ISSN: 2754-6659

Journal of Artificial Intelligence & Cloud Computing

SCIENTIFIC Research and Community

Review Article

Open Access

Automated SLA Monitoring in AWS Cloud Environments - A Comprehensive Approach Using Dynatrace

Lakshmi Narasimha Rohith Samudrala

AVCO Consulting, Inc, USA

ABSTRACT

In the quickly developing landscape of cloud computing, ensuring consistent performance, reliability, and security is critical for organizations leveraging cloud services. Service Level Agreements (SLAs) form the foundation of this assurance, detailing the expected service levels across various metrics such as uptime, response time, error rate, and throughput. This paper presents a comprehensive framework for automating SLA monitoring within Amazon Web Services (AWS) environments using Dynatrace, a leading software intelligence platform. By integrating AWS with Dynatrace, organizations can achieve real-time observability and proactive management of SLAs through the use of Service Level Indicators (SLIs) and Service Level Objectives (SLOs). The proposed methodology not only facilitates continuous monitoring and early detection of potential SLA breaches but also enables IT teams to take corrective actions promptly, thereby minimizing impact on end-users and optimizing resource utilization. The paper further explores the implementation of automated SLA monitoring in AWS, including the creation of SLOs, the application of AI-driven analytics for anomaly detection, and the development of dashboards for comprehensive SLA management. Through this approach, organizations can significantly enhance service quality, align cloud resources with business goals, and maintain a competitive edge in a dynamic cloud environment.

***Corresponding author**

Lakshmi Narasimha Rohith Samudrala, AVCO Consulting, Inc, USA.

Received: March 03, 2023; **Accepted:** March 08, 2023; **Published:** March 13, 2023

Keywords: Automated SLA Monitoring, AWS Cloud Environments, SLAs, SLIs, SLOs, Dynatrace, Cloud Observability, Real-Time Monitoring, AI-Driven Analytics, Predictive Maintenance, Performance Management, Infrastructure Monitoring, Proactive Management, Multi-Cloud Monitoring

Introduction

In today's dynamic cloud computing landscape, the ability to ensure consistent performance, reliability and security of these cloud-based environments becomes paramount. Service Level Agreements (SLA) serve as an agreed upon commitment between cloud service providers and the customers using the service [1]. SLAs detail the expected level of service across various metrics such as uptime, throughput, response time, error rate [2]. These agreements are vital for maintaining customer trust and ensuring robust and reliable operations.

In the context of Amazon Web Services (AWS), monitoring SLAs involves tracking a wide variety of metrics that measure the health of all different services and components in AWS like EC2 instances, RDS latency, Lambda Functions, etc [3]. However, manual monitoring of these metrics can be labor-intensive, prone to errors, and reactive rather than proactive. To address these challenges, the use of an automated monitoring tool like Dynatrace is very beneficial. Dynatrace provides a comprehensive observability by integrating into AWS to automatically monitor and manage SLAs across the cloud environments [4].

Dynatrace provides all the monitored AWS metrics as Service Level Indicators (SLI) [5]. These SLIs can be used to define the SLA objective in Dynatrace. These objectives are called as Service Level Objective (SLO). The SLOs provide automatic monitoring and alerting on the objective of the agreed upon SLA [6].

Key Terminology

- **Service Level Agreement (SLA):** A formal agreement between a service provider and a customer that defines the expected performance and availability of a service provided. SLAs often include specific metrics, such as uptime, throughput, error rate and response times. The agreed upon performance must be met by the service provider in order to avoid penalties or service credits [6-8].
- **Service Level Indicators (SLI):** A measurable metric that provides data on the performance or health of a service such as percentage requests successfully processed under a given timeframe [6,8].
- **Service Level Objectives (SLO):** A target value or range defined for an SLI. This defines the acceptable level of performance for the service provided. For example, 95% of request must respond under 200ms [6,8].
- **Amazon Web Services (AWS):** A comprehensive cloud computing platform provided by Amazon that offers a wide array of services, including computing power, storage options, and networking capabilities, among others.

- **Dynatrace:** A software intelligence platform that provides application performance management (APM), infrastructure monitoring, and digital experience management, with the capability to integrate deeply with cloud computing environments like AWS for automated monitoring and observability.
- **Error Budget:** An error budget is a concept used to quantify the permissible amount of failure or downtime that a service can experience while still meeting its agreed-upon service levels [6,8].
- **Burn Rate:** The burn rate refers to the speed at which a service is consuming its error budget [6,8].

