

AWS Lambda Integration in DataDog vs New Relic

CloudWatch

- CloudWatch is the base line for all purpose monitoring in AWS but it can't monitor non-AWS cloud services or user managed infrastructure. This is a major limiter which hold enterprises with sizable infrastructure, as well as hybrid and multi-cloud applications to solely depend over it.
- For advanced analysis and automation, CloudWatch requires users to export data to other services and to develop custom scripts, lambda functions, business intelligence or machine learning models.
- Organizations depend upon third-party tools such as DataDog or New Relic that offers capabilities that CloudWatch lacks.
- DataDog and New Relic provides unified infrastructure and application performance monitoring suites that include an extensive set of service integration, pre-built dashboards and data analysis models.

Cost Model

- DataDog
 - DataDog has many pricing plans to fit user needs. It calculates fees based on product usage during each calendar month.
 - Most Common Pricing units are as follows
 - Infrastructure Monitoring
 - APM
 - Log Management
 - Security Monitoring
 - Synthetic Monitoring
 - Network Performance Monitoring
 - Real User Monitoring
 - Continuous Profiler
- New Relic
 - Compared to DataDog, New Relic offers two pricing plans
 - New Relic One Pricing Plan
 - Relic Original Pricing Plan
 - Original Pricing Plan was based on subscriptions to specific products like
 - APM
 - Mobile
 - Infrastructure
 - New Relic One Pricing Plan is rather simpler than original pricing plan. It bills users on pricing tier (Standard, Pro, Enterprise), GB of data ingested, and the count of billable.

Key Differences between DataDog and New Relic

Datadog vs. New Relic

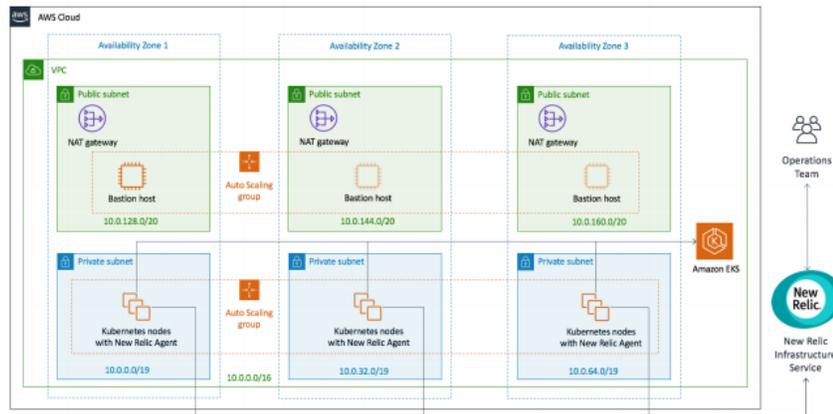
	DATADOG	NEW RELIC
AWS SUPPORT*	<ul style="list-style-type: none"> Support for 70 AWS services 	<ul style="list-style-type: none"> Support for 31 AWS services
OTHER INTEGRATIONS	<ul style="list-style-type: none"> Azure and GCP 	<ul style="list-style-type: none"> Azure, GCP, Java, most popular databases and scripting languages
LOGGING	<ul style="list-style-type: none"> Aggregation across systems, apps and cloud services Visual UI with automatic tagging and event correlation 	<ul style="list-style-type: none"> Aggregation across supported platforms
MONITORING AND METRICS	<ul style="list-style-type: none"> Anomaly detection Outlier detection (instance groups) Trending and forecasting 	<ul style="list-style-type: none"> Anomaly detection Custom metrics can display relationships and correlations between individual measures
APPLICATION TRACES	<ul style="list-style-type: none"> End-to-end transaction tracing across distributed systems App performance summaries with auto-generated service overviews Charts and alerts on key parameters like latency, error rate or performance outside N-percentile boundary Support for open source tracing libraries to provide granular app telemetry 	<ul style="list-style-type: none"> Application tracing that graphically shows calls to different functions, APIs, databases and external systems to help understand application flow and troubleshoot problems Distributed tracing that tracks application requests/ transactions end-to-end as they cross from system to system; this feature is critical to identify latency bottlenecks in composite, distributed applications
DASHBOARDS	<ul style="list-style-type: none"> Real-time charts across data sources Flexible filtering allows charting data by host, device or any other tag Support for computed metrics such as rates, ratios, averages or integrals Customizable via UI or code/configuration file 	<ul style="list-style-type: none"> Customizable dashboards can display data from any source and be built via a graphical UI
ALERTS	<ul style="list-style-type: none"> Alerts on any metric on any host or cluster Notifications via email, PagerDuty, Slack and other collaboration systems Alert syntax enables custom triggers using multiple conditions and conditional logic 	<ul style="list-style-type: none"> Unified UI to manage alerts across infrastructure, applications, Web and mobile apps Notifications via external platforms such as PagerDuty, Campfire, HipChat and Slack Customized incident rollups and prioritized search terms to trigger alerts to any supported notification platform
TEAM COLLABORATION	<ul style="list-style-type: none"> Commenting feature allows annotating logs, metrics, charts and threaded discussions in context 	<ul style="list-style-type: none"> Team dashboards to display SLIs, SLOs and SLAs Can send notifications to several external systems including email, Opsgenie, PagerDuty, Slack, Campfire and Webhooks
EXTENSIBILITY	<ul style="list-style-type: none"> API enables integrating custom applications, accessing or exporting data to other systems 	<ul style="list-style-type: none"> APIs enable programmatic data queries; dashboard creation, updates and deletion; integration with external services; and writing synthetic tests for application performance measurements

