



CSCI 5902 Adv. Cloud Architecting
Fall 2023
Instructor: Lu Yang

Module 9 Implementing Elasticity, Hight Availability, & Monitoring (Sec 5) &
Module 10 Decoupled Architecture (Sec 1-5)
Nov 20, 2023

Housekeeping items and feedback

1. Start recording
2. The final is:
9:30-11:30am, Dec 11
CHEB room 170
3. Release more practice tests released on Brightspace
4. Do not always keep your resources running for the term project. Use IaC to keep your infrastructure.
5. CPC and SAA voucher request spreadsheet is up. The deadline to sign up is Dec 4.
6. PIER tour is still on



Recap of our last lecture

Module overview



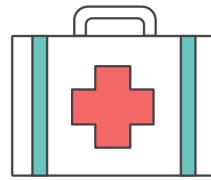
Sections

1. Architectural need
2. Scaling your compute resources
3. Scaling your databases
4. Designing an environment that's highly available
5. Monitoring

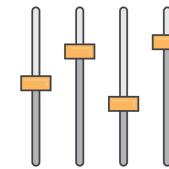
Module 9: Implementing Elasticity, High Availability, and Monitoring

Section 5: Monitoring

Monitoring usage, operations, and performance



Operational Health



Resource Utilization



Application Performance

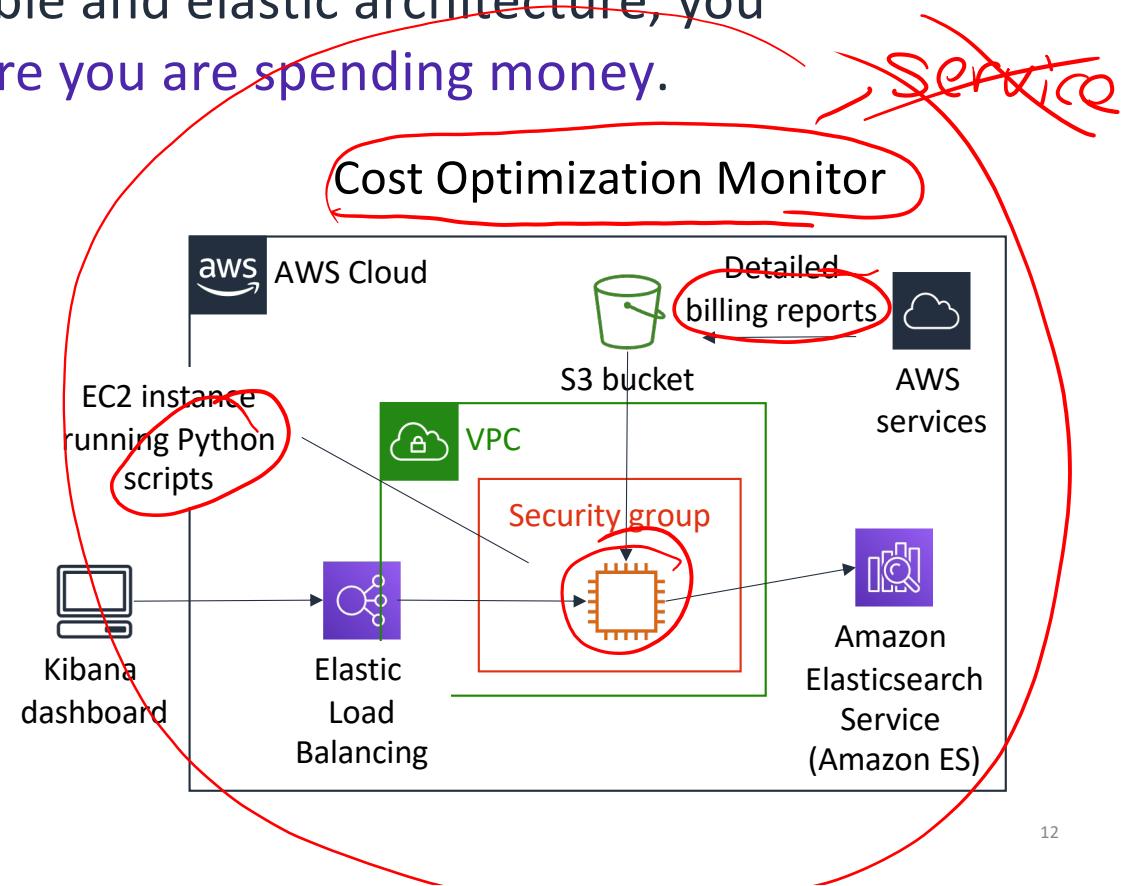
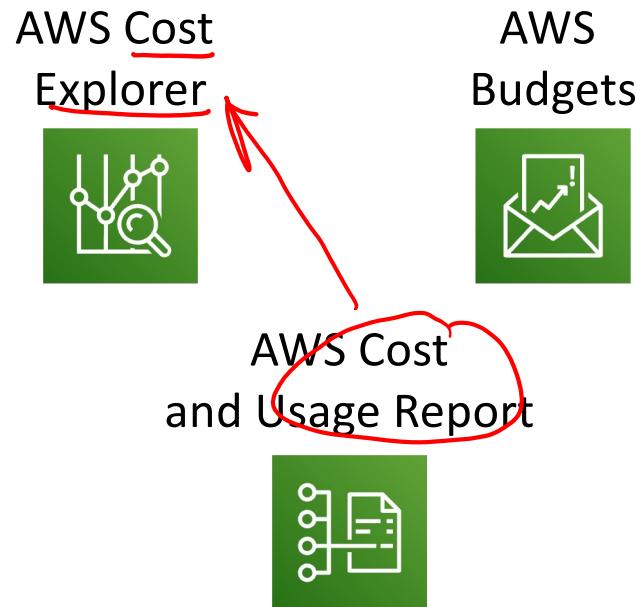


Security Auditing

Monitoring your costs



To create a more flexible and elastic architecture, you should know where you are spending money.



Amazon CloudWatch



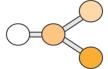
Amazon
CloudWatch

- Collects and tracks metrics for your resources and applications
- Helps you correlate, visualize, and analyze metrics and logs
- Enables you to create alarms and detect anomalous behavior
- • Can send notifications or make changes to resources that you are monitoring