Literature Review

The evolution of cloud computing has led to the widespread adoption of Service Level Agreements (SLAs) as a mechanism to ensure the reliability of cloud services. SLAs are formal agreements between service providers and consumers, defining the quality the provided service is expected to meet [3]. Multiple studies have explored various aspects of SLA monitoring and management within cloud environments, highlighting the challenges and proposing frameworks to enhance the reliability and effectiveness of cloud services [2].

One important area of focus is the dynamic monitoring of SLAs to ensure that service providers meet their contractual agreements [9]. For instance, a cloud service monitoring system using web services technology has been proposed for dynamic SLA monitoring [1]. This approach highlights the importance of service reliability through continuous oversight, ensuring that the quality of service promised in the SLA is delivered consistently [9].

In the context of smart grids and energy systems, the integration of multiple stakeholders in a deregulated market poses unique challenges for SLA monitoring. The introduction of new stakeholders brings new challenges for monitoring key performance indicators (KPIs) across pre-negotiated SLAs. To address this, a framework has been proposed for active monitoring based on SLAs. This allows automated and trustworthy coordination of information among stakeholders. This framework highlights the necessity of SLA-based monitoring in environments where multiple parties interact, ensuring that all negotiated SLAs are met [2,7].

The importance of SLA management is also apparent in web services, where the ability to guarantee service quality can substantially enhance service delivery [7]. SLA contracts in web services define a range of quality metrics that ensure services are executed effectively. Managing SLAs in this context requires careful consideration of the SLA life cycle, which extends beyond the service runtime environment [10]. A specialized SLA management platform has been designed to support the definition, registration, and runtime monitoring of SLAs, emphasizing the need for a structured approach to managing the entire SLA life cycle in web services [10].

The dynamic nature of cloud computing further complicates SLA monitoring, especially when consumers use services from multiple providers. Where each cloud environment has their own distinct monitoring policies. This makes proving SLA violations particularly challenging. To assist consumers in overcoming such issues, advanced SLA management strategies that incorporate logical representation and reasoning techniques have been proposed [11]. By leveraging these techniques, the proposed monitoring approach allows for semantic modeling of SLAs, enabling more accurate detection of SLA violations and ensuring a reliable Quality of Service (QoS) [10].

In summary, the literature on SLA monitoring in cloud environments highlights the critical need for dynamic, automated, and trustworthy monitoring systems. As cloud computing continues to evolve, the development of sophisticated and AI-driven SLA management frameworks that can adapt to the intricacies of modern cloud services will be essential for maintaining service reliability and meeting user expectations.

Methodology

This paper presents a comprehensive approach to automating Service Level Agreement (SLA) monitoring specifically within AWS cloud environments using Dynatrace. The primary goal is to ensure that SLAs are consistently met and preemptive warnings are generated when SLAs are at the risk of being breached, allowing IT teams the necessary time to make corrective actions and avoid negative impacts to end-users and business operations. By integrating AWS with Dynatrace, organizations can achieve highly efficient and proactive approach to maintaining service quality and reliability.

Implementation

The implementation of automated SLA monitoring involves of the following steps:

Integrate AWS with Dynatrace

Monitoring AWS using Dynatrace contains the following steps: First, create the necessary IAM Roles. Dynatrace requires two specific IAM role. The first role is assumed by the Dynatrace ActiveGate and provides the necessary permissions to access the cloud watch metrics and other AWS services necessary for monitoring. The second role is assumed by the monitored AWS account and it allows Dynatrace to gather data from the specific AWS services within the account.

Next, connect the AWS Account to Dynatrace. After the AWS access is granted to Dynatrace via IAM, the AWS account can be connected to Dynatrace from the "Cloud and Virtualization" section in Dynatrace. The "AWS" tab under cloud and virtualization section in Dynatrace provides an option to connect a new AWS instance. In order to make this connection, the earlier created monitoring role would be needed along with the account ID for the AWS account that is to be monitored. Upon providing these details, the connect option can be used to verify the connectivity between AWS and Dynatrace. Once the connection is successfully verified and saved, the AWS account related monitoring data would be populated in the "AWS" tab of Dynatrace under "Infrastructure Observability" section [3,4].

