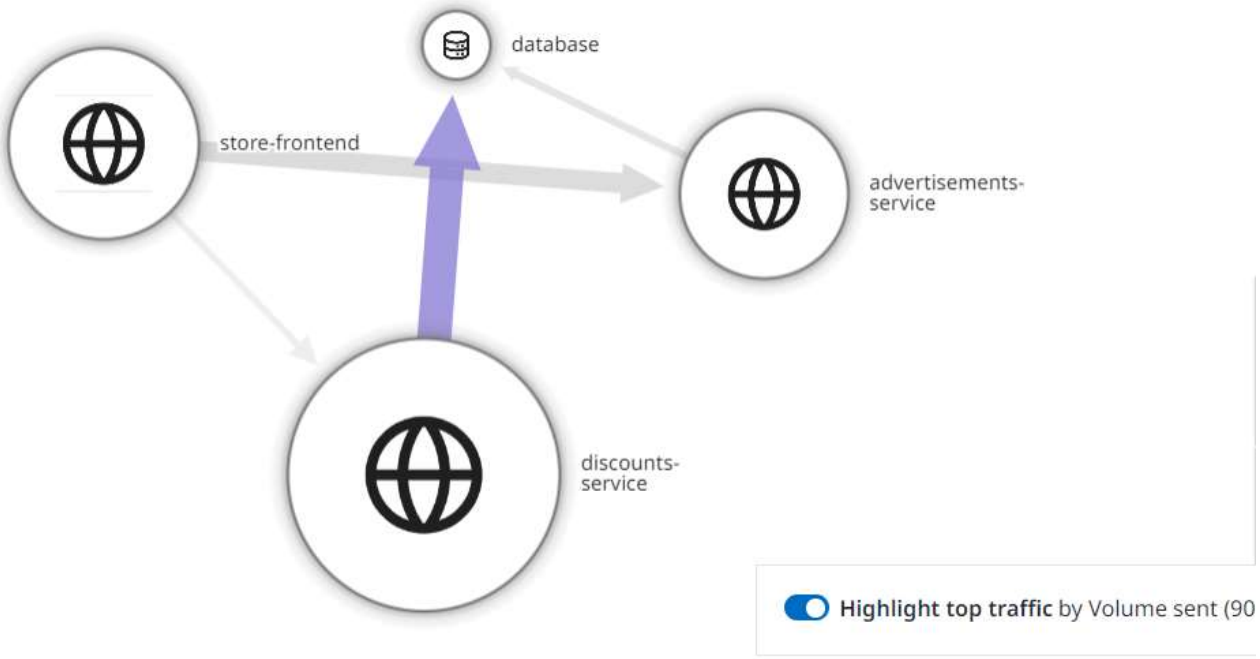


Network Monitoring

- NPM is built on eBPF, which enables detailed visibility into network flows at the Linux kernel level. Consequently, NPM is powerful and efficient with extremely low overhead.
- NPM provides useful domain name information, such as the name resolution of the external IP addresses your services are connecting to, and the health of the DNS servers they use.



Network Map


database
store-frontend
advertisements-service
discounts-service
Highlight top traffic by Volume sent (90

Network Monitoring : Setup

1	Copy the System-probe configuration file
	sudo -u dd-agent install -m 0640 /etc/datadog-agent/system-probe.yaml.example /etc/datadog-agent/system-probe.yaml
2	Edit /etc/datadog-agent/system-probe.yaml to set the enable flag to true:
	network_config: # use system_probe_config for Agent's older than 7.24.1 ## @param enabled - boolean - optional - default: false ## Set to true to enable Network Performance Monitoring. # enabled: true
3	Restart the agent
	sudo systemctl restart datadog-agent

Watchdog

Watchdog

- Watchdog is Datadog's AI engine, providing you with **automated alerts**, **insights**, and **root cause analyses** that draw from observability data across the entire Datadog platform.
- Watchdog continuously monitors your infrastructure and calls attention to the signals that matter most, helping you to detect, troubleshoot, and resolve issues.
- Watchdog proactively computes a baseline of expected behavior for your systems, applications, and deployments. This baseline is then used to detect anomalous behavior.