How CloudWatch responds



Metrics



Logs



Alarms



Events



Rules

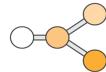


Targets

CloudWatch metrics



Metrics



Logs



Alarms



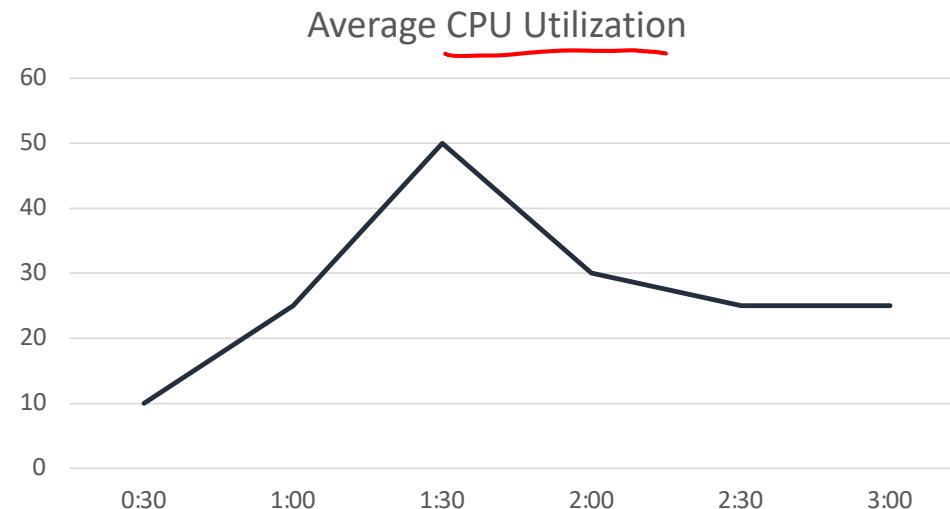
Events



Rules



Targets

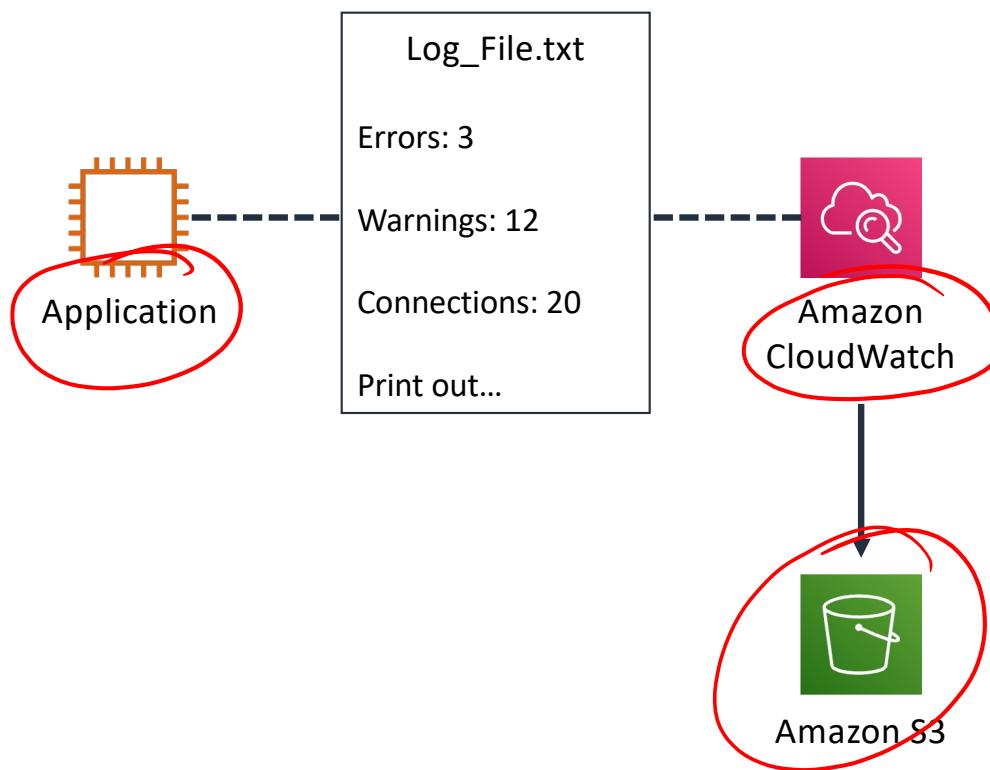


Metric data is kept for 15 months

Amazon CloudWatch Logs



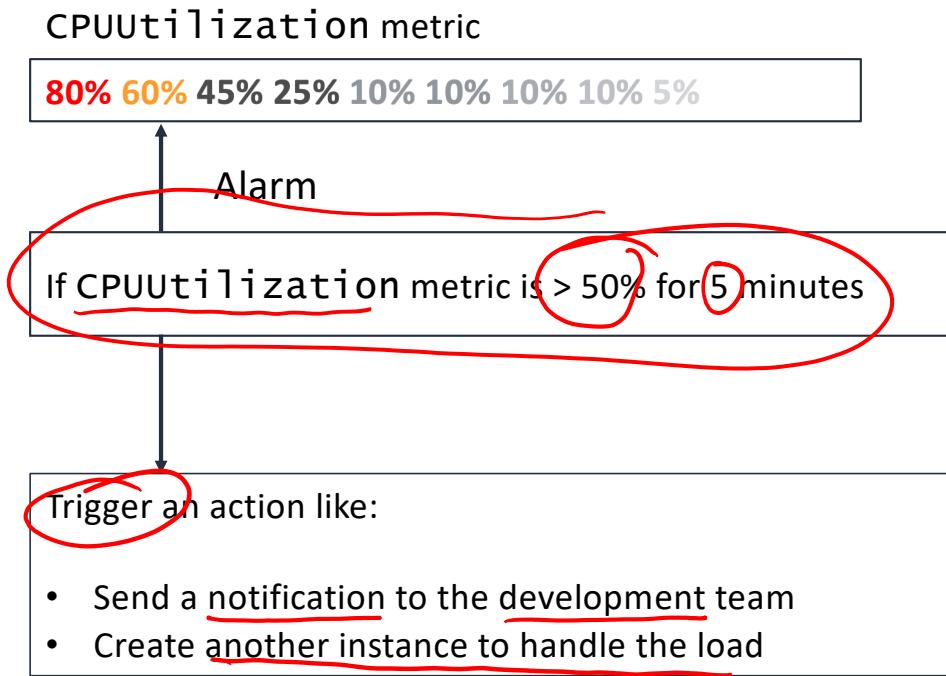
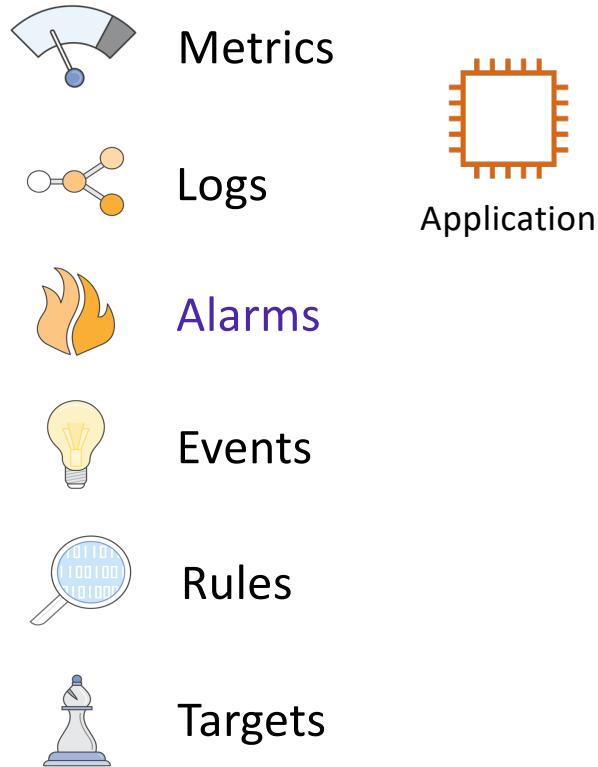
-  Metrics
-  Logs
-  Alarms
-  Events
-  Rules
-  Targets



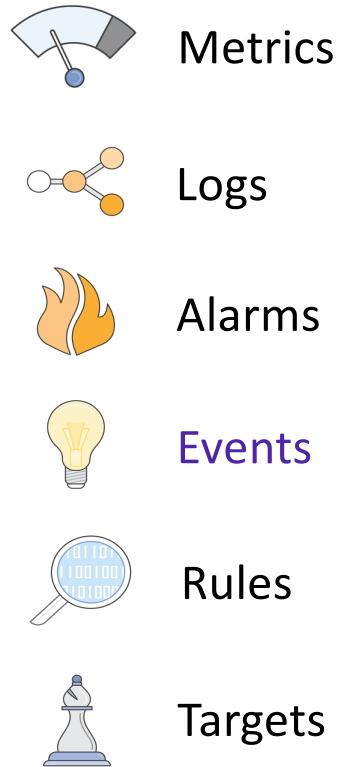
Source examples

- VPC Flow Logs
- Amazon Route 53
- Elastic Load Balancing access logs

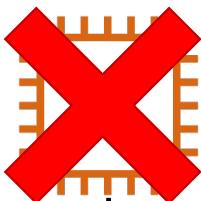
CloudWatch alarms



Amazon EventBridge events



Event: EC2
instance
termination



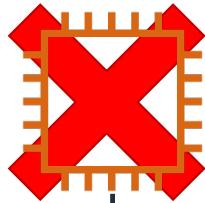
Event examples

- Change in AWS resource, such as –
 - Console sign-in
 - EC2 instance state change
 - EC2 Auto Scaling state change
 - EBS volume creation
- AWS API call
- Events from SaaS partners
- Events from your own applications

Amazon EventBridge rules



Event



Rule example

```
{  
  "source": [  
    "aws_ec2"  
  ],  
  "detail-type": [  
    "EC2 Instance State-change Notification"  
  ],  
  "detail": {  
    "state": [  
      "terminated"  
    ]  
  }  
}
```

