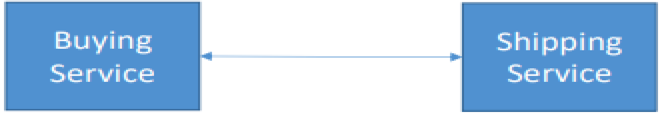
Cloud Integration Section

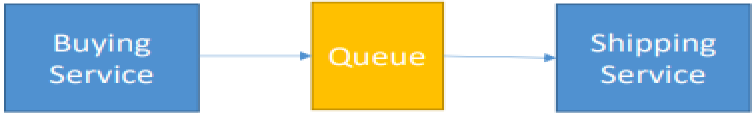
Section Introduction

- When we start deploying multiple applications, they will inevitably need to communicate with one another
- There are two patterns of application communication

1) Synchronous communications (application to application)



2) Asynchronous / Event based (application to queue to application)

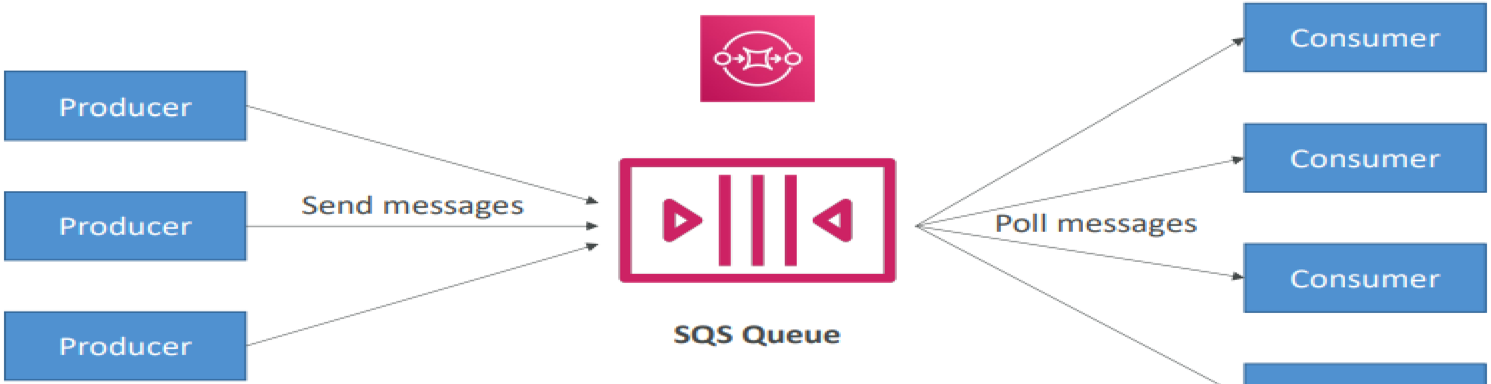


Section Introduction

- Synchronous between applications can be problematic if there are sudden spikes of traffic
- What if you need to suddenly encode 1000 videos but usually it's 10?

- In that case, it's better to **decouple** your applications:
  - using SQS: queue model
  - using SNS: pub/sub model
  - using Kinesis: real-time data streaming model
- These services can scale independently from our application!

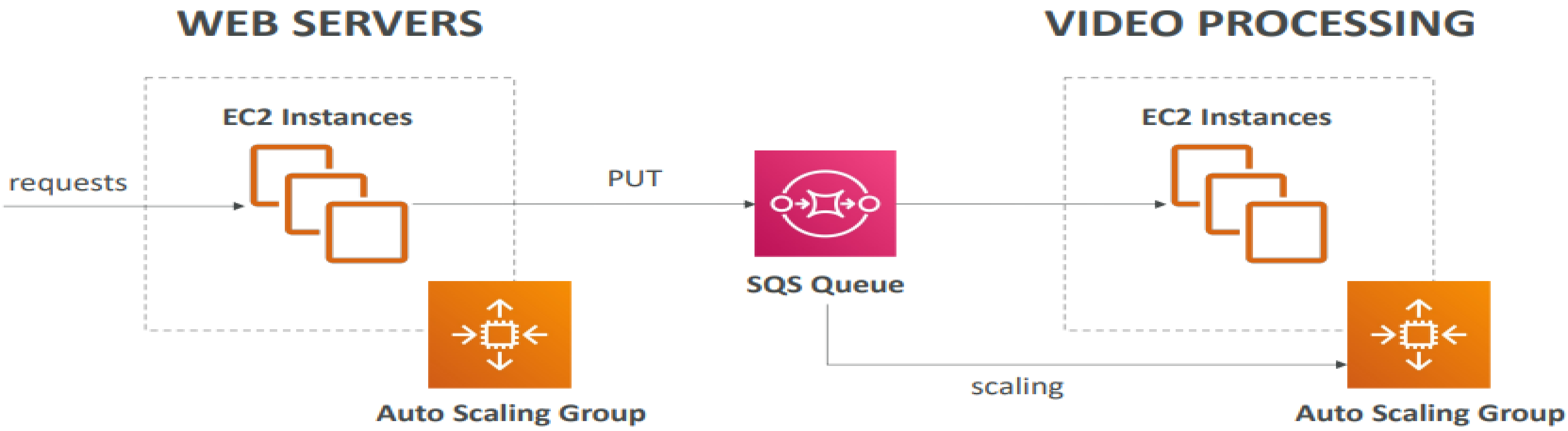
Amazon SQS – Simple Queue Service
What's a queue?