Watchdog Alerts

- Watchdog proactively looks for anomalies on your systems and applications. Each anomaly is then displayed in the Watchdog Alert Explorer.

1. **Status**: The anomaly can be `ongoing`, `resolved`, or `expired`. (An anomaly is `expired` if it has been ongoing for over 48 hours.)
2. **Timeline**: Describes the time period over which the anomaly occurs.
3. **Message**: Describes the anomaly.
4. **Graph**: Visually represents the anomaly.
5. **Tags**: Shows the scope of the anomaly.
6. **Impact** (when available): Describes which users, views, or services the anomaly affects.

Watchdog Alerts

Watchdog observes the patterns in:

Logs
- New error logs
- Increasing Error logs

APM metrics
- Error rate
- Latency
- Hits (request rate)

Infrastructure
- System, for host-level memory usage (memory leaks) and TCP retransmit rate.
- Redis
- PostgreSQL
- NGINX
- Docker
- Kubernetes
- Amazon Web Services:
- S3
- ELB/ALB/NLB
- CloudFront
- DynamoDB
- RDS
- ECS
- Lambda

Watchdog Impact Analysis

Whenever Watchdog finds an APM anomaly, it simultaneously analyzes a variety of latency and error metrics that are submitted from the RUM SDKs to evaluate if the anomaly is adversely impacting any web or mobile pages visited by your users.

If Watchdog determines that the end-user experience is impacted, it provides a summary of the impacts in Watchdog APM Alert. This includes:

- A list of impacted RUM views
- An estimated number of impacted users
- A link to the list of impacted users, so that you can reach out to them, if needed.

Watchdog RCA

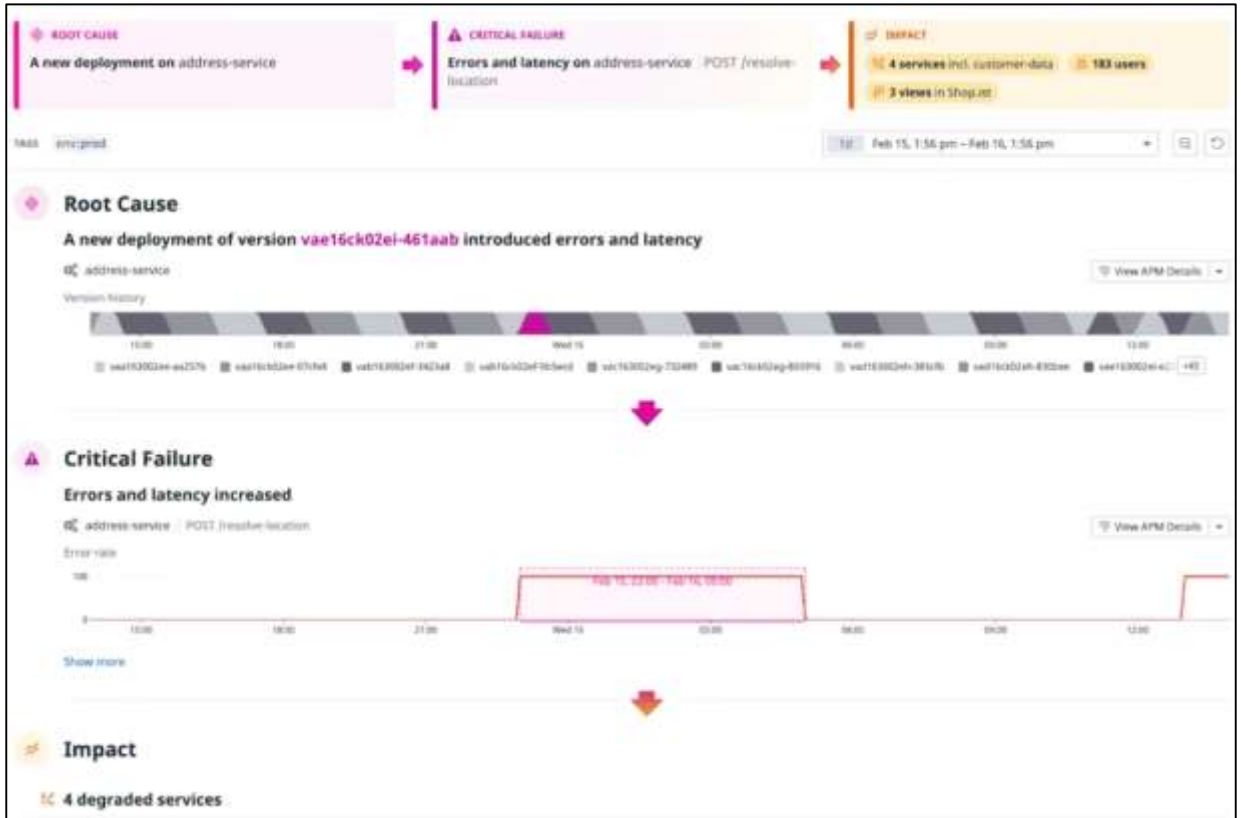
Watchdog Root Cause Analysis (RCA) helps you reduce mean time to recovery (MTTR) by automating preliminary investigations during incident triage. The Watchdog AI engine identifies interdependencies between application performance anomalies and related components to draw causal relationships between symptoms. Whenever Watchdog finds an APM anomaly, it starts a root cause analysis in an attempt to provide deeper insight into the cause and/or effects of the anomaly.