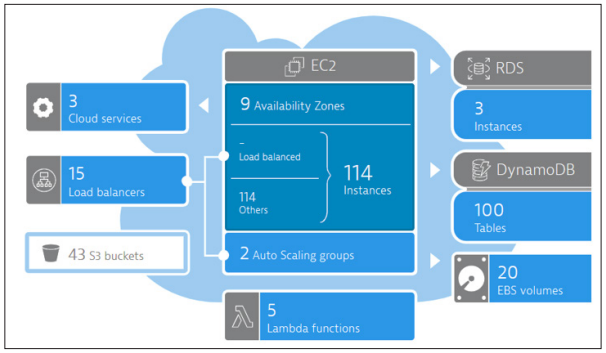

Figure 1: AWS Services Monitored by Dynatrace

Figure 1 depicts AWS account related monitoring data populated in the "AWS" tab under "Infrastructure Observability" section in Dynatrace.

Identify and Define AWS-Specific SLAs and their Corresponding SLIs to Measure Key Aspects of the Service

While considering reliability and performance of cloud-based services hosted on AWS the following factors could prove to be key for SLA considerations:

• Uptime/ Availability

Objective: Ensure that services are available 99.9% of the time.
SLI: builtin:host.availability
Calculation: Uptime percentage is calculated by dividing the total available time minus downtime by the total available time, then multiplying by 100.

• Response Time

Objective: Maintain an average response time of ≤ 200ms.
SLI: builtin:service.response.time
Calculation: Average response time is calculated by dividing the sum of all response times by the total number of requests.

• Latency

Objective: Ensure 99th percentile latency is ≤ 300ms.
SLI: builtin:service.latency
Calculation: 99th percentile latency is calculated by identifying the value below which 99% of the latency measurements fall.

• Throughput

Objective: Achieve a throughput of ≥ 1,000 requests per second.
SLI: builtin:service.request.count.total
Calculation: Throughput is calculated by dividing the total number of requests by the total time in seconds.

• Error Rate

Objective: Maintain an error rate of ≤ 1% across all services.
SLI: builtin:service.errors.perminute
Calculation: Error rate is calculated by dividing the total number of errors by the total number of requests, then multiplying by 100.

• Function Invocation Success Rate

Objective: Ensure that the Lambda functions successfully executes 99% or higher
SLI: builtin:cloud.aws.lambda.errorsRate
Calculation: Success rate is calculated subtracting errored invocations from total invocations and dividing that by the total number of invocations and multiplying by 100.

• Function Latency (Duration)

Objective: Ensure 95% or more of invocations are completed under 500ms
SLI: ext:cloud.aws.lambda.duration
Calculation: The success of the objective is calculated by getting a count of invocations that have completed under 500ms and dividing the count by the total number of invocations, then multiplying by 100

• Lambda Concurrency Limits

Objective: Ensure that functions scale appropriately without hitting the concurrency limits (0% throttling).
SLI: builtin:cloud.aws.lambda.throttlers, builtin:cloud.aws.lambda.invocations
Calculation: Throttling percentages can be calculated by dividing the number of throttles by the total number of invocations, then multiplying by 100.

Establish the SLO in Dynatrace

Dynatrace provides a user-friendly interface to create an SLO. The SLO creation widget in Dynatrace can be found in the "Automations" section of the menu. The handy menu search makes it easy to find the SLO creation UI in Dynatrace. Figure 2 shows the menu search functionality in Dynatrace. Where in "SLO" is typed to find the "Service-Level Objective" option.

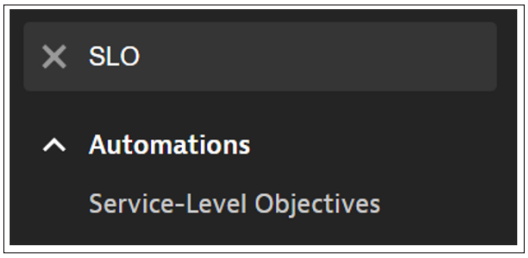

Figure 2: Open Service-Level Objective Creation User-Interface in Dynatrace

A new SLO creation can either be done using a template or by defining a metric. Using a template provides a starting point for the development of the SLO. In this case, the metric definition approach is used.