Amazon SQS – Standard Queue

- Oldest AWS offering (over 10 years old)
- Fully managed service (~serverless), use to **decouple** applications
- Scales from 1 message per second to 10,000s per second
- Default retention of messages: 4 days, maximum of 14 days
- No limit to how many messages can be in the queue
- **Messages are deleted after they're read by consumers**
- Low latency (<10 ms on publish and receive)
- **Consumers share the work to read messages & scale horizontally**

SQS to decouple between application tiers

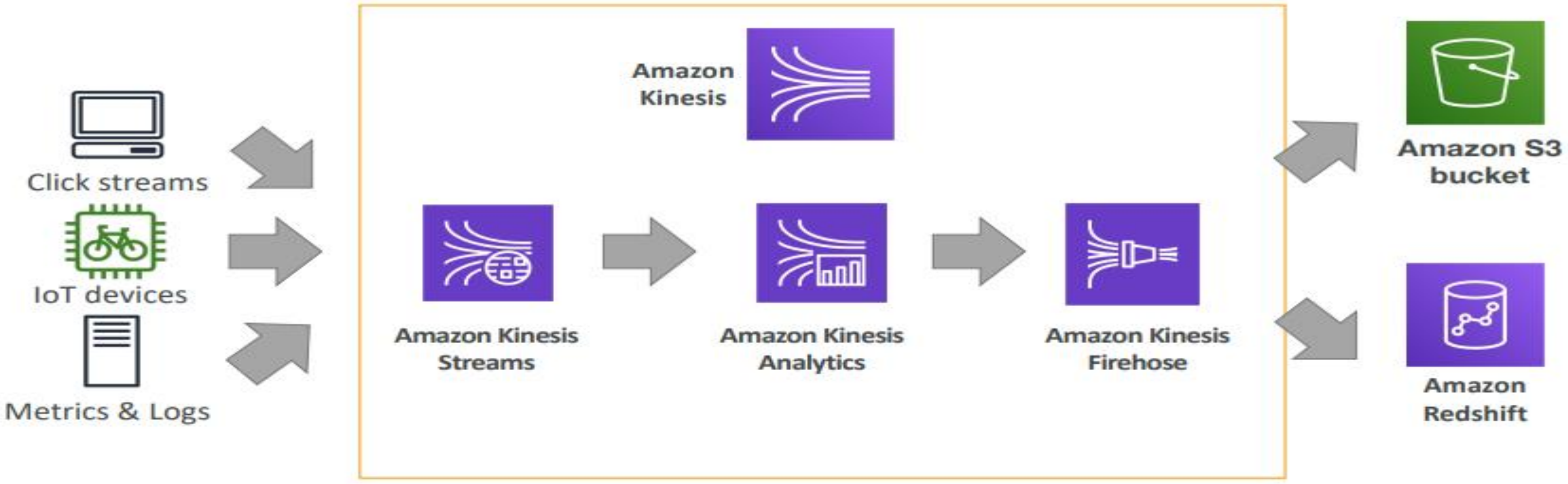


Amazon Kinesis

- For the exam: **Kinesis = real-time big data streaming**
- **Managed service to collect, process, and analyze real-time streaming data at any scale**

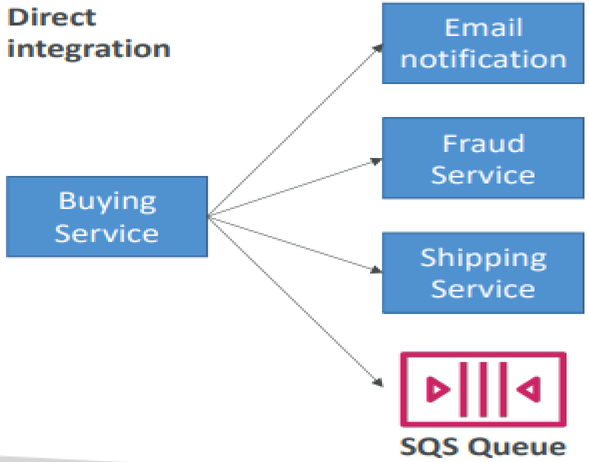
- Too detailed for the Cloud Practitioner exam but good to know:
  - **Kinesis Data Streams:** low latency streaming to ingest data at scale from hundreds of thousands of sources
  - **Kinesis Data Firehose:** load streams into S3, Redshift, ElasticSearch, etc…
  - **Kinesis Data Analytics:** perform real-time analytics on streams using SQL
  - **Kinesis Video Streams:** monitor real-time video streams for analytics or ML

