

# Cloud Monitoring

- In this section, we will learn how to keep track of and monitor our cloud deployments strategically.

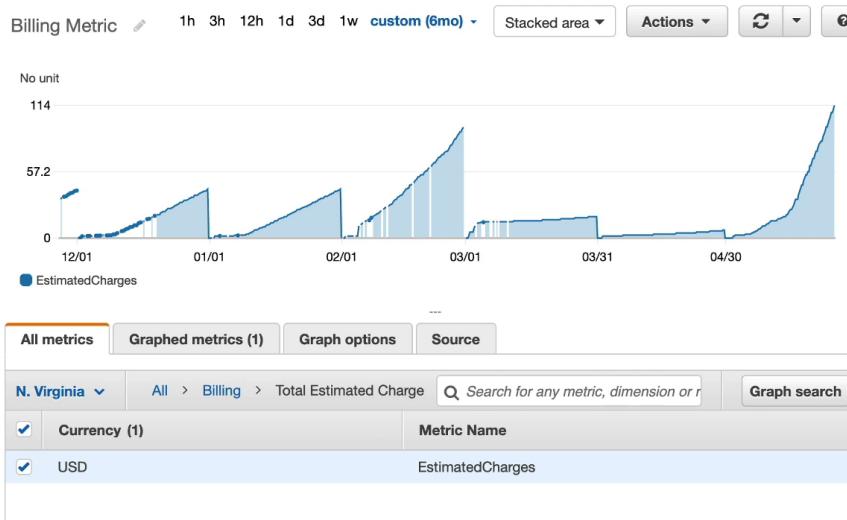
## CloudWatch

- CloudWatch Metrics
- CloudWatch Alarms
- CloudWatch Logs
- CloudWatch Events (EventBridge)

### CloudWatch Metrics

- CloudWatch Metrics are data points/parameters used to quantify the performance of your resources, such as CPU utilization, network traffic, or disk usage.
- You can use CloudWatch Metrics to monitor your AWS resources in real-time, set up alarms, and take automated actions based on defined thresholds.
- CloudWatch provides metrics for **every** service in aws.
- Metrics have timestamps
- You can create a CloudWatch Metrics Dashboard to visualise and monitor your usage.

# Example: CloudWatch Billing metric (us-east-1)



- **Important CloudWatch Metrics to monitor different services:**

- EC2 instances: CPU Utilization, Status Checks, Network (not RAM)
  - Default metrics every 5 minutes
  - Option for Detailed Monitoring (\$\$\$): metrics every 1 minute
- EBS volumes: Disk Read/Writes
- S3 buckets: BucketSizeBytes, NumberOfObjects, AllRequests
- Billing: Total Estimated Charge (only in us-east-1)
- Service Limits: how much you've been using a service API
- Custom metrics: push your own metrics

## CloudWatch Alarms

- Alarms in CloudWatch allow you to monitor a single metric over a specified time period and perform one or more actions based on the value of the metric relative to a threshold over time.

- Alarms are used to trigger notifications set for any metrics.
- Whenever a metric goes above its set threshold, you will be notified. This will help with either cost optimization or will help you scale your application as per demand.
- Alarm actions can include:
  - Auto scaling of ec2 instances.
  - EC2 actions like stop, terminate, reboot or recover.
  - SNS notifications
- You have various options to create the alarm like %, max, min, etc. and define a period over which you want to evaluate an alarm like an hour, a day, even a minute.
- **Alarm states are: Ok (all good), Insufficient\_Data (can't be determined), Alarm (bad)**
- **Note:** CloudWatch billing data alarm is available only in the US-East-1 region.

## CloudWatch Logs

- CloudWatch Logs is a service provided by AWS that allows you to collect, store, and analyze log files from your applications, servers, and AWS resources.
- CloudWatch Logs is your tool for collecting and understanding the story your applications and systems are telling you through their logs. It's a valuable resource for troubleshooting, monitoring, and maintaining the health of your AWS resources.
- These logs prove to be handy especially when troubleshooting is required, you can simply go through log lines which will point out where the issue is arising from and fix it.