Upon opening a new SLO creation page, name/ define the SLO and name the metric that is to be created. Note that the metric name cannot be modified once created. It is recommended to follow a naming convention for metrics across the platform. Figure 3 shows the name, description of the SLO and name of metric for the SLO.

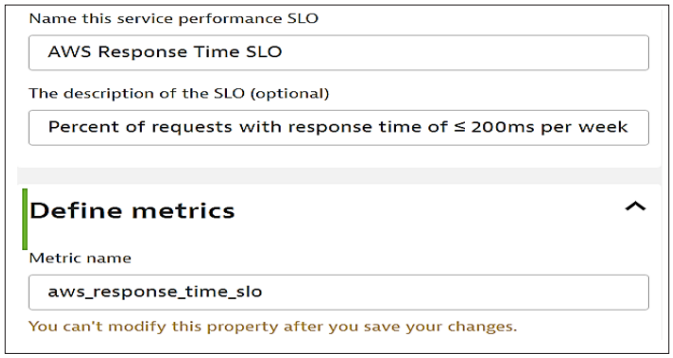

Figure 3: Name/ Description of the SLO and Name of the Metric

After naming the metric, an expression for the objective has to be defined. The metric expression is the calculation for the SLO using the selected Dynatrace metrics (SLI).

Consider the following examples:

Response Time: The Response Time SLO utilizes "builtin:service.response.time" SLI. The percentage for response time is calculated by creating a partition of services that have an average response time of less than 200ms, marked as "good" and dividing the count by the total number of services, multiplied by 100. Figure 4 showcases the aforementioned calculation as metric expression.

```
((builtin:service.response.time:avg:partition(
"latency",value("good",lt(200000))):splitBy():
count:default(0))/(builtin:service.response.ti
me:avg:splitBy():count)*
(100)):default(100,always)
```

Figure 4: Metric Expression for Response Time SLO

Function Invocation Success Rate: The function invocation success rate uses "builtin:cloud.aws.lambda.errorsRate" metric as SLI. The success rate is automatically calculated by the metric out of the box in Dynatrace. Therefore, no additional calculation is required in this case. Figure 5 showcases how this SLI translates into a metric expression.

```
builtin:cloud.aws.lambda.errorsRate:splitBy()
```

Figure 5: Metric Expression for the Function Invocation Success Rate SLO

Post the definition of the metric expression a couple of filters may need to be defined. First, is a timeframe filter. The timeframe filter allows to define the timeframe for which the SLO needs to be evaluated. The timeframe for the SLO depends on the use case. It is generally recommended to have one week rolling timeframe. The timeframe selector is capable of handling simple to complex expressions. For example, "-1w" for the last 1 week or "-10d to now" for the last 10 days to now. Figure 6 shows the timeframe selector evaluating the SLO for 1 week.

Figure 6: SLO Evaluation Timeframe Selector

Second, is an option filter that can be used to narrow down the scope of the SLO. The entity selector needs an entity type for the SLO and an entity filter that would narrow the scope of the SLO. In Figure 7, the scope is narrowed down to a particular management zone for the entity type "AWS_LAMBDA_FUNCTION".

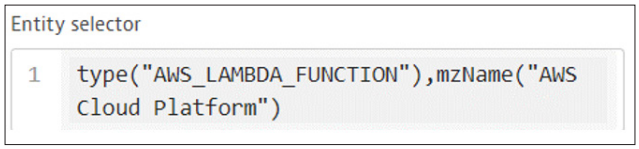

Figure 7: Entity Selector for the SLO

The final step in creating an SLO is defining the success criteria. The success criteria section of SLO has two parts. First, is setting the "Target" and "Warning" thresholds. If the percentage is lower than "Target", then the SLO is evaluated as "failure" and if the percentage is higher than "Warning", then the SLO is evaluated as "good". The second step of the success criteria and the final step in the SLO creation is enabling the calculation of error budget burn and modifying the burn rate if needed. Figure 8 below provides an example of the success criteria creation for the "Response Time" SLO.