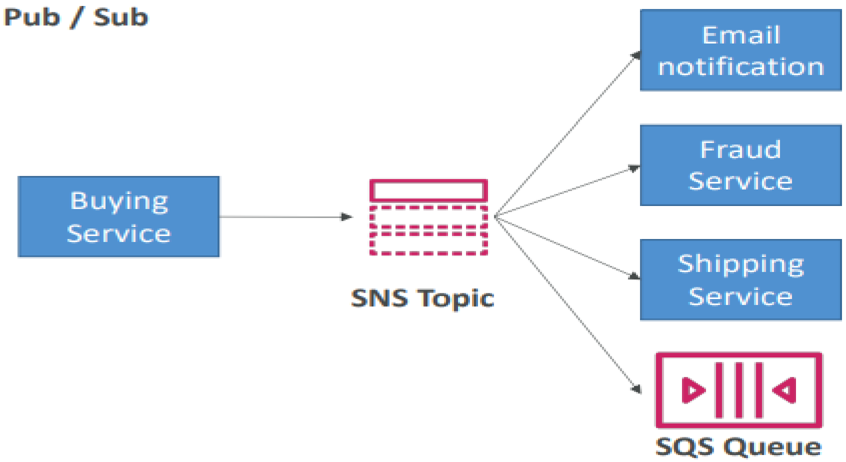
Kinesis (high level overview)



Amazon SNS

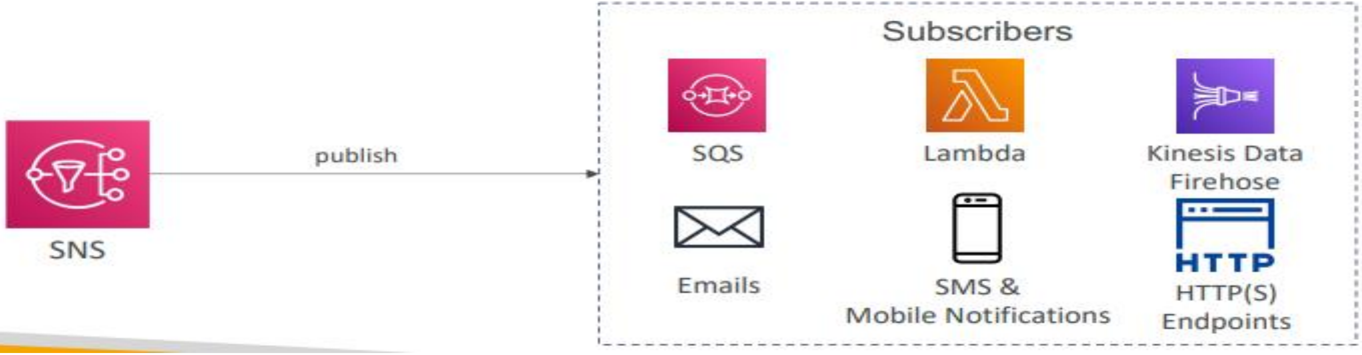
- What if you want to send one message to many receivers?





Amazon SNS

- The "event **publishers**" only sends message to one SNS topic
- As many "event **subscribers**" as we want to listen to the SNS topic notifications
- Each subscriber to the topic **will get all the messages**
- Up to 12,500,000 subscriptions per topic, 100,000 topics limit



Amazon MQ

- SQS, SNS are "cloud-native" services: proprietary protocols from AWS
- Traditional applications running from on-premises may use open protocols such as: MQTT, AMQP, STOMP, Openwire, WSS
- **When migrating to the cloud**, instead of re-engineering the application to use SQS and SNS, we can use Amazon MQ
- **Amazon MQ is a managed message broker service for**

RabbitMQ

ACTIVEMQ

- Amazon MQ doesn't "scale" as much as SQS / SNS
- Amazon MQ runs on servers, can run in Multi-AZ with failover
- Amazon MQ has both queue feature (~SQS) and topic features (~SNS)

