
aws
training and
certification

AWS Certified SysOps Administrator- Associate

Week 6 Content Review

April 2023 Accelerator Cohort

aws training and certification

About the Exam

aws training and certification

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

AWS Certified SysOps Administrator - Associate

About the Exam

- 180 minutes
- 65 ‘Scoring Opportunities’
  - Scored 100 to 1000 (720+ pass)
- $150
- Multiple Response, Individual response questions, and Lab Activities
- In-Person & Remote proctoring available

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Certified SysOps Admin - Associate

Key Exam Topics

Domains Covered:	% of Exam
Domain 1: Monitoring, Logging, and Remediation	20%
Domain 2: Reliability and Business Continuity	16%
Domain 3: Deployment, Provisioning, and Automation	18%
Domain 4: Security and Compliance	16%
Domain 5: Networking and Content Delivery	18%
Domain 6: Cost and Performance Optimization	12%
Total:	**100%**

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Certified SysOps Admin - Associate

Helpful Resources

Training

- AWS Partner Accreditation: Technical
- AWS SysOps Admin – Accelerator Learning plan
- Add-on Subscription Learning Plan

White Papers

- Overview of Amazon Web Services
- AWS Well-Architected Framework
- Management and Governance Lens
- AWS Global Infrastructure
- Shared Responsibility Model
- How AWS Pricing Works
- AWS Architecture Center
- Secure Content Delivery with Amazon CloudFront
- IPv6 on AWS
- Overview of Deployment options on AWS
- Organizing your AWS Environment using multiple accounts

Exam Preparation

- Twitch Power Hours
- Sample Questions
- Schedule an Exam

Looking for more ***Practice Exams***?

*Check out our **Skill Builder Subscription** (information on the next slide)*

aws training and certification

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Skill Builder Subscription

The Skill Builder subscription provides access to official AWS Certification practice exams, self-paced digital training content including open-ended challenges, self-paced labs, and game-based learning. ***Please note, the Skill Builder subscription is not required for this Accelerator program.***

Free digital training

LINK HERE

Special features include:

- 500+ digital courses
- Learning plans
- 10 Practice Question Sets
- *AWS Cloud Quest*

Individual subscription

LINK HERE

Everything in free digital training, plus:

- AWS Cloud Quest (3 additional roles)
- AWS Certification Official Practice Exams
- Exam prep courses
- 100+ AWS Builder Labs
- AWS Jam Journey (lab-based challenges)

Access ***65*** SysOps Admin – Associate Practice Exam Questions with feedback on your answer choices

Individual subscriptions are priced at $29 USD per month (*Flexibility to cancel anytime*) or $299 USD per year.

aws training and certification

Loosely Coupled Architecture

aws training and certification

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

Monolithic Applications

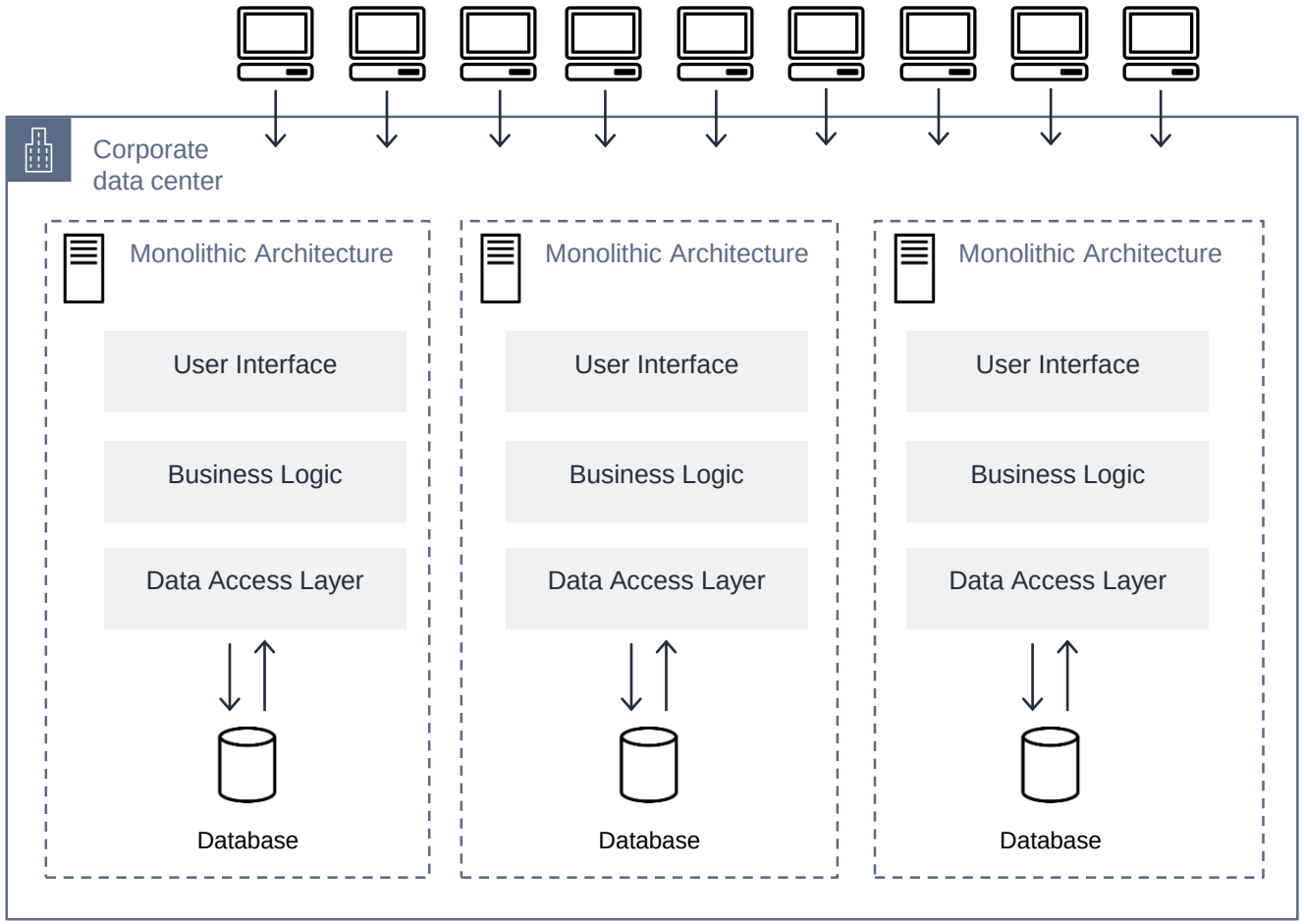
Single, unified application with all components tightly coupled, working from a shared database

Challenges

Resource management – Massive application requiring resources across memory, storage, and network bandwidth.

Inefficient Scaling - monoliths up or out requires scaling of the entire application, even if single module needs the resources

Complexity – Tightly coupled modules grow increasingly complex over time. Maintenance and updates are exceedingly resource intensive, with large impacts to business agility.



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Microservice Applications

Break apart a monolith to build more scalable, fault tolerant applications

Use individual components and services, loosely coupled, which operate independently

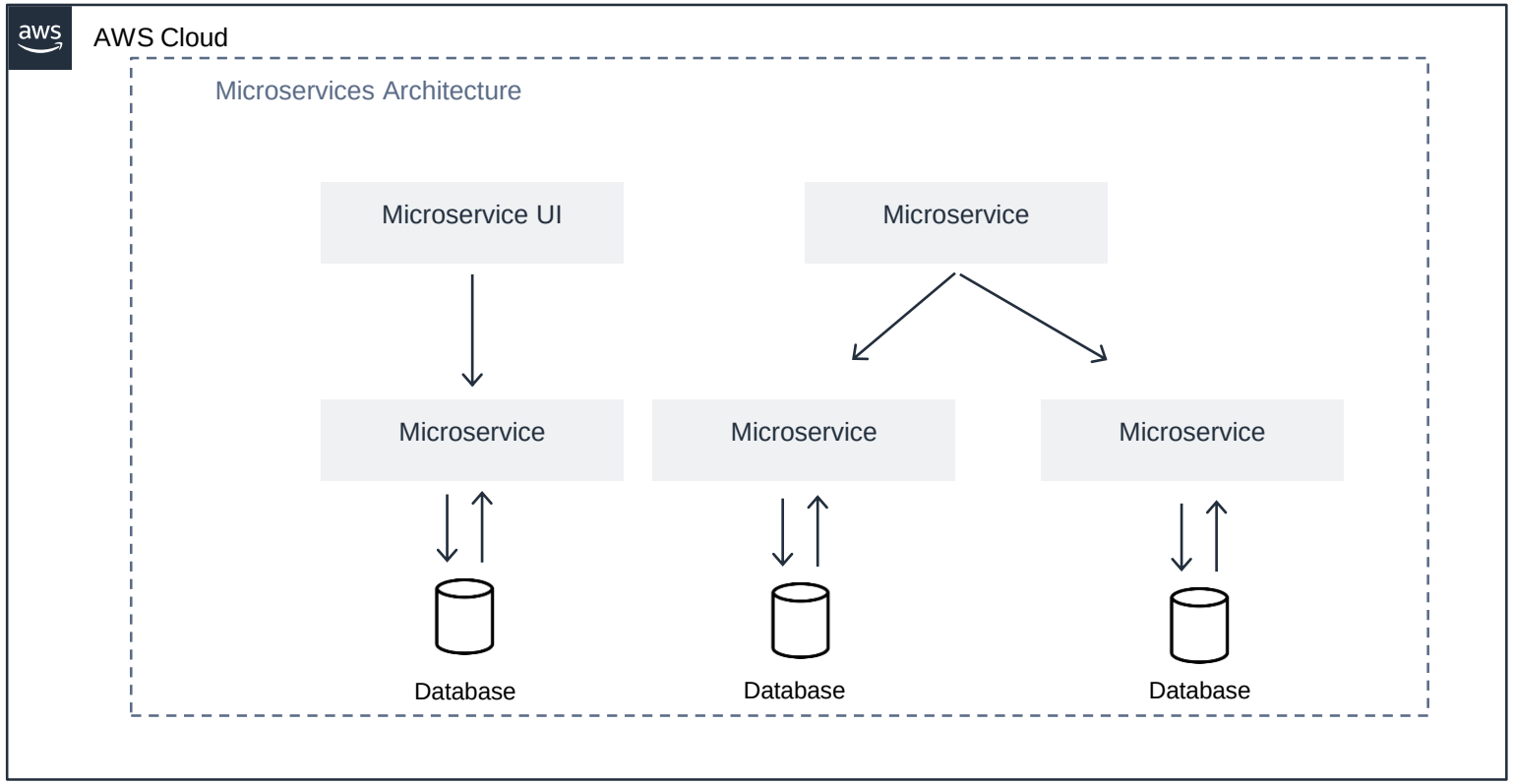
Summary

Decompose monoliths or architect loose coupling from the beginning.

Outcomes

Rapid adjustments to fluctuating business demand but without interrupting core activities, such as high scalability, improved resiliency, continuous delivery, and failure isolation.

Faster innovation because each microservice can be individually tested and deployed.



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Event-Driven Architecture

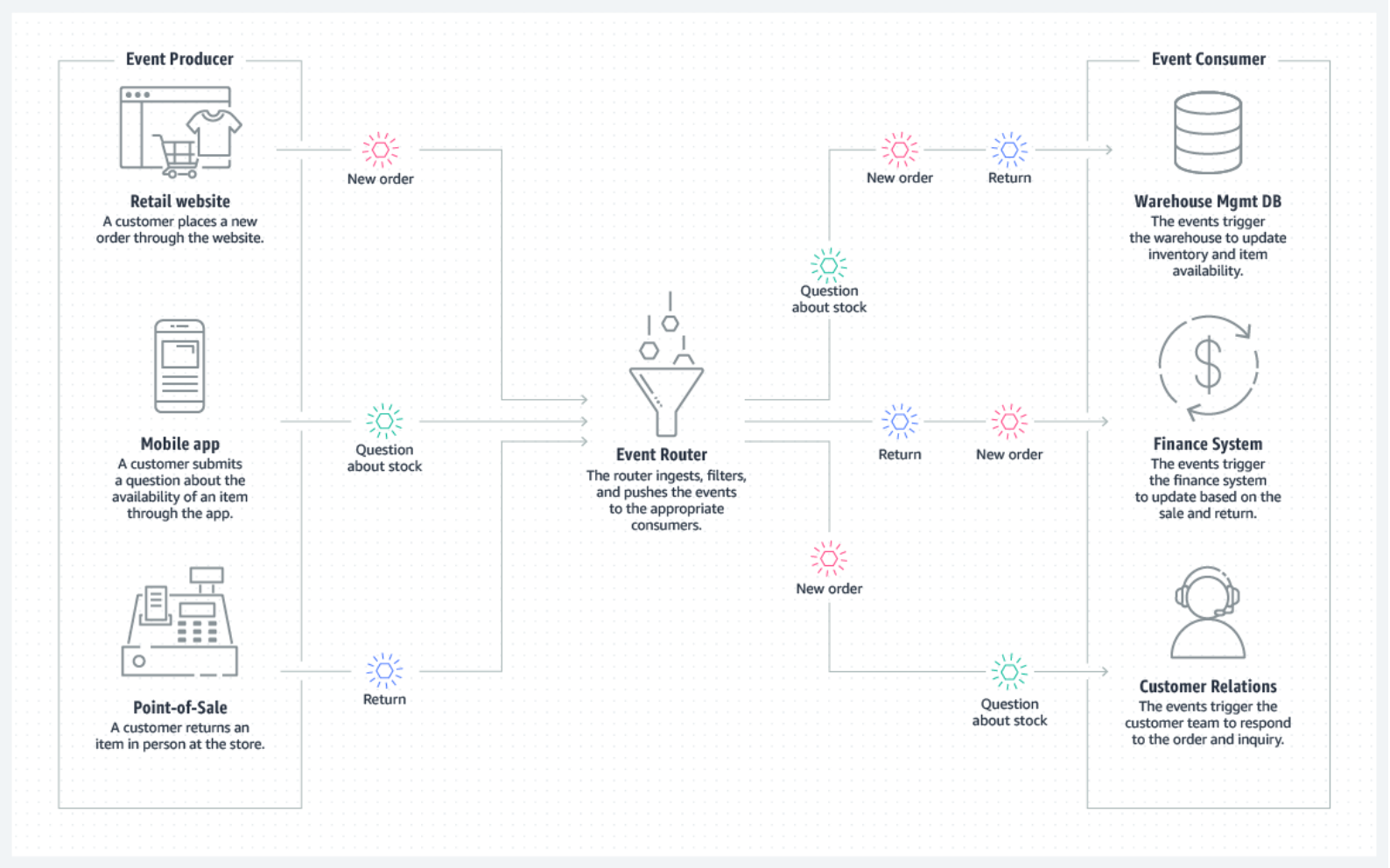
Decoupled systems that run in response to events

How it Works

Uses events to trigger and communicate between decoupled services and is common in modern applications built with microservices. Event-driven architectures have three key components: event producers, event routers, and event consumers.

Benefits

- Scale and fail independently
- Audit with ease
- Develop with agility
- Cut costs



aws training and certification

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon Simple Queue Service

Fully managed message queuing service

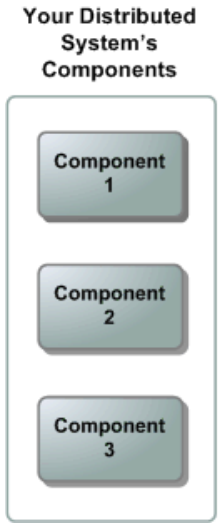
Amazon SQS Enables you to decouple and scale microservices, distributed systems, and serverless applications.

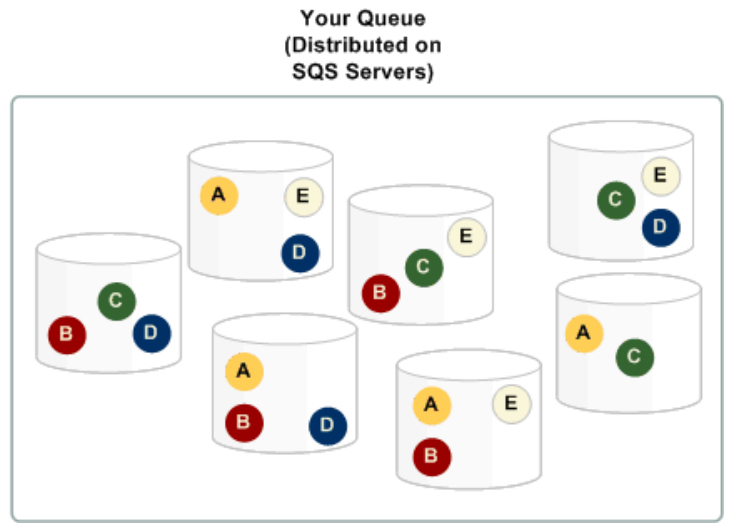
How it Works

AWS CloudFormation lets you model, provision, and manage AWS and third-party resources by treating infrastructure as code.

Use Cases

Common use cases for CloudFormation include managing infrastructure with DevOps through automated, test and deploy infrastructure templates and scaling production stacks at scale.





aws training and certification

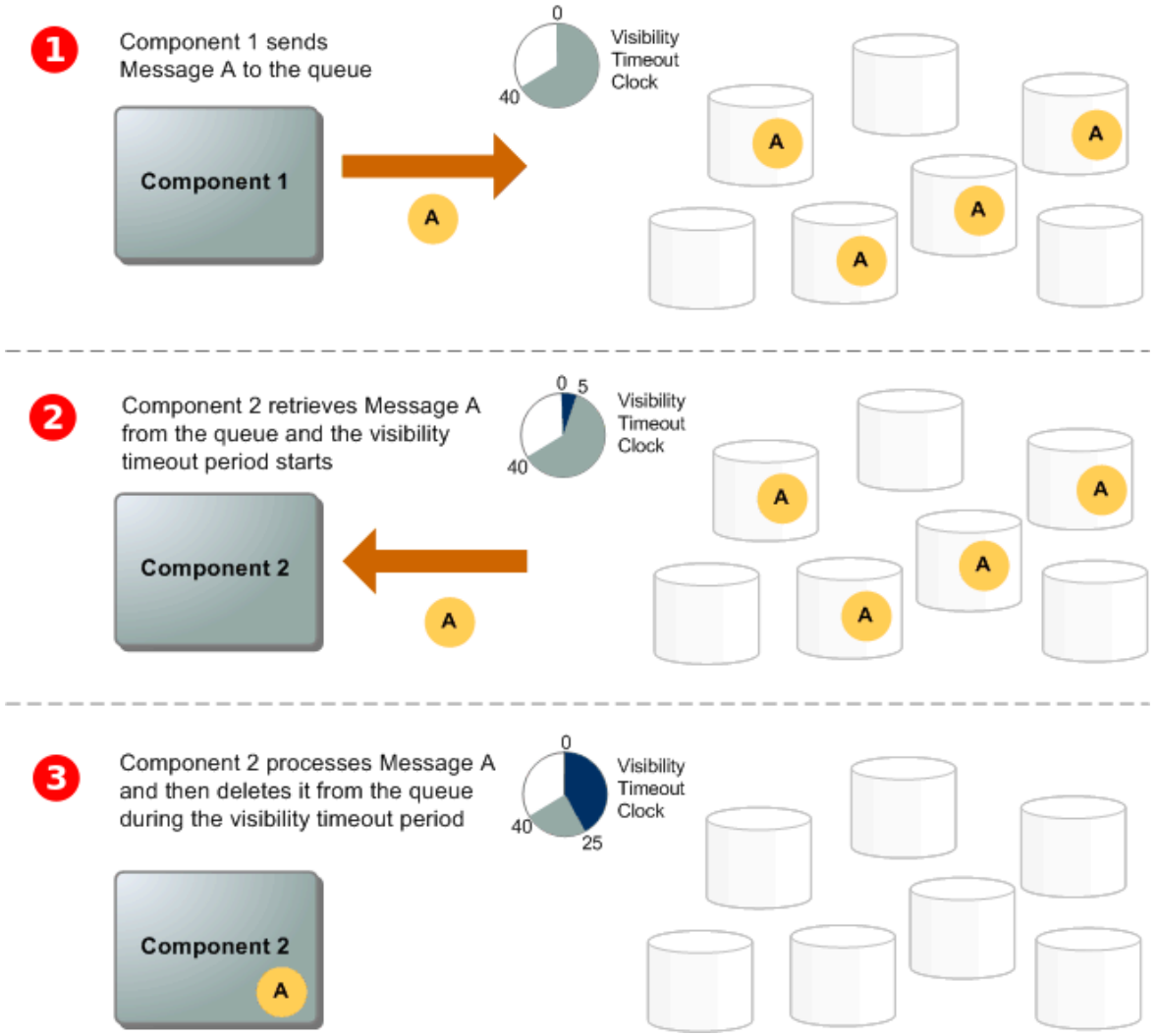
© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon SQS Message Lifecycle

1. Producer sends message to a queue, and message distributed across Amazon SQS servers redundantly.

2. When a consumer (component 2) is ready to process messages, it consumes messages from the queue, and message A is returned. While message A is being processed, it remains in the queue and isn't returned to subsequent receive requests for the duration of the visibility timeout.

3. The consumer (component 2) deletes message A from the queue to prevent the message from being received and processed again when the visibility timeout expires.



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon SQS Standard Queues

SQS Default queue type, nearly unlimited API calls per second

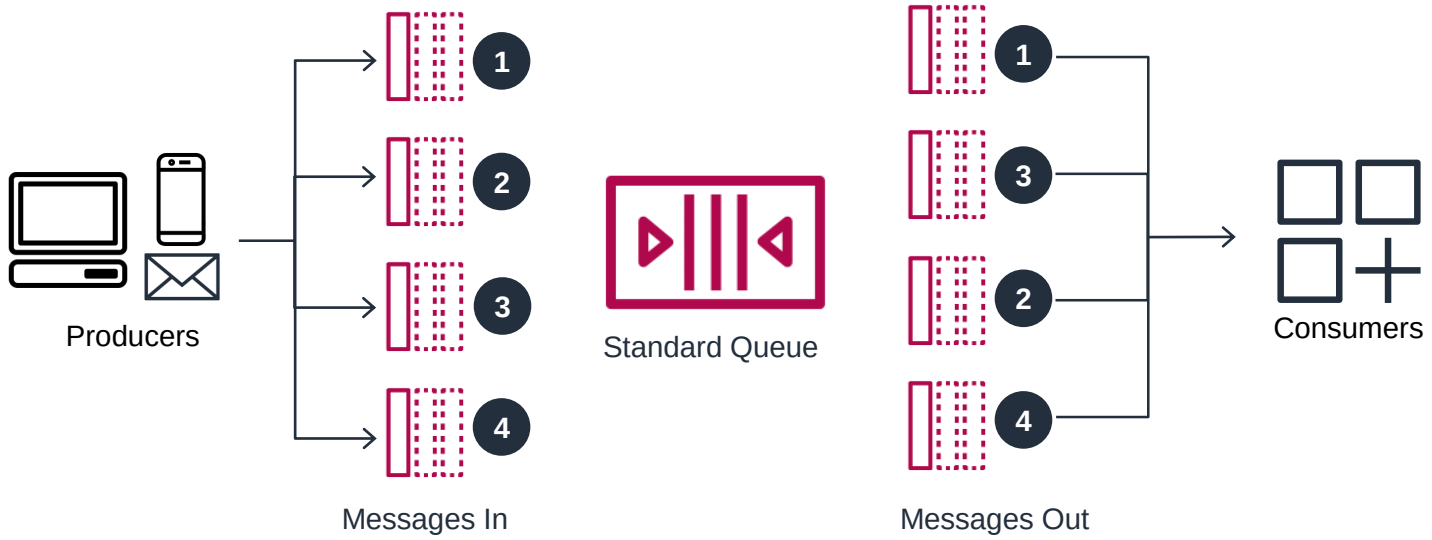
Supports at-least-once message delivery, *best-effort* ordering

Message Ordering

A standard queue makes a best effort to preserve the order of messages, but more than one copy of a message might be delivered out of order.

At-least-once delivery

Amazon SQS stores copies of your messages on multiple servers for redundancy and high availability. On rare occasions, one of the servers that stores a copy of a message might be unavailable when you receive or delete a message.



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon SQS First-In-First-Out (FIFO) Queues

Exactly-one processing, limited transactions per second.

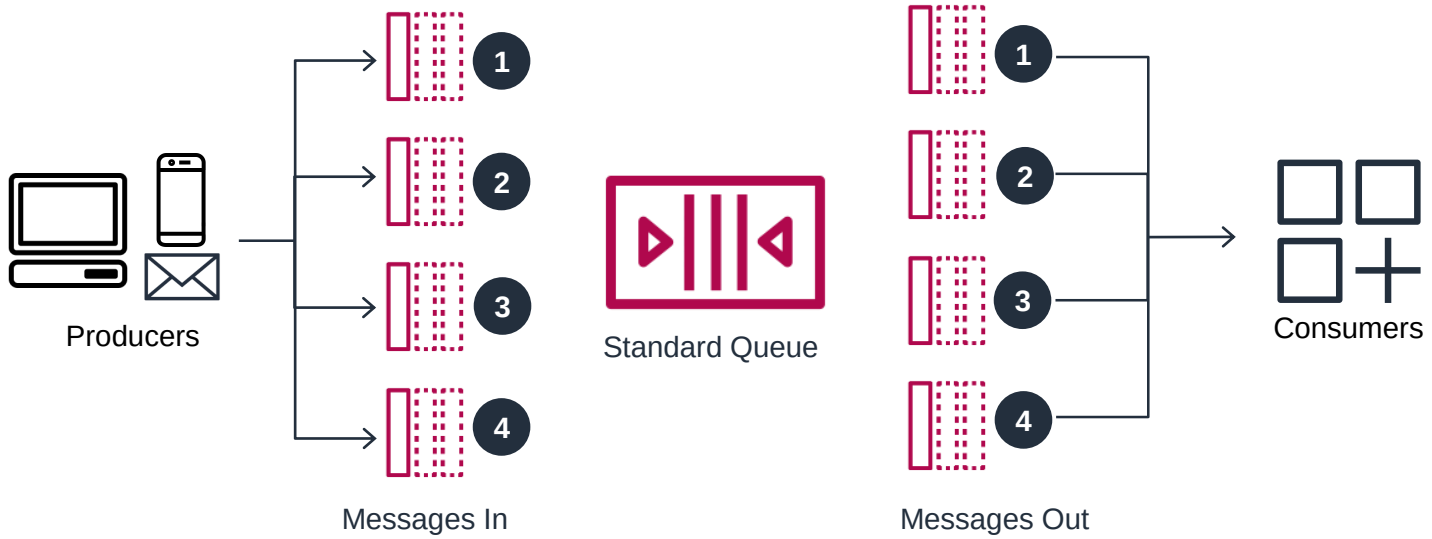
When order of operations or events is critical; or when duplicates can't be tolerated

Message Ordering

The order in which messages are sent and received is strictly preserved and a message is delivered once and remains available until a consumer processes and deletes it.

Exactly-once processing

FIFO queues don't introduce duplicate messages. FIFO queues help you avoid sending duplicates to a queue.



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

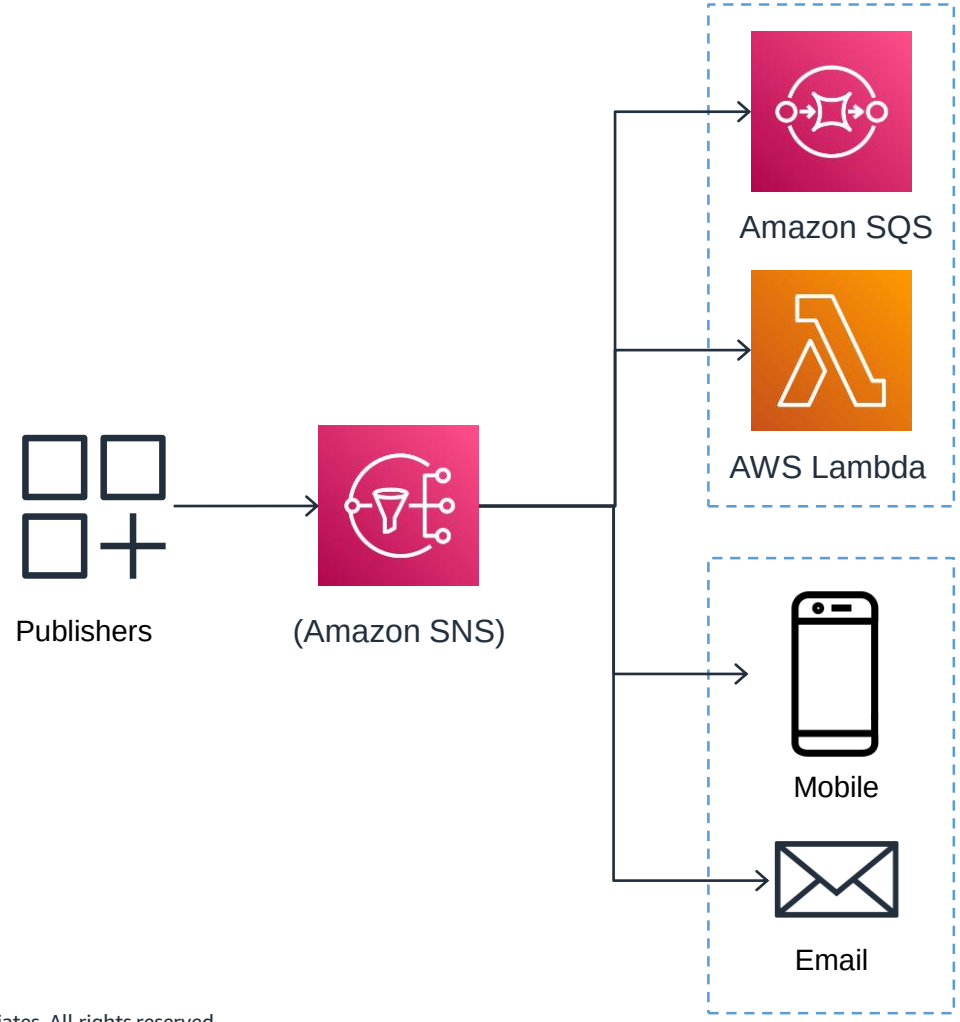
Amazon Simple Notification Service

Fully managed pub/sub messaging, SMS, email, and mobile push notifications

How it Works

Application to Application (A2A) pub/sub provides topics for many-to-many messaging across distributed systems, microservices, and event-driven serverless applications. Publisher systems can use topics to fan out messages to large number of subscriber systems.

Application to Person (A2P) enables message sending to users at scale via SMS, Mobile push, and email.



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon SNS Pub/Sub

Asynchronous event notifications

Publish/subscribe messaging, is a form of asynchronous service-to-service communication used in serverless and microservices architectures.

How it Works

Any message published to a topic is immediately received by all of the subscribers to the topic. Pub/sub messaging can be used to enable event-driven architecture or to decouple applications in order to increase performance, reliability and scalability.

Pub/sub model allows messages to be broadcast to different parts of a system asynchronously. To broadcast a message, a component called a publisher simply pushes a message to the topic. All components that subscribe to the topic will receive every message that is broadcast.

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon EventBridge

Amazon EventBridge is a serverless event bus service that you can use to connect your applications with data from a variety of sources.

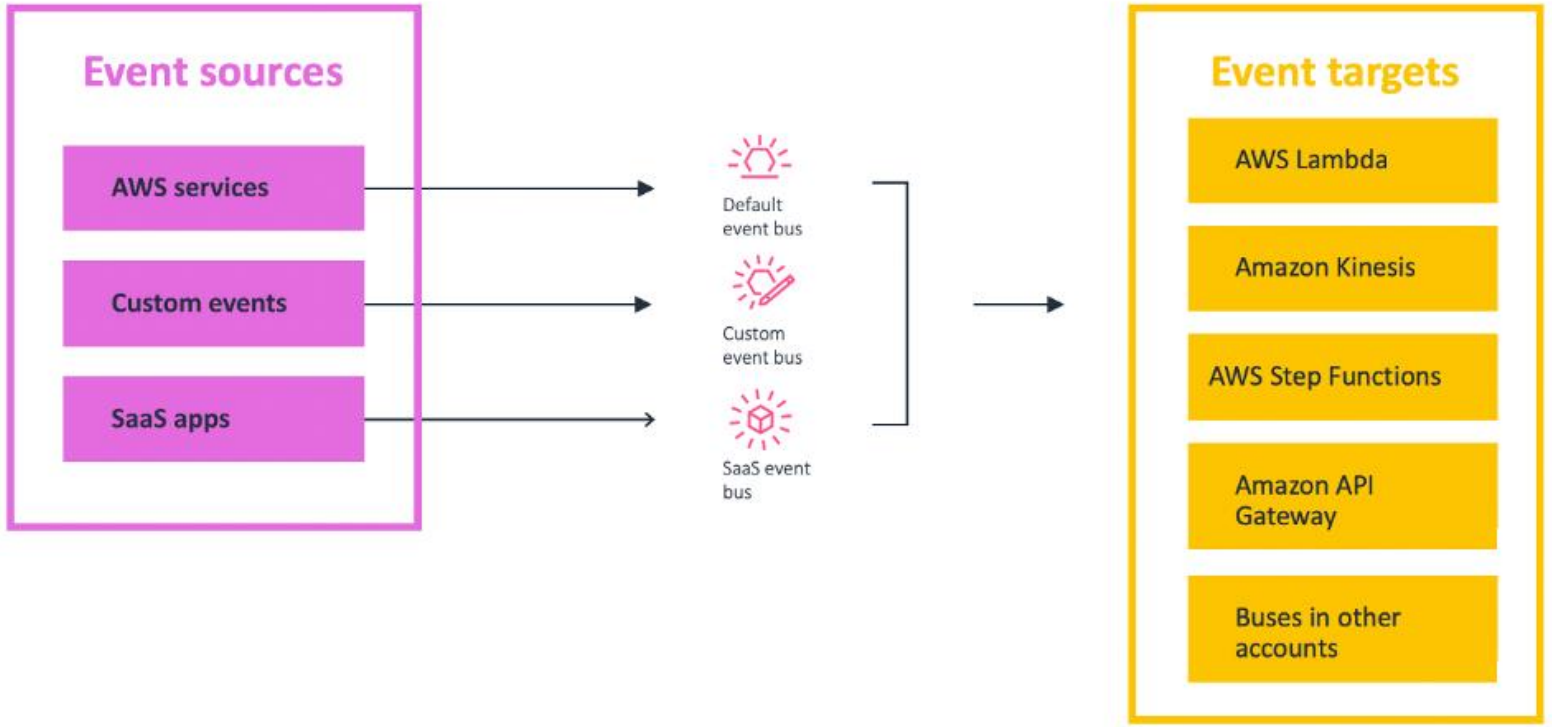
What is it?

Amazon EventBridge is a serverless event bus service that you can use to connect your applications with data from a variety of sources. EventBridge delivers a stream of real-time data from your applications, software as a service (SaaS) applications, and AWS services to targets such as AWS Lambda functions, HTTP invocation endpoints using API destinations, or event buses in other AWS accounts.

How it works

EventBridge receives an event, an indicator of a change in environment, and applies a rule to route the event to a target. Rules match events to targets based on either the structure of the event, called an event pattern, or on a schedule.

For example, when an Amazon EC2 instance changes from pending to running, you can have a rule that sends the event to a Lambda function.



Lambda & Troubleshooting

aws training and certification

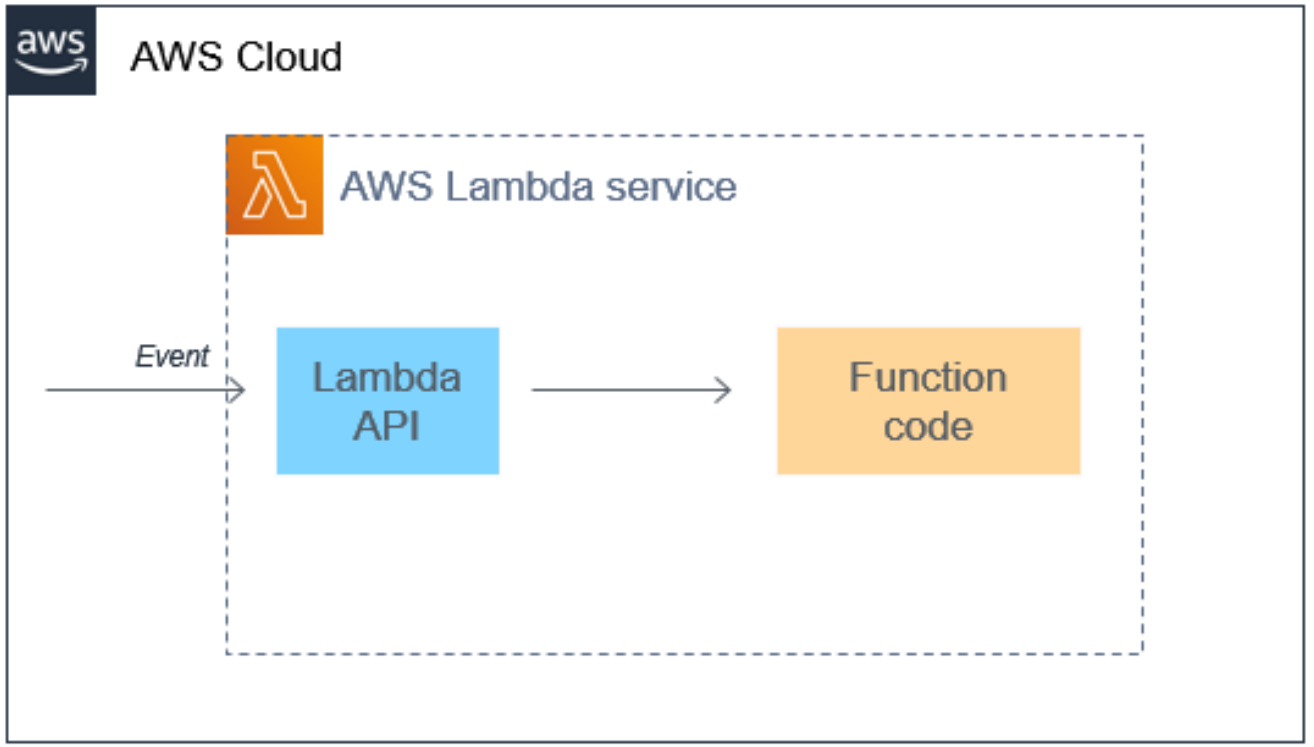
© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Lambda

On-demand compute service, executing code (functions) in response to events.

How does it work?

Most AWS services generate events, and many can act as an event source for Lambda. Within Lambda, your code is stored in a code deployment package. All interaction with the code occurs through the Lambda API and there is no direct invocation of functions from outside of the service.

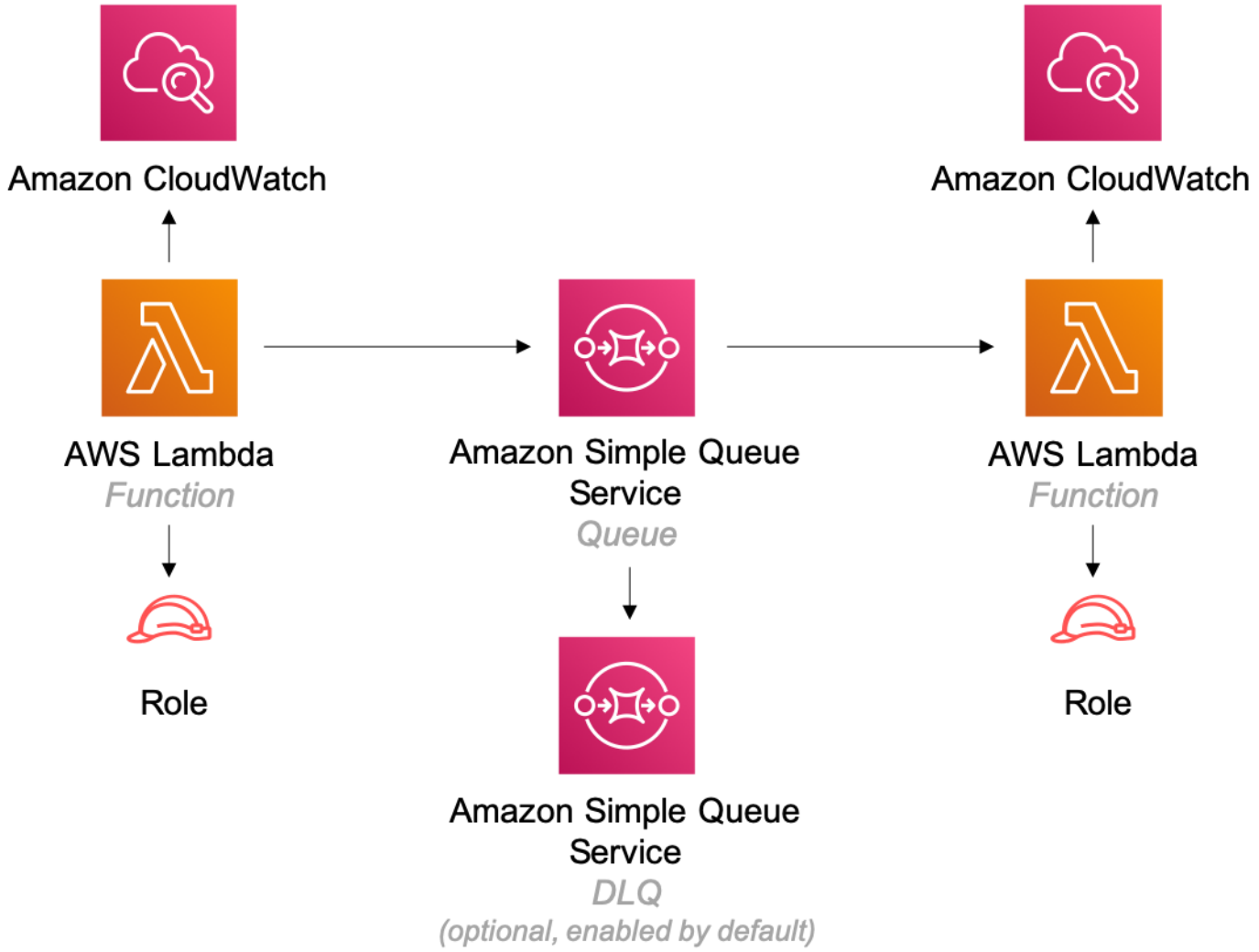


© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Lambda w/ SQS

Example architecture

1. AWS Lambda function sends messages to a queue
2. Amazon SQS queue holds messages
3. AWS Lambda function consumes messages from the queue
4. Activity from both functions monitored with Amazon CloudWatch



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Troubleshooting – Deployment

Lambda deploys function updates by launching new instances; errors prevent new version deployment

Direct Deployment – Changes deployed to function directly from Lambda API, or with a client (e.g., AWS CLI).

Error Tracing – Errors will be shown directly in the output

Indirect Deployment – Changes deployed indirectly via other AWS services (AWS CloudFormation, AWS CodeDeploy, AWS CodePipeline).

Error Tracing – Logs or event stream from that service will have Lambda response

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Deployment Errors - General

Permission is denied / Cannot load such file

Example Errors

Error: *EACCES: permission denied, open '/var/task/index.js'*

Error: *cannot load such file – function*

Error: *[Errno 13] Permission denied: '/var/task/function.py'*

Resolution

Lambda runtime doesn't have permissions to read the files referenced in your deployment package.

Use **chmod** *command to change file mode:*
Chmod –R o+rX

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Deployment Errors - General

Error occurs when calling the UpdateFunctionCode

Example Errors

Error: *An error occurred (RequestEntityTooLargeException) when calling the UpdateFunctionCode operation*

Resolution

Uploads of deployment package or layer archive as ZIP file directly to Lambda has **50 MB** limit.

Larger files may be stored in Amazon S3 and shared with S3Bucket & S3Key parameters.

Note – Binary ZIP converted to base64 upon upload, increase size approx. 30%.

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Deployment Errors – Amazon S3

Error Code PermanentRedirect

Example Error

Error: *Error occurred while GetObject. S3 Error Code: PermanentRedirect. S3 Error Message: The bucket is in this region: us-east-2. Please use this region to retry the request*

Resolution

Upload of deployment package from an Amazon S3 bucket requires the bucket and function to be in same Region.

Issue occurs when Amazon S3 object specified in a call to **UpdateFunctionCode**, or when package and deploy commands used in AWS CLI or AWS SAM CLI.

Create deployment artifact bucket for each Region where applications are developed.

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Deployment Errors – General

Cannot find, cannot load, unable to import, class not found, no such file or directory

Example Errors

Error: *Cannot find module 'function'*

Error: *cannot load such file – function*

Error: *Unable to import module 'function'*

Error: *Class not found: function.Handler*

Error: *fork/exec /var/task/function: no such file or directory*

Error: *Unable to load type 'Function.Handler' from assembly 'Function'.*

Context

Name of file or class in function's handler configuration doesn't match code.

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Deployment Errors – General

Undefined method handler

Example Errors

Error: *index.handler is undefined or not exported*

Error: *Handler 'handler' missing on module 'function'*

Error: *undefined method `handler' for #<LambdaHandler:0x000055b76ccebf98>*

Error: *No public method named handleRequest with appropriate method signature found on class function.Handler*

Error: *Unable to find method 'handleRequest' in type 'Function.Handler' from assembly 'Function'*

Context

Name of handler method in function's handler configuration doesn't match the code.

Each runtime defines naming convention for handlers. Handler is method in function's code that the runtime runs when function is invoked.

Lambda provides a library for some languages. The interface expects handler methods to have a specific name.

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Deployment Errors – Lambda

Layer conversion failed

Example Error

Error: *Lambda layer conversion failed. For advice on resolving this issue, see the Troubleshoot deployment issues in Lambda page in the Lambda User Guide.*

Resolution

Lambda functions configured with a layer will merge the layer with the function code.

Error indicates that this merge has failed to complete.

- Delete any unused files from your layer
- Delete any symbolic links in your layer
- Rename any files that have the same name as a directory in any of your function's layers

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Deployment Errors – Lambda

InvalidParameterValueException or RequestEntityTooLargeException

Example Errors

Error: *InvalidParameterValueException: Lambda was unable to configure your environment variables because the environment variables you have provided exceeded the 4KB limit. String measured: {"A1":"uSFeY5cyPiPn7AtnX5BsM...*

Error: *RequestEntityTooLargeException: Request must be smaller than 5120 bytes for the UpdateFunctionConfiguration operation*

Resolution

- Maximum size of variables object stored in the function's configuration must not exceed 4096 bytes.
- Inclusive of key names, values, quotes, commas, and brackets.
- Total size of the HTTP request body is also limited.

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Deployment Errors – Lambda

InvalidParameterValueException

Example Error

Error: *InvalidParameterValueException: Lambda was unable to configure your environment variables because the environment variables you have provided contains reserved keys that are currently not supported for modification.*

Context

Lambda reserves some environment variable keys for internal use.

Example - AWS_REGION is used by the runtime to determine the current Region and cannot be overridden.

Other variables (PATH) are used by the runtime but can be extended in your function configuration.

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Deployment Errors – Lambda

Concurrency and memory quotas

Example Errors

Error: *Specified ConcurrentExecutions for function decreases account's UnreservedConcurrentExecution below its minimum value*

Error: *'MemorySize' value failed to satisfy constraint: Member must have value less than or equal to 3008*

Context

Occur when concurrency or memory quotas for account are exceeded. New AWS accounts have reduced quotas.

Concurrency: Error if you try to create a function using reserved or provisioned concurrency, or if your per-function concurrency request (PutFunctionConcurrency) exceeds your account's concurrency quota. *Increase can be requested*.

Memory: Errors occur if the amount of memory allocated to the function exceeds your account's memory quota. *Cannot be increased*.

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Troubleshooting – Invocation

Invoked Lambda functions validate requests and check for scaling capacity first

Direct Invocation – Changes deployed to function directly from Lambda API, or with a client (e.g., AWS CLI).

Error Tracing – Errors will be shown directly in the invocation response from lambda

Asynchronous invocation – Invoked via event source mapping or through another service.

Error Tracing – Errors can be found in logs, a dead-letter queue, or failed-event destination.

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Invocation Errors – IAM

lambda:InvokeFunction not authorized

Example Error

Error: *User: arn:aws:iam::123456789012:user/developer is not authorized to perform: lambda:InvokeFunction on resource: my-function*

Context

Your user, or the role that you assume, must have permission to invoke a function. This requirement also applies to Lambda functions and other compute resources that invoke functions. Add the AWS managed policy AWSLambdaRole to your user, or add a custom policy that allows the lambda:InvokeFunction action on the target function.

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Invocation Errors – Lambda

Operation cannot be performed ResourceConflictException

Error: *User: arn:aws:iam::123456789012:user/developer is not authorized to perform: lambda:InvokeFunction on resource: my-function*

Context: When you connect a function to a virtual private cloud (VPC) at creation, the function enters a Pending state while Lambda creates elastic network interfaces. While Pending, you can't invoke or modify. If connected after creation, you can invoke, but not modify code or configuration.

Function is stuck in Pending

Error: *A function is stuck in the Pending state for several minutes.*

Resolution: If stuck in *Pending* state for more than six minutes, three API operations can unblock:
UpdateFunctionCode; UpdateFunctionConfiguration; PublishVersion
Lambda will cancel the pending operation and puts function in the *Failed* state.

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Invocation Issues – Lambda

One function is using all concurrency

Issue: *One function is using all of the available concurrency, causing other functions to be throttled.*

Context: Split AWS account's available concurrency into pools with reserved concurrency. Reserved concurrency ensures that a function can always scale to (but not beyond) assigned concurrency.

Alias routing with provisioned concurrency

Issue: *Provisioned concurrency spillover invocations during alias routing.*

Context: Lambda uses a simple probabilistic model to distribute the traffic between the two function versions. At low traffic levels, you might see a high variance between the configured and actual percentage of traffic on each version. If your function uses provisioned concurrency, you can avoid spillover invocations by configuring a higher number of provisioned concurrency instances during the time that alias routing is active.

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Invocation Issues – General

Cannot invoke function with other accounts or services

Issue: *You can invoke your function directly, but it doesn't run when another service or account invokes.*

Context: You grant other services and accounts permission to invoke a function in the function's resource-based policy. If the invoker is in another account, that user must also have permission to invoke functions.

Function invocation is looping

Issue: *Function is invoked continuously in a loop.*

Context: Typically occurs when function manages resources in the same AWS service that triggers it.

To stop the function from running, on the function configuration page, choose **Throttle**. Then, identify the code path or configuration error that caused the recursive invocation.

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Invocation Issues – EFS

Function could not mount the EFS file system

Error: *EFSMountFailureException: The function could not mount the EFS file system with access point arn:aws:elasticfilesystem:us-east-2:123456789012:access-point/fsap-015cxmplb72b405fd.*

Context: Check the function's permissions, and confirm that its file system and access point exist and are ready for use.

Function could not connect to the EFS file system

Error: *EFSMountConnectivityException: The function couldn't connect to the Amazon EFS file system with access point arn:aws:elasticfilesystem:us-east-2:123456789012:access-point/fsap-015cxmplb72b405fd. Check your network configuration and try again.*

Context: The function couldn't establish a connection to the function's file system with the NFS protocol (TCP port 2049). Check the security group and routing configuration for the VPC's subnets.

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Troubleshooting – Execution

Function execution errors can be caused by issues with your code, function configuration, downstream resources, or permissions.

Direct Invocation – Changes deployed to function directly from Lambda API, or with a client (e.g., AWS CLI).

Error Tracing – Errors will be shown directly in the response from lambda

Asynchronous invocation – Invoked via event source mapping or through another service.

Error Tracing – Errors can be found in logs, a dead-letter queue, or failed-event destination.

Status code – Function code or Lambda runtime errors will have a **200 OK** response.
Error in response indicated by **X-Amz-Function-Error** header.

400 and 500-series codes reserved for invocation errors.

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Execution Issues – Lambda

Execution takes too long

Example Issues	Context
Issue: *Function execution takes too long; longer than on local machine.*	Memory or processing power available to function can cause extended execution times. Configure the function to have additional memory. This increases both memory and CPU available to the function.

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Execution Issues – Lambda

Logs or traces don't appear

Example Issues

Issue: *Logs don't appear in CloudWatch Logs.*

Issue: *Traces don't appear in AWS X-Ray.*

Solution

Function needs permission to call CloudWatch Logs and X-Ray. Update **execution role** to grant permission.
Add the following managed policies to enable logs and tracing:

- AWSLambdaBasicExecutionRole
- AWSXRayDaemonWriteAccess

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Execution Issues – Lambda

The function returns before execution finishes

Example Issue

Issue: (Node.js) *Function returns before code finishes executing*

Solution

When you make a network call or perform another operation that requires waiting for a response, libraries return an object called a promise that tracks the progress of the operation in the background.

To wait for the promise to resolve into a response, use the await keyword. This blocks your handler code from executing until the promise is resolved into an object that contains the response. If you don't need to use the data from the response in your code, you can return the promise directly to the runtime.

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Troubleshooting – Networking

Network connectivity issues with functions' access to AWS services and the internet

Functions and VPC's

1. Functions run in an internal VPC w/ connectivity to AWS services and the internet.
2. Functions can be configured to connect to VPC within your account.
    1. YOU manage functions' internet access and network connectivity w/ VPC resrouces.

Possible issues

- VPC routing configuration
- Security Group rules
- AWS IAM Role permissions
- Network address translation
- Resource availability (IP addresses, network interfaces)

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Networking Issues – VPC

Function loses internet access or times out

Example Errors / Issues

Issue: *Your Lambda function loses internet access after connecting to a VPC.*

Error: *Error: connect ETIMEDOUT 176.32.98.189:443*

Error: *Error: Task timed out after 10.00 seconds*

Error: *ReadTimeoutError: Read timed out. (read timeout=15)*

Solution

All function outbound requests go through the VPC, when connected to one. To connect to the internet, configure your VPC to send outbound traffic from the function's subnet to a NAT gateway in a public subnet.

If some of your TCP connections are timing out, this may be due to packet fragmentation. Lambda functions cannot handle incoming fragmented TCP requests, since Lambda does not support IP fragmentation for TCP or ICMP.

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Networking Issues – VPC

Function needs access to AWS services without using the internet

Error: *EFSMountFailureException: The function could not mount the EFS file system with access point arn:aws:elasticfilesystem:us-east-2:123456789012:access-point/fsap-015cxmplb72b405fd.*

Context: Check the function's permissions, and confirm that its file system and access point exist and are ready for use.

Elastic network interface limit reached

Error: *ENILimitReachedException: The elastic network interface limit was reached for the function's VPC.*

Context: When you connect a Lambda function to a VPC, Lambda creates an elastic network interface for each combination of subnet and security group attached to the function. The default service quota is 250 network interfaces per VPC. To request a quota increase, use the Service Quotas console.

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Serverless Storage

aws training and certification

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

Amazon RDS (Relational Database Service)

Set up, operate, and scale a fully managed RDS with just a few clicks

Amazon Aurora PostgreSQL-Compatible Edition

Amazon Aurora MySQL-Compatible Edition

MySQL

PostgreSQL

MariaDB

Microsoft SQL Server

ORACLE

Easy to administer

Easily deploy and maintain hardware, OS, and DB software, with built-in monitoring

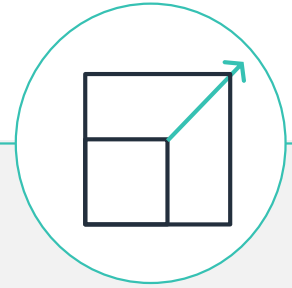
Secure and compliant

Data encryption at rest and in transit, with industry compliance and assurance programs

Available and durable

Automatic Multi-AZ data replication, with automated backup, snapshots, and failover

Performant and scalable



Scale compute and storage with a few clicks, plus minimal downtime for your application

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon RDS Multi-AZ

Highly available, durable relational databases deployed across up to three AZs