Watchdog RCA requires the use of APM. In order for Watchdog to take full advantage of all relevant Datadog telemetry for impacted services, Datadog recommends that you set up unified tagging.

Watchdog RCA considers the following sources of data in its analysis:

- APM error rate, latency, and hit rate metrics
- APM deployment tracking
- APM traces
- Agent based infrastructure metrics, including CPU usage, memory usage, and disk usage
- AWS instance status check metrics
- Log pattern anomalies

Watchdog RCA



Watchdog RCA

Watchdog supports four types of root causes:

- Version changes, as captured by APM Deployment Tracking
- Traffic increases, as captured by hit rate metrics on your APM-instrumented services
- AWS instance failures, as captured by Amazon EC2 integration metrics
- Running out of disk space, as captured by system metrics from the Datadog agent

Watchdog Insights

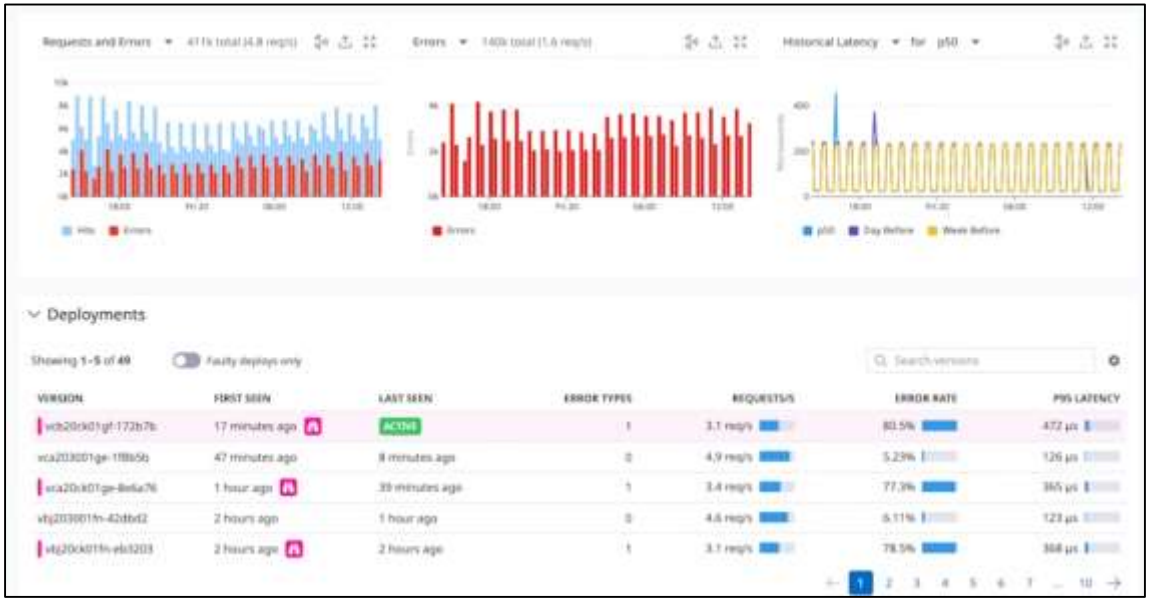
Throughout most of Datadog, Watchdog returns two types of insights:

- **Anomalies**: All the pre-calculated Watchdog alerts matching the active search query that Watchdog found by scanning your organization's data. Access the full list in the Watchdog Alert explorer.

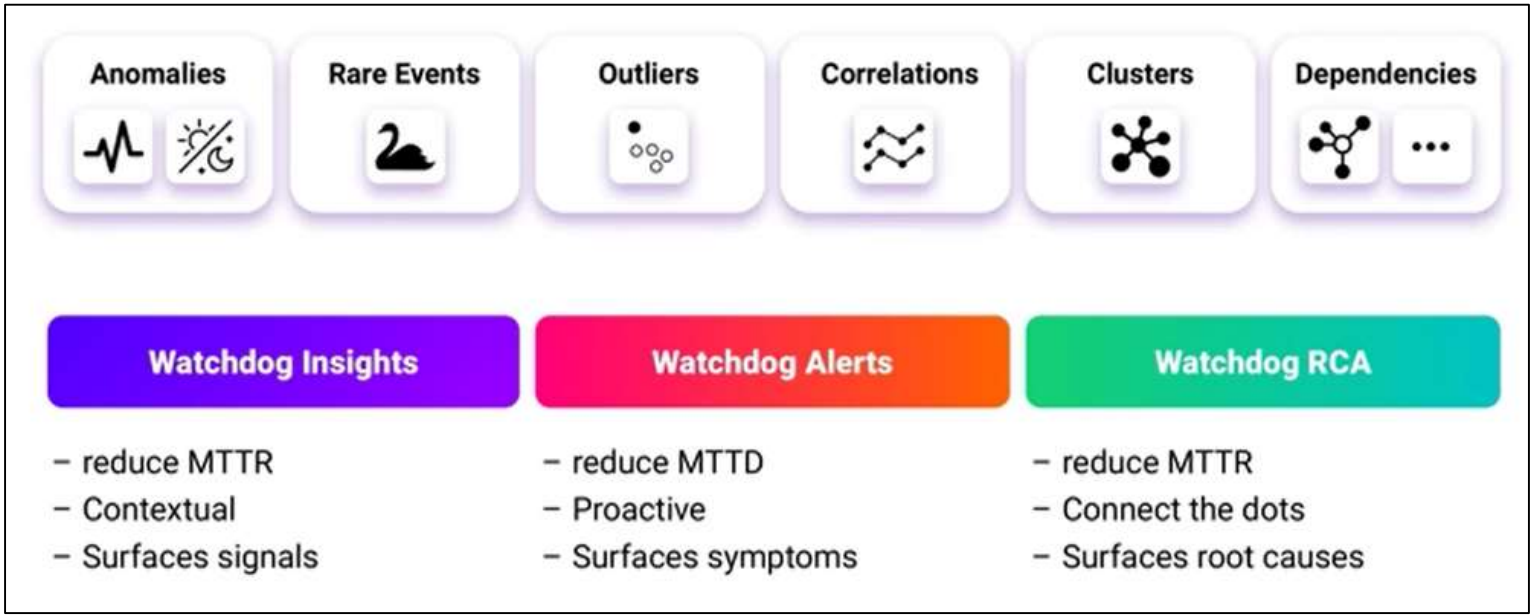
- **Outliers**: Tags that appear too frequently in some event types (for example, errors) or drive some continuous metrics upwards (for example, latency). Outliers are dynamically calculated on the data matching the active query and the time frame.

