1. Overview

1. Amazon CloudWatch monitors your Amazo**n** Web Services (AWS) resources and the applications you run on AWS in **real time**.

2. With CloudWatch we can get:
    a. Resource Utilization (System-wide visibility into resource utilization)
    b. Resource Optimization
    c. Application Performance
    d. Application Monitoring
    e. Operational Health

3. Cloudwatch can be used **on premise**.
    You need to install CloudWatch Agent. You can download the CloudWatch agent package using either Systems Manager Run Command or an Amazon S3 download link

4. CloudWatch helps to perform the following operations:

    1. **Collect**: CloudWatch collects monitoring and operational data in the following forms from different AWS resources and services:
        a. Logs
        b. Metrics
        c. Events

    2. **Monitor**: CloudWatch monitors your application using a unified view called Dashboards and also you can set Alarms too.
        a. Dashboards
        b. Alarms

    3. **Actions**: CloudWatch performs actions based on Alarm and CloudWatch Events. (Like restarting an instance)
        a. Alarms
        b. Events

    4. **Analyze**: You can analyze the logs using custom processing or using Athena

2. Collect

1. CloudWatch Metrics

1. CloudWatch collects data in the form of Metrics and CloudWatch provides metrics for every service in AWS.

2. Metric is a variable monitor (CPU Utilization, NetworkIn..)

3. Metrics belong to namespaces. Dimension is an attribute of metrics (instance id, environment).

4. You can have upto **10 dimensions** per metric.

There are two types of Metrics:

1. Default Metrics:

Provided by AWS and **it is FREE**

2. Custom Metrics:

We need to gerentate

You **can NOT publish custom metrics from the AWS console**. You need to use **AWS CLI to publish it.**

A custom metric can be one of the following:

1. *Standard resolution*

With data having one-minute granularity

2. *High resolution:*

With data at a granularity of **one second.** Use API call **PutMetricData**

How would you set a CustomMetric with High Resolution? Set **Storage Resolution parameter** to 1 using the **PutMetricRequestAPI**

Are high-resolution custom metrics priced differently than regular custom metrics? No, high-resolution custom metrics are priced in the same manner as standard 1-minute custom metrics.

You **CAN NOT Manually** delete metrics, they are removed based on the schedule:

1. 1 minutes metrics: Available for **15 days**
2. 5 minute metrics: Available for **63 days**
3. 1 hour metrics : Available for **15 months**

Metrics are stored separately in Regions, but you can use CloudWatch **Cross-Region functionality to aggregate statistics from different Regions.**

Refer: https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/Cross-Account-Cross-Region.html

StatsD and **CollectD**

In October 2018, AWS introduced the ability to use **StatsD** and **CollectD** to collect custom metrics to be consumed by Amazon CloudWatch

StatsD

1. You can retrieve additional custom metrics from your applications or services using the CloudWatch agent with the `StatsD` protocol.
2. This is supported in **Windows and Linux**
3. StatsD is a popular open-source solution that can gather metrics from a wide variety of applications.
4. **StatsD is especially useful for instrumenting your own metrics.**
5. **For an example of using the CloudWatch agent and StatsD together, see How to better monitor your custom application metrics using Amazon CloudWatch Agent.**

Collectd

1. You can retrieve additional metrics from your applications or services using the CloudWatch agent with the collectd protocol
2. **This is supported only on Linux servers**.
3. collectd is a popular open-source solution with plug-ins that can gather **system statistics for a wide variety of applications.**

2. ClodWatch Logs

Applications can send logs to CloudWatch using **SDK**

The Amazon CloudWatch Logs agent default sending time is **every 5 seconds**; however, this is configurable.

To send logs to CloudWatch, make sure **IAM permissions are correct**

By default, CloudWatch **stores the log data indefinitely,** and the retention can be changed for each log group at any time. You can define log expiration policies (never expire or 30 days)

Log data is **encrypted in transit and at rest within Amazon CloudWatch**. This requires **no** special configuration on the part of a system administrator.