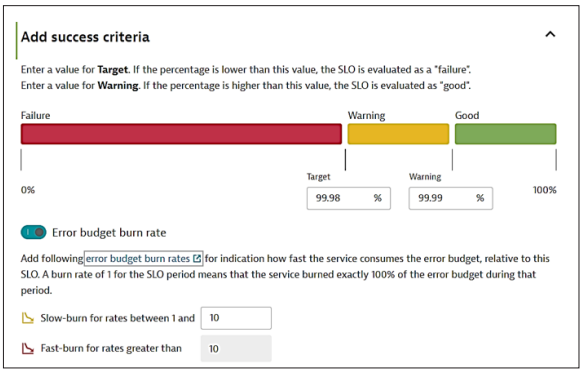

Figure 8: Success Criteria for the Response Time SLO

After successfully entering all the needed details for the creation of the SLO, the "Evaluate" button on the page can be used to shows an evaluation of the new SLO based on the entered values. Figure 9 shows the evaluation of the "Response Time" SLO.

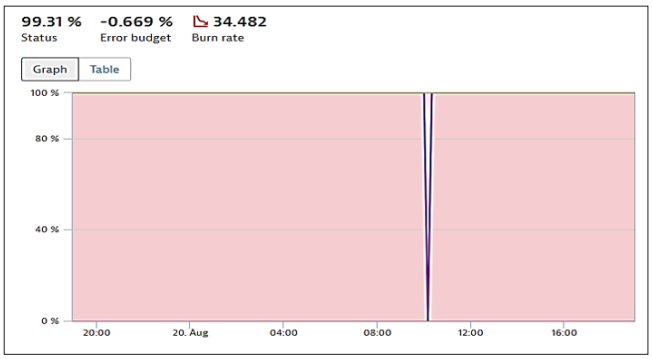

Figure 9: Evaluation of the Response Time SLO

Successfully created SLOs show up under the Service-Level Objective page in Dynatrace. The Service-Level Objective page in Dynatrace provides an overview of the SLO at a glance, it contains information such as the color-coded status of the SLO (Green = Good status, Yellow = Warning status, Red = Failure status), Error budget for the SLO, Target threshold, Warning threshold, Open/total problem (alerts generated for the SLO), Evaluation timeframe.

Create Alerts for the SLO

Alerts for the defined SLOs can be created from the Service-Level Objective page in Dynatrace. The perform certain actions on the SLO. One such action is creation of alerts for the SLO. Figure 10 shows the "Actions" option and the actions that can be performed for the SLO.

Figure 10: "Actions" Option and the Allowed Options

Clicking on the "Create alert" action, opens an alert creation widget. This widget allows to choose between alerting on the status of the SLO or the burn rate. The title of the alert and the threshold at which the alert should trigger can also be defined here. This can be used o generate warning and critical alerts for the SLO by defining the alert trigger thresholds.

Create Dashboard for SLA Monitoring

One of the actions that can be performed from the "Actions" option in the SLO overview page is the creation and pinning of the SLO to a dashboard. Clicking on the "Pin to Dashboard" action provides a list of dashboards, once the dashboard is selected the SLO tile is pinned to the dashboard. If a pre-existing dashboard is already not present, then a dashboard can be created from the "Pin to Dashboard" widget directly. The pinned tile provides color-coded percentages for status and error budget. It also provides the target for the SLO. As the SLO status directly corelates to the SLA adherence, adding all the SLOs to the dashboard provides a single pane of glass for monitoring the SLA adherence status of AWS in the organization.

Conclusion

In conclusion, automating SLA monitoring for AWS cloud environments using Dynatrace provides a powerful and efficient approach to maintaining high levels of service quality, reliability and performance. By integrating AWS with Dynatrace, organizations can achieve near real-time view into the health of their cloud services. The establishment of well-defined SLOs (Service-Level Objectives) with AWS specific metrics in Dynatrace ensures that key aspects of AWS such as uptime, response times, error rate, latency, throughput, lambda functions invocation success rate, lambda functions response time and throttling are continuously monitored and managed.

This approach enhances the ability to detect potential issues before they can impact the end-users. It also enabled proactive management of resources, leading to a better cost efficiency and stronger alignment with business goals [12]. With the use of Dynatrace's advanced monitoring capabilities IT teams can focus on strategic initiatives knowing that the AWS environments are consistently meeting the agreed upon service levels.

Ultimately, the implementation of automated SLA monitoring in AWS using Dynatrace represents a significant step forward in cloud service management, offering a robust framework for ensuring that services remain reliable, performant and secure in an increasingly dynamic and complex cloud landscape.