Integration Section – Summary

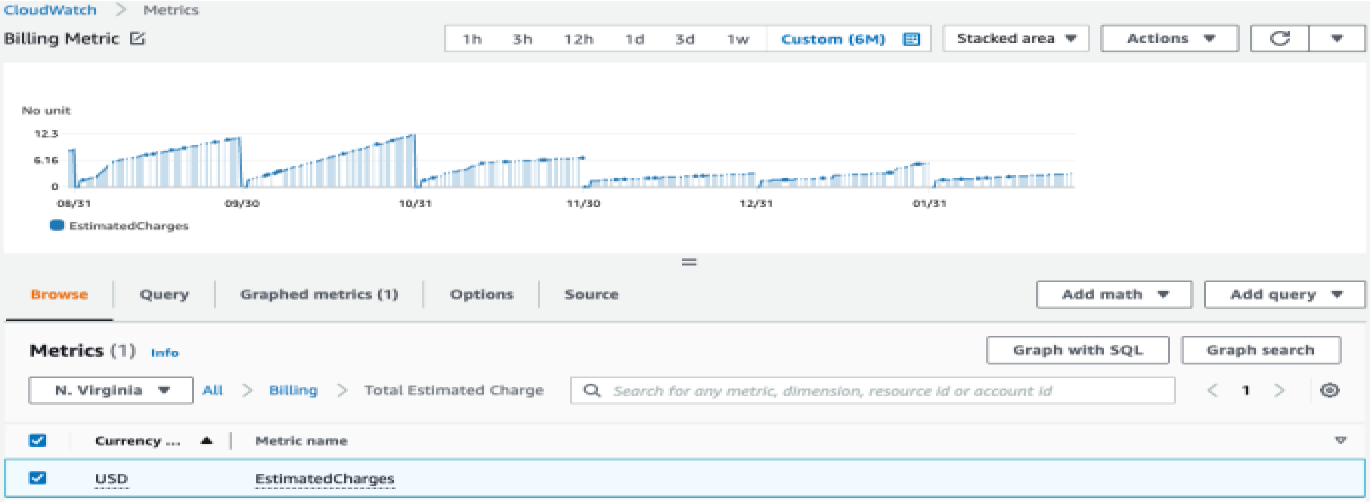
- **SQS:**
  - Queue service in AWS
  - Multiple Producers, messages are kept up to 14 days
  - Multiple Consumers share the read and delete messages when done
  - Used to **decouple** applications in AWS
- **SNS:**
  - Notification service in AWS
  - Subscribers: Email, Lambda, SQS, HTTP, Mobile…
  - Multiple Subscribers, send all messages to all of them
  - No message retention
- **Kinesis**: real-time data streaming, persistence and analysis
- **Amazon MQ**: managed message broker for ActiveMQ and RabbitMQ in the cloud (MQTT, AMQP.. protocols)

Cloud Monitoring Section

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

Example: CloudWatch Billing metric



Metrics (1) Info

N. Virginia ▼ All > Billing > Total Estimated Charge

☑	Currency ... ▲	Metric name
☑	USD	EstimatedCharges

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

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

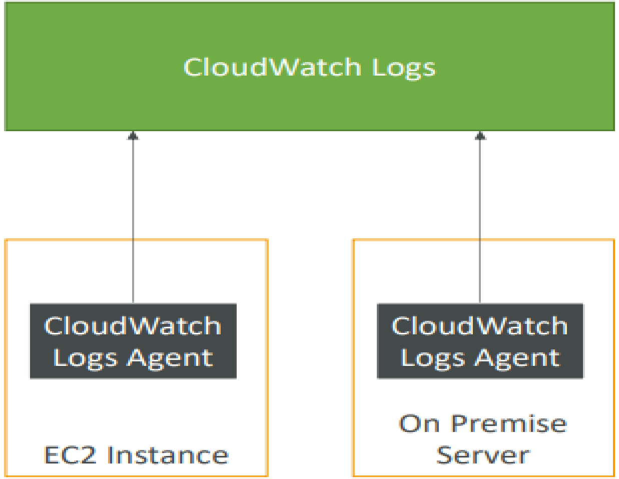
Amazon CloudWatch Logs

- 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

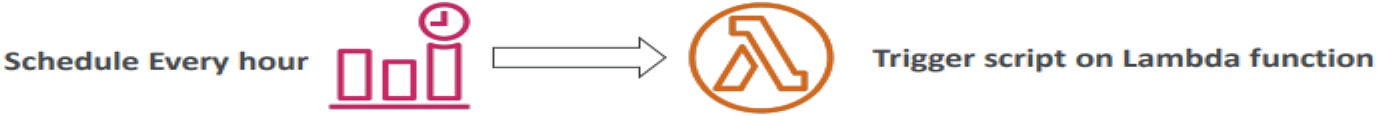
CloudWatch Logs for EC2

- 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



Amazon EventBridge (formerly CloudWatch Events)

