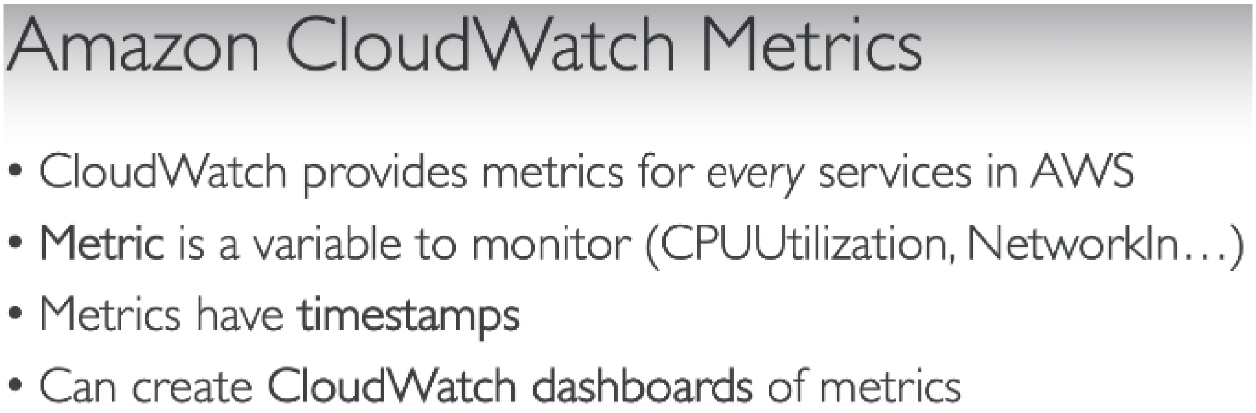
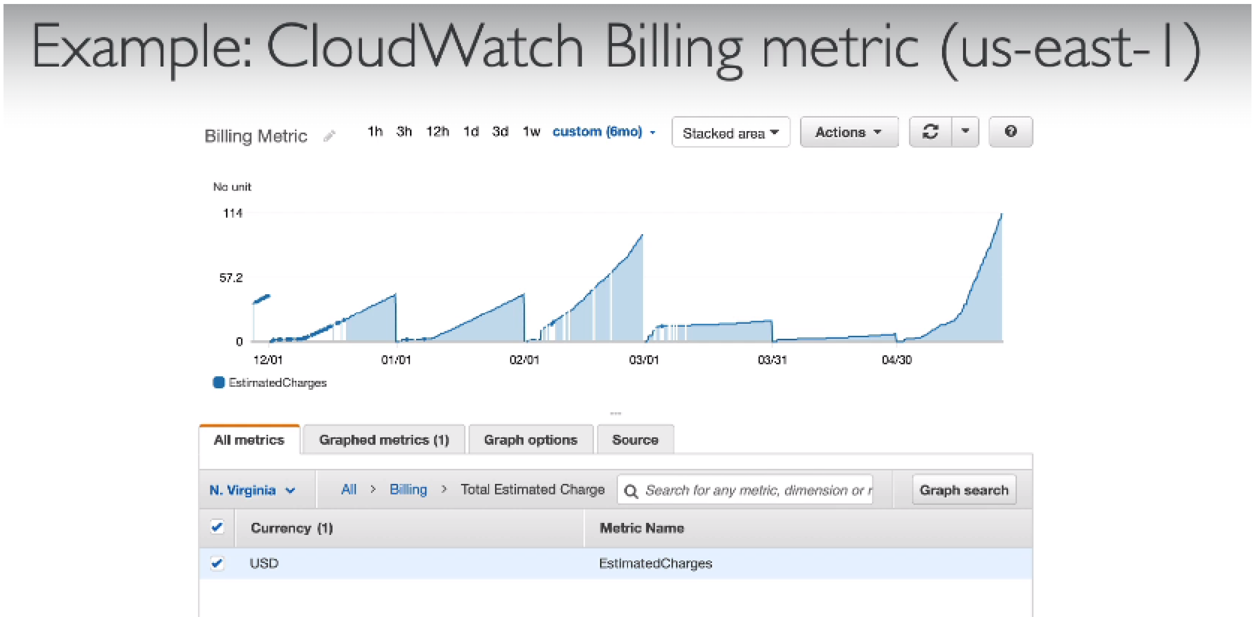
CloudWatch Metrics & CloudWatch Alarm

1. In this section, we're going to know how we can get a better idea and a better picture of the performance of our cloud deployments.
2. 

Amazon CloudWatch Metrics

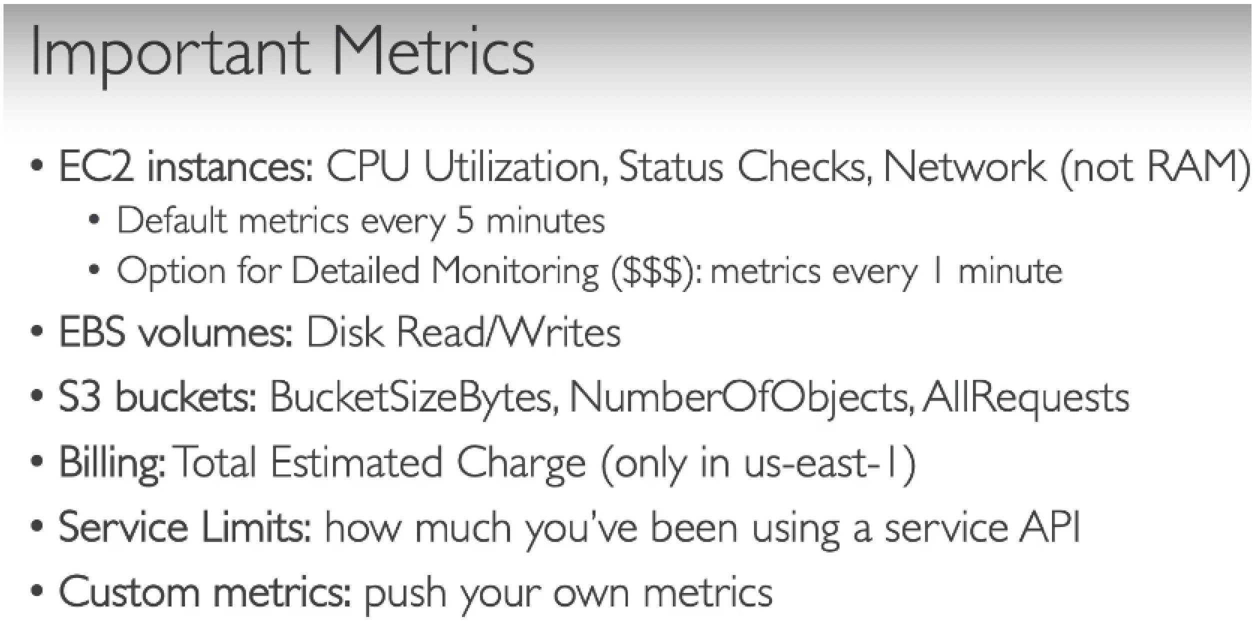
- CloudWatch provides metrics for *every* services in AWS
- **Metric** is a variable to monitor (CPUUtilization, NetworkIn…)
- Metrics have **timestamps**
- Can create **CloudWatch dashboards** of metrics

3. 

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

☑	Currency (1)	Metric Name
☑	USD	EstimatedCharges

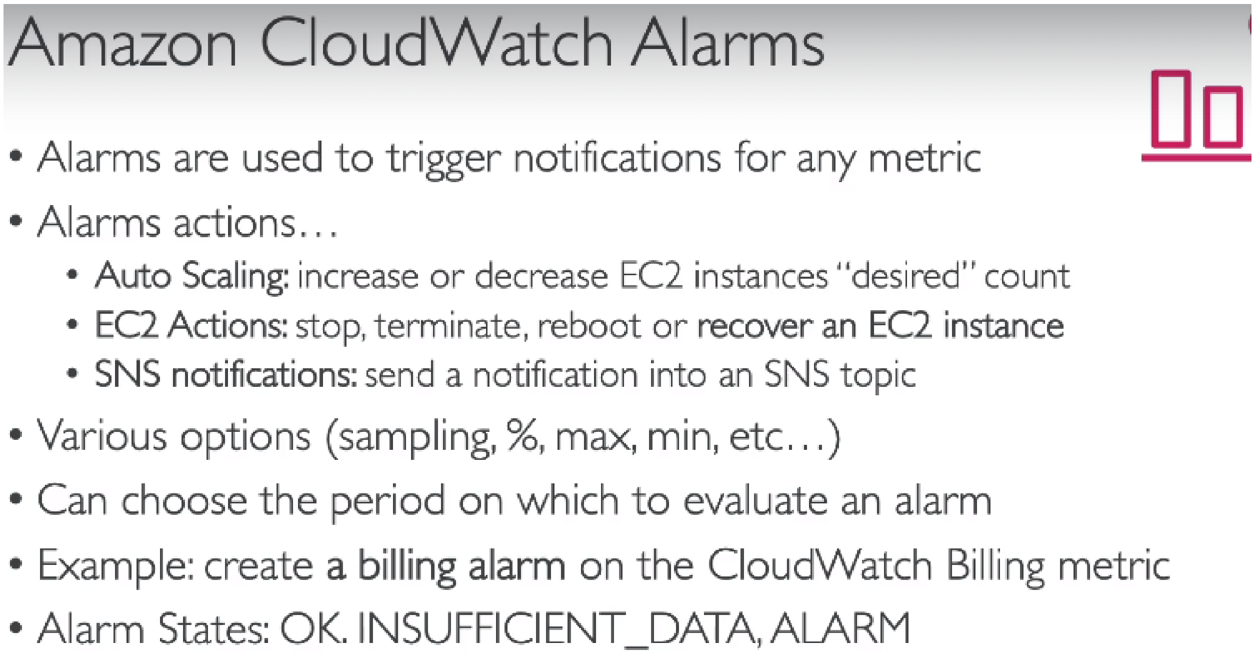
4. So this metric is only available in us-east-l, so only in one region, and it represents the total amounts you have spent on your AWS cloud…after every month it resets to zero
5. For example in our ec2 instance..we have metrics like CPU utilization…based on the utilization..we can scale up or down..
6. We can check the status of our ec2 instance and also how much network is coming in and out



Important Metrics

- **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

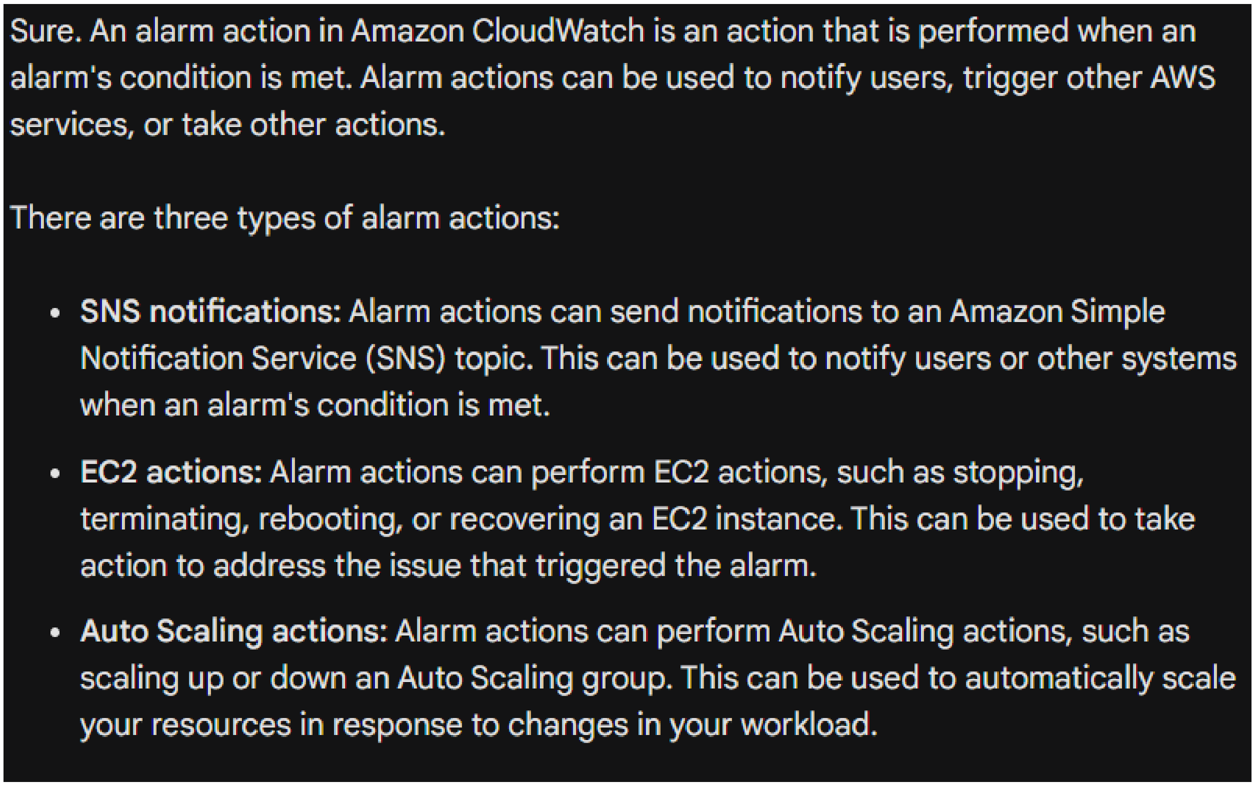
7.



