

AWS CloudWatch 🍀 : CMAA | Metric | Alarms | Namespaces | Events



Amazon Cloudwatch

When an application is designed, we don't have an idea of how the application is going to perform in the future, what impact it will create, how many people will engage with the application.

Considering global deployment of the application it is little tough to manage the crashing of the application. Hence the users may start facing problems with the application. This happens because of short-sightedness and sometimes because of lack of the budget.

Considering microservices architecture for such application with thousands of APIs which works in tandem with each other. In this case there could be failure related to the API on the service, the authentication can fail and also sometimes it can be the issue of CPU utilization and memory consumption which makes the application to crash.

If this problem would have been identified on a timely basis or at an early time then the mess or the whole crash of application would have been restricted by taking necessary steps at the early crash stage.

For that we need a service that can help us collect the data points or logs. This can help us monitor the current state or the state over a period of time and help us create action items to mitigate the issue and the same way lets us analyze the data we have in order to avoid such issues in the future and that's where **cloud watch** comes into the picture.

AWS CloudWatch gives us a proper logging mechanism and monitoring mechanism. It allows to collect the data points, monitor them act on them automatically and this will help us avoid the problems that may occur with the application in the coming future.

What is Amazon CloudWatch?

Amazon CloudWatch is a monitoring service provided by Amazon Web Services (AWS) that provides data and operational insights for various AWS resources. It helps you monitor your resources, applications and services running on the AWS cloud, so you can troubleshoot issues and ensure high availability.

CloudWatch collects monitoring and operational data in the form of logs metrics and events and visualizes it using automated dashboards so you can get a unified view of your AWS resources applications and services that run in AWS and on-premise.

It works on the Principal Collect, Monitor, Act, Analyze(CMAA).

Working of AWS CloudWatch :



1. **Collect:** The first step is to collect data from various sources, such as logs, metrics, and events, in order to get a complete picture of the behavior and performance of your IT systems and infrastructure.
2. **Monitor:** Once data is collected, the next step is to monitor it in real-time in order to identify any potential issues or problems. This includes setting up alerts and notifications to proactively notify you when certain conditions are met.
3. **Analyze:** The third step is to analyze the data in order to gain insights into the performance and behavior of your systems. This includes looking at trends, identifying patterns, and performing root cause analysis to determine the underlying cause of any issues.
4. **Act:** The final step is to take action based on the insights gained from the analysis. One can automate actions based on specific events or conditions. This includes making changes to your systems, such as optimizing settings or deploying new software updates, in order to improve performance and reliability.



The **CMAA methodology** is a continuous process that allows you to proactively manage and monitor your IT systems and infrastructure in real-time, ensuring that any potential issues are quickly identified and resolved. By using this framework, you can improve the availability, performance, and security of your systems, and ultimately deliver a better user experience.

AWS CloudWatch :



This is a basic flow of the AWS CloudWatch. As you can see in the diagram the very first step here is the Application Monitoring.

Application Monitoring

AWS CloudWatch collects all the data points and logs from the application for which we are using the AWS CloudWatch system. And helps us to get an idea of the application, its performance and monitor them act on them automatically and this will help us avoid the problems that may occur with the application in the coming future.

System-wide Visibility

In AWS CloudWatch you an exposure to monitor and get data and about all the tiers of the application so that we won't miss out any services that we have been using in the various tiers of the application.

Resource Optimization

In resource optimization Auto-Scaling comes into the picture. When there is peak CPU utilization then we can increase the number of instances and reduce as the CPU utilization comes down.

Unified Operational Health

And all these things helps the system to have an Unified Operational Health.

Here are some key concepts related to Amazon CloudWatch:

Metrics:

Metrics are data points that measure the behavior and performance of AWS resources. Examples of metrics include CPU utilization, disk read/write operations, and network traffic.

So the basic idea here is that if we want to judge the current state of an instance or resource we need a benchmark isn't it for example if we consider the CPU utilization of the instance goes above 85 percent then I want to scale a new resource so what we will say then the benchmark here is CPU utilization and the threshold value that I have here is 85 percent.

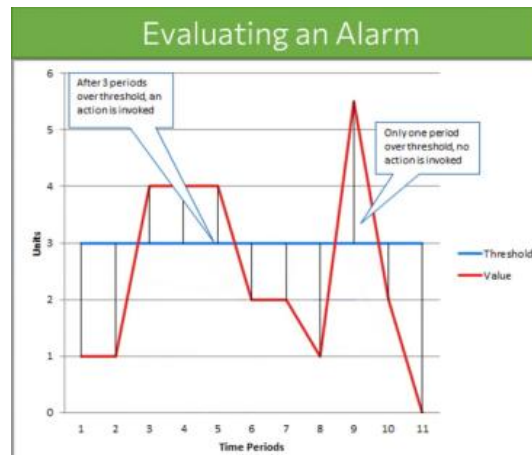
If the CPU utilization of the instance goes above 85 percent then we want to scale a new resource so 85 percent becomes the threshold and the benchmark on which we are trying to judge the resource state i.e. CPU utilization.