Use Case

- **Deploying New Relic Infrastructure on AWS**

- **Architecture**

- The architecture is highly available that spans in three Availability Zones.



- A VPC configured with public and private subnets according to AWS best practices, to provide you with your own virtual network on AWS.
 - In the public subnets, you have to manage NAT gateways to allow outbound internet access for resources in the private subnets.
 - In the public subnets, a Linux bastion host in an Auto Scaling group will be used to allow inbound Secure Shell (SSH) access to Amazon EC2 instances in public and private subnets.
 - An Amazon EKS cluster.
 - In the private subnets, you will have Kubernetes nodes.
 - In the cluster, kube-state-metrics will be present. Kube-state-metrics is an open source project designed to generate metrics derived from the state of Kubernetes objects. The New Relic integration polls the Kubernetes API for kube-state-metrics and uses those metrics for trending, tracking, and correlation.
 - New Relic Infrastructure agent, for collecting telemetry from the Amazon EKS cluster

- **Steps to follow**

- Sign in to your AWS account
 - Choose one of the following deployment options



[Deploy New Relic Infrastructure into a new VPC on AWS](#)



[Deploy New Relic Infrastructure into an existing VPC on AWS](#)

- Check the region that's displayed on the upper right corner of the navigation bar, and change as per your requirement.
- On the **Select Template** page, keep the default setting for the template URL, and then choose Next.
- On the Specify Details page, change the stack name if needed. Review the parameters for the template. Provide values for the parameters that require input. When you finish reviewing and customizing the parameters, choose Next.
- We can deploy New Relic Infrastructure in either
 - New VPC
 - Existing VPC
- On the Options page, you can specify tags (**key-value pairs**) for resources in your stack and set advanced options. It will help you to differentiate during multiple projects. When you're done, choose Next.
- On the Review page, review and confirm the template settings. Under Capabilities, select the check box to acknowledge that the template will create IAM resources.
- Choose Create to deploy the stack
- Monitor the status of the stack. When the status is CREATE_COMPLETE, the New Relic Infrastructure cluster is ready
- Use the URLs displayed in the Outputs tab for the stack to view the resources that were created.
- Test the deployment
- If you not seeing data in your new Relic Account

```
kubectl get pods --all-namespaces -o wide | grep newrelic
```

- You can confirm kube-state-metrics installed successfully

```
kubectl get pods --all-namespaces | grep kube-state-metrics
```

Author: Giten Mitra, Cloud Solution Architect - MIND

Date: 06/01/2021