## Purpose and Importance:

### 1. Centralized Logging:

- Imagine you have multiple servers or applications running. CloudWatch Logs lets you centralize all the logs generated by these different components in one place.

### 2. Troubleshooting and Debugging:

- When something goes wrong with your applications or servers, logs can be like a detective's notebook. CloudWatch Logs helps you collect these detective notes in an organized manner so that you can analyze them when needed.

### 3. Security and Compliance:

- Logs often contain information about who accessed your systems, what actions were taken, and more. CloudWatch Logs can be used for auditing purposes, helping you meet security and compliance requirements.

## How It Works:

### 1. Log Groups:

- In CloudWatch Logs, logs are organized into "Log Groups." A log group is like a container for logs related to a specific application, service, or system.

### 2. Log Streams:

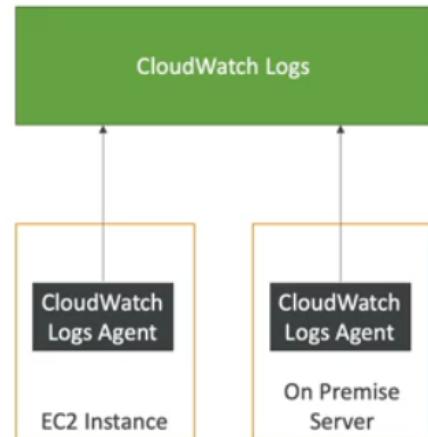
- Within each log group, you have "Log Streams." Think of log streams as individual logs for a specific instance or task. Each log stream represents a sequence of log events.

### 3. Log Events:

- Log events are the actual log entries. These can include information about what's happening in your application or system, such as error messages, status updates, or user activities.

- CloudWatch Logs collect log files from:

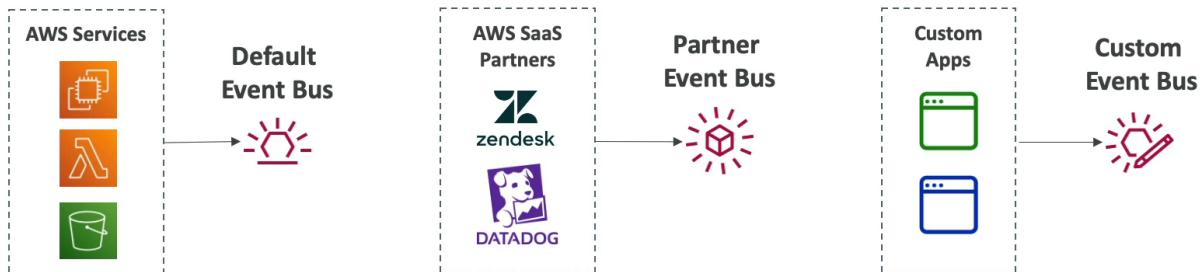
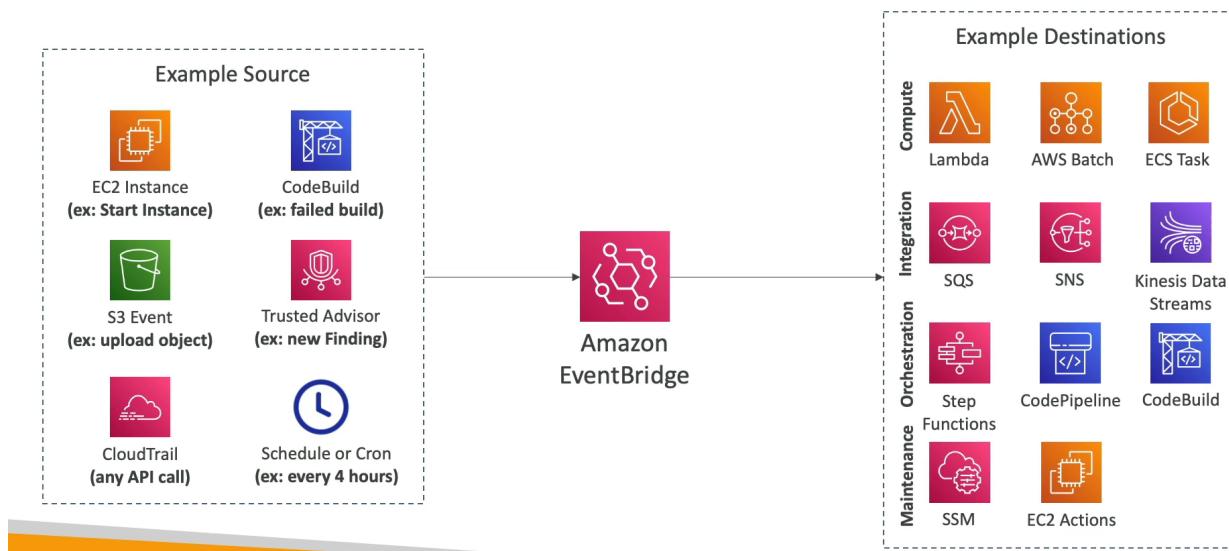
- Elastic Beanstalk: collection of logs from application
  - ECS: collection from containers
  - AWS Lambda: collection from function logs
  - CloudTrail based on filter
  - CloudWatch log agents: on EC2 machines or on-premises servers
  - Route53: Log DNS queries
- 
- The **CloudWatch Logs Agent** can be installed on your ec2 instance or on premises server (it's hybrid) to fetch log files from the server and send to the CloudWatch Logs service.
    - By default, no logs from your EC2 instance will go to CloudWatch
    - You need to run a CloudWatch agent on EC2 to push the log files you want
    - Make sure IAM permissions are correct
    - The CloudWatch log agent can be setup on-premises too



## EventBridge (CloudWatch Events)

- Amazon EventBridge is a fully managed event bus service that makes it easy to connect different AWS services and your own applications using events (data from one service triggering an action in another).
- EventBridge uses an event bus, which is a central hub where events from various sources (called "event producers") are sent and then consumed by different targets (called "event consumers").

- With EventBridge, you can react to the events happening within your AWS account. You can create rules that match events and trigger actions in response.
- Usage:** Automate responses to specific events, schedule recurring tasks, and integrate with other AWS services. For example, you can automatically stop or terminate instances based on a schedule or an event.
- One imp use case is scheduling of CRON jobs and reacting to Event Patterns.



## Schedule pattern

Occurrence | [Info](#)

You can define an one-time or recurrent schedule.

One-time schedule

Recurring schedule

### Schedule type

Choose the schedule type that best meets your needs.

Cron-based schedule

A schedule set using a cron expression that runs at a specific time, such as 8:00 a.m. PST on the first Monday of every month.

Rate-based schedule

A schedule that runs at a regular rate, such as every 10 minutes.

### Amazon EventBridge:

Amazon EventBridge is a fully managed event bus service designed to facilitate the flow of events within an event-driven architecture. It serves as a central hub, collecting events from various sources, including AWS services, custom applications, and external events. EventBridge uses event rules to define conditions for routing events to specific targets, enabling seamless integration between different components of a system. Its focus is on managing the orchestration and flow of events across services.

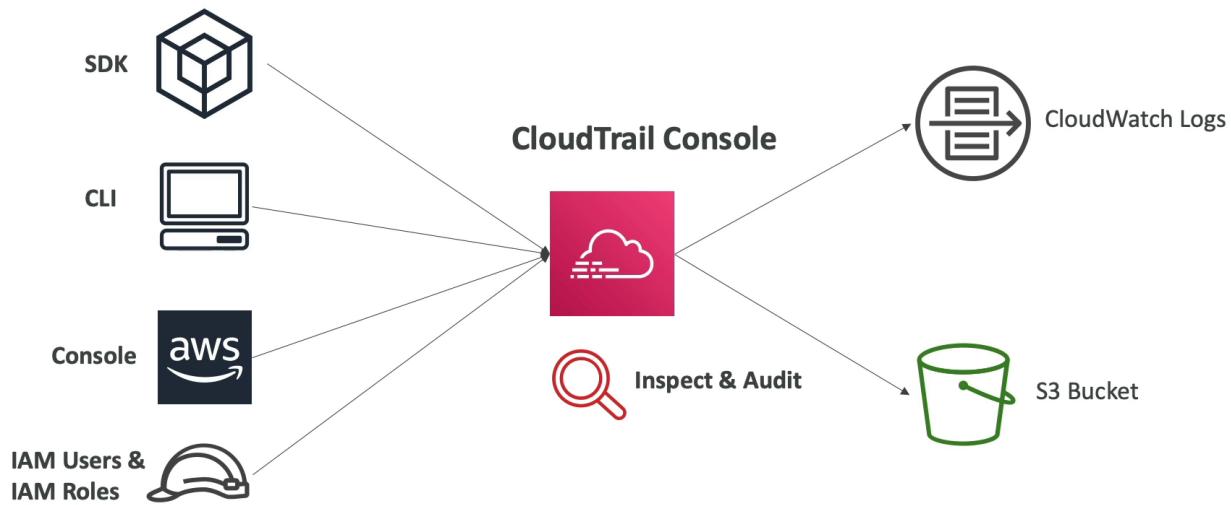
### AWS Lambda:

AWS Lambda, on the other hand, is a serverless compute service that allows you to run code in response to events. While Lambda can be triggered by events from various sources, including EventBridge, its primary purpose is the execution of functions.

Lambda functions can be used to process events, perform computations, and execute code without the need to manage underlying infrastructure. It provides a serverless environment for running code snippets in response to specific events, making it a versatile compute service in event-driven architectures.

## AWS CloudTrail

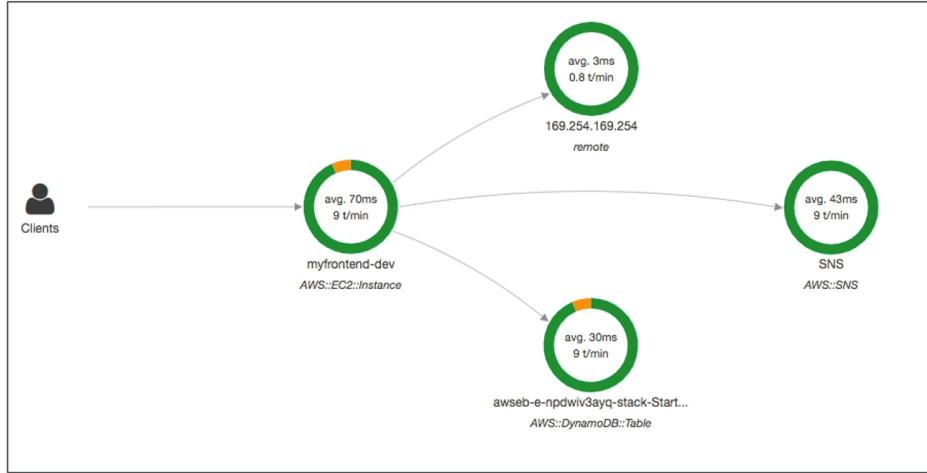
- Imagine you have a security camera for your AWS account. CloudTrail is like that security camera—it records every time someone interacts with your AWS account, like making changes or accessing services. It keeps a detailed history of who did what, when they did it, and what exactly they did. This helps you keep track of actions, detect any unusual behavior, and ensure the security of your AWS resources. Hence, the name **CloudTrail**.
- Provides governance, compliance and audit for your AWS Account
- CloudTrail is enabled by default!
- Get an history of events / API calls made within your AWS Account by:
  - Console
  - SDK
  - CLI
  - AWS Services
- All these logs of actions made within your AWS account can be sent to two locations: either CloudWatch logs or amazon S3.
- A trail can be applied to all regions (by default) or pinned to a single region.
- If a resource is deleted or modified then to find out who, what, when — use CloudTrail.



## AWS X-Ray

- AWS X-Ray is a distributed tracing service that helps developers analyze and debug production, distributed applications.
- **X-Ray traces requests as they travel through your application, providing a visual representation of the components that are involved and the time they take. This aids in identifying performance bottlenecks and improving application performance.**
- Debugging in Production, the good old way:
  - Test locally
  - Add log statements everywhere
  - Re-deploy in production
- Log formats differ across applications and log analysis is hard.
- Debugging: one big monolith “easy”, distributed services “hard”
- No common views of your entire architecture
- Enter... AWS X-Ray!

## Visual analysis of our applications



## AWS X-Ray advantages

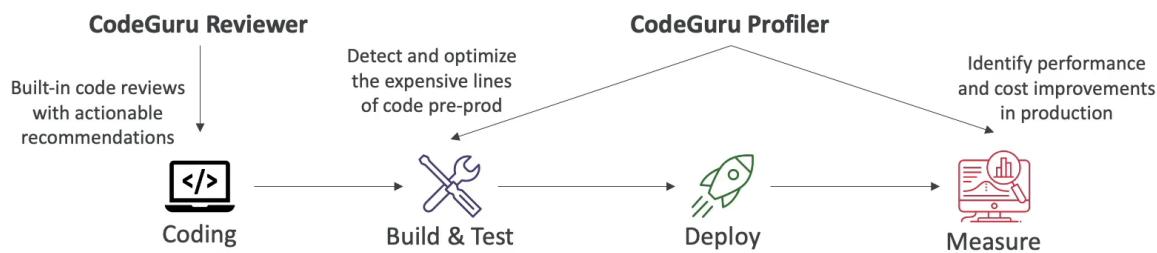


- Troubleshooting performance (bottlenecks)
- Understand dependencies in a microservice architecture
- Pinpoint service issues
- Review request behavior
- Find errors and exceptions
- Are we meeting time SLA?
- Where I am throttled?
- Identify users that are impacted

## Amazon CodeGuru

- AWS CodeGuru is a machine learning-based service that provides automated code reviews and performance recommendations.
- CodeGuru helps developers improve code quality by identifying issues, suggesting improvements, and providing insights into code performance.

- It supports both Java and Python.
- An ML-powered service for automated code reviews and application performance recommendations
- Provides two functionalities
  - **CodeGuru Reviewer:** automated code reviews for static code analysis (development)
  - **CodeGuru Profiler:** visibility/recommendations about application performance during runtime (production)



## Amazon CodeGuru Reviewer

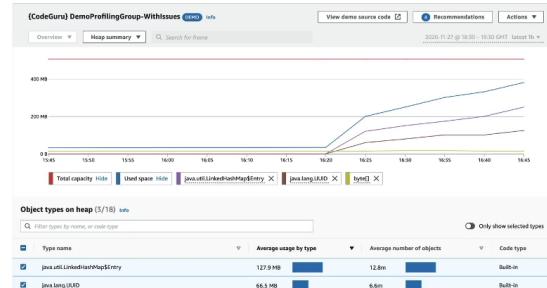
- Identify critical issues, security vulnerabilities, and hard-to-find bugs
- Example: common coding best practices, resource leaks, security detection, input validation
- Uses Machine Learning and automated reasoning
- Hard-learned lessons across millions of code reviews on 1000s of open-source and Amazon repositories
- Supports Java and Python
- Integrates with GitHub, Bitbucket, and AWS CodeCommit

The screenshot shows a detailed analysis report for a GitHub repository named "RepositoryAnalysis-amazon-codeguru-reviewer-sample-app-master-mw2tsa5600000000". The report includes the following sections:

- Details:** Status (Completed), Recommendations (4), Metered lines of code (80), Time created (10 Nov 2020 08:08:47 AM GMT-0800), Last updated (10 Nov 2020 08:11:44 AM GMT-0800).
- Recommendations (4):** A search bar for recommendations, followed by two items:
  - EventHandler.java Line: 79**: A warning about waiting for a resource before it runs, with a link to developer.waits-in-the-aws-sdk-for-java/.
  - EventHandler.java Line: 100**: A note that this code might not produce accurate results if the operation returns paginated results instead of all results, with a link to developer.waits-in-the-aws-sdk-for-java/.
- Was this helpful?**: Buttons for upvote and downvote.
- EventHandler.java Line: 100**: A note that this code uses an outdated API, with a link to developer.waits-in-the-aws-sdk-for-java/.

# Amazon CodeGuru Profiler

- Helps understand the runtime behavior of your application
- Example: identify if your application is consuming excessive CPU capacity on a logging routine
- Features:
  - Identify and remove code inefficiencies
  - Improve application performance (e.g., reduce CPU utilization)
  - Decrease compute costs
  - Provides heap summary (identify which objects using up memory)
  - Anomaly Detection
- Support applications running on AWS or on-premise
- Minimal overhead on application



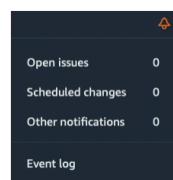
## AWS Health Dashboard

- The AWS Health Dashboard provides personalized information about the status of AWS services and regions.
- **Usage:** You can use the dashboard to stay informed about ongoing and past issues, planned maintenance, and general service health. It helps you understand the impact on your resources and plan accordingly.

### AWS Health Dashboard — Your Account

- Previously called AWS Personal Health Dashboard (PHD)
- AWS Account Health Dashboard provides alerts and remediation guidance when AWS is experiencing events that may impact you.
- While the Service Health Dashboard displays the general status of AWS services, Account Health Dashboard gives you a personalized view into the performance and availability of the AWS services underlying your AWS resources.
- The dashboard displays relevant and timely information to help you manage events in progress and provides proactive notification to help you plan for scheduled activities.
- Can aggregate data from an entire AWS Organization

- Global service
- Shows how AWS outages directly impact you & your AWS resources
- Alert, remediation, proactive, scheduled activities



Event log						
Event	Status	Event category	Region / Zone info	Start time	Last update time	Affected resources
Operational issue - EC2 (Ohio)	Closed	Issue	us-east-2	December 24, 2022 at 2:25:00 AM UTC	December 24, 2022 at 2:38:53 AM UTC	-
Operational issue - Codestar (Oregon)	Closed	Issue	us-west-2	December 21, 2022 at 5:05:57 PM UTC	December 21, 2022 at 4:50:47 PM UTC	-
Operational issue - Amplify (N. Virginia)	Closed	Issue	us-east-1	December 17, 2022 at 2:24:17 PM UTC	December 17, 2022 at 2:45:21 PM UTC	-
Operational issue - Multiple services (Singapore)	Closed	Issue	ap-southeast-1	December 13, 2022 at 10:00:55 PM UTC	December 14, 2022 at 1:01:16 AM UTC	-

The screenshot shows the AWS Health Dashboard with the "Service history" section selected. It displays a table of AWS services and their status across various regions over the last 12 months. The table includes columns for Service, RSS feed, and a grid of status icons for each day from December 6 to December 12.

Service	RSS	Today	11 Dec	10 Dec	9 Dec	8 Dec	7 Dec	6 Dec
Amazon API Gateway (Canada-Central)	RSS	Green	Green	Green	Green	Green	Green	Green
Amazon API Gateway (N. California)	RSS	Green	Green	Green	Green	Green	Green	Green
Amazon API Gateway (N. Virginia)	RSS	Green	Green	Green	Green	Green	Green	Green
Amazon API Gateway (Ohio)	RSS	Green	Green	Green	Green	Green	Green	Green
Amazon API Gateway (Oregon)	RSS	Green	Green	Green	Blue	Green	Green	Green
Amazon AppFlow (Canada-Central)	RSS	Green	Green	Green	Green	Green	Green	Green
Amazon AppFlow (N. California)	RSS	Green	Green	Green	Green	Green	Green	Green
Amazon AppFlow (N. Virginia)	RSS	Green	Green	Green	Green	Green	Green	Green
Amazon AppFlow (Ohio)	RSS	Green	Green	Green	Green	Green	Green	Green
Amazon AppFlow (Oregon)	RSS	Green	Green	Green	Green	Green	Green	Green
Amazon AppStream 2.0 (Canada-Central)	RSS	Green	Green	Green	Green	Green	Green	Green

## Summary

- CloudWatch:
  - Metrics: monitor the performance of AWS services and billing metrics
  - Alarms: automate notification, perform EC2 action, notify to SNS based on metric
  - Logs: collect log files from EC2 instances, servers, Lambda functions...
  - Events (or EventBridge): react to events in AWS, or trigger a rule on a schedule
- CloudTrail: audit API calls made within your AWS account
- CloudTrail Insights: automated analysis of your CloudTrail Events
- X-Ray: trace requests made through your distributed applications
- AWS Health Dashboard: status of all AWS services across all regions
- AWS Account Health Dashboard: AWS events that impact your infrastructure
- Amazon CodeGuru: automated code reviews and application performance recommendations



### Good job!

The CloudWatch Alarms feature allows you to watch CloudWatch metrics and to receive notifications when the metrics fall outside of the levels (high or low thresholds) that you configure.

This was discussed in Lecture 158: [CloudWatch Metrics & CloudWatch Alarms Overview](#) >

Question 1:

Which CloudWatch feature would you use to **trigger notifications** when a metric reaches a threshold you specify?

CloudWatch Events

CloudWatch Logs

CloudWatch Alarms

CloudWatch Triggers



**Good job!**

AWS X-Ray helps developers analyze and debug production, distributed applications, such as those built using a microservices architecture.

Question 2:

Which AWS service helps developers analyze and debug production as well as distributed applications?

CloudWatch

X-Ray

Service Health Dashboard

CloudTrail



**Good job!**

AWS Health Dashboard provides alerts and remediation guidance when AWS is experiencing events that may impact you.

Question 3:

Which AWS service provides alerts and remediation guidance when AWS is experiencing events that may impact you?

Service Health Dashboard

CloudWatch

AWS Health Dashboard

CloudTrail

**Good job!**

Amazon CodeGuru is a developer tool that provides intelligent recommendations to improve code quality and identify an application's most expensive lines of code.

**Question 6:**

Which AWS service automatically analyzes code and provides performance recommendations?

 X-Ray CodePipeline **CodeGuru****Good job!**

You can use Amazon CloudWatch Logs to monitor, store, and access your log files from Amazon Elastic Compute Cloud (Amazon EC2) instances, AWS CloudTrail, Route 53, and other sources.

**Question 7:**

How would you describe Amazon CloudWatch Logs?

 **A single, highly scalable service that centralizes the logs from all of your systems, applications, and AWS services that you use** A service that provides a real-time stream of system events that describe changes in AWS resources A service that enables governance, compliance, operational auditing, and risk auditing of your AWS account A service that lets you run code without provisioning or managing servers

**Good job!**

CloudTrail can record the history of events/API calls made within your AWS account, which will help determine who or what deleted the resource. You should investigate it first.

**Question 8:**

If a resource is deleted in AWS, which service should you use to investigate first?

CloudTrail

CloudWatch Logs

Personal Health Dashboard