CloudWatch Custom Metrics

CloudWatch Custom Metrics allows you to monitor and collect custom data from your applications, services, and resources within the AWS ecosystem. While AWS provides default metrics for many of its services, sometimes you need to monitor specific aspects of your applications or services that are not covered by default metrics. This is where custom metrics come into play.

Here's how can work with CloudWatch Custom Metrics:

Instrumentation: ***You need to instrument your applications or services to send custom metrics to CloudWatch. This can be done using the AWS SDK or Command Line Interface (CLI) to push data points to CloudWatch.***

Data Points: ***Custom metrics are composed of data points, each representing a single measurement at a specific point in time. You can send these data points to CloudWatch using the PutMetricData API.***

Namespace: ***Custom metrics are organized within namespaces. Namespaces act as containers for metrics and help in categorizing them. You need to specify a namespace when sending custom metrics to CloudWatch.***

Dimensions: ***Dimensions are key-value pairs associated with your metrics. They provide additional information to help you uniquely identify and categorize your metrics. For example, you might have a dimension for InstanceID or Environment.***

Retention: ***CloudWatch retains custom metric data for up to 15 months by default. You can adjust the retention period based on your requirements.***

Monitoring: ***Once you've sent custom metrics to CloudWatch, you can create alarms, dashboards, and visualizations to monitor and analyze the data. Alarms can be configured to trigger notifications or automated actions based on predefined thresholds.***

Cost: ***There is a cost associated with sending custom metrics data to CloudWatch, so you should be mindful of how much data you're sending and how frequently. You can refer to the AWS pricing page for CloudWatch to estimate costs.***

When using CloudWatch Custom Metrics, it's important to design your monitoring strategy carefully to ensure that you're collecting the right data to meet your monitoring and troubleshooting needs without incurring unnecessary costs. Additionally, consider security best practices to ensure that only authorized systems or users can send custom metrics to CloudWatch.

AWS provides detailed documentation and commands to work with CloudWatch Custom Metrics efficiently.
Here's how you can access them:

1) ***AWS CloudWatch Documentation: https://docs.aws.amazon.com/cloudwatch/***
2) ***Collect metrics using the CloudWatch monitoring scripts: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring-scripts-intro.html***

AWS has deprecated the use of the CloudWatch monitoring scripts for collecting metrics. These scripts were previously provided by AWS to help users collect system-level metrics from their EC2 instances and on-premises servers for monitoring with CloudWatch. This page provides information about those monitoring scripts for customers who are still using them, but the monitoring scripts are no longer available.

Now AWS strongly recommends that to use the CloudWatch agent to collect metrics and logs.

Now here are the Steps followed by AWS Documentation: ***Collecting metrics using the CloudWatch monitoring scripts.***

Step 1) *Launch the Instance.*

This Monitoring script only supports or was tested on Instance using the following systems (AMIs):

- ***Amazon Linux 2***
- ***Amazon Linux AMI 2014.09.2 and later***
- ***Red Hat Enterprise Linux 6.9 and 7.4***
- ***SUSE Linux Enterprise Server 12***
- ***Ubuntu Server 14.04 and 16.04***

Step 2) *After launching the Instance need to associate an IAM role with EC2 instances or on-premises servers so that the CloudWatch monitoring scripts have the necessary permissions to call AWS actions.*

- *Create an IAM role for Ec2 Service*
- *Create a custom policy with the following permission and attach it to the IAM Role*
  a. ***cloudwatch:PutMetricData***
  b. ***cloudwatch:GetMetricStatistics***
  c. ***cloudwatch:ListMetrics***
  d. ***ec2:DescribeTags***
- *Now Attach the IAM role to Instance.*

Step 3) *Connect to the Instance.*

Step 4) *We may need to* ***install*** *additional* ***Perl modules*** *to ensure that the CloudWatch monitoring scripts function properly. These Perl modules provide the necessary functionality for the scripts to interact with AWS services and collect metrics****.***

Run the command following command:

sudo yum install -y perl-Switch perl-DateTime perl-Sys-Syslog perl-LWP-Protocol-https perl-Digest-SHA.x86_64

Step 5) *After installing Perl we need to download CloudWatch Monitoring Script.*

- *To download the Monitoring Script :*

curl https://aws-cloudwatch.s3.amazonaws.com/downloads/CloudWatchMonitoringScripts-1.2.2.zip -O

- *To Unzip the downloaded package and install it :*
  ***unzip CloudWatchMonitoringScripts-1.2.2.zip && ***
  ***rm CloudWatchMonitoringScripts-1.2.2.zip && ***
  cd aws-scripts-mon

Here Our Setup for the Collecting metrics using the CloudWatch monitoring scripts is completed.