Watchdog Insights

Whenever code is deployed, Watchdog compares the performance of the new code version with previous versions to spot new types of errors or increases in error rates introduced in a deployment. If Watchdog determines that a new deployment is faulty, details about the affected service appears on the APM service page, as well as the resource page of the affected endpoints.



Watchdog


Anomalies
Rare Events
Outliers
Correlations
Clusters
Dependencies
Watchdog Insights
Watchdog Alerts
Watchdog RCA
– reduce MTTR
– Contextual
– Surfaces signals
– reduce MTTD
– Proactive
– Surfaces symptoms
– reduce MTTR
– Connect the dots
– Surfaces root causes

Datadog Security

Datadog Security

- Datadog Security offers real-time threat detection and continuous configuration audits across applications, hosts, containers, and cloud infrastructure components. And because it is integrated within the larger Datadog observability platform, Datadog Security helps DevOps teams incorporate steps to evaluate and resolve security concerns into existing workflows.

Datadog Security

Datadog Security includes:

Application Security Management (ASM)

- A Datadog product that leverages the Datadog Trace library to let teams defend against and detect application-level attacks targeting code-level vulnerabilities

Cloud Security Information and Event Management (SIEM)

- A Datadog product, powered by Datadog Log Management, that detects real-time threats to your applications and infrastructure

Cloud Security Management Misconfigurations (CSM Misconfigurations)

- A component of Datadog's Cloud Security Management product that uses data from existing cloud integrations to perform continuous configuration checks across your cloud accounts, hosts, and containers.

Cloud Security Management Threats (CSM Threats)

- A built-in feature of the Datadog platform, enabled in the Datadog Agent (7.27.0+), that monitors file, network, and process activity across your environment to detect real-time threats to your infrastructure.

Datadog Security

Application Security Management (ASM)

- Datadog ASM provides visibility into application-level attacks that aim to exploit code-level vulnerabilities and into any bad actors targeting your systems. In addition, ASM detects the risks built into your applications, for example, through vulnerable libraries and dependencies the application uses at runtime.

- Datadog ASM works by drawing upon the same tracing libraries as those used by Datadog Application Performance Monitoring (APM). APM uses these libraries to record traces about each application request. ASM, meanwhile, uses the tracing libraries to monitor your traffic and flag attack attempts.

- ASM can flag attack attempts by virtue of the fact that Datadog formalizes known attack patterns as rules and regularly updates these rules as new attack patterns are identified. (You can also create custom rules to cover your own use cases.) When a rule is triggered, a security signal is created. The signals identify meaningful threats for your review, rather than assessing each individual attack attempt.

Datadog Security

Cloud Security Management Misconfigurations (CSM Misconfigurations)

Datadog Cloud Security Management (CSM) looks for real-time threats throughout your environment, including infrastructure components such as AWS EC2 instances, Docker containers, and Kubernetes cluster nodes. As a part of the Datadog platform, you can combine this real-time threat detection with metrics, logs, traces, and other telemetry to see the full context surrounding a potential attack on your workloads.

CSM Threats relies on the Datadog Agent for its real-time monitoring capabilities. More specifically, CSM leverages the Datadog Agent to perform four types of monitoring:

- Process Execution Monitoring to watch process executions for malicious activity on hosts or containers in real time.
- File Integrity Monitoring to watch for changes to key files and directories on hosts or containers in real time.
- DNS Activity Monitoring to watch network traffic for malicious activity on hosts and containers in real time.
- Kernel Activity Monitoring to watch for kernel-layer attacks like process hijacking, container breakouts, and more in real time.