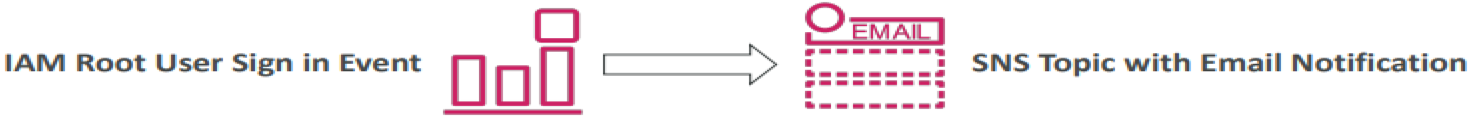
- Schedule: Cron jobs (scheduled scripts)



Schedule Every hour

Trigger script on Lambda function

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

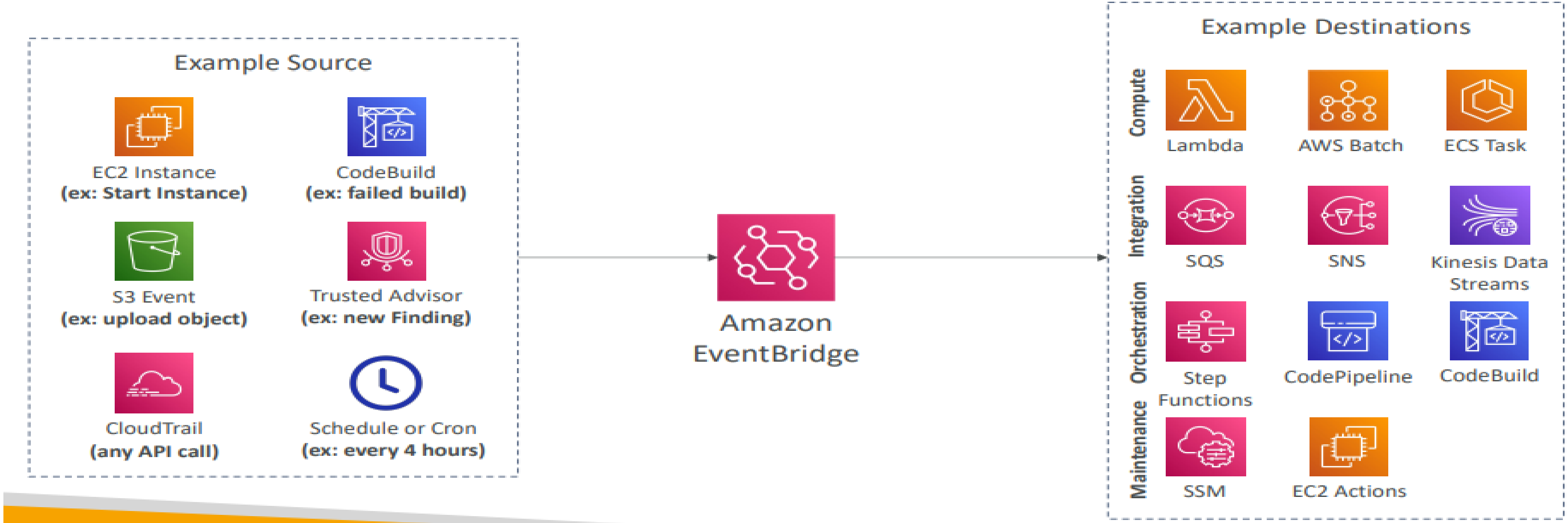


IAM Root User Sign in Event

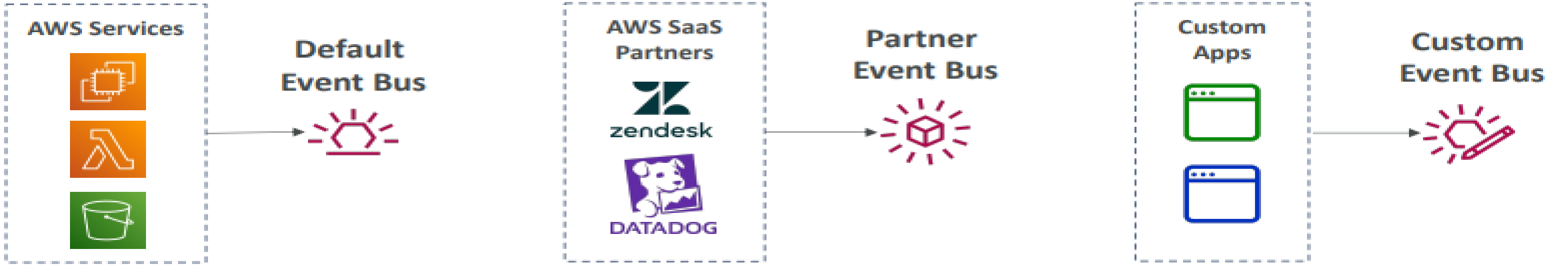
SNS Topic with Email Notification

- Trigger Lambda functions, send SQS/SNS messages…

Amazon EventBridge Rules