CloudWatch logs uses **KMS** to secure data at-rest

Amazon CloudWatch Logs Insights:

Allows you to run **interactive queries against your logs and create visualizations** that include dashboards in Amazon CloudWatch

Allows you to query large sets of logs. You can **even use regex to extract data** from event fields

NOTE: AWS CloudWatch Metrics Filters **DO NOT** support regex filters. This can be used to trigger alarms

Question: After installing the Amazon CloudWatch agent on the EC2 instance, the anticipated system logs are not received by CloudWatch Logs. What are the possible reasons?

a. CloudWatch agent does not support the OS used.
b. The EC2 instance is in a private subnet, and VPC doesn't have a NAT gateway. *

The Amazon CloudWatch Logs agent for the Windows can be used to send **IIS logs** to CloudWatch

ClodWatch logs can go to:

1. Batch exporter to S3 for archival
2. Stream to Elasticsearch cluster for further analysis

Logs Storage architecture:

1. Logs groups: arbitrary name, usually representing an application
2. Log stream: instance within application / log files / containers

If you want to secure the logs, you can do encryption of logs using KMS at the Group level

CloudWatch Logs can use filter expressions:

- Find specific IP inside logs
- Metric filters can be used to trigger alarms

CloudWatch can collect logs from:

- Elastic BeanStalk : collection of logs from application
- ECS: collection from containers
- AWS Lambda: collection from function logs
- VPC Flow Logs: VPC Specific logs
- API Gateway:
- ClodTrail based filter
- ClodWatch log agents: for example on EC2 machines
- Route53: Logs DNS queries

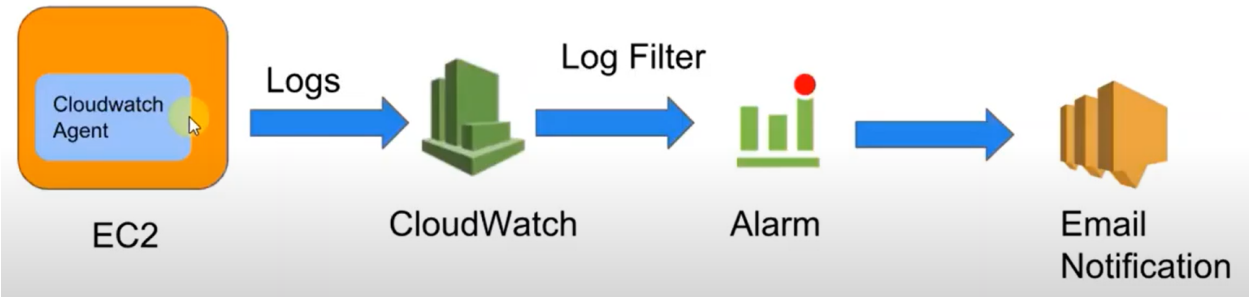
CloudWatchReadOnlyAccess: The managed policy CloudWatchReadOnlyAccess will give users the ability to view the metrics in CloudWatch without needing to gain access to the other AWS services.

You can choose to filter statistics by:

1. Specific Amazon EC2 instance
2. An AutoScaling group
3. By an AMI

Amazon CloudWatch Snapshot graph: Allows you to display charts on a webpage or a 3rd party tool. This functionality is not provided by Amazon CloudWatch Logs, the Amazon CloudWatch Logs Agent, or Amazon CloudWatch APIs.

You can create Alarm based on Log Filter from CloudWatch



3. Monitoring

3.1 CloudWatch Monitoring: There are 2 types of monitoring:

1. Basic Monitoring
2. Detailed Monitoring

1 Basic Monitoring

- Enabled By Default
- Data is available in **5 minutes** period with no charge

2 Detailed Monitoring

- **Must be enabled at the instance level**
- Data is available in **1 minutes** period with additional charge

3.2. CloudWatch Dashboards

If you want to have quick ways to see key metrics, the dashboards are a great way to do that.

Dashboards are multi-region and can display any widget in any region. To add the widget, change the region that you need to add then add the widget to the dashboard.