Step 6) *Now run the following to collect all available memory metrics and send them to CloudWatch.*

./mon-put-instance-data.pl --mem-util --mem-used --mem-avail

Step 7) *Go to the CloudWatch Service and Click on All metrics, you can see the custom metrics are added.*

Step 8) *Click on System/Linux then click on Instance ID*

Step 9) *Select all Metrics, and now you can monitor custom metrics.*

Step 10) *Now set a cron schedule for metrics report to CloudWatch. (otherwise, we need to run the every time).*

- *Run command: crontab –e*
- *Add the command to crontab: [* **** * * * * /home/ec2-user/aws-scripts-mon/ ./monput-instance-data.pl --mem-util --mem-used --mem-avail*** *] and save it.*

Step 10) *Now restart the crontab, by running the command: [* ***sudo systemctl restart crond*** *]*

CloudWatch Agent

The CloudWatch Agent is a lightweight, unified agent provided by AWS for collecting and publishing system and application metrics to Amazon CloudWatch. It replaces the previously used monitoring scripts for EC2 instances and on-premises servers, offering improved functionality and flexibility.

Unified Metric Collection: The CloudWatch Agent can collect both system-level metrics (CPU, memory, disk utilization, etc.) and custom metrics from applications and services running on your EC2 instances or on-premises servers.

Custom Metrics: In addition to system metrics, the CloudWatch Agent allows you to collect custom metrics generated by your applications or services. This provides greater insight into the performance and behavior of your applications.

- ***Flexible Configuration:*** *The CloudWatch Agent supports flexible configuration options, allowing you to customize the metrics and logs collected, as well as the frequency of data collection.*
- ***Automatic Discovery:*** *The agent can automatically discover and configure metrics collection for supported services running on your instances, such as Apache, NGINX, MySQL, and more.*
- *Integration with CloudWatch Logs:* *In addition to metrics, the CloudWatch Agent can also collect and publish log files to CloudWatch Logs, enabling centralized log management and analysis.*
- ***Cross-Platform Support:*** *The CloudWatch Agent is compatible with a wide range of operating systems, including Amazon Linux, Ubuntu, CentOS, Red Hat Enterprise Linux (RHEL), Debian, and Windows.*
- ***Simplified Installation and Updates:*** *The CloudWatch Agent provides a simple installation process and supports automatic updates, making it easy to deploy and maintain across your fleet of instances.*
- ***Cost-Effective:*** *The CloudWatch Agent is offered at no additional charge, and you only pay for the metrics and logs ingested into CloudWatch.*

Now we will Download and Install the CloudWatch Agent and configure it, then monitor the Ec2 Instance using this CloudWatch Agent.

Step 1) ***Launch the Instance with Ubuntu AMI and Configure the following Settings***

- ***Allow SSH and HTTP rule***
- ***Add user data :***

```
#!/bin/bash
sudo apt update -y
sudo apt install apache2 -y
sudo systemctl start apache2
```

- ***Add CloudWatch Full Access role to Instance.***

Step 2) ***Connect to the Instance.***

Step 3) ***Now will install the CloudWatch Agent. Run The following Commands***

- ***wget https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb***

This is the URL of the CloudWatch Agent Debian package for Ubuntu 64-bit. It points to the latest version available on Amazon S3.

- ***sudo dpkg -i -E ./amazon-cloudwatch-agent.deb***

This command will install the CloudWatch Agent on your system

Step 4) ***Now we need to Configure the CloudWatch Agent***

Run the following command:

- ***sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-config-wizard***

When you run this command, the wizard will prompt you with a series of questions to configure the agent.

Once the configuration is complete, the wizard will create the configuration file at the specified location.

- ***cat /opt/aws/amazon-cloudwatch-agent/bin/config.jso*** *-- > to check configuration file*
- *file_path": "/var/log/apache2/access.log*

Give this path during configuration, the file path of the Apache access log file, which is used to record details of HTTP requests made to the Apache web server.

- ***sudo apt-get update –y***

This command updates the package lists for repositories.

- ***sudo apt-get install collectd***

Update Package Lists: sudo apt-get update is executed to update the package lists for repositories.

Install Collectd: collectd package and its dependencies are installed on your system.

Completion: Once installation is complete, collectd is available for collecting system statistics and metrics.

- ***sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a status***

To check the status of the CloudWatch Agent and display whether it is running or stopped on your system.

Step 5) ***Copy the Public IP Address of Instance and browse it 3 or 5 times.***

Step 6) ***Now Go to The CloudWatch Service.***

- *Click on Logs*
- *Click on logs Groups.*

You will see log group is created.