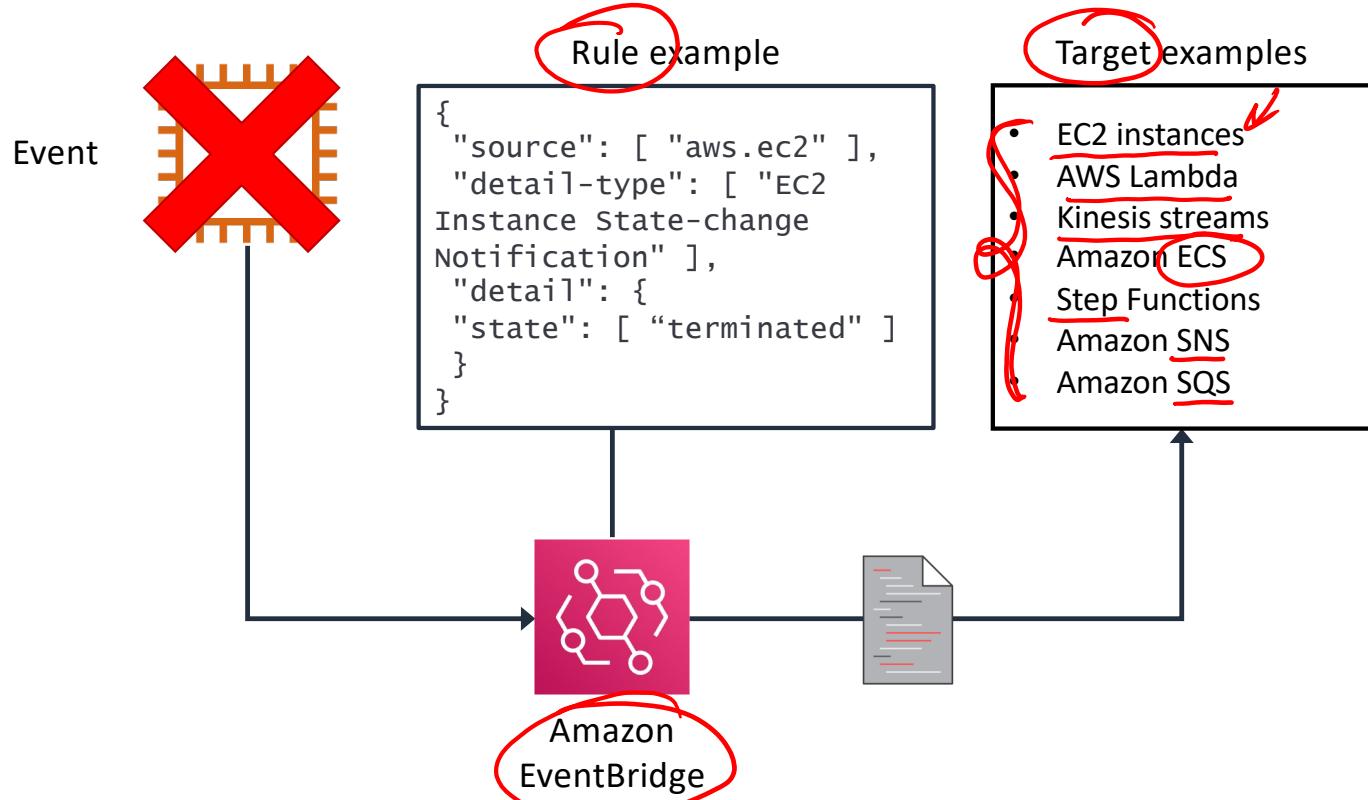


Amazon
EventBridge

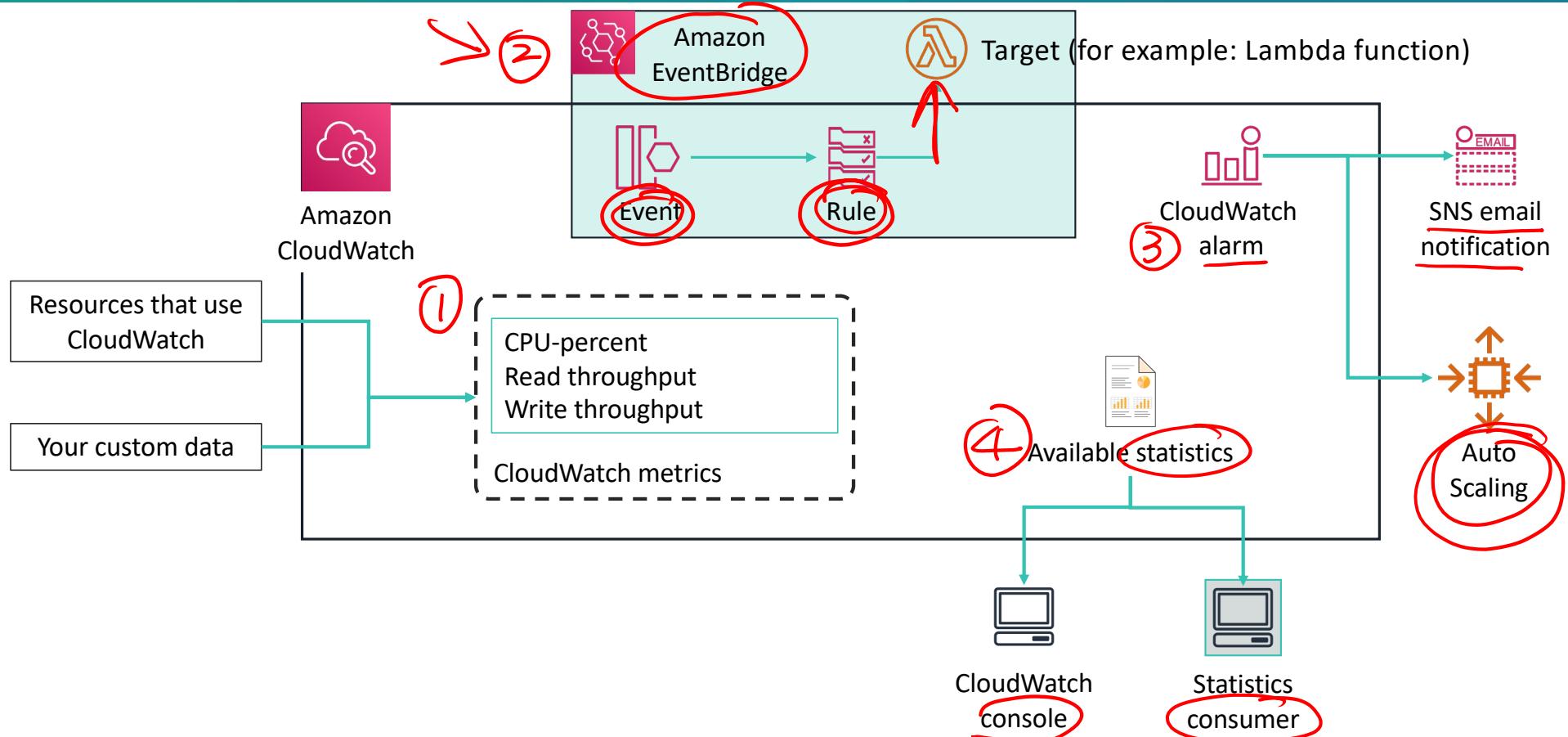
Amazon EventBridge targets



- Metrics
- Logs
- Alarms
- Events
- Rules
- Targets



How CloudWatch and EventBridge work



AWS Cloudwatch Console Walkthrough: ([https://www.youtube.com/ watch?v=k7wulrHU4UY](https://www.youtube.com/watch?v=k7wulrHU4UY))



Section 5 key takeaways



23



- AWS Cost Explorer, AWS Budgets, AWS Cost and Usage Report, and the Cost Optimization Monitor can help you understand and manage the cost of your AWS infrastructure.
- CloudWatch collects monitoring and operational data in the form of logs, metrics, and events. It visualizes the data by using automated dashboards so you can get a unified view of your AWS resources, applications, and services that run in AWS and on-premises.
- EventBridge is a serverless event bus service that connects your applications with data from various sources. EventBridge delivers a stream of real-time data from your own applications, SaaS applications, and AWS services. It then routes that data to targets.

© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Module 9: Implementing Elasticity, High Availability, and Monitoring

Module wrap-up

Module summary



In summary, in this module, you learned how to:

- Use Amazon EC2 Auto Scaling within an architecture to promote elasticity
- Explain how to scale your database resources
- Deploy an Application Load Balancer to create a highly available environment
- Use Amazon Route 53 for DNS failover
- Create a highly available environment
- Design architectures that use Amazon CloudWatch to monitor resources and react accordingly

AWS Academy Cloud Architecting

Module 10: Building Decoupled Architectures

Module overview



Sections

1. Architectural need
2. Decoupling your architecture
3. Decoupling with Amazon Simple Queue Service (Amazon SQS)
4. Decoupling with Amazon Simple Notification Service (Amazon SNS)
5. Sending messages between cloud applications and on-premises with Amazon MQ

Module objectives



At the end of this module, you should be able to:

- Differentiate between tightly and loosely coupled architectures
- Identify how Amazon SQS works and when to use it
- Identify how Amazon SNS works and when to use it
- Describe Amazon MQ

Module 12: Building Decoupled Architectures

Section 1: Architectural need

Café business requirement



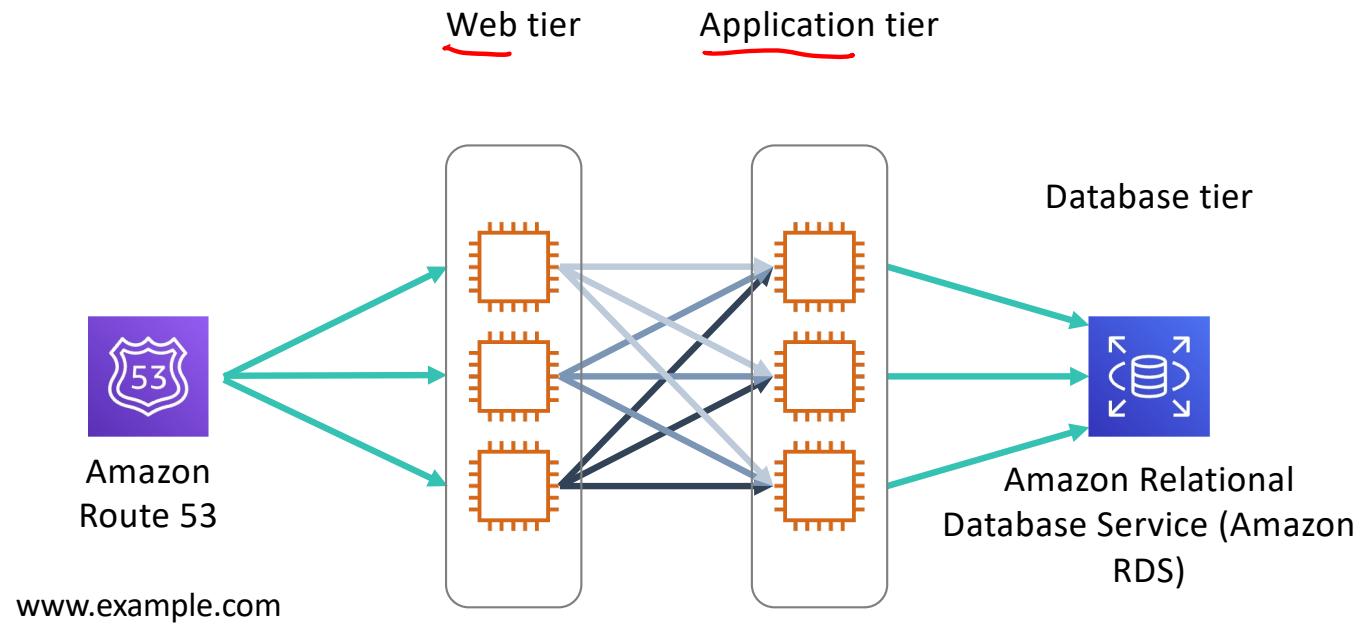
The café's architecture now supports hundreds of thousands of users. However, it's difficult to make changes to one layer of the application without affecting the other layers.