Dashboards are **Global**

Dashboard can include graphs from **Different Regions**

You can set up automatic refresh (10s, 1m, 2m, 5m, 15m)

Things to consider when creating a Custom dashboard with metric data from different regions

1. **Detailed Monitoring** is required to be enabled.
2. IAM users who have permission to use:
    a. **PutMetricData** AND
    b. **Put Dashboard** can create customized dashboard

Pricing:
3 dashboards (up to 50 metrics) are free
$3/dashboard/month

3.3. ClodWatch Alarms

Alarms perform **one or more actions** based on the value of the metric relative to a threshold over a number of time periods

CloudWatch Alarm history is stored for **only 14 days**.

You **can't** modify the Alarm name after you create an alarm

You can also add alarms to CloudWatch Dashboards and monitor them visually.

How to simulate an Alarm? **AWS CLI set-alarm** command

What does **AWS CLI Dry run** do?
This helps to **check if you have the required permission** for an action **without** making the actual request.

We can use Alarms to perform actions, for example:

a. We can say AutoScaling group to add a new instance when the CPU utilization is over 80%
b. EC2 actions: We can say reboot the EC2 instance when the CPU > 95% for 30 minutes
c. Billing Alarm: We can set a billing alarm to notify when the cost is going above the specified amount

States of Alarm:

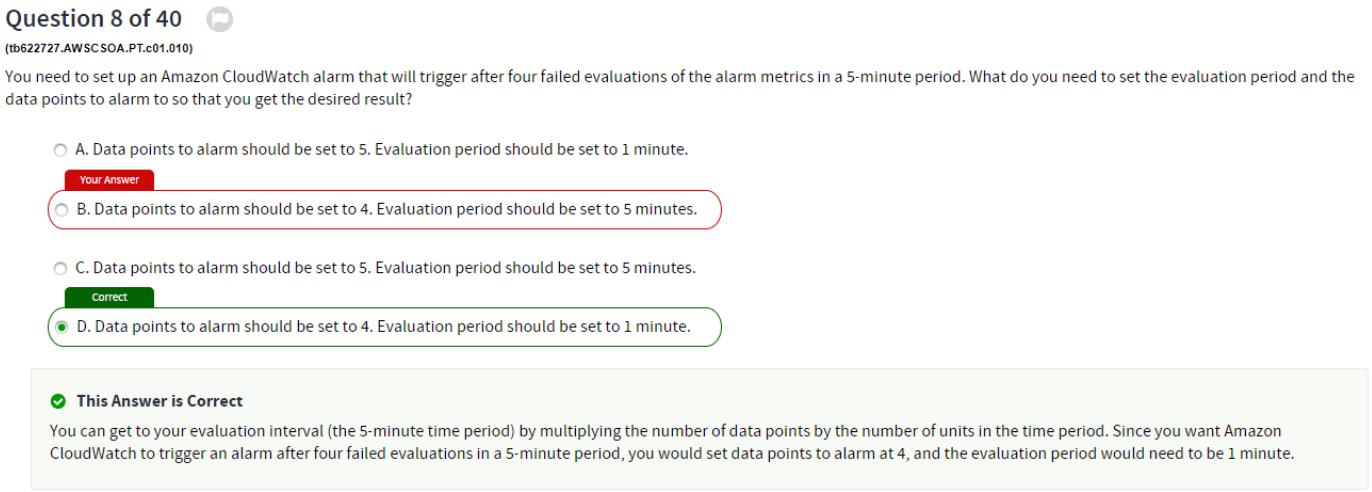
You can have upto **5000** Amazon CloudWatch alarms per **Single Region**