Amazon CloudWatch Alarms

- Alarms are used to trigger notifications for any metric
- Alarms actions…
  - **Auto Scaling:** increase or decrease EC2 instances "desired" count
  - **EC2 Actions:** stop, terminate, reboot or **recover an EC2 instance**
  - **SNS notifications:** send a notification into an SNS topic
- Various options (sampling, %, max, min, etc…)
- Can choose the period on which to evaluate an alarm
- Example: create **a billing alarm** on the CloudWatch Billing metric
- Alarm States: OK. INSUFFICIENT_DATA, ALARM

8.
9. We have 3 alarm states The alarm state can be OK when everything is green,
10. INSUFFICIENT_DATA when there's not enough data points to figure out if it should be green or bad, and then ALARM when it's bad.

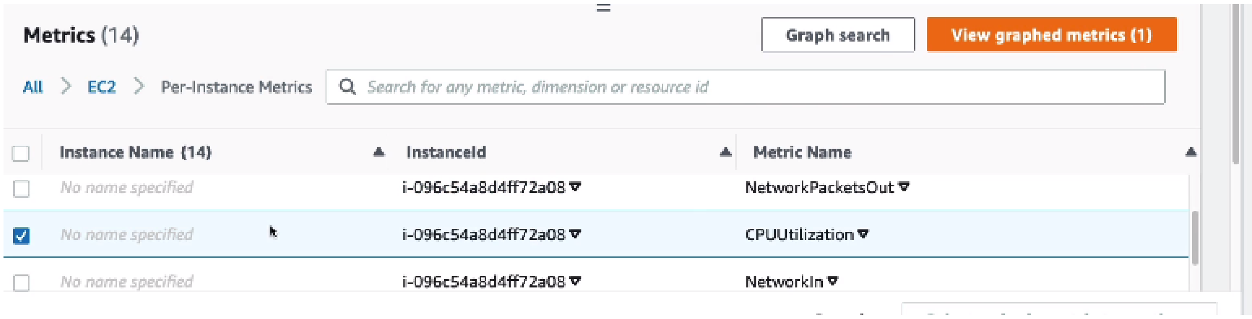
11. 

Sure. An alarm action in Amazon CloudWatch is an action that is performed when an alarm's condition is met. Alarm actions can be used to notify users, trigger other AWS services, or take other actions.

There are three types of alarm actions:

- **SNS notifications:** Alarm actions can send notifications to an Amazon Simple Notification Service (SNS) topic. This can be used to notify users or other systems when an alarm's condition is met.
- **EC2 actions:** Alarm actions can perform EC2 actions, such as stopping, terminating, rebooting, or recovering an EC2 instance. This can be used to take action to address the issue that triggered the alarm.
- **Auto Scaling actions:** Alarm actions can perform Auto Scaling actions, such as scaling up or down an Auto Scaling group. This can be used to automatically scale your resources in response to changes in your workload.

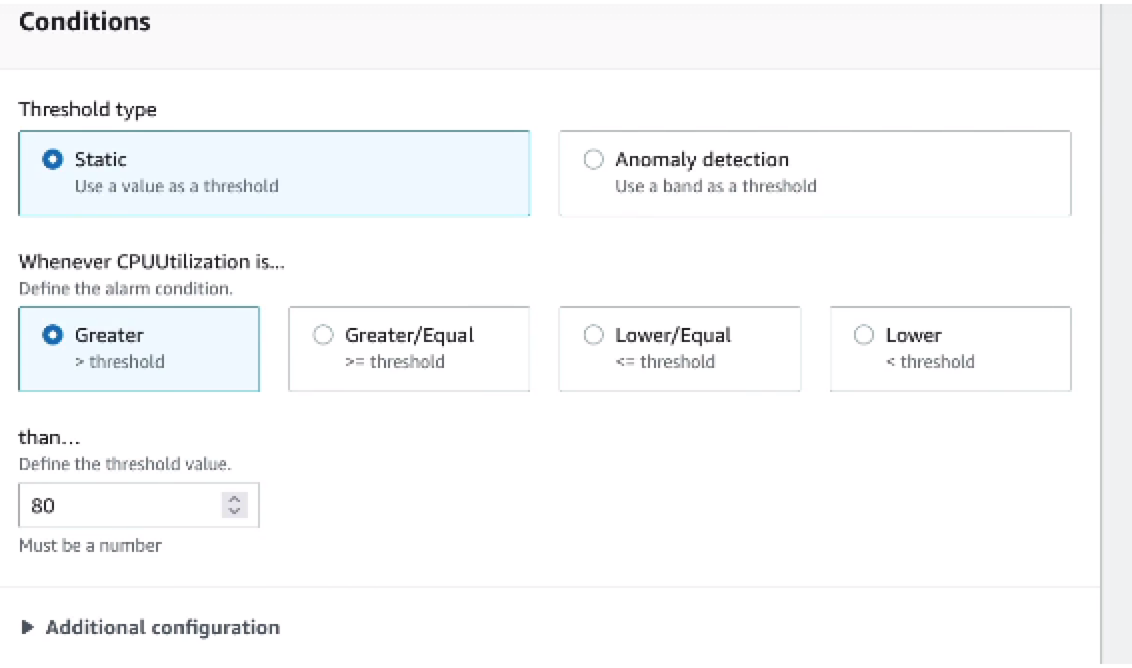
Cloud Watch metric's hands on

1. First head to cloud watch and then go to metrics..Here we can every metrics for our account
2. For example ..check s3 metrics…it has metrics like "number of bytes uses, num of objects etc
3. Next we will create an alarm..for creating an alarm go to alarms and select metrics
4. 

Metrics (14)
Graph search
View graphed metrics (1)
All > EC2 > Per-Instance Metrics
Search for any metric, dimension or resource id

☐	Instance Name (14)	InstanceId	Metric Name
☐	No name specified	i-096c54a8d4ff72a08	NetworkPacketsOut
☑	No name specified	i-096c54a8d4ff72a08	CPUUtilization
☐	No name specified	i-096c54a8d4ff72a08	NetworkIn

5. Here we have created a metric for an ec2 instance on CPU utilization
6. Next choose math on how to calculator metric



Conditions

Threshold type

● Static
Use a value as a threshold

○ Anomaly detection
Use a band as a threshold

Whenever CPUUtilization is...
Define the alarm condition.

● Greater
> threshold

○ Greater/Equal
>= threshold

○ Lower/Equal
<= threshold

○ Lower
< threshold

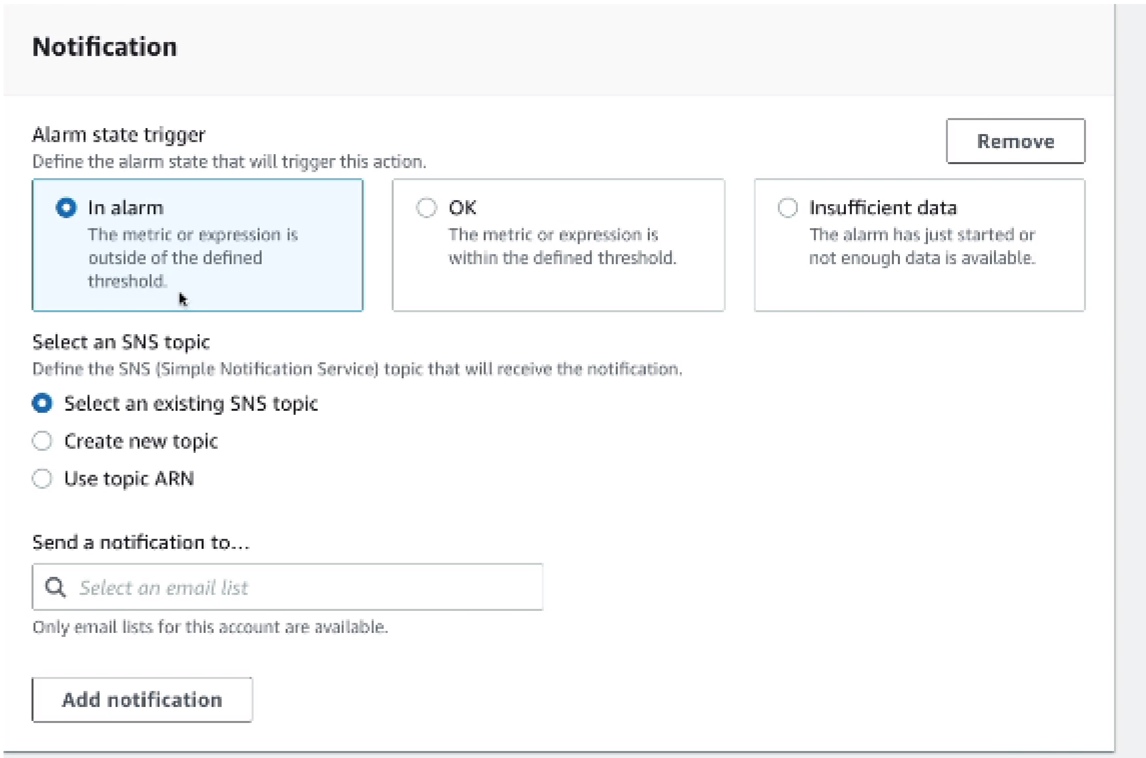
than...
Define the threshold value.

80

Must be a number

▶ Additional configuration

7.
8. If the cpu utilization is more than 80%..make it as a alarm
9. Next choose an alert



Notification

Alarm state trigger
Define the alarm state that will trigger this action.

● In alarm
The metric or expression is outside of the defined threshold.

○ OK
The metric or expression is within the defined threshold.

○ Insufficient data
The alarm has just started or not enough data is available.

Select an SNS topic
Define the SNS (Simple Notification Service) topic that will receive the notification.

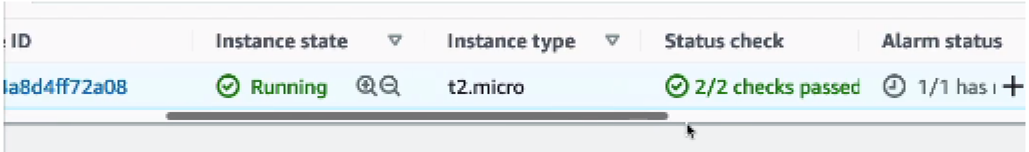
● Select an existing SNS topic
○ Create new topic
○ Use topic ARN

Send a notification to...

Select an email list

Only email lists for this account are available.

10.
11. We can also directly create an alarm from our ec2 instance..by clicking on alarm status



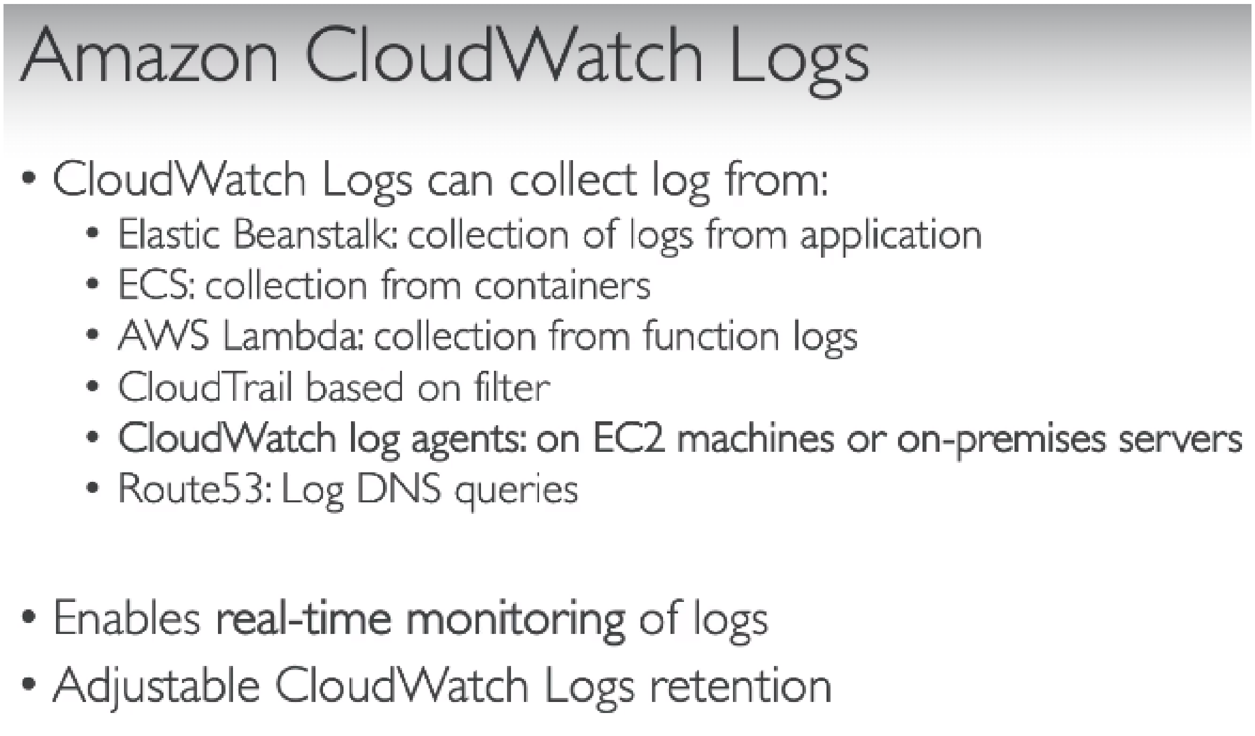
ID	Instance state	Instance type	Status check	Alarm status
a8d4ff72a08	✓ Running	t2.micro	✓ 2/2 checks passed	1/1 has i +

12.