Module 12: Building Decoupled Architectures

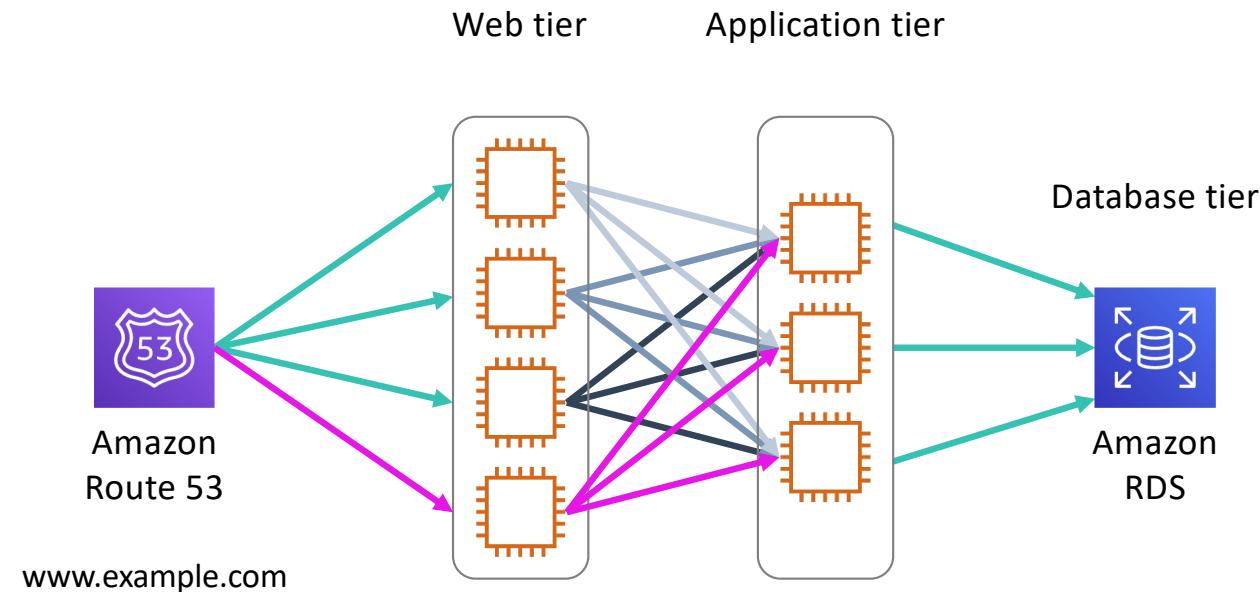
Section 2: Decoupling your architecture

Tightly coupled architectures



Components are **strongly connected** to each other.

Tightly coupled architectures impede scaling



Adding resources increases complexity and impedes scaling.

Forms of system coupling



Application-level coupling:

Relates to managing incoming and outgoing dependencies

Platform coupling:

Relates to interoperability of heterogeneous systems components

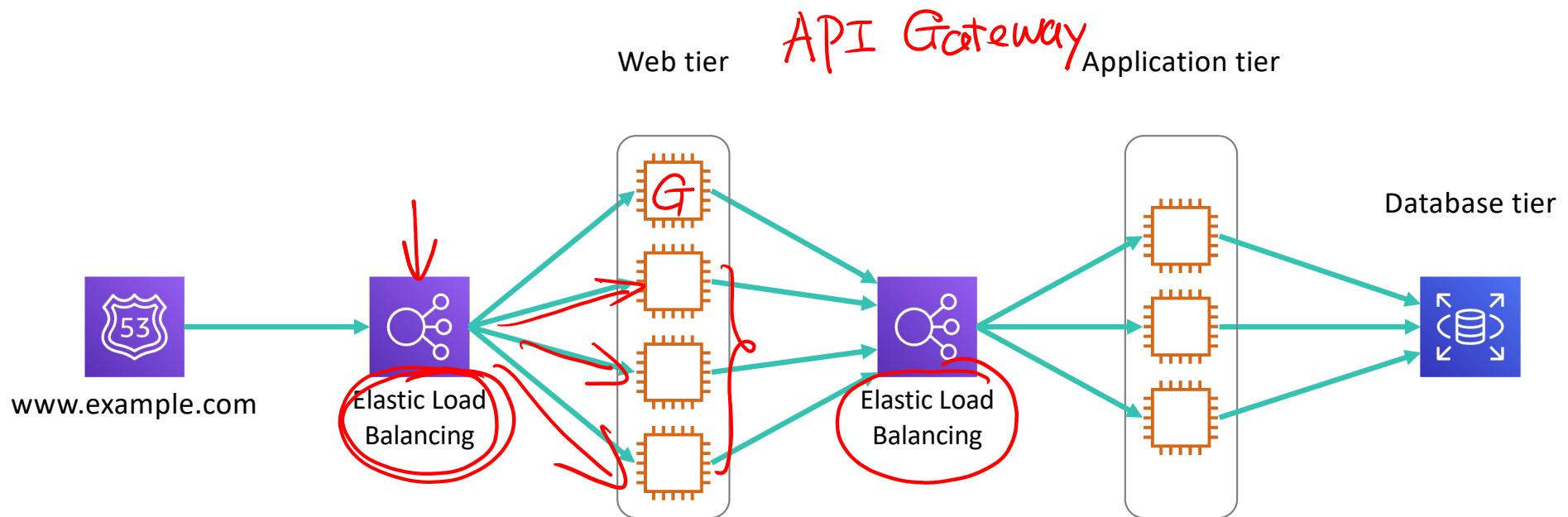
Spatial coupling:

Relates to managing components at network-topology level or protocol level

Temporal (runtime) coupling:

Refers to the ability of a system component to do meaningful work while it performs a synchronous, blocking operation

Loosely coupled architectures



Use **managed solutions** as **intermediaries** between layers.

Section 2 key takeaways



36



- Tightly coupled systems have chains of tightly integrated components and impede scaling
- You can implement loose coupling in your system by using managed solutions (such as Elastic Load Balancing) as intermediaries between layers

Module 12: Building Decoupled Architectures

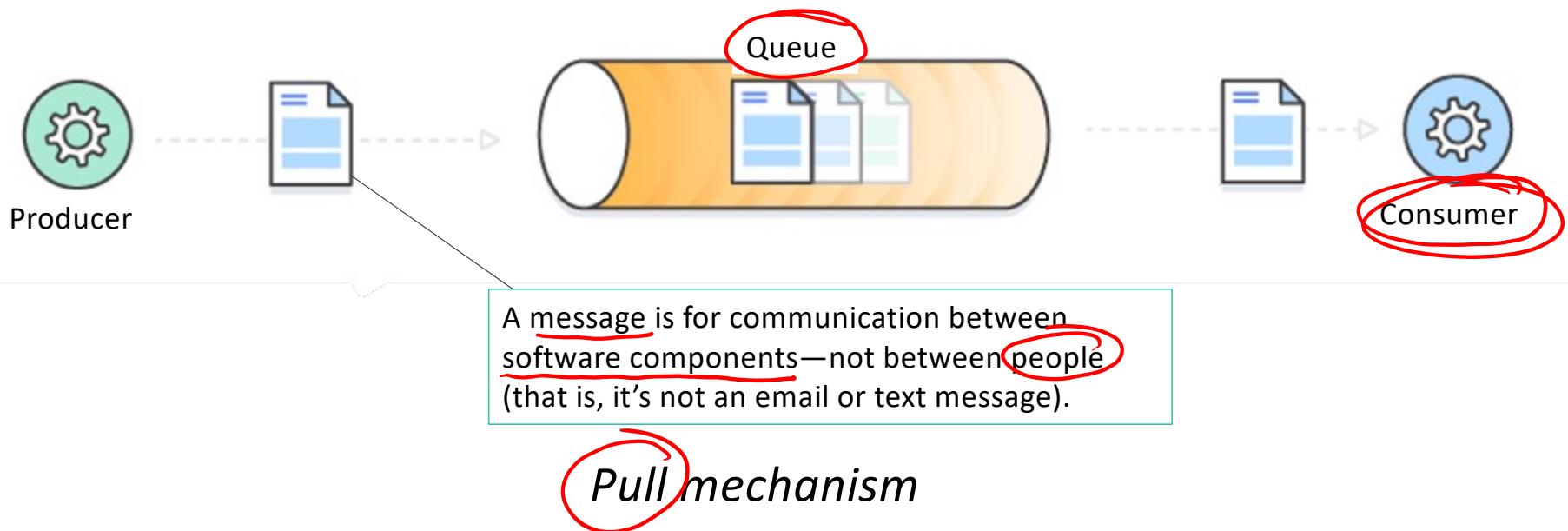
Section 3: Decoupling with Amazon SQS

Message queues for decoupling architectures



Producer – Application component
that produces messages and
adds them to queue

Consumer – Application component
that polls queue for messages
and processes them





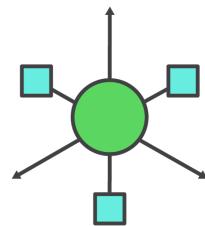
Amazon Simple
Queue Service
(Amazon SQS)

- Fully managed message queueing service
- Uses a pull mechanism
- Messages are encrypted and stored until they are processed and deleted
 - In transit using SSL/TLS
 - At rest using Amazon SQS key (SSE-SQS) or KMS (SSE-KMS)
- Acts as a buffer between producers and consumers
- Could be unlimited throughput
- Message retention: default 4 days, up to 14 days
- Maximum message size: 256KB

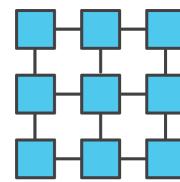
Achieve loose coupling with Amazon SQS



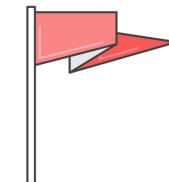
With Amazon SQS, you can:



Use asynchronous processing to get your responses from each step quickly

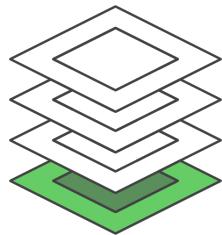


Handle performance and service requirements by increasing the number of job instances

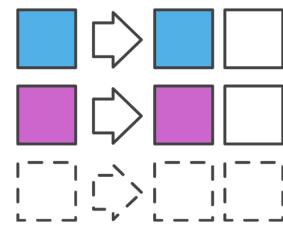


Easily recover from failed steps because messages will remain in the queue

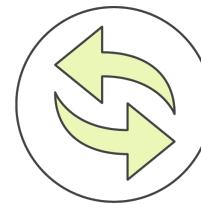
Amazon SQS general use cases



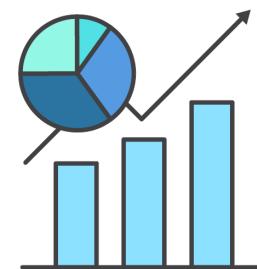
Work queues



Buffering batch
operations



Request
offloading

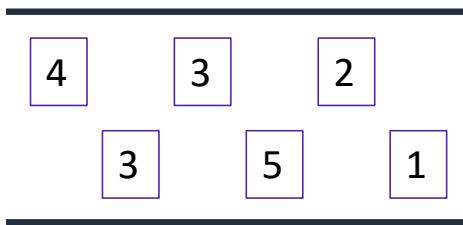


Trigger Amazon
EC2 Auto Scaling

Queue types



Standard queues



First in, first out (FIFO) queues



- At-least-once delivery
- Best-effort ordering
- Nearly unlimited throughput

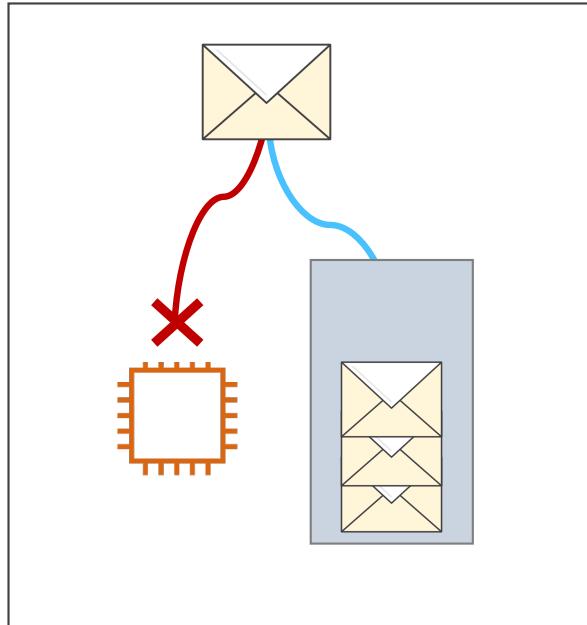
- First-in-first-out delivery
- Exactly once processing
- High throughput

300 messages/sec
batch 10 messages/operation 3000/sec

Amazon SQS features (1 of 3)



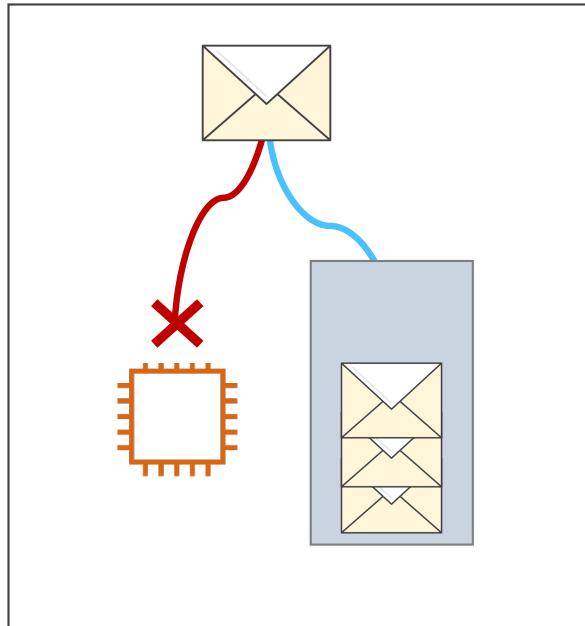
Dead-letter queue support



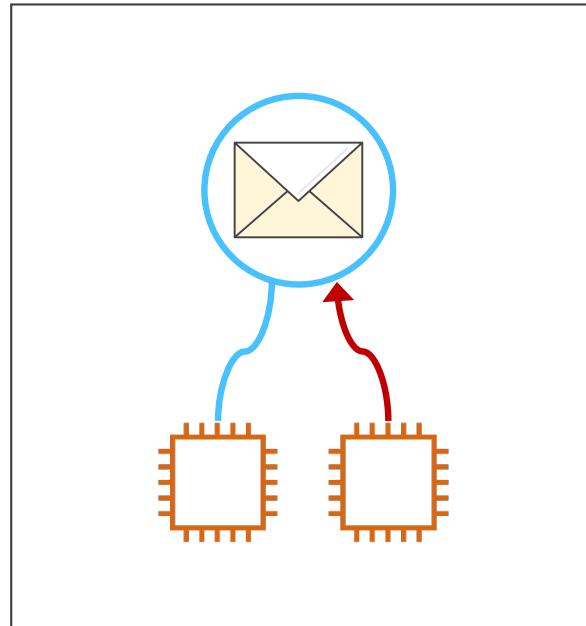
Amazon SQS features (2 of 3)



Dead-letter queue support



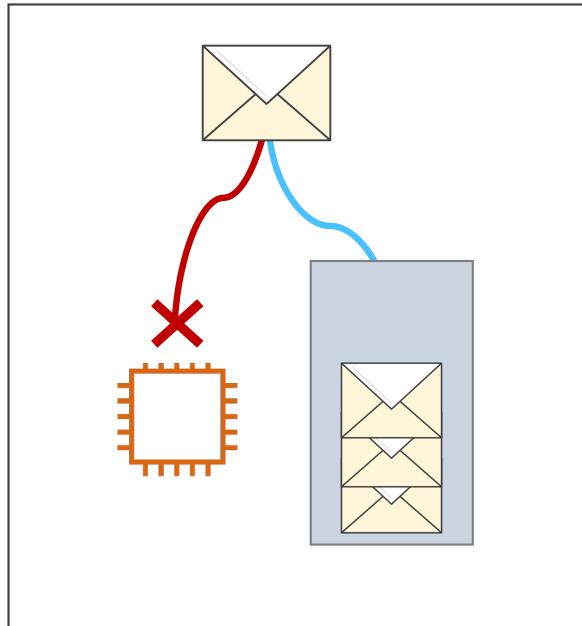
Visibility timeout



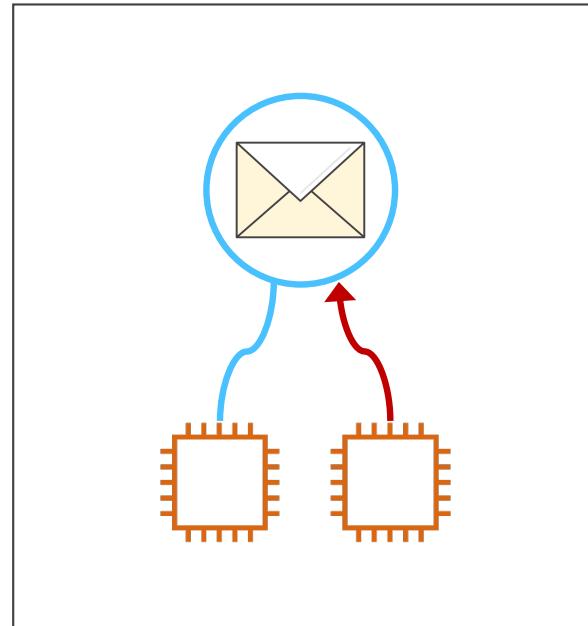
Amazon SQS features (3 of 3)



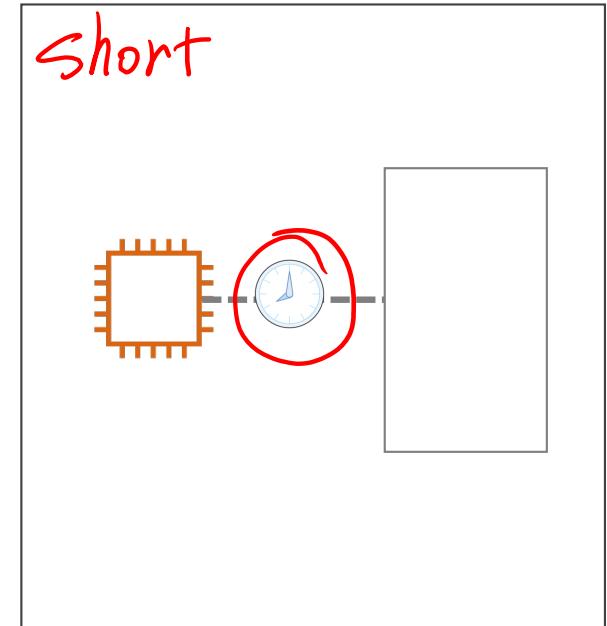
Dead-letter queue support



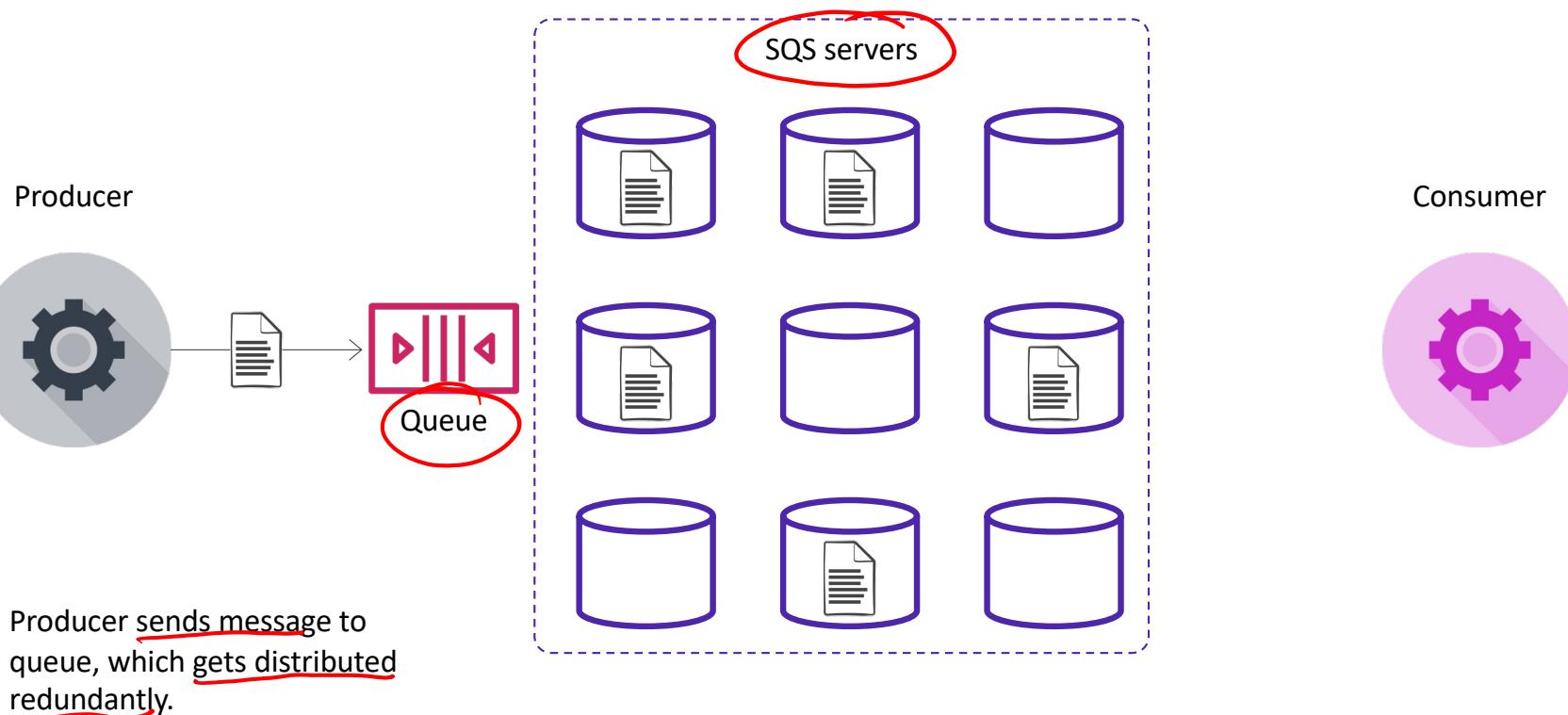
Visibility timeout



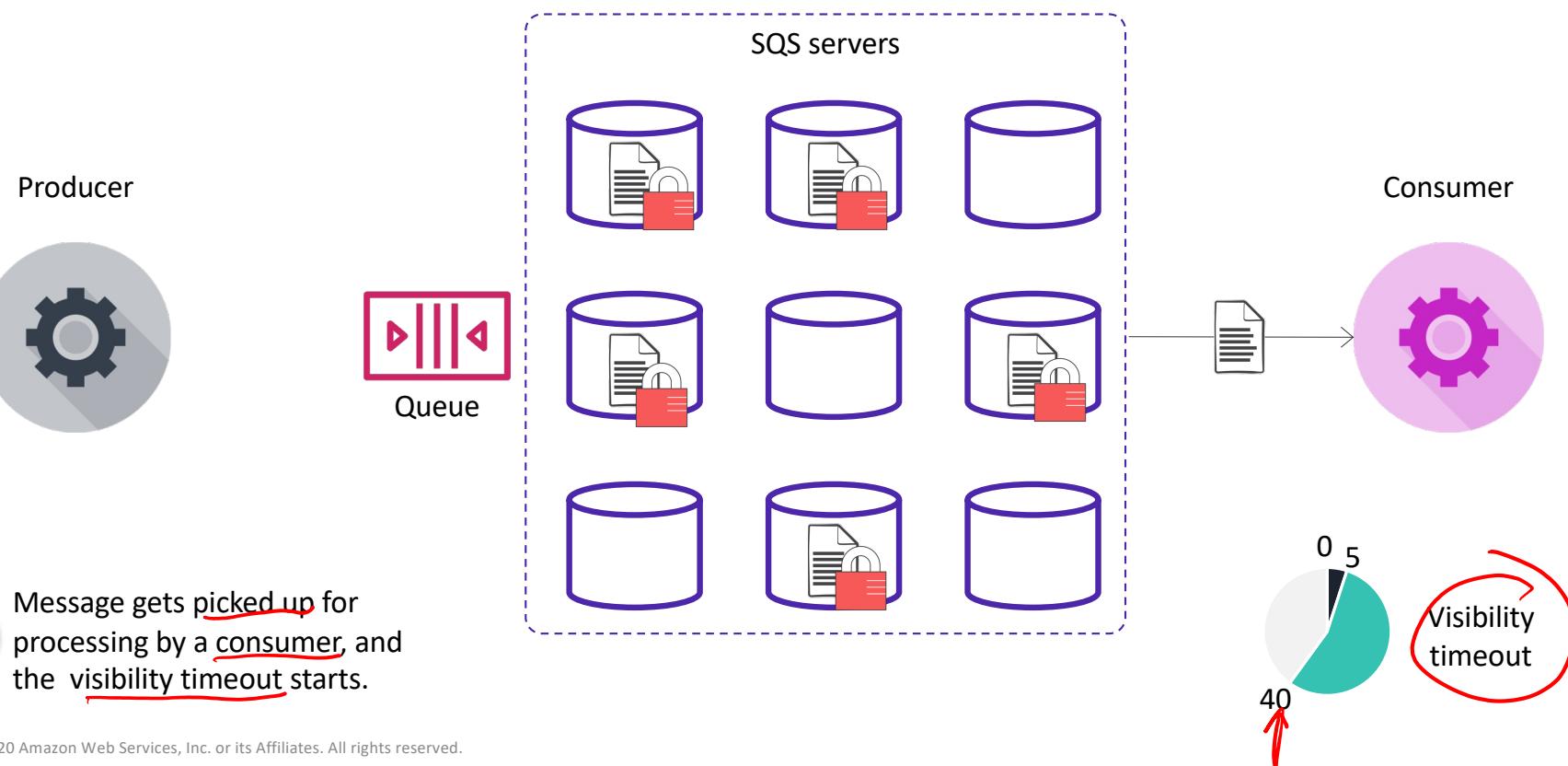
Long polling



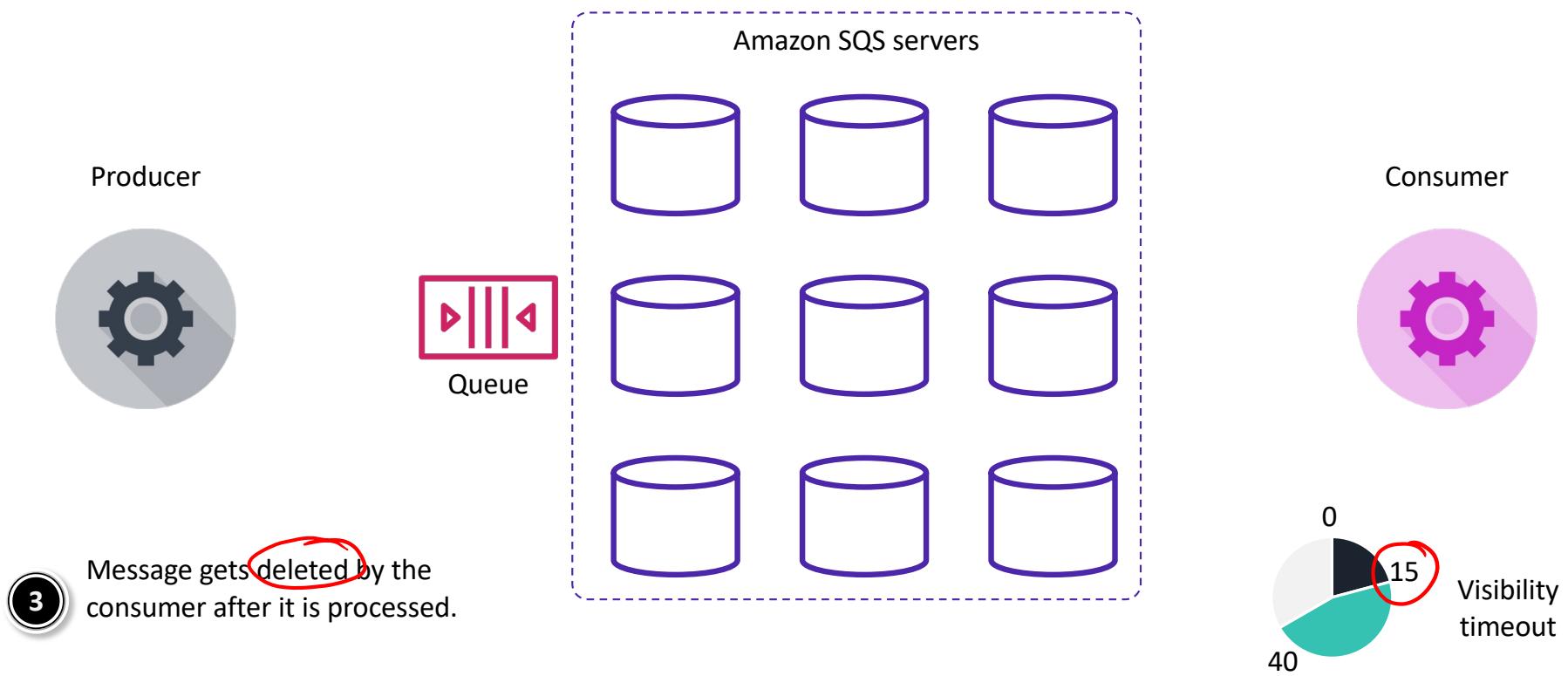
Amazon SQS message lifecycle: Create



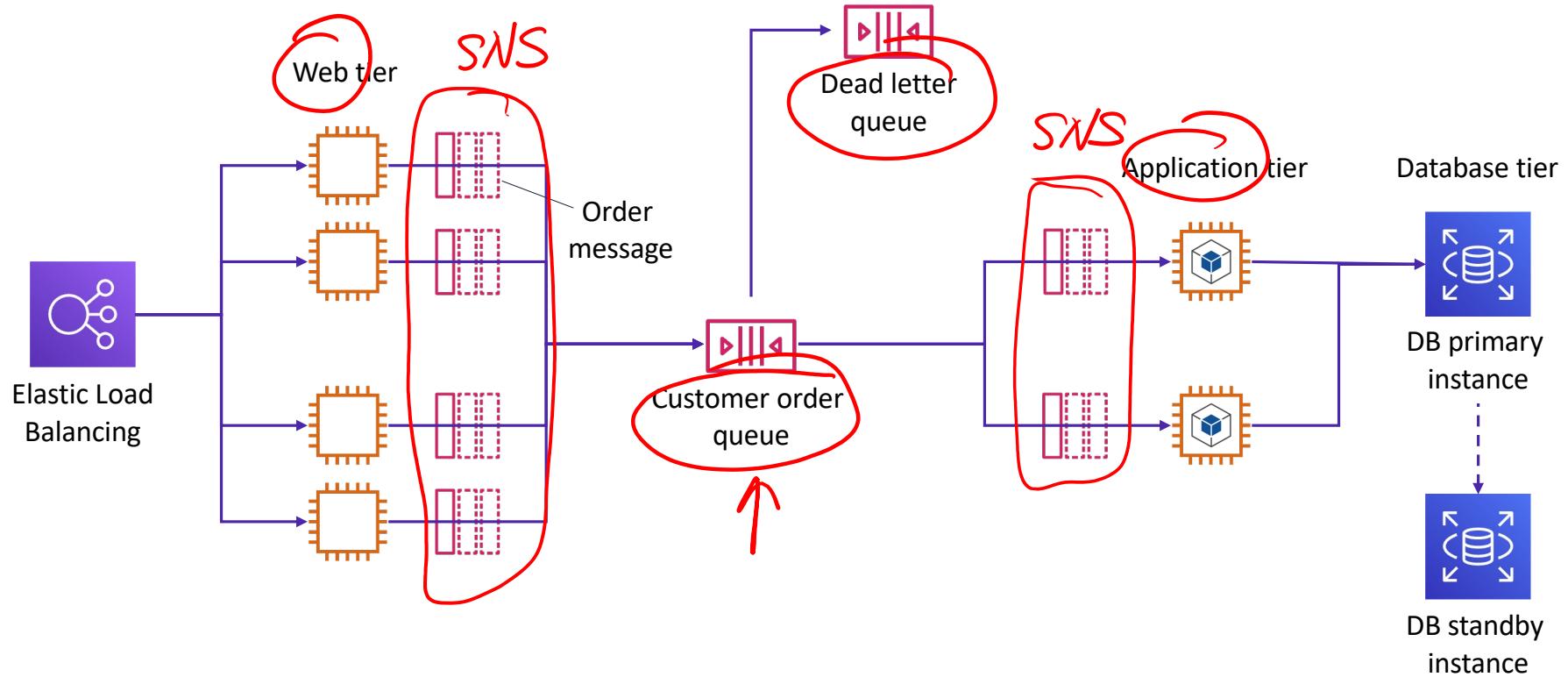
Amazon SQS message lifecycle: Process



Amazon SQS message lifecycle: Delete



Decoupling example: Using Amazon SQS



Demonstration:
Amazon SQS
(https://www.youtube.com/watch?v=PXX8_3ENc2o)

50



© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Section 3 key takeaways



- Amazon SQS is a fully managed, message-queuing service that enables you to decouple application components so they run independently.
- Amazon SQS supports standard and FIFO queues.
- A producer sends a message to a queue. A consumer processes and deletes the message during the visibility timeout.
- Messages that cannot be processed can be sent to a dead letter queue.
- Long polling is a way to retrieve a large number of messages from your SQS queues.

Module 12: Building Decoupled Architectures

Section 4: Decoupling with Amazon SNS

Pub/sub messaging



Publisher – Component that pushes a message to a topic



Publisher



Publisher

Subscriber – Component that subscribes to a topic



Subscriber



Subscriber



Subscriber

Topic

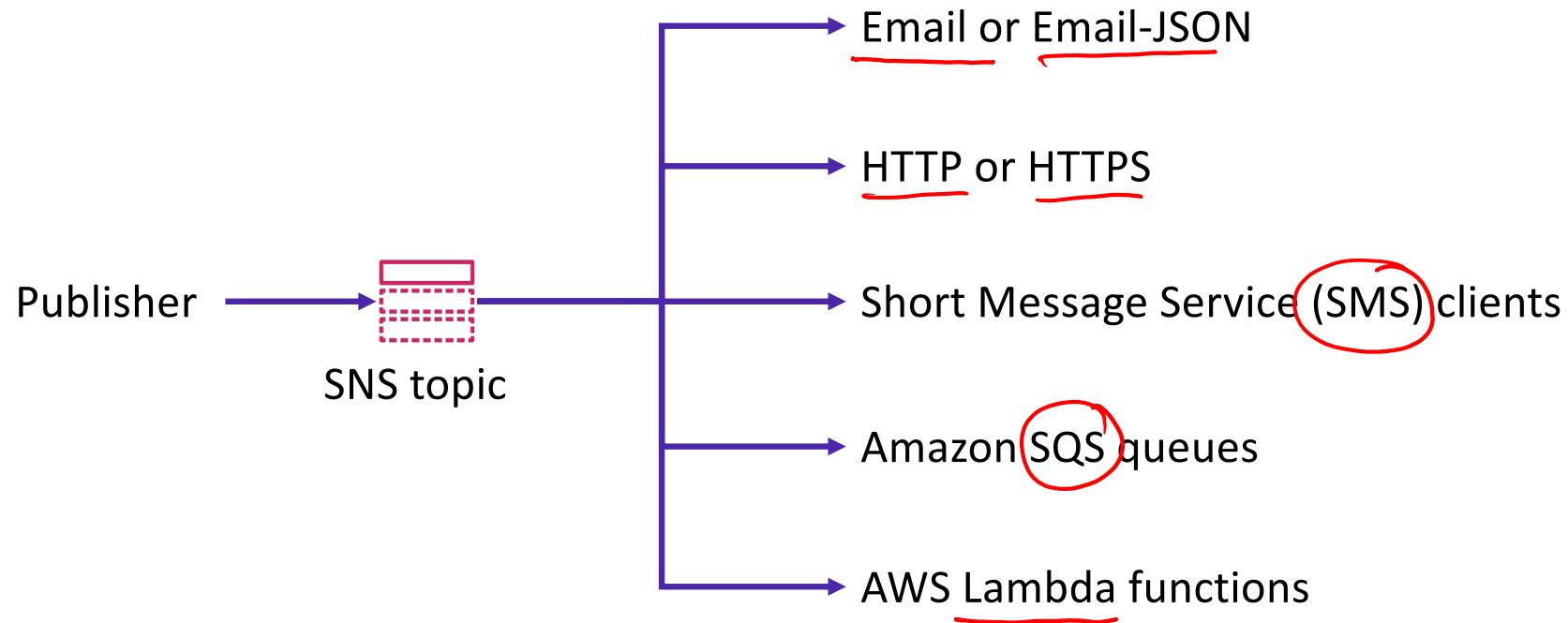
Push mechanism



Amazon Simple
Notification Service
(Amazon SNS)

- Is a highly available, durable, secure, and fully managed **pub/sub messaging** service
- Uses a **push** mechanism
- Supports **encrypted topics** using **(CMKs)** customer master keys
- A single message can only be sent to one SNS topics
- Up to 12,500,000 subscriptions per topic
- Up to 100,000 topics per account

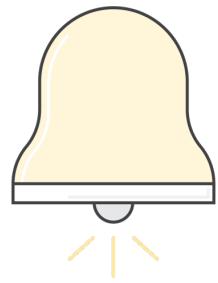
Supported transport protocols



General use cases for Amazon SNS



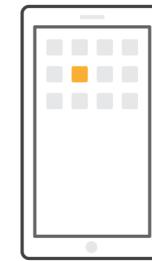
Application and system alerts



Push email and text messaging



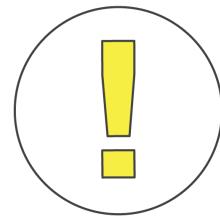
Mobile push notifications



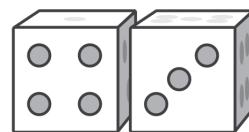
Amazon SNS considerations



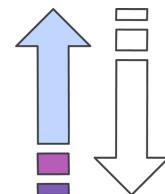
Single published
message



No recall options

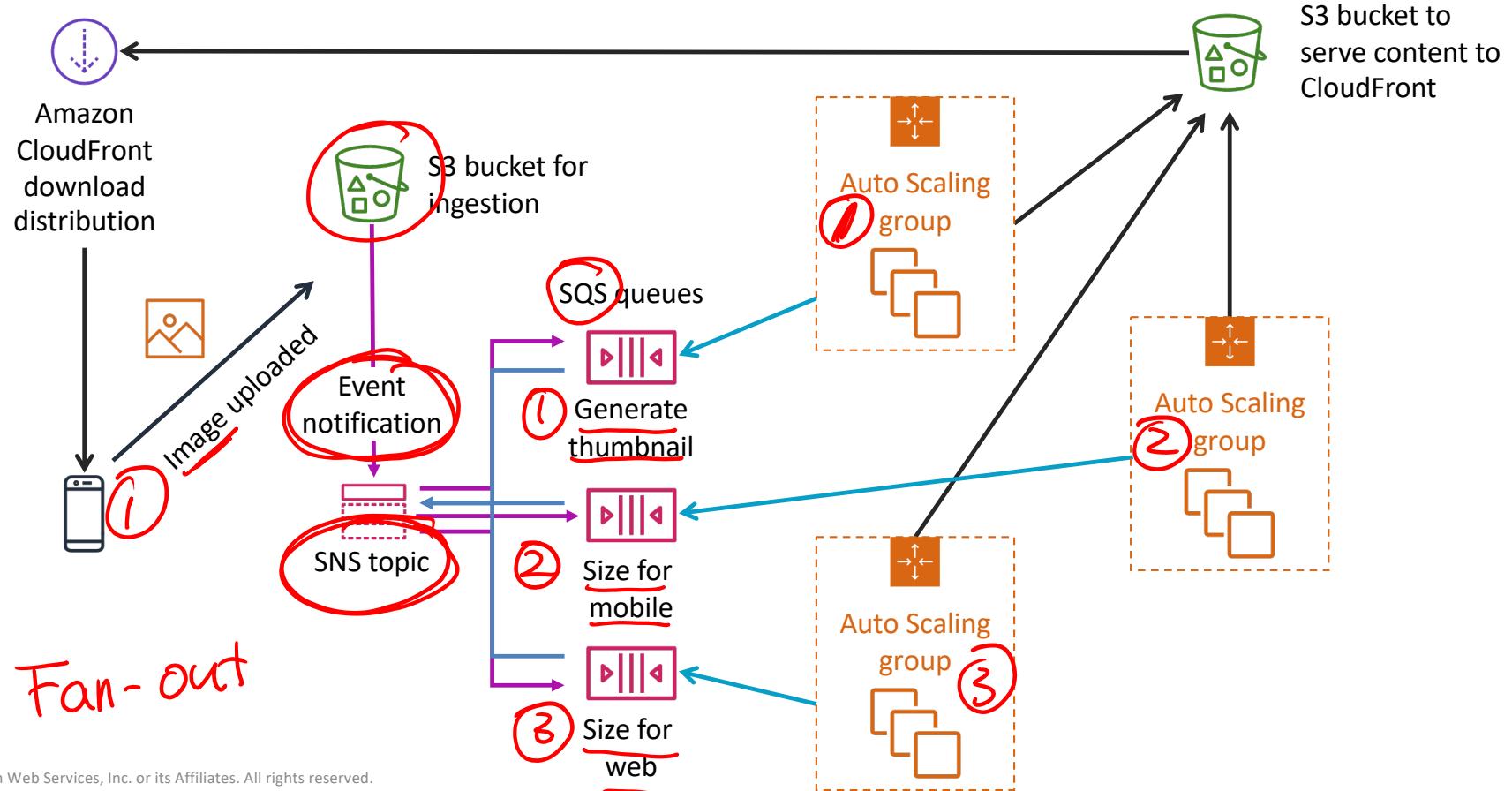


Order and
delivery not
guaranteed



Retry policy for
each delivery
protocol

Decoupling example: Using Amazon S3 with Amazon SNS



Amazon SNS versus Amazon SQS



Feature	Amazon SNS (Publisher/Subscriber)	Amazon SQS (Producer/Consumer)
Producer/consumer	<u>Publish/subscribe</u>	<u>Send/receive</u>
Delivery mechanism	Push (passive)	Poll (active)
Distribution model	<u>Many to many</u>	<u>One to one</u>
Message persistence	No	Yes

Section 4 key takeaways



62



- Amazon SNS is a web service that you can use to set up, operate, and send notifications from the cloud
- Amazon SNS follows the pub/sub messaging paradigm
- When you use Amazon SNS, you create a topic and set policies that restrict who can publish or subscribe to the topic
- You can use topics to decouple message publishers from subscribers, fan-out messages to multiple recipients at one time, and eliminate polling in your applications
- AWS services can publish messages to your SNS topics to trigger event-driven computing and workflows

Module 12: Building Decoupled Architectures

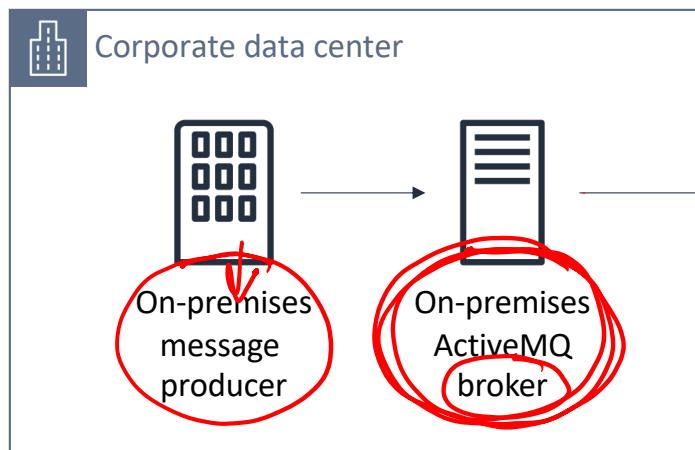
Section 5: Sending messages between cloud applications and on-premises with Amazon MQ



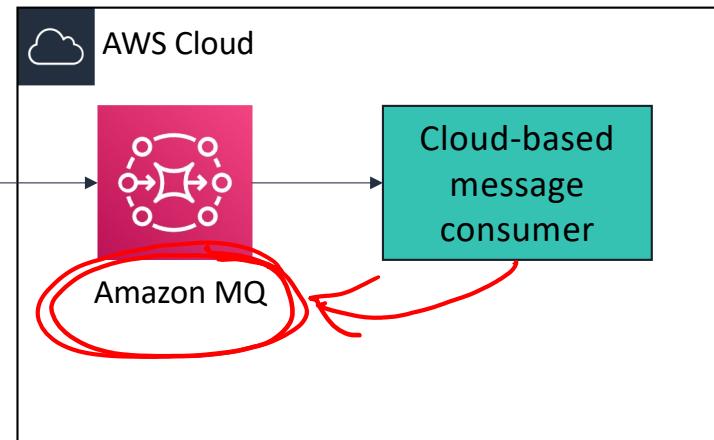
Amazon
MQ

- Is a managed message broker service for Apache ActiveMQ
- Manages the provisioning, setup, and maintenance of ActiveMQ
- Simplifies message migration to the cloud
- Is compatible with open-standard APIs and protocols
 - JMS, NMS, AMQP, STOMP, MQTT, and WebSockets

Amazon MQ use case: Hybrid cloud environment



Message producer
is on-premises



Message consumer
is cloud-based

Amazon MQ versus Amazon SQS and Amazon SNS



Amazon MQ	Amazon SQS and SNS
For <u>application migration</u>	For <u>born-in-the-cloud</u> applications
Protocols: JMS, NMS, AMQP, STOMP, MQTT, and WebSockets	Protocol: <u>HTTPS</u>
Feature-rich	Nearly unlimited throughput
<u>Pay per hour and pay per GB</u>	<u>Pay per request</u>
Can do pub/sub	Cannot do pub/sub in Amazon SQS, but you can do pub/sub in Amazon SNS

Section 5 key takeaways



- Amazon MQ is a managed message-broker service for Apache ActiveMQ that enables you to set up and operate message brokers in the cloud
- Amazon MQ manages the provisioning, setup, and maintenance of ActiveMQ, which is a popular open-source message broker
- Amazon MQ is compatible with open standard APIs and protocols (that is, JMS, NMS, AMQP, STOMP, MQTT, and WebSockets)
- You can use Amazon MQ to integrate on-premises and cloud environments by using the network of brokers feature of ActiveMQ

Module 10: Building Decoupled Architectures

Module wrap-up

Module summary



In summary, in this module, you learned how to:

- Differentiate between tightly and loosely coupled architectures
- Identify how Amazon SQS works and when to use it
- Identify how Amazon SNS works and when to use it
- Describe Amazon MQ