Amazon EventBridge

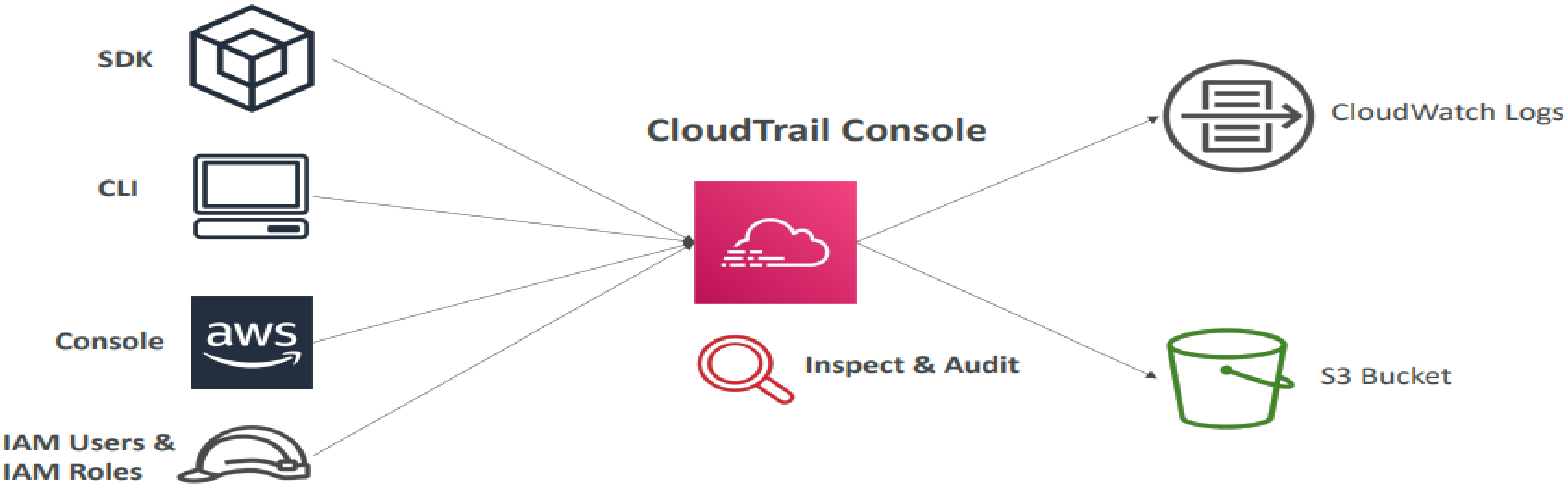


- **Schema Registry**: model event schema
- You can **archive events** (all/filter) sent to an event bus (indefinitely or set period)
- Ability to **replay archived events**

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!

CloudTrail Diagram

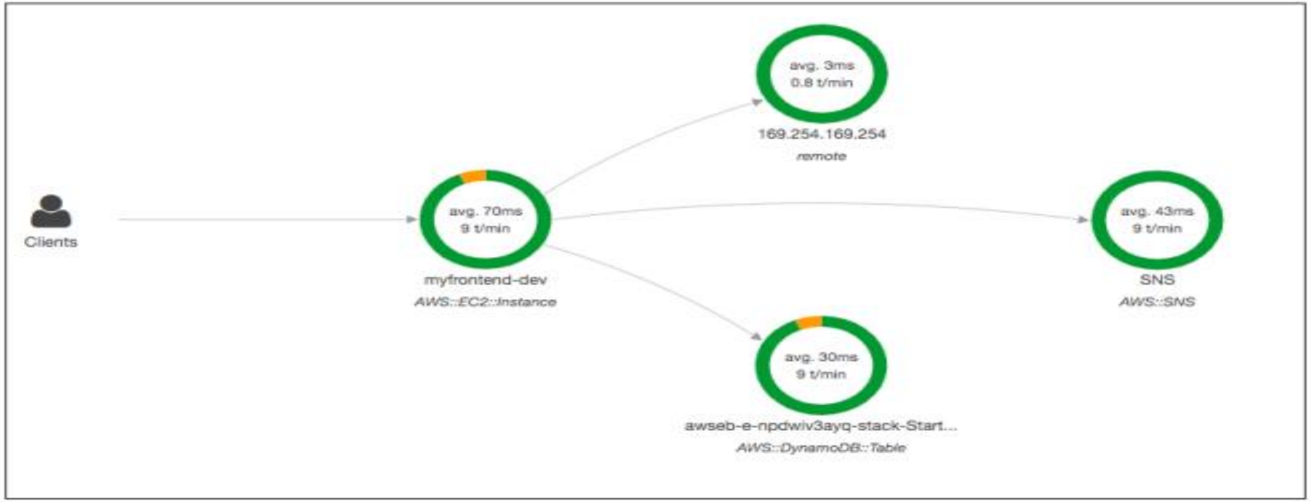


AWS X-Ray

- 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!