Alarms:

Alarms are used to trigger actions based on the value of a metric. For example, you can create an alarm that sends an email notification when the CPU utilization of an EC2 instance exceeds a certain threshold.

We can use alarms to automatically initiate actions on your behalf for any certain actions that has occurred.

Alarms keep focus on a single metric over a period of time. Once the action is occurred we can notify the same to SNS topic or to an Auto-scaling group to take certain necessary actions.



When you create the CloudWatch alarm, you need to set the 3 settings:

1. **Time Period:** It is the length of time to evaluate the metric or expression to create each individual data point for an alarm.
2. **Evaluation period:** The number of the most recent periods, or data points, to evaluate when determining alarm state.
3. **Data-points to Alarm:** The number of data points within the Evaluation Period that must be breaching to cause the alarm to go to the alarm state.

Namespaces :

A namespace is a container for CloudWatch metrics.

A metric is a set of data points that are recorded over time and represent a specific aspect of a system or application. A namespace is used to organize metrics and to ensure that metrics with the same name do not overlap.

CloudWatch provides several default namespaces for AWS services, such as Amazon EC2, Amazon S3, and Amazon RDS. However, you can also create custom namespaces for your own applications and services.

When you create a custom namespace, you must choose a name that is unique within your AWS account. You can then use the CloudWatch API to publish metrics to your namespace, or you can use CloudWatch agents or integrations to automatically collect and publish metrics from your applications and services.

Once you have created a namespace and published metrics to it, you can use CloudWatch to create alarms, dashboards, and other visualizations based on the data in the metrics. This can

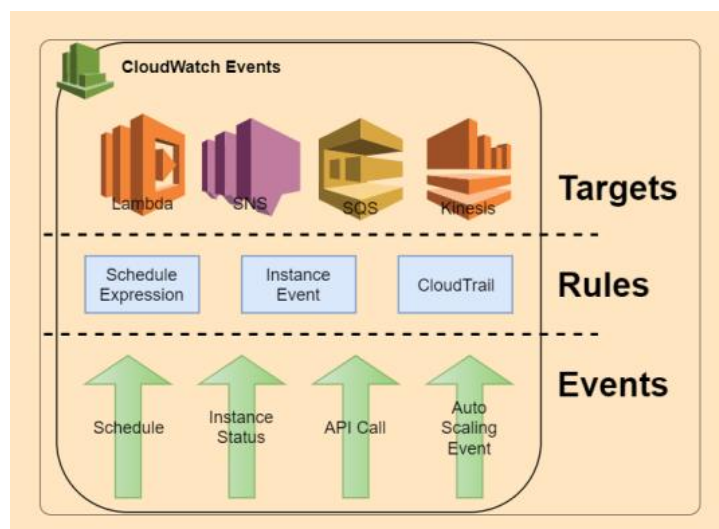
help you monitor the health and performance of your systems and applications and quickly identify and respond to issues.

Example : myapplication-cpu-utils, myapplication-cpu-memory

Service	Namespace
Amazon EC2	AWS/EC2
Amazon EC2 Elastic Graphics	AWS/ElasticGPUs
Amazon EC2 Spot Fleet	AWS/EC2Spot
Amazon EC2 Auto Scaling	AWS/AutoScaling

Events:

Events are a way to react to changes in your AWS environment, such as the creation or termination of an EC2 instance. You can use CloudWatch Events to automate your responses to events.



Events are used to monitor and respond to changes in your AWS resources and applications. CloudWatch Events allows you to track changes in your AWS resources, such as when an EC2 instance is launched or terminated, or when a new object is added to an S3 bucket. You can also create rules to trigger events based on application-level events, such as when a file is added to a specific folder on a server.

CloudWatch Events are made up of two main components: rules and targets. A rule defines the event pattern to match, and a target defines the action to take when the rule matches an event. Targets can include Lambda functions, SNS topics, SQS queues, and other AWS services.

When a CloudWatch Event matches a rule, the associated target is triggered, which can then perform a variety of actions, such as sending a notification or invoking a Lambda function to perform additional processing.

CloudWatch Events can be used for a variety of use cases, such as triggering automated workflows, sending notifications, or responding to security threats.

Logs:

Logs are records of events that have occurred in your AWS environment. CloudWatch Logs allows you to store, search, and visualize logs from multiple AWS resources in a central place.

Dashboards:

Dashboards are customizable views that allow you to display metrics, logs, and alarms in a single place. You can create multiple dashboards and share them with other users.

Billing Alarms:

Billing alarms are used to monitor your AWS costs and usage. You can set an alarm to notify you when your monthly costs exceed a certain threshold.

Insights:

CloudWatch Insights is a search and analysis tool that allows you to quickly search and visualize CloudWatch logs.

In summary, CloudWatch helps you monitor and manage your AWS resources and applications by collecting and analyzing data from various sources and providing tools to help you take actions based on that data.

Please, feel free to drop any questions in the comments below. I would be happy to answer them.

If this post was helpful, please do follow and click the clap

_Thank you for reading

_Rajani