Future Work

The evolving landscape of cloud computing and service level management presents numerous opportunities for advancements in the domain of automated SLA monitoring, particularly within AWS environments using Dynatrace [11].

Enhanced AI-Driven Anomaly Detection

Although Dynatrace currently offers an AI-based anomaly detection, future research could focus on enhancing predictive and preventive capabilities of the AI-Engine. Using AI and ML to analyze historic data and identify patterns allows organizations to detect and predict potential SLA breaches. Therefore, allowing them to take preemptive actions before service degradation occurs. This would further reduce downtime and improve overall service reliability.

Impact of Edge Computing on SLA Monitoring

With the growth of edge computing and IoT deployments, there is a need for research into how automated SLA monitoring can be adapted to these decentralized and often resource-constrained environments. Future work could explore the application of Dynatrace in monitoring SLAs for edge devices and IoT networks, ensuring reliability and optimal performance in highly distributed systems.

Dynamic SLA Adjustment Based on Real-Time Conditions

Due to the dynamic nature of cloud environments, there is a need for adaptive monitoring strategies that can automatically adjust SLOs based on changing workloads and resource availability [8]. Research in this area could focus on developing adaptive algorithms that respond to real-time changes in cloud infrastructure, ensuring that monitoring remains effective even as the environment evolves [10].

Evolution of SLIs

As cloud environments continue to grow, the scope for further research and development in automated SLA monitoring within AWS using platforms like Dynatrace is enormous. One key area for future work involves enhancing the Service Level Indicators (SLIs), as the services and features provided by AWS grows at a rapid pace. There is a need to develop more sophisticated metrics that can capture the nuances of these services. This would require an ongoing effort to refine existing monitoring frameworks and integrate them seamlessly with new AWS offerings [13].

References

1. Xuan Liu, Feng Xu (2013) Cloud Service Monitoring System based on SLA. IEEE Conference Publication https://ieeexplore.ieee.org/document/6636434.
2. Mahbub K, Spanoudakis G, Tsigkritis T (2011) Translation of SLAs into Monitoring Specifications. Springer eBooks 79-101.
3. (2017) Set up Dynatrace Managed for AWS monitoring. Dynatrace Managed Docs https://docs.dynatrace.com/managed/shortlink/aws-managed-deployment.
4. (2023) Amazon Web Services monitoring. Dynatrace Docs https://docs.dynatrace.com/docs/platform-modules/infrastructure-monitoring/cloud-platform-monitoring/aws-monitoring.
5. Zenuni A (2022) Service Level Objectives (SLOs) at Scale (Tips and Tricks). Dynatrace News https://www.dynatrace.com/news/blog/slos-at-scale/.
6. Gunja S (2022) What are SLOs? How service-level objectives work with SLIs to deliver on SLAs. Dynatrace News https://www.dynatrace.com/news/blog/what-are-slos/.
7. Shahid H, Rune G, Arshad S, Lars N (2014) SLA conceptual framework for coordinating and monitoring information flow in Smart Grid. IEEE Conference Publication https://ieeexplore.ieee.org/document/6816470.
8. Judkowitz J, Carter M (2018) SRE fundamentals: SLAs vs SLOs vs SLIs. Google Cloud Blog https://cloud.google.com/blog/products/devops-sre/sre-fundamentals-slis-slas-and-slos.
9. Taher L, Achraf M, Walid G, Samir T, Faiez G (2017) Cloud SLA modeling and monitoring. IEEE Conference Publication https://ieeexplore.ieee.org/document/8035003.
10. Shu Zhang, Meina Song (2010) An architecture design of life cycle based SLA management. IEEE Conference Publication https://ieeexplore.ieee.org/document/5440283.
11. Giuseppe A, Alessio B, Walter de Donato, Antonio P (2012) Cloud monitoring: Definitions, issues and future directions.

IEEE Conference Publication https://ieeexplore.ieee.org/document/6483656.

12. Rajarajeswari CS, Aramudhan M (2013) Resource provisioning for SLA management. IEEE Conference Publication https://ieeexplore.ieee.org/document/6938707.
13. Fei G, Xuesong Q, Luoming M (2010) Business-driven adaptive SLO maintenance in service level management. IEEE Conference Publication https://ieeexplore.ieee.org/document/5486585/.