AWS X-Ray
Visual analysis of our applications

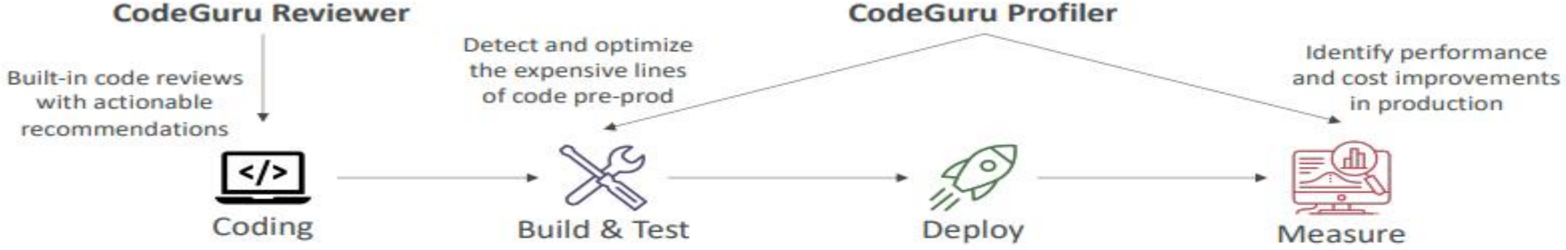


AWS X-Ray advantages

- 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

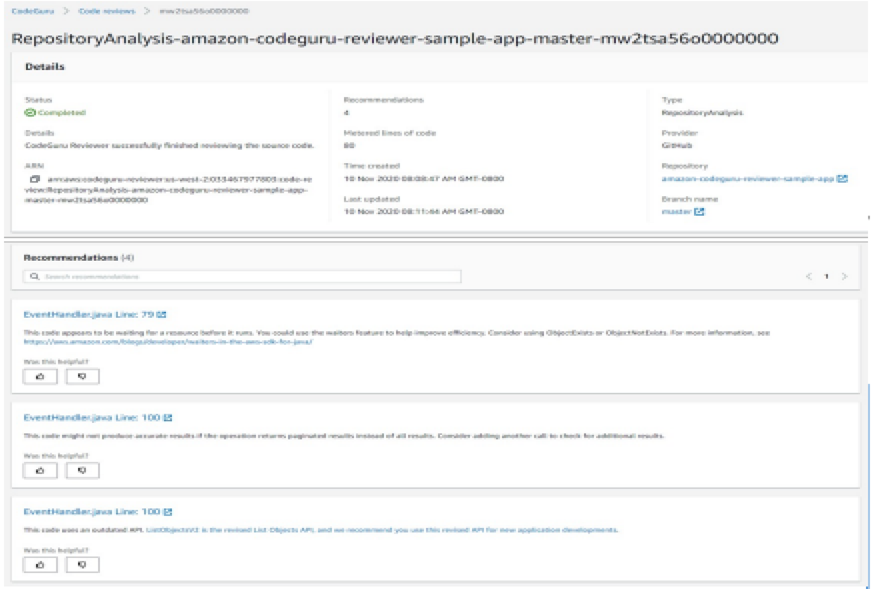
- An ML-powered service for **automated code reviews** and **application performance recommendations**
- Provides two functionalities
  - **CodeGuru Reviewer:** automated code reviews for static code analysis (development)
  - **CodeGuru Profiler:** visibility/recommendations about application performance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



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

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

Details
CodeGuru Reviewer successfully finished reviewing the source code.