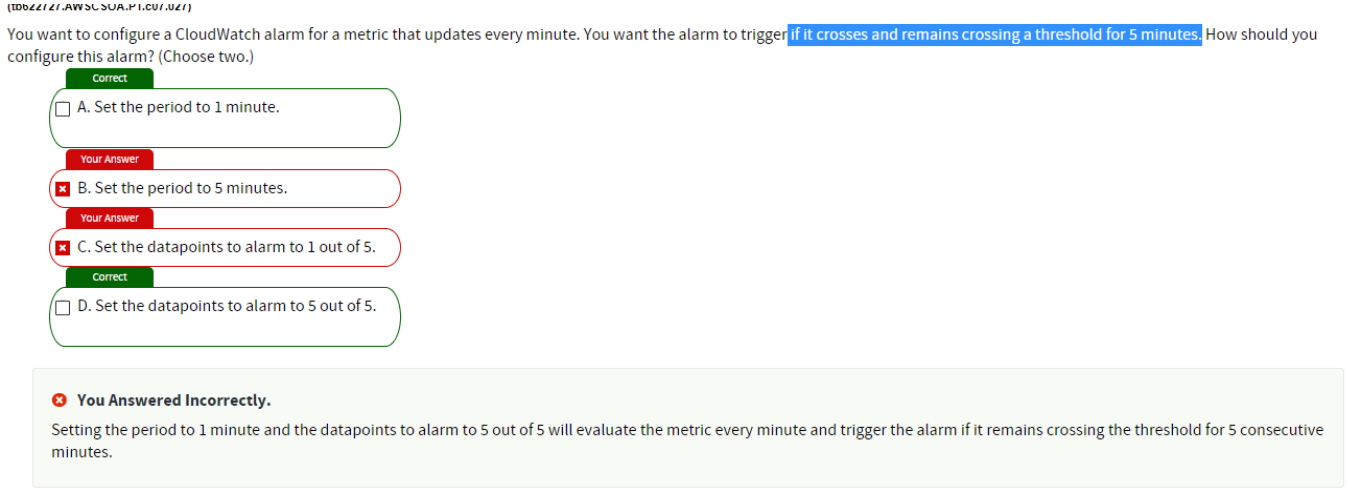
Alarm Evaluation Interval Calculation

Evaluation interval = number of data points * number of unit time period

You need to set up an Alarm that will trigger after 4 failed evaluation of the alarm metrics in a 5-minute data points to alarms so that you get the desired results.

So, here the number of data points are 4. If each time the data point gets triggered in the unitime of 1 minutes. In the 5th minutes when the data point reached the alarm will get triggered





Reasons for getting INSUFFICIENT_DATA from Alarm

Teva, you need to ask an IMPORTANT QUESTION whether that service is **just started or it has been running?**

If it's just barely started: The alarm has only just been started, so it doesn't have enough data to determine if the state should be OK or Alarm

If the instance is an existing one: Not enough data for the metric to determine whether it should be OK or Alarm

4. Act

Cloudwatch Alarm is covered in the previous section under monitoring

4.1. CloudWatch Events

- **Events** - An event indicates a change in your AWS environment, such as EC2 instance state change from pending to running
- **Targets** - A target processes events. Target can be Amazon EC2, or AWS Lambda functions
- **Rules** - A Rule matches incoming events and routes them to targets for processing. A single rule can route to multiple targets, all of which as processes parallel

5. Analyze

Process CloudWatch log events through Kinesis and Lambda for custom processing and analysis

Metric math

enables you to query multiple CloudWatch metrics and use math expressions to create a new time series based on these metrics.

Amazon CloudWatch snapshot graphs

Amazon CloudWatch snapshot graphs allow you to display charts on a web page or a third-party tool.

6. CLI Commands

1. AWS cloudwatch list-instances --namespace AWS/EC2
2. This is for checking instance status and system status
   aws ec2 describe-instance-status
3. This is for disabling the monitoring in AWS Cloudwatch based on the given instance ID:
   aws ec2 unmonitor-instances --instance-ids <instance-id>

7. The unified CloudWatch agent enables you to do the following

- Collect more system-level metrics from Amazon EC2 instances across operating systems. The metrics can include in-guest metrics, in addition to the metrics for EC2 instances. The additional metrics that can be collected are listed in Metrics Collected by the CloudWatch Agent.
- Collect system-level metrics from on-premises servers. These can include servers in a hybrid environment as well as servers not managed by AWS.
- Retrieve custom metrics from your applications or services using the StatsD and collectd protocols. StatsD is supported on both Linux servers and servers running Windows Server. collectd is supported only on Linux servers.
- Collect logs from Amazon EC2 instances and on-premises servers, running either Linux or Windows Server.