13. Billing alarms will be available only in US-east-1 region..

AWS CloudWatch Logs

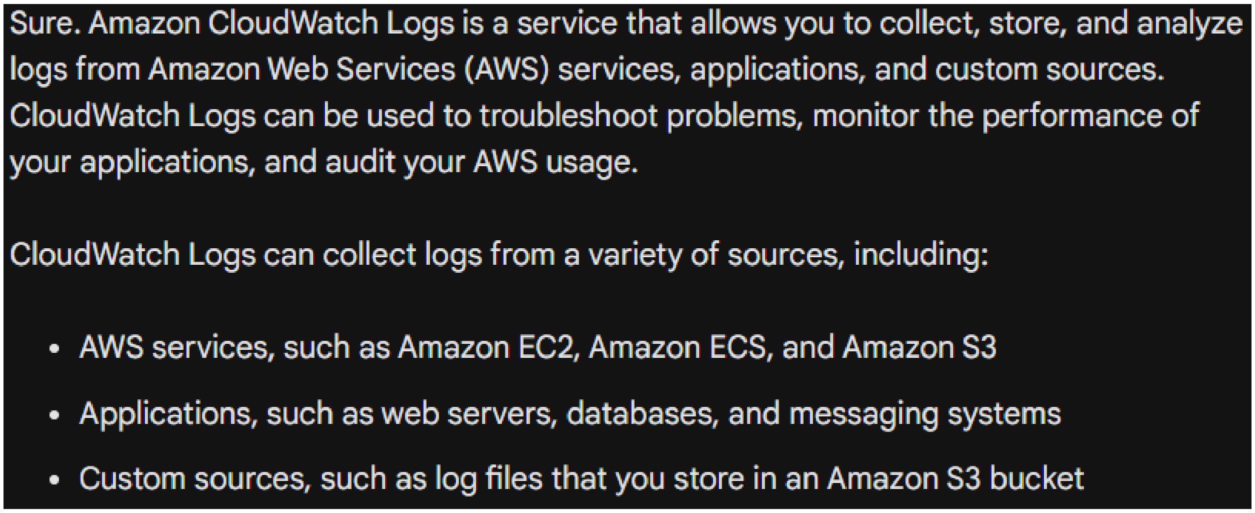
1. Logs helps us to debug ..if there are any errors ..or if we want to know how a thing works



Amazon CloudWatch Logs

- CloudWatch Logs can collect log 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
- Enables **real-time monitoring** of logs
- Adjustable CloudWatch Logs retention

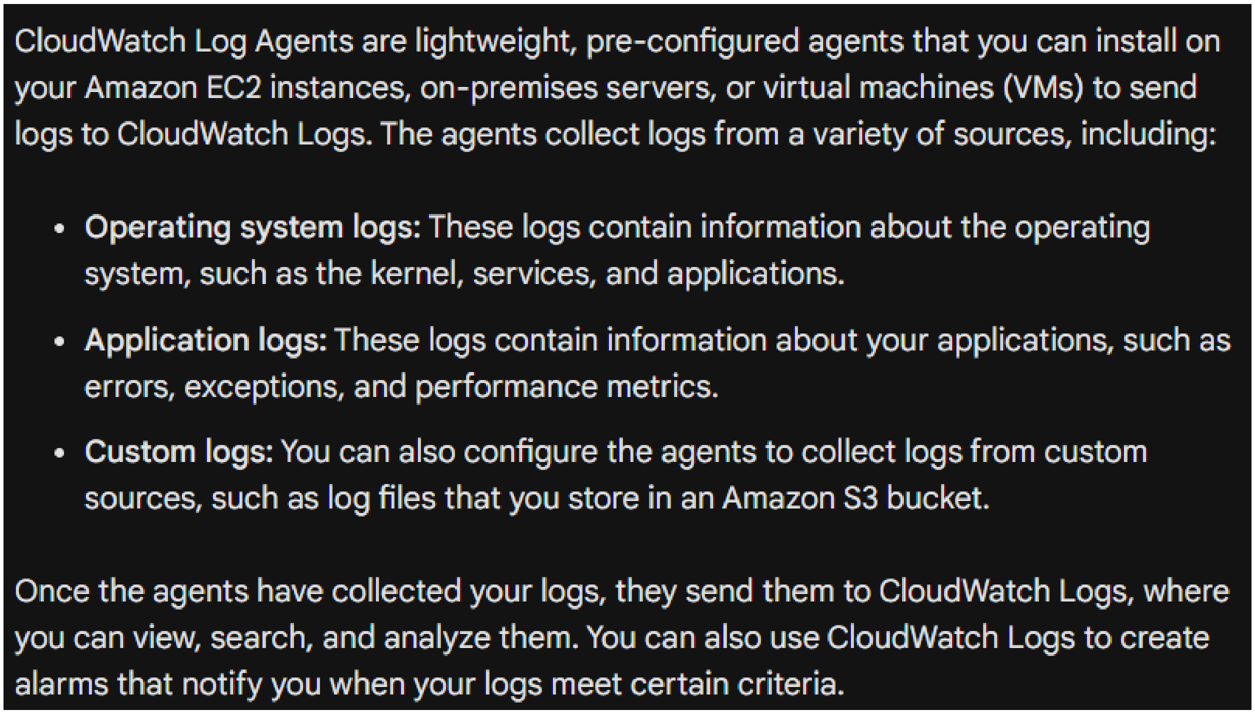
2.
3. We can also retain our cloudwatch logs ..infinitel
4.



Sure. Amazon CloudWatch Logs is a service that allows you to collect, store, and analyze logs from Amazon Web Services (AWS) services, applications, and custom sources. CloudWatch Logs can be used to troubleshoot problems, monitor the performance of your applications, and audit your AWS usage.

CloudWatch Logs can collect logs from a variety of sources, including:

- AWS services, such as Amazon EC2, Amazon ECS, and Amazon S3
- Applications, such as web servers, databases, and messaging systems
- Custom sources, such as log files that you store in an Amazon S3 bucket

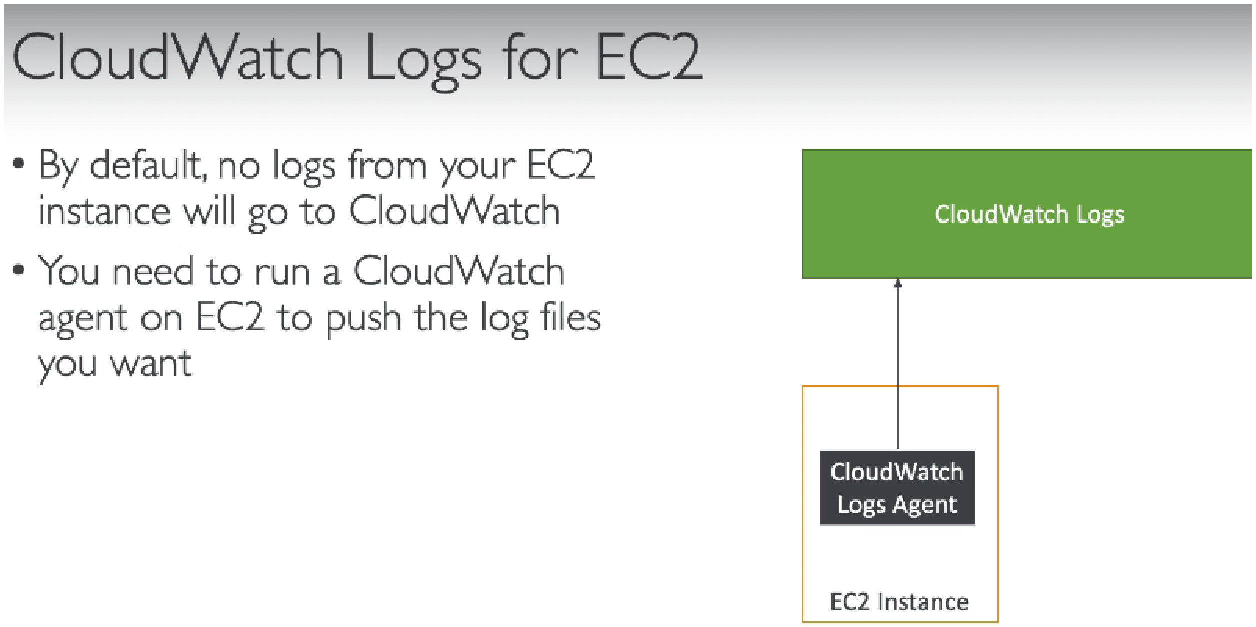
5. 

CloudWatch Log Agents are lightweight, pre-configured agents that you can install on your Amazon EC2 instances, on-premises servers, or virtual machines (VMs) to send logs to CloudWatch Logs. The agents collect logs from a variety of sources, including:

- **Operating system logs:** These logs contain information about the operating system, such as the kernel, services, and applications.
- **Application logs:** These logs contain information about your applications, such as errors, exceptions, and performance metrics.
- **Custom logs:** You can also configure the agents to collect logs from custom sources, such as log files that you store in an Amazon S3 bucket.

Once the agents have collected your logs, they send them to CloudWatch Logs, where you can view, search, and analyze them. You can also use CloudWatch Logs to create alarms that notify you when your logs meet certain criteria.

6. By default we don't get logs for our ec2 instance

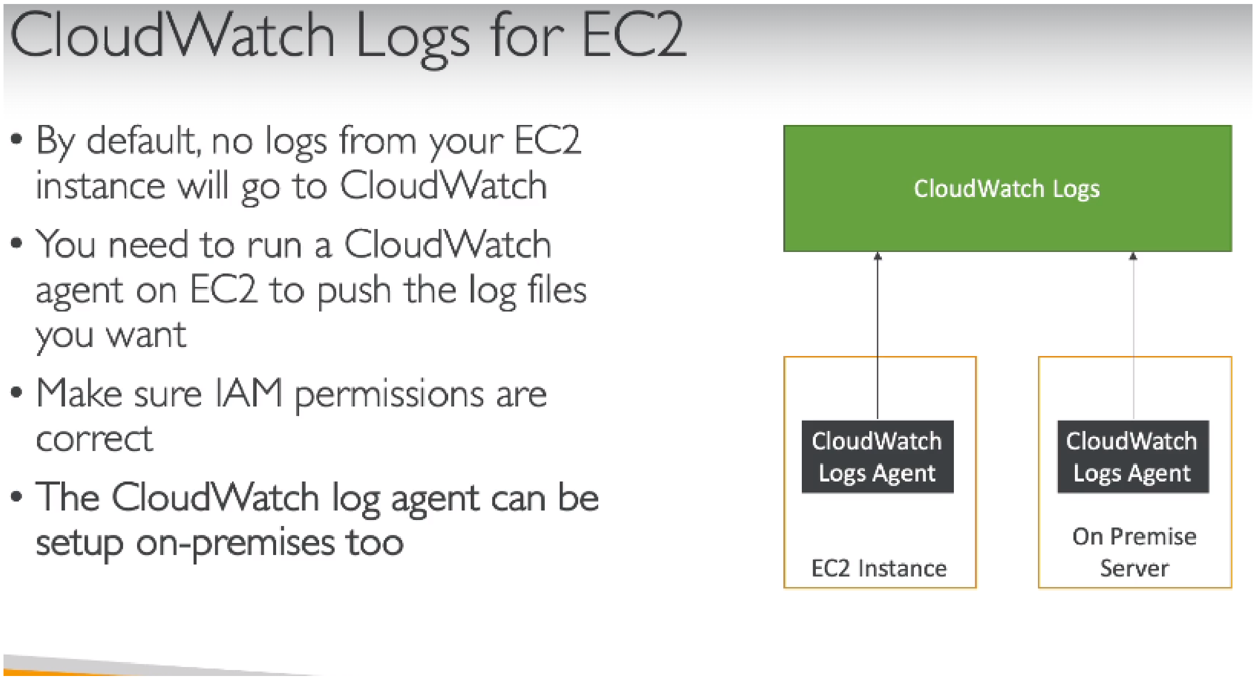


CloudWatch Logs for EC2

- 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

7.

8. For this to work we need to make sure that our EC2 instance has a proper instance role with the correct IAM permissions to send the log data into CloudWatch Logs.

9. 

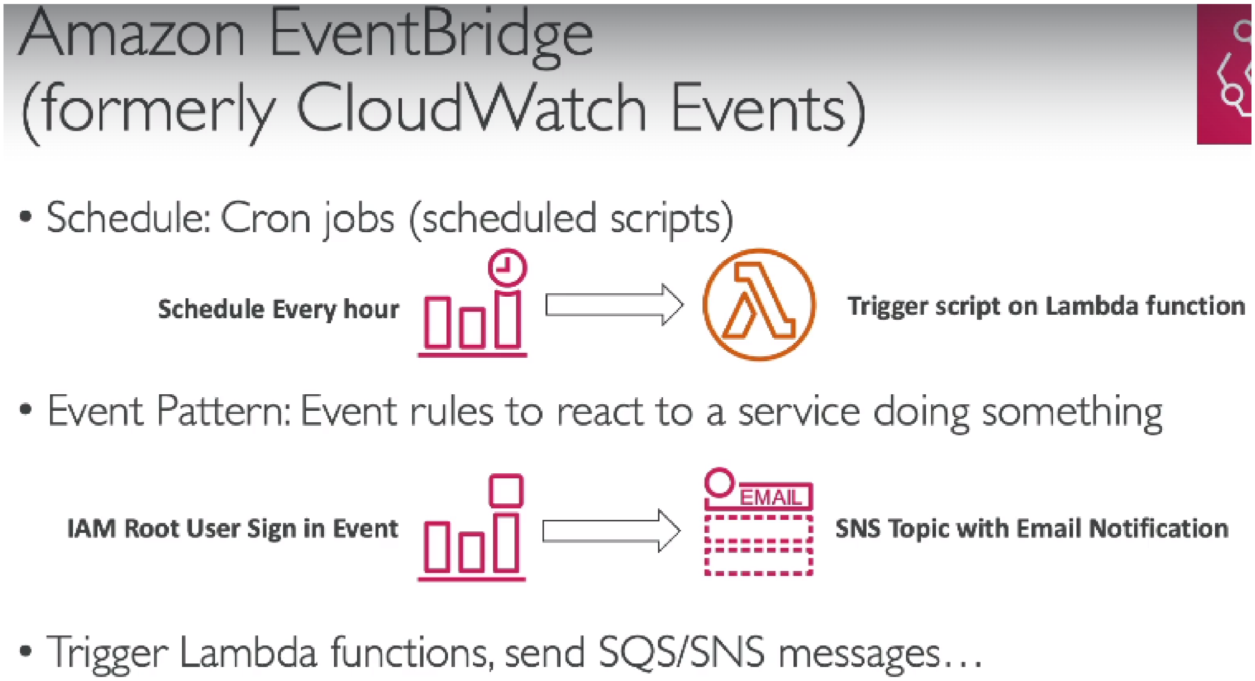
CloudWatch Logs for EC2

- 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

Cloudwatch logs HandsOn

1. We have log groups..and by default when we create a lambda..it creates a log group for us
2. In lambda..if there are any errors in our code..we can check the logs
3. Check online for more handsOn

Amazon EventBridge



Amazon EventBridge
(formerly CloudWatch Events)

- Schedule: Cron jobs (scheduled scripts)

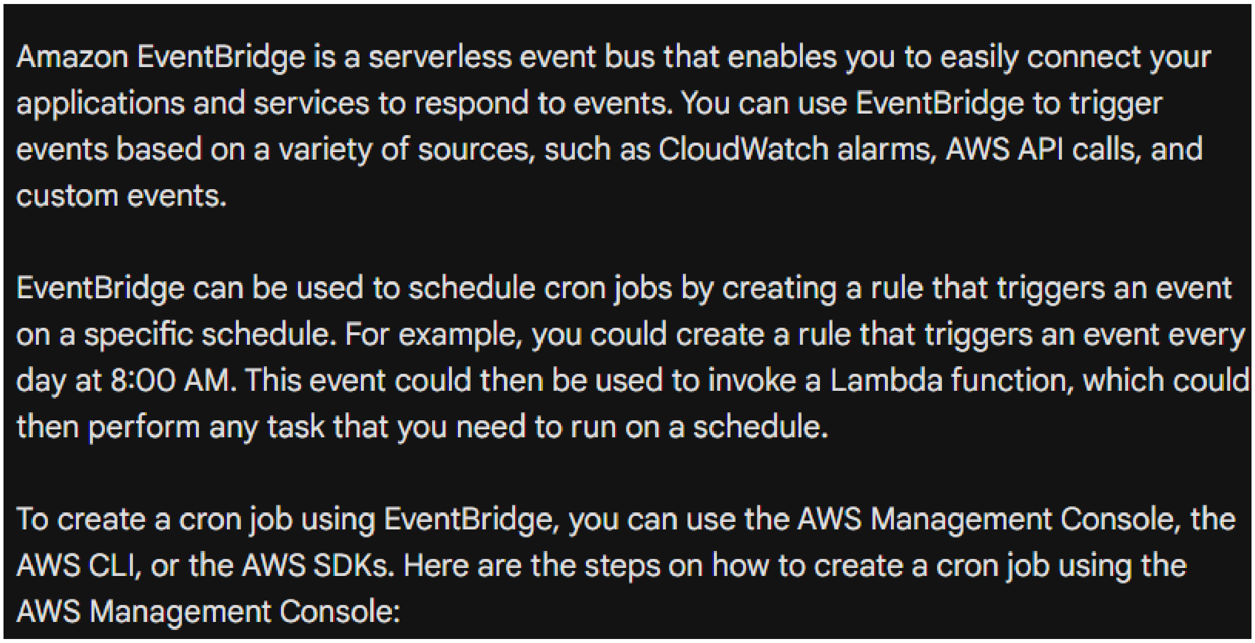
Schedule Every hour
Trigger script on Lambda function

- Event Pattern: Event rules to react to a service doing something

IAM Root User Sign in Event
EMAIL
SNS Topic with Email Notification

- Trigger Lambda functions, send SQS/SNS messages…

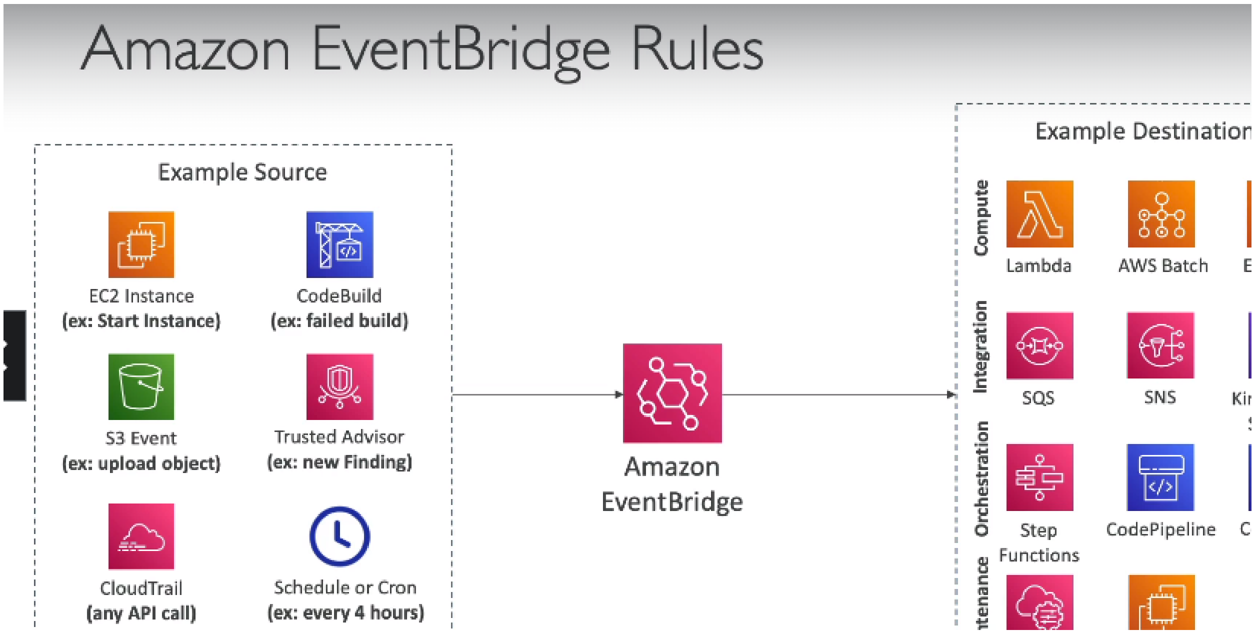
1.
2. Suppose if any IAM user logs into root user..we can trigger an event and send SNS topic notification
3.



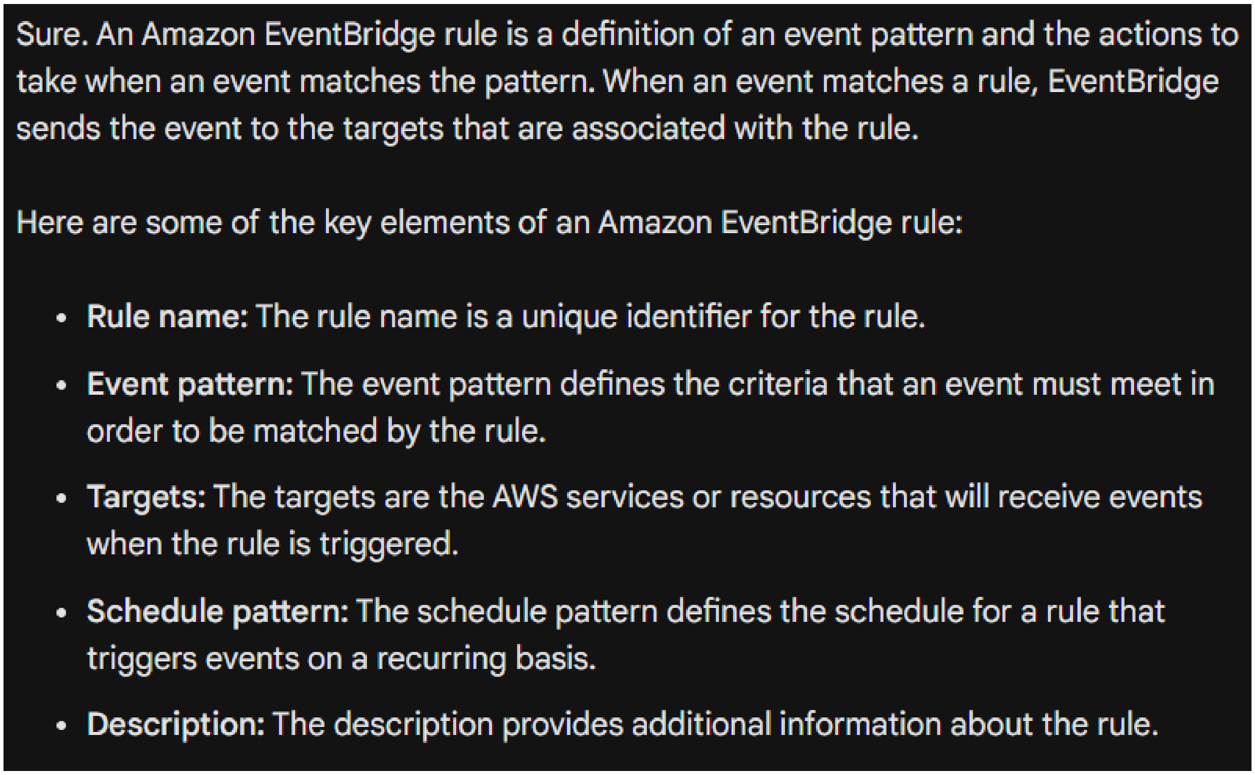
Amazon EventBridge is a serverless event bus that enables you to easily connect your applications and services to respond to events. You can use EventBridge to trigger events based on a variety of sources, such as CloudWatch alarms, AWS API calls, and custom events.

EventBridge can be used to schedule cron jobs by creating a rule that triggers an event on a specific schedule. For example, you could create a rule that triggers an event every day at 8:00 AM. This event could then be used to invoke a Lambda function, which could then perform any task that you need to run on a schedule.

To create a cron job using EventBridge, you can use the AWS Management Console, the AWS CLI, or the AWS SDKs. Here are the steps on how to create a cron job using the AWS Management Console:

4. 

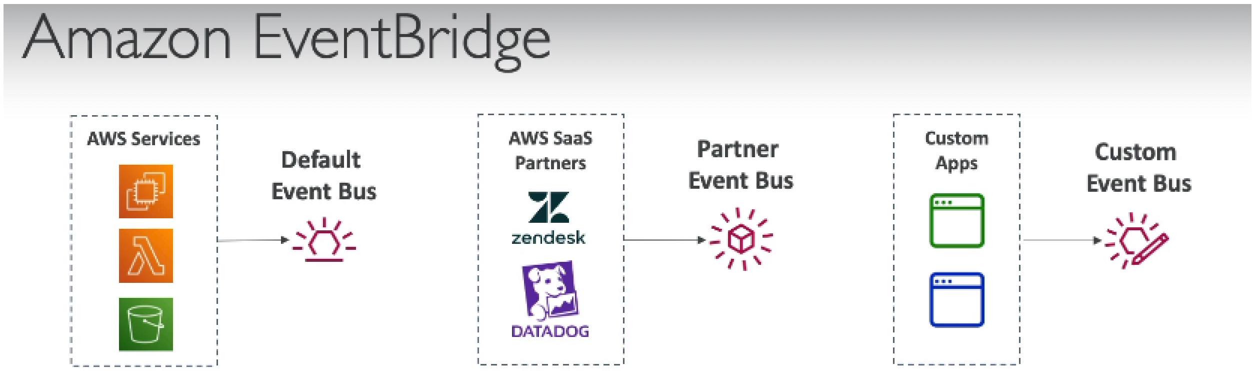
Amazon EventBridge Rules

5. 

Sure. An Amazon EventBridge rule is a definition of an event pattern and the actions to take when an event matches the pattern. When an event matches a rule, EventBridge sends the event to the targets that are associated with the rule.

Here are some of the key elements of an Amazon EventBridge rule:

- **Rule name:** The rule name is a unique identifier for the rule.
- **Event pattern:** The event pattern defines the criteria that an event must meet in order to be matched by the rule.
- **Targets:** The targets are the AWS services or resources that will receive events when the rule is triggered.
- **Schedule pattern:** The schedule pattern defines the schedule for a rule that triggers events on a recurring basis.
- **Description:** The description provides additional information about the rule.



Amazon EventBridge

AWS Services

Default Event Bus

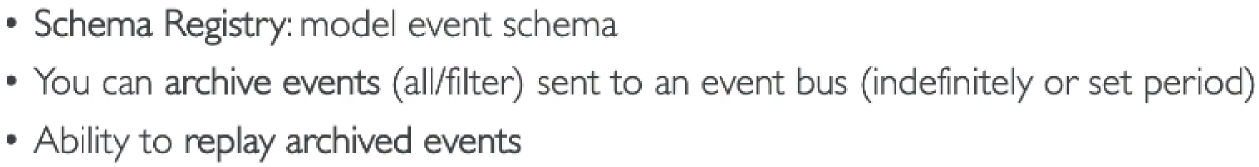
AWS SaaS Partners

Partner Event Bus

Custom Apps

Custom Event Bus

6.
7. For example, if you're using Zendesk, or Datadog, or others that are partnered with AWS, then they can send their own events into your account through a partner event bus, and, therefore, you can react to events happening outside of AWS as well.



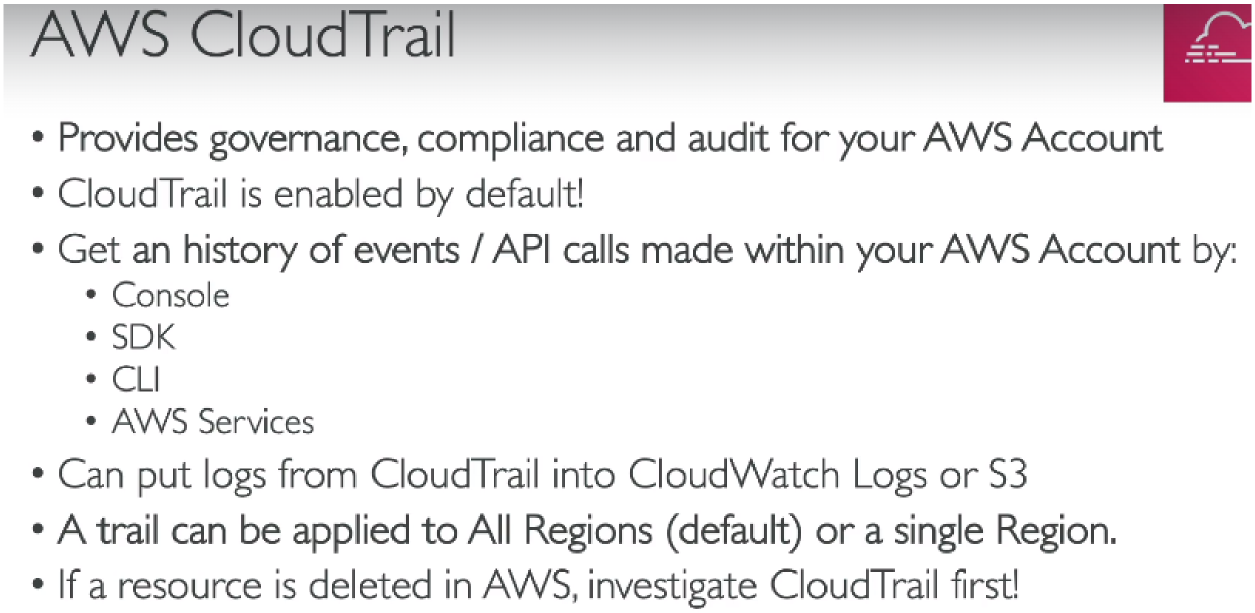
- Schema Registry: model event schema
- You can archive events (all/filter) sent to an event bus (indefinitely or set period)
- Ability to replay archived events

8.

Event Bridge HandsOn

1. Refer Online

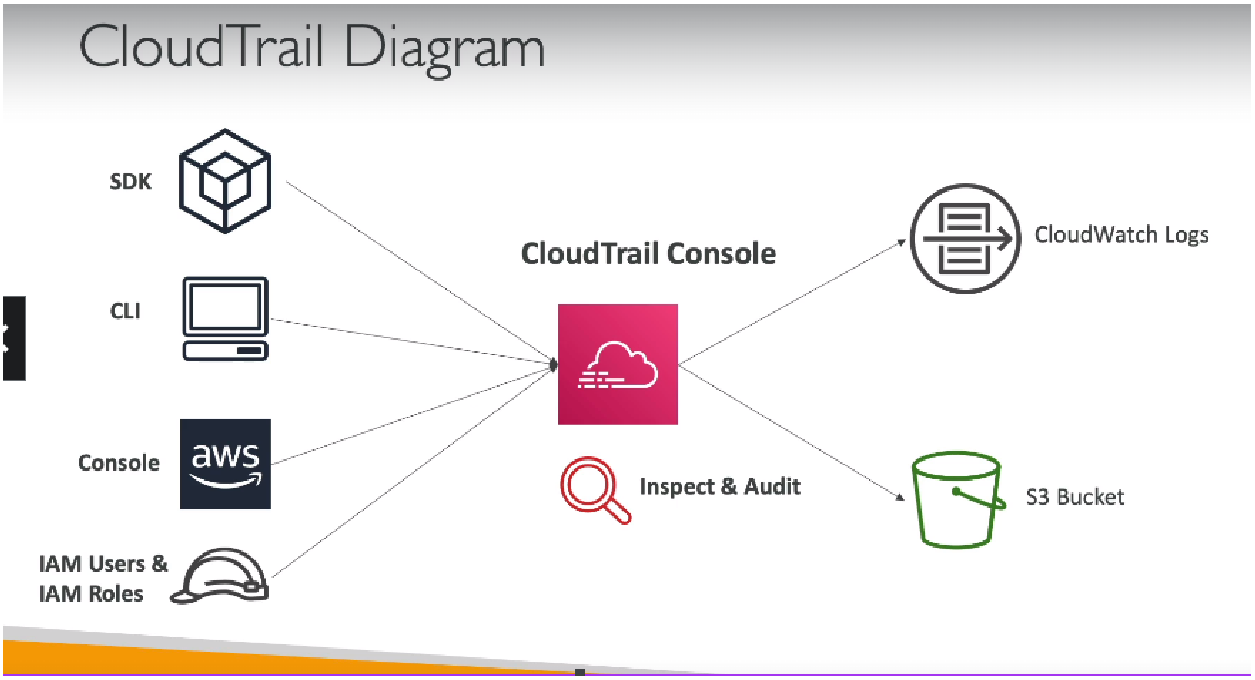
AWS CLoudTrail



AWS 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
- Can put logs from CloudTrail into CloudWatch Logs or S3
- A trail can be applied to All Regions (default) or a single Region.
- If a resource is deleted in AWS, investigate CloudTrail first!

1.
2. It stores the events..such if someone logs into console..etc
3. We can also investigate the things inside aws using cloudtrail
4. a user has deleted something. How would we know what has been deleted and who deleted it and when? Then the answer is going to be CloudTrail.
5. From within the CloudTrail console we can have information about usage of the SDK, CLI and console, as well as any IAM users and IAM roles and all the API calls they make, then the CloudTrail consult will display it.
6. But if you want long term retention of data what you can do is that you can send them to CloudWatch Logs or to your S3 bucket for longer term retention.

7. 

CloudTrail Diagram

SDK

CLI

Console

IAM Users &
IAM Roles

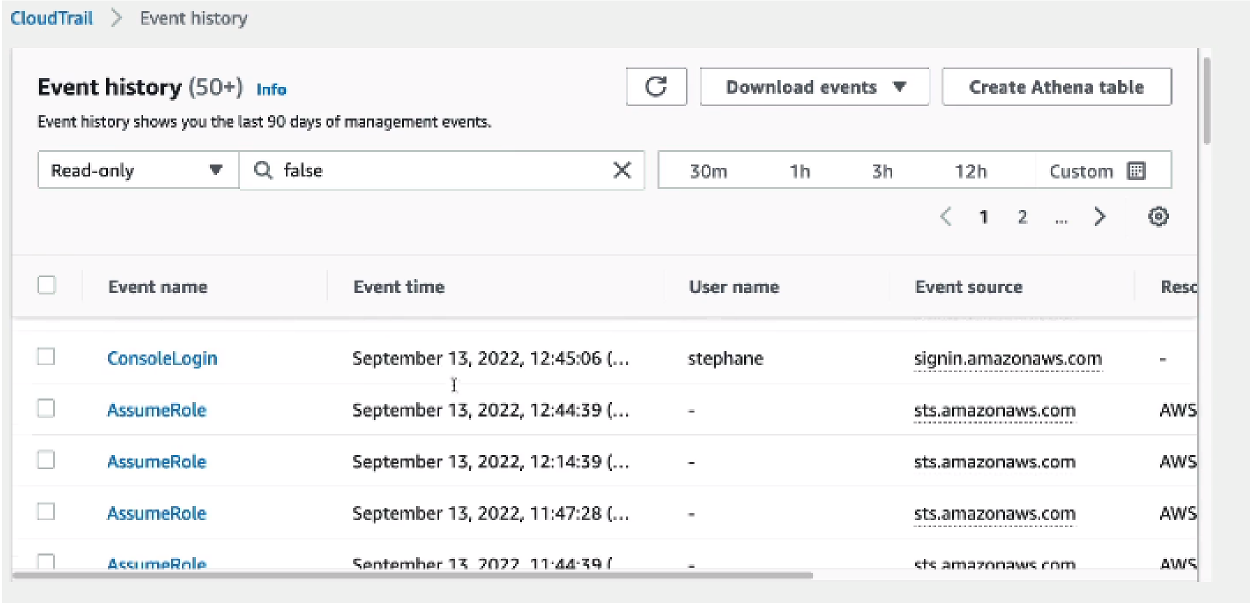
CloudTrail Console

Inspect & Audit

CloudWatch Logs

S3 Bucket

CloudTrail HandsOn:

1. 

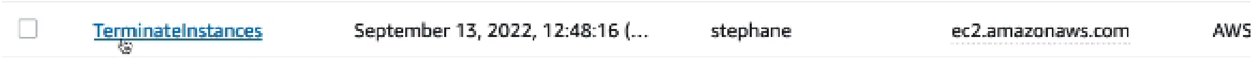
Event history (50+) Info

Event history shows you the last 90 days of management events.

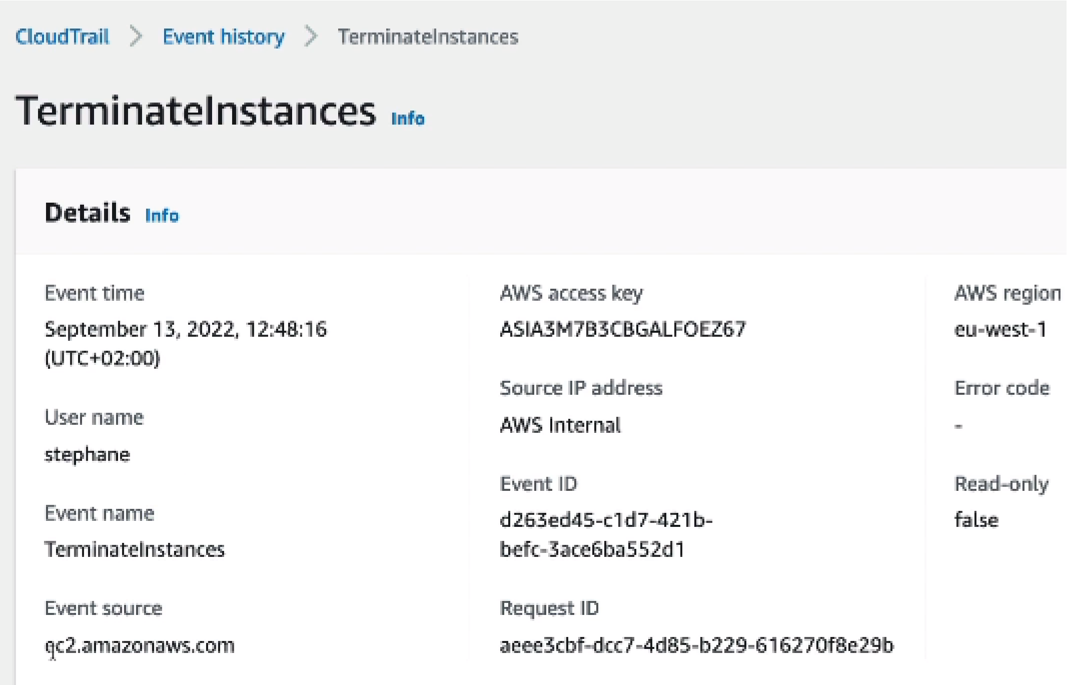
Read-only: false

□	Event name	Event time	User name	Event source	Resc
□	ConsoleLogin	September 13, 2022, 12:45:06 (...	stephane	signin.amazonaws.com	-
□	AssumeRole	September 13, 2022, 12:44:39 (...	-	sts.amazonaws.com	AWS
□	AssumeRole	September 13, 2022, 12:14:39 (...	-	sts.amazonaws.com	AWS
□	AssumeRole	September 13, 2022, 11:47:28 (...	-	sts.amazonaws.com	AWS
□	AssumeRole	September 13, 2022, 11:44:39 (...	-	sts.amazonaws.com	AWS

2. Here we have terminated an instance …and it has recorded that event



□	TerminateInstances	September 13, 2022, 12:48:16 (...	stephane	ec2.amazonaws.com	AWS

3. 

CloudTrail > Event history > TerminateInstances

TerminateInstances Info

Details Info

Event time
September 13, 2022, 12:48:16 (UTC+02:00)

User name
stephane

Event name
TerminateInstances

Event source
ec2.amazonaws.com

AWS access key
ASIA3M7B3CBGALFOEZ67

Source IP address
AWS Internal

Event ID
d263ed45-c1d7-421b-befc-3ace6ba552d1

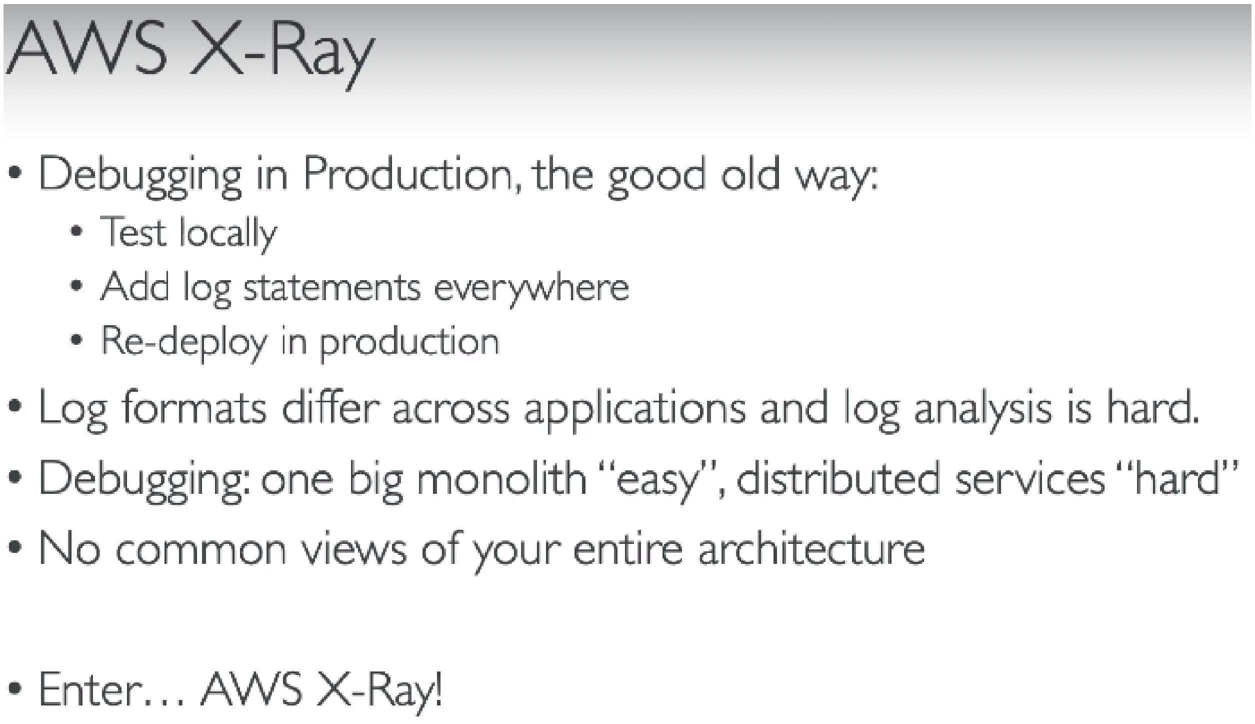
Request ID
aeee3cbf-dcc7-4d85-b229-616270f8e29b

AWS region
eu-west-1

Error code
-

Read-only
false

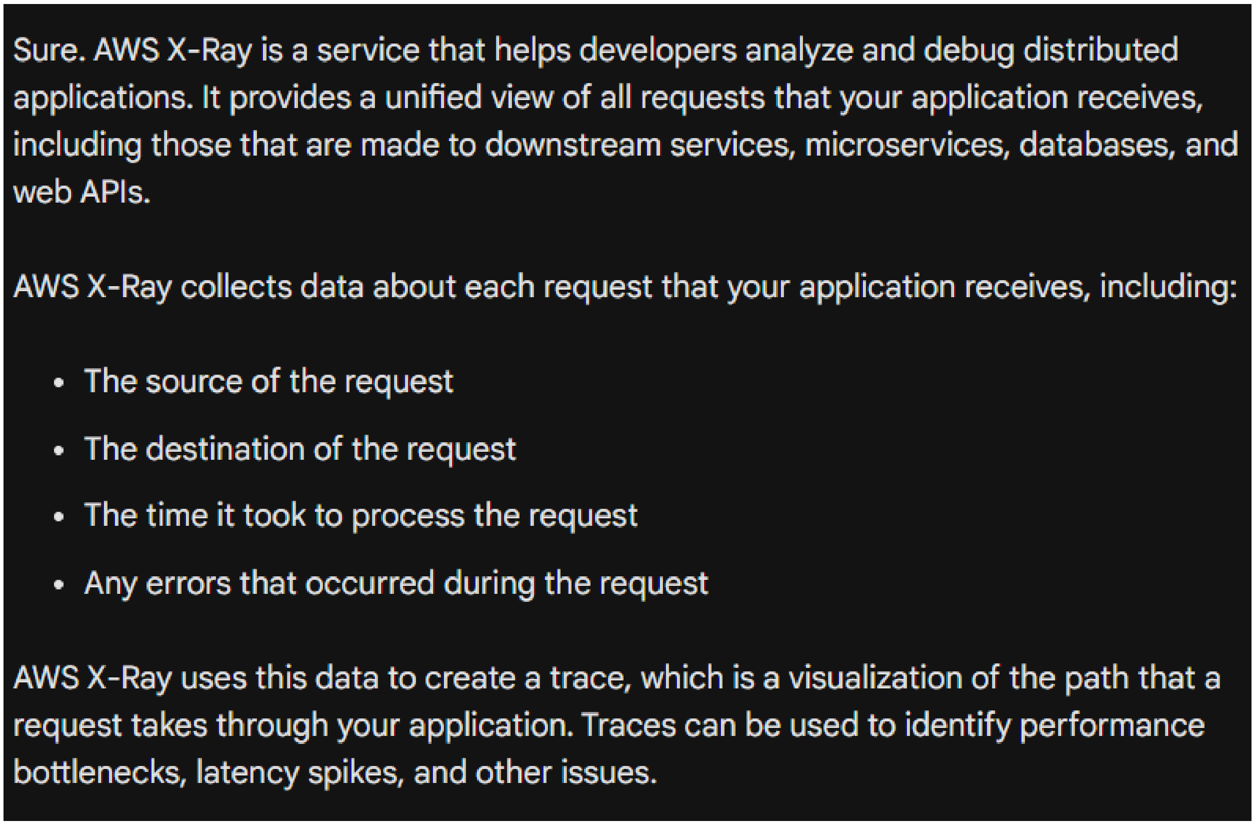
AWS X-Ray

1. 

AWS X-Ray

- 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!

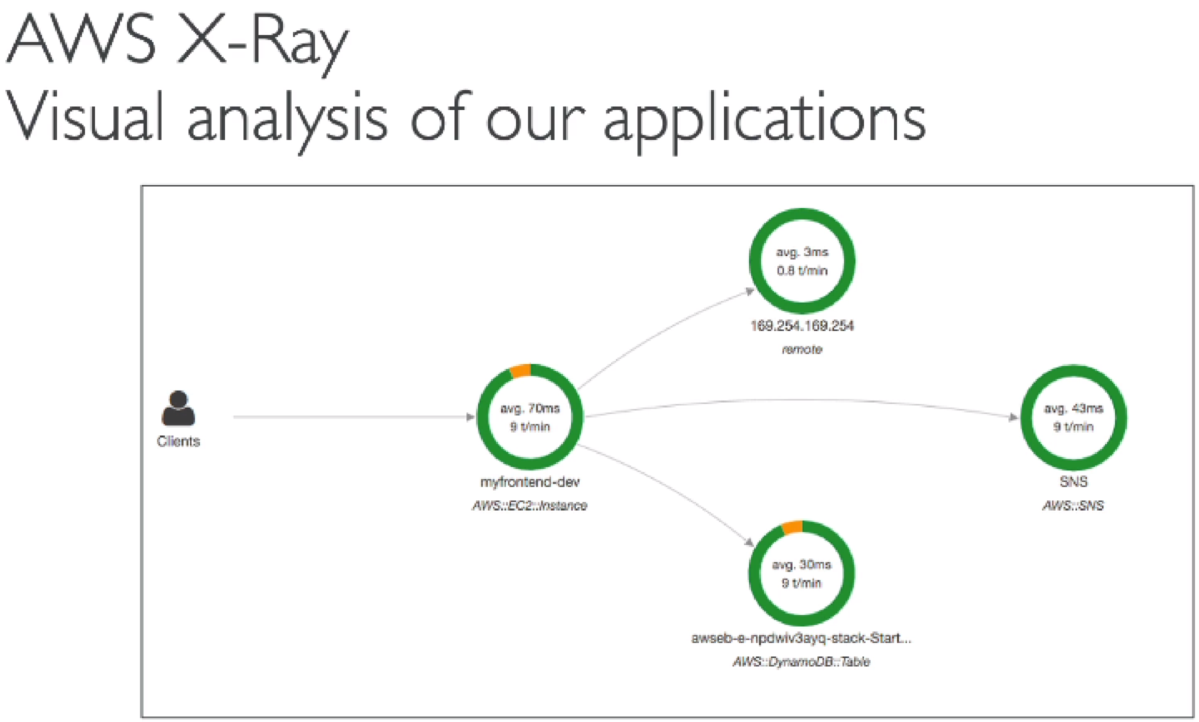
2. Doing log analysis is very hard because we have to combine everything. So, if you have one application that's called a big monolith, so, one giant application, it's sort of easy to do debugging.
3. But if you have distributed services they're connected through SQS queues, SNS topics, they're decoupled and so on, it becomes really, really hard to trace and see what is happening within your system.
4. 

Sure. AWS X-Ray is a service that helps developers analyze and debug distributed applications. It provides a unified view of all requests that your application receives, including those that are made to downstream services, microservices, databases, and web APIs.

AWS X-Ray collects data about each request that your application receives, including:

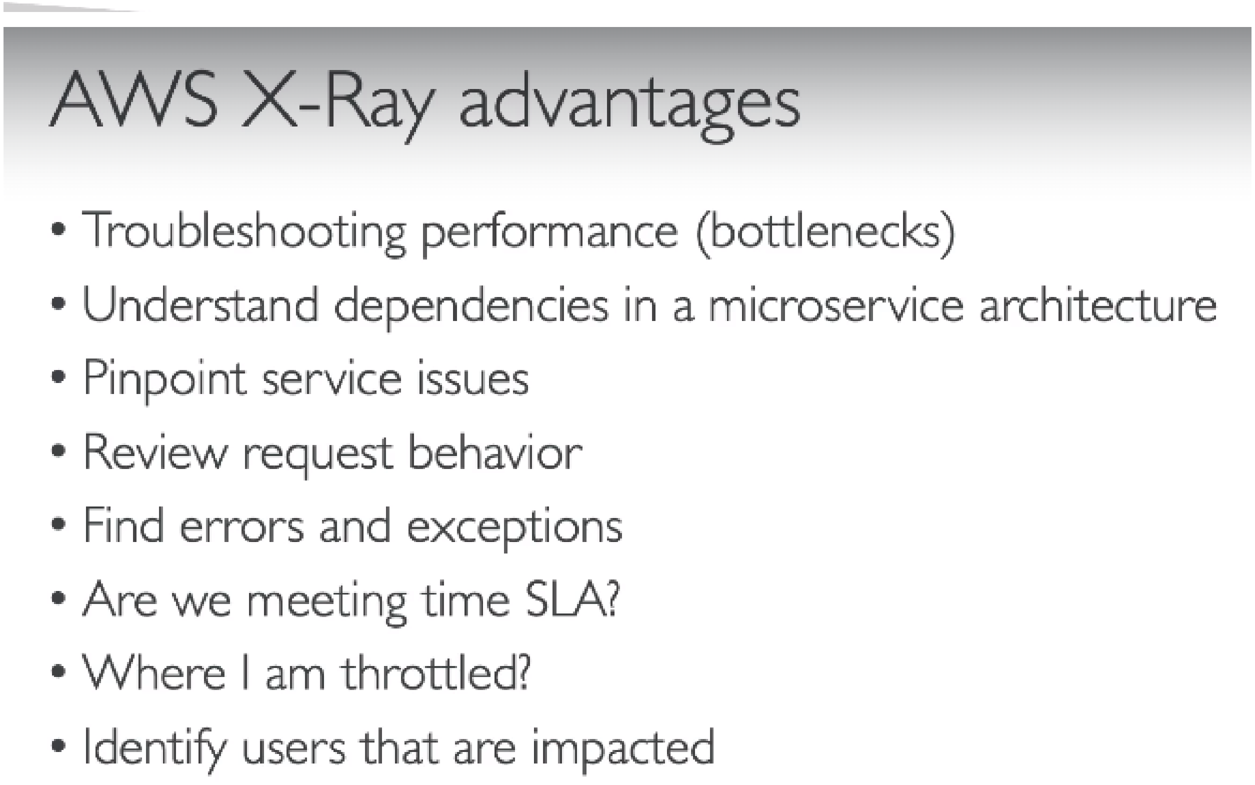
- The source of the request
- The destination of the request
- The time it took to process the request
- Any errors that occurred during the request

AWS X-Ray uses this data to create a trace, which is a visualization of the path that a request takes through your application. Traces can be used to identify performance bottlenecks, latency spikes, and other issues.



AWS X-Ray
Visual analysis of our applications

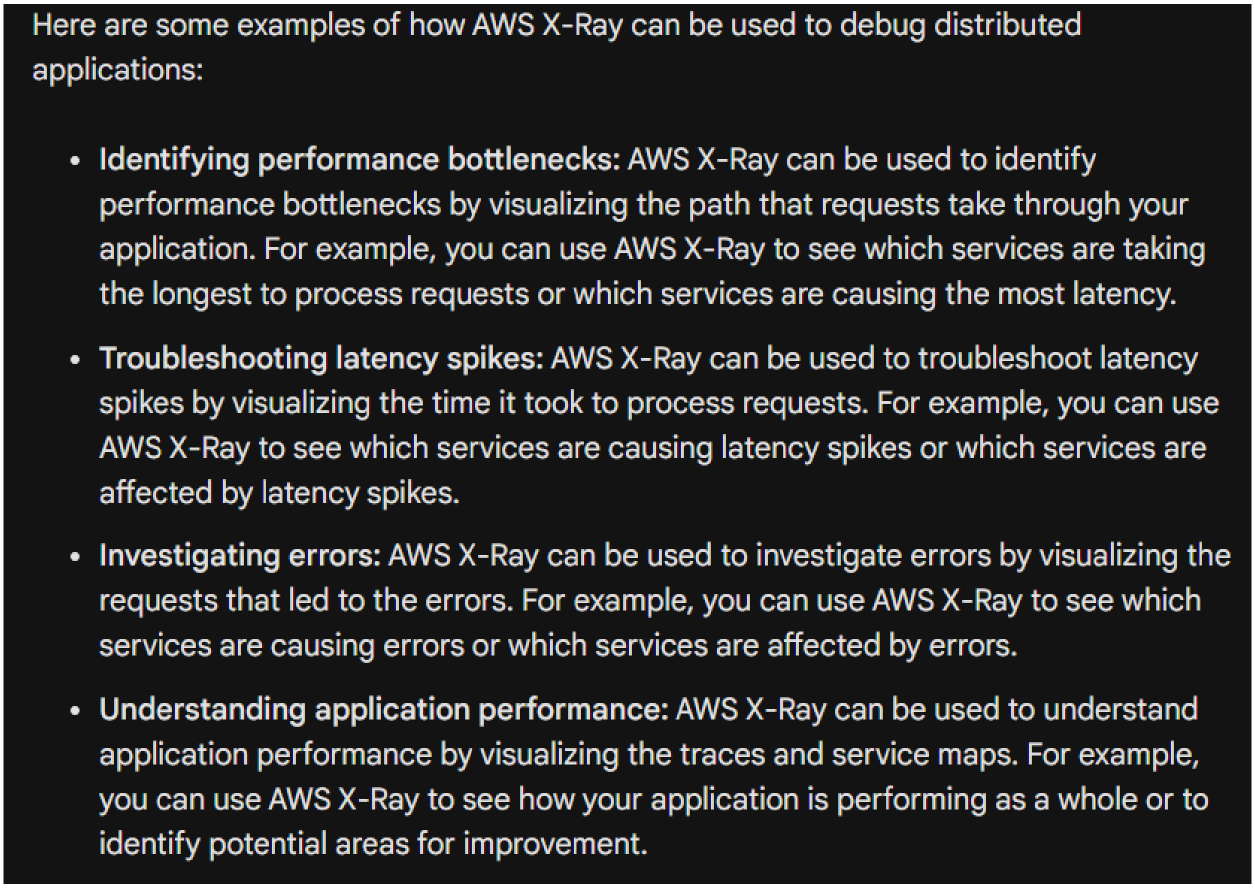
5.



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

6.

7. 

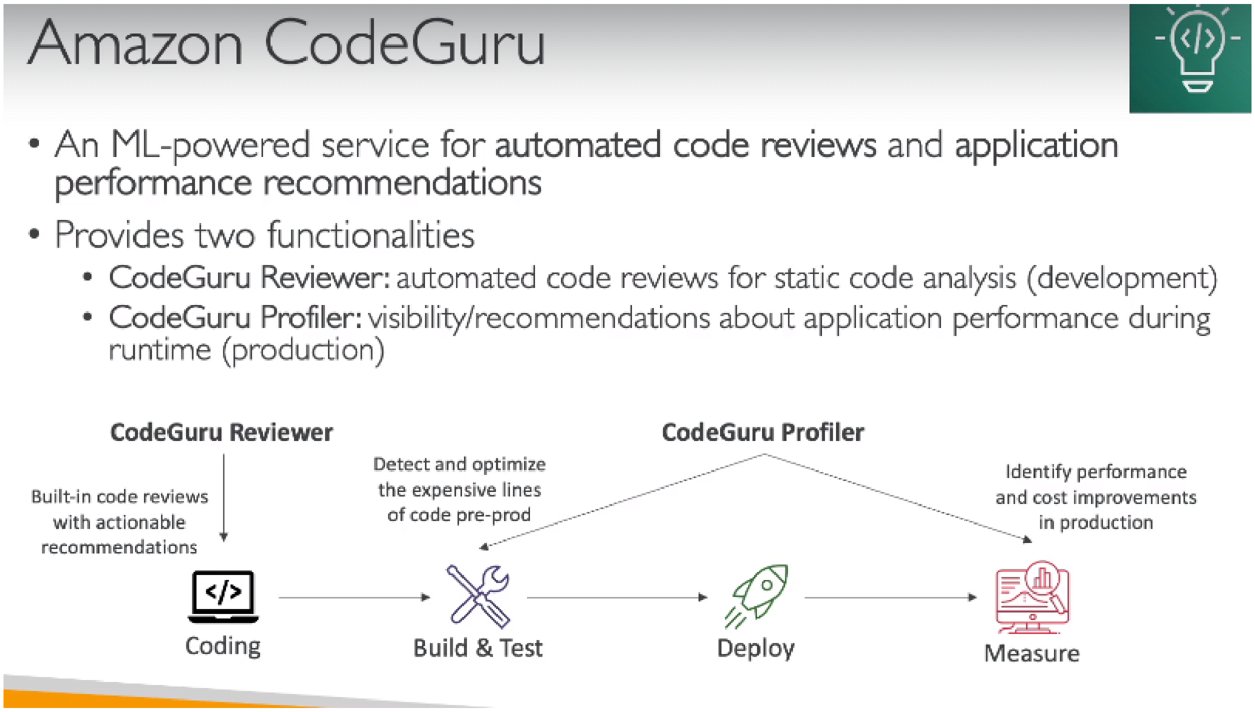
Here are some examples of how AWS X-Ray can be used to debug distributed applications:

- **Identifying performance bottlenecks:** AWS X-Ray can be used to identify performance bottlenecks by visualizing the path that requests take through your application. For example, you can use AWS X-Ray to see which services are taking the longest to process requests or which services are causing the most latency.
- **Troubleshooting latency spikes:** AWS X-Ray can be used to troubleshoot latency spikes by visualizing the time it took to process requests. For example, you can use AWS X-Ray to see which services are causing latency spikes or which services are affected by latency spikes.
- **Investigating errors:** AWS X-Ray can be used to investigate errors by visualizing the requests that led to the errors. For example, you can use AWS X-Ray to see which services are causing errors or which services are affected by errors.
- **Understanding application performance:** AWS X-Ray can be used to understand application performance by visualizing the traces and service maps. For example, you can use AWS X-Ray to see how your application is performing as a whole or to identify potential areas for improvement.

Amazon Code Guru

1. It is an ML-powered services..which helps us to automate code review and gives performance recommendation
2. Usually when we deploy code..our co-workers will check for any errors…instead of that codeguru handles it with ML-algorithms

3.



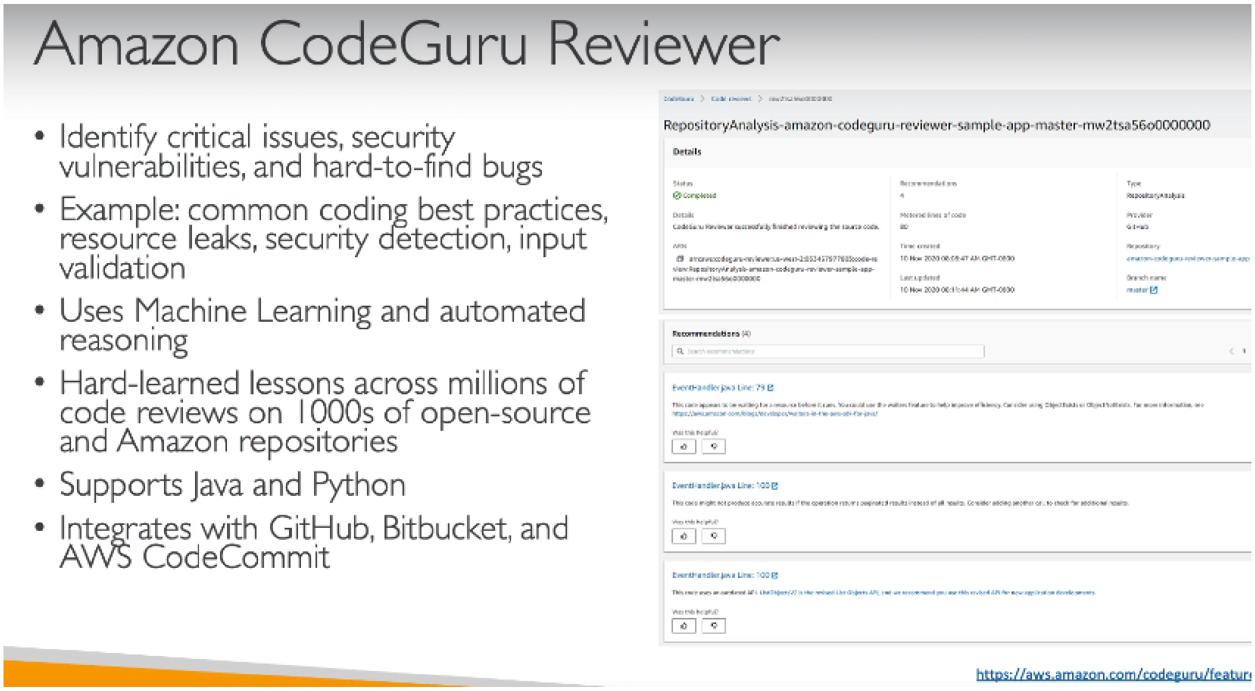
Amazon CodeGuru

- 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)

CodeGuru Reviewer
Built-in code reviews with actionable recommendations
CodeGuru Profiler
Detect and optimize the expensive lines of code pre-prod
Identify performance and cost improvements in production
Coding
Build & Test
Deploy
Measure

4. CodeGuru Reviewer really looks at your commits,
5. so whenever you push your code, and it tells you the lines of code that are probably wrong, so it could be very, very handy,so you can identify critical issues, security vulnerabilities, and hard-to-find bugs.

6.



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

RepositoryAnalysis-amazon-codeguru-reviewer-sample-app-master-mw2tsa56o0000000

Details

Status: Completed
Recommendations: 4
Type: RepositoryAnalysis
Details: CodeGuru Reviewer successfully finished reviewing the source code.
Metered lines of code: 80
Provider: GitHub
Time created: 10 Nov 2020 08:08:47 AM GMT-0800
Last updated: 10 Nov 2020 08:11:44 AM GMT-0800
Branch name: master

Recommendations (4)

EventHandler.java Line: 79

This code appears to be waiting for a resource before it runs. You could use the waiters feature to help improve efficiency. Consider using ObjectExists or ObjectNotExists. For more information, see https://aws.amazon.com/blogs/developer/waiters-in-the-aws-sdk-for-java/

Was this helpful?

EventHandler.java Line: 100

This code might not produce accurate results if the operation returns paginated results instead of all results. Consider adding another call to check for additional results.

Was this helpful?

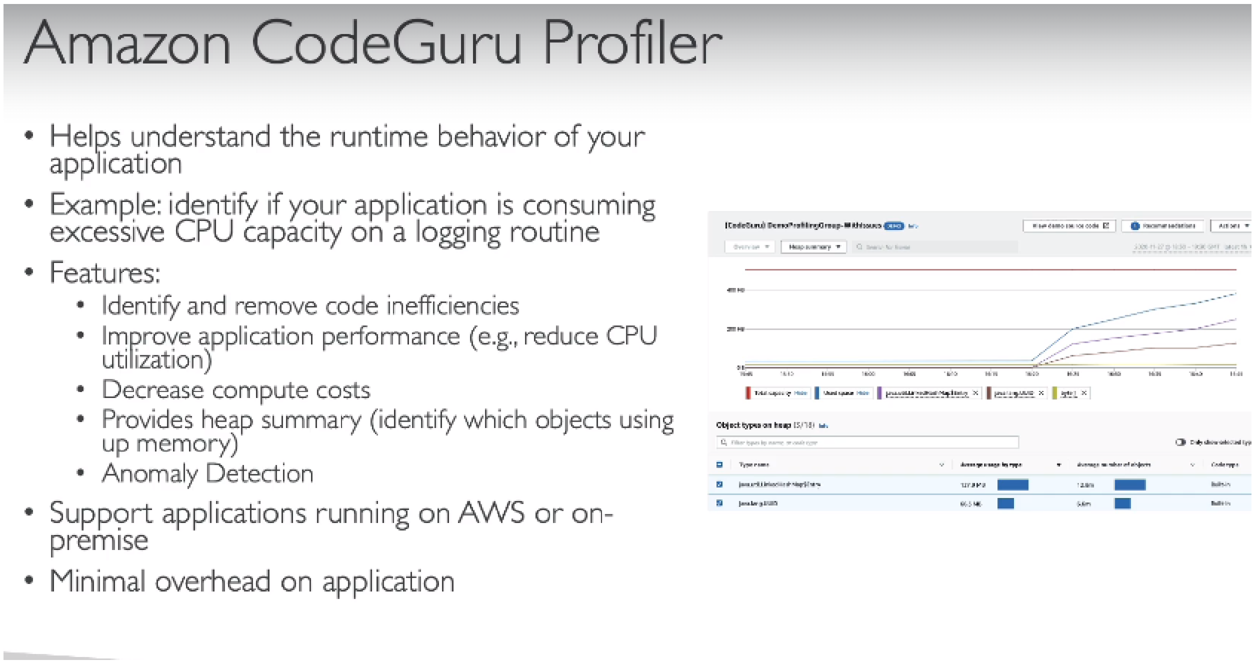
EventHandler.java Line: 100

This code uses an outdated API. ListObjectsV2 is the revised List Objects API, and we recommend you use this revised API for new application development.

Was this helpful?

https://aws.amazon.com/codeguru/featur

7. CodeGuru profiler



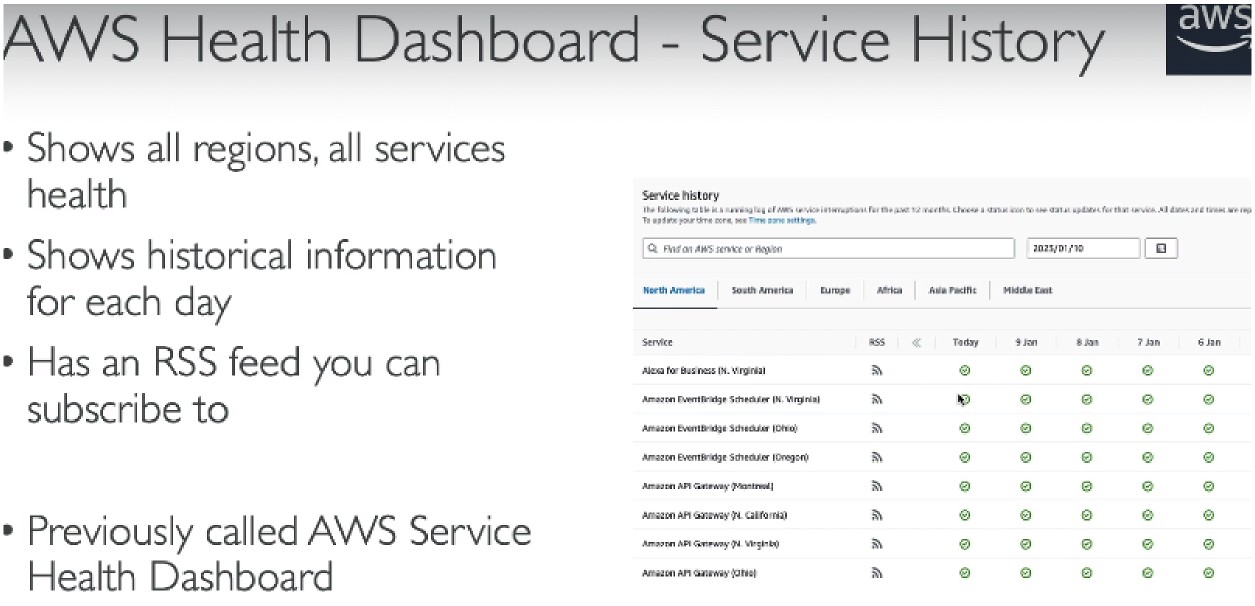
Amazon CodeGuru Profiler

- 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

8.

AWS Health Dashboard - Service History

1. It gives the health of all regions



AWS Health Dashboard - Service History

- Shows all regions, all services health
- Shows historical information for each day
- Has an RSS feed you can subscribe to
- Previously called AWS Service Health Dashboard

Service history

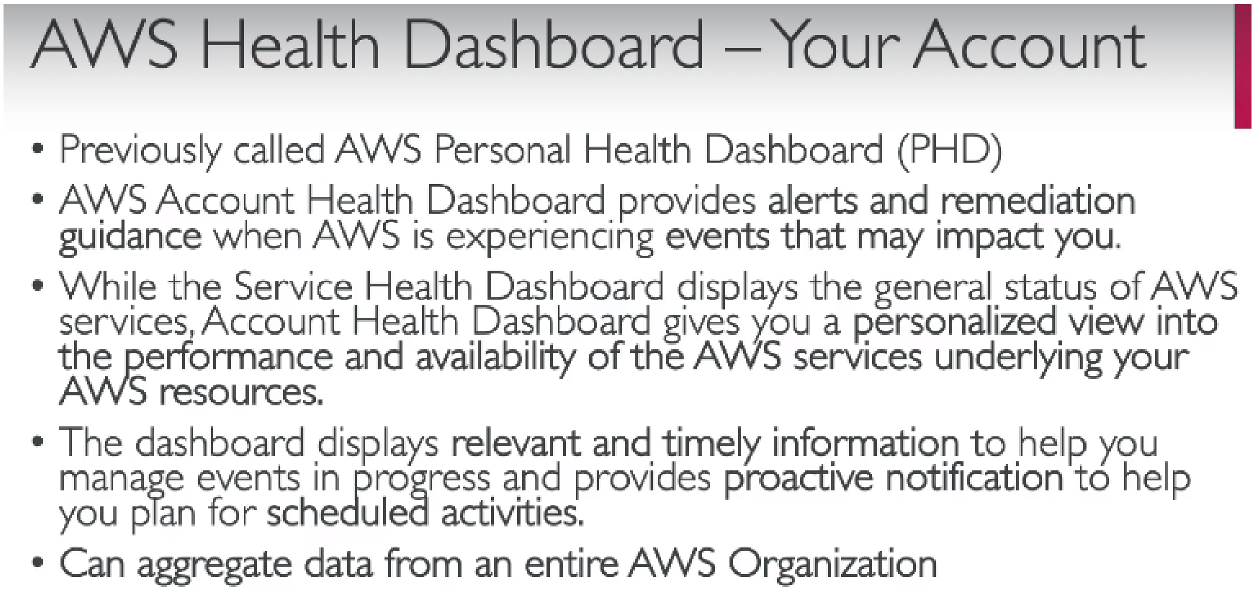
The following table is a running log of AWS service interruptions for the past 12 months. Choose a status icon to see status updates for that service. All dates and times are rep
To update your time zone, see Time zone settings.

Find an AWS service or Region | 2023/01/10

North America | South America | Europe | Africa | Asia Pacific | Middle East

Service	RSS	«	Today	9 Jan	8 Jan	7 Jan	6 Jan
Alexa for Business (N. Virginia)	☰		✓	✓	✓	✓	✓
Amazon EventBridge Scheduler (N. Virginia)	☰		✓	✓	✓	✓	✓
Amazon EventBridge Scheduler (Ohio)	☰		✓	✓	✓	✓	✓
Amazon EventBridge Scheduler (Oregon)	☰		✓	✓	✓	✓	✓
Amazon API Gateway (Montreal)	☰		✓	✓	✓	✓	✓
Amazon API Gateway (N. California)	☰		✓	✓	✓	✓	✓
Amazon API Gateway (N. Virginia)	☰		✓	✓	✓	✓	✓
Amazon API Gateway (Ohio)	☰		✓	✓	✓	✓	✓

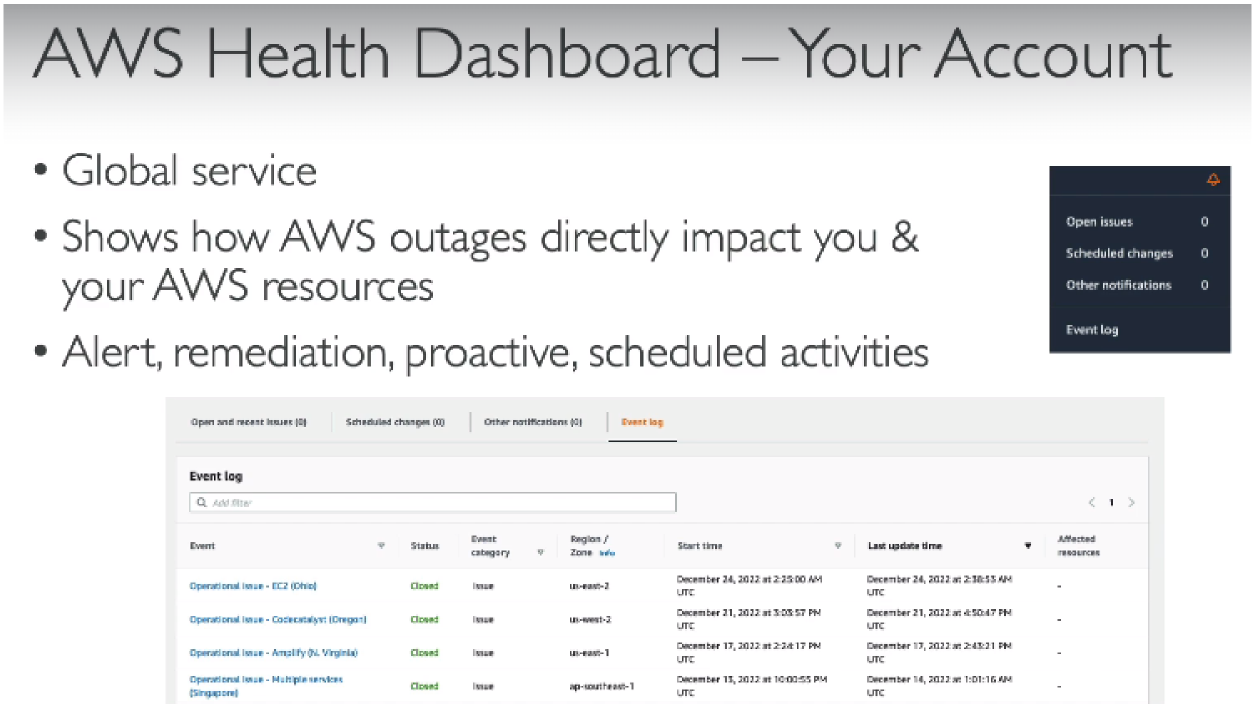
2.



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**

3.



AWS Health Dashboard – Your Account

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

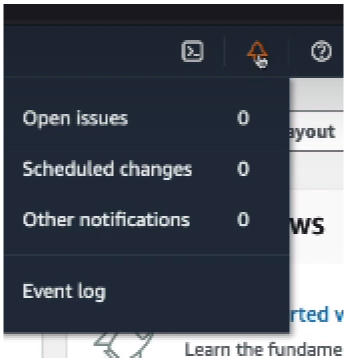
Open issues	0
Scheduled changes	0
Other notifications	0
Event log	

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 - Codecatalyst (Oregon)	Closed	Issue	us-west-2	December 21, 2022 at 3:03:57 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 2:34:17 PM UTC	December 17, 2022 at 2:43:21 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	-

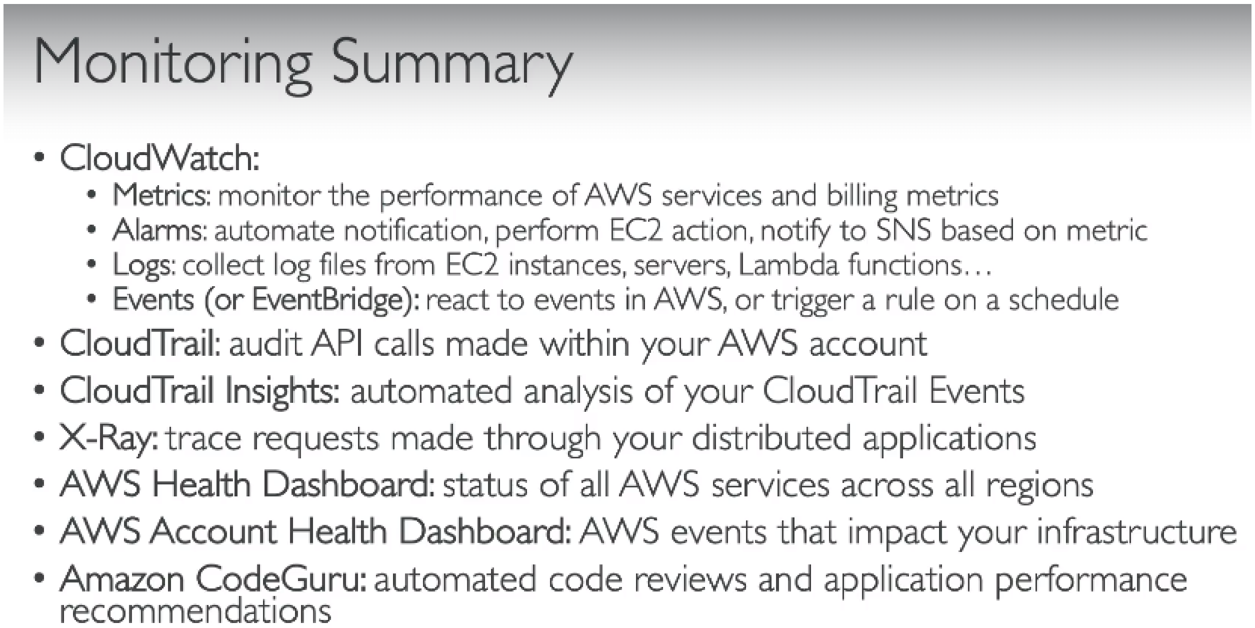
4.

AWS healthLog Hands On:

1. 

2. Click on event logs..
3. And on the left side we can check service health and our account health

Summary:



Monitoring 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

1.