ARN
arn:aws:codeguru-reviewer:us-west-2:033467577805:code-review:RepositoryAnalysis-amazon-codeguru-reviewer-sample-app-master-mw2tsa56o0000000

Recommendations
4

Metered lines of code
88

Time created
10 Nov 2020 08:08:47 AM GMT-0800

Last updated
10 Nov 2020 08:11:44 AM GMT-0800

Type
RepositoryAnalysis

Provider
GitHub

Repository
amazon-codeguru-reviewer-sample-app

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

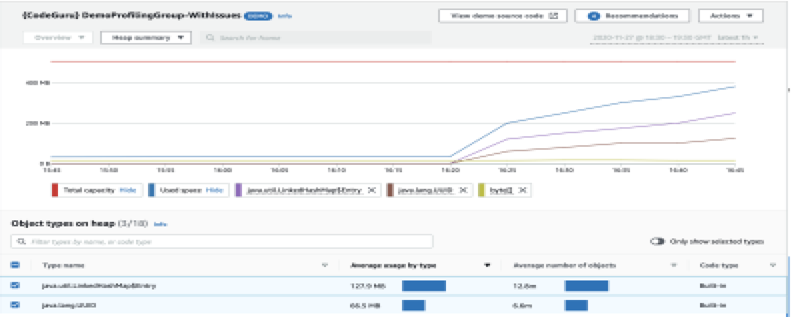
This code uses an outdated API. ListObjectsV2 is the revised List Objects API, and we recommend you use this revised API for new application developments.

Was this helpful?

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

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



Object types on heap (3/10)

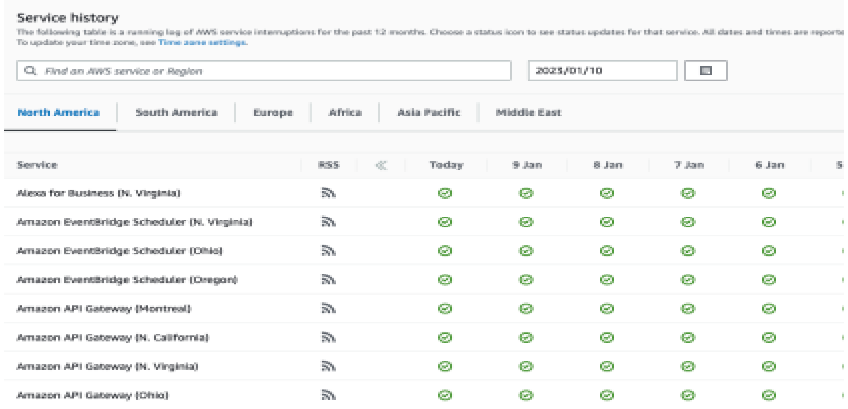
	Type name	Average usage by type	Average number of objects	Code type
☑	java.util.LinkedHashMap$Entry	127.9 MB	12.8m	Built-in
☑	java.lang.UUID	66.5 MB	6.6m	Built-in

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

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 reporte
To update your time zone, see Time zone settings.

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

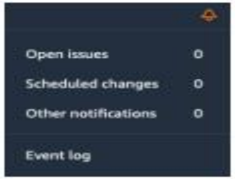
Service	RSS	«	Today	9 Jan	8 Jan	7 Jan	6 Jan	5
Alexa for Business (N. Virginia)	📶		✓	✓	✓	✓	✓	
Amazon EventBridge Scheduler (N. Virginia)	📶		✓	✓	✓	✓	✓	
Amazon EventBridge Scheduler (Ohio)	📶		✓	✓	✓	✓	✓	
Amazon EventBridge Scheduler (Oregon)	📶		✓	✓	✓	✓	✓	
Amazon API Gateway (Montreal)	📶		✓	✓	✓	✓	✓	
Amazon API Gateway (N. California)	📶		✓	✓	✓	✓	✓	
Amazon API Gateway (N. Virginia)	📶		✓	✓	✓	✓	✓	
Amazon API Gateway (Ohio)	📶		✓	✓	✓	✓	✓	

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

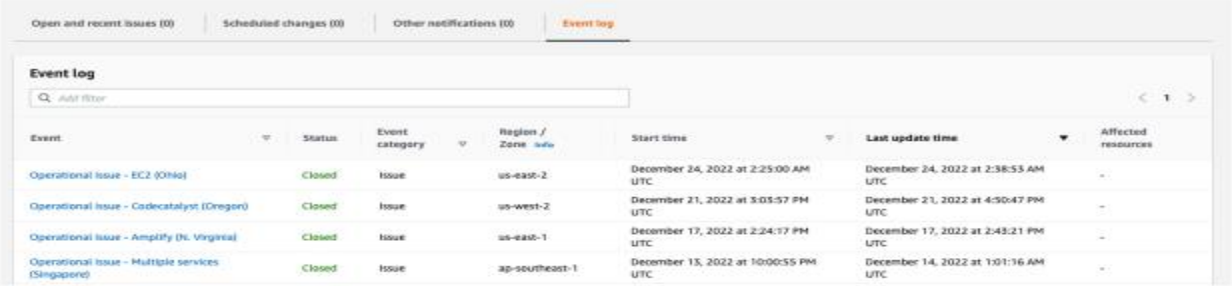
AWS Health Dashboard – Your Account

- 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 3:05:57 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 2:43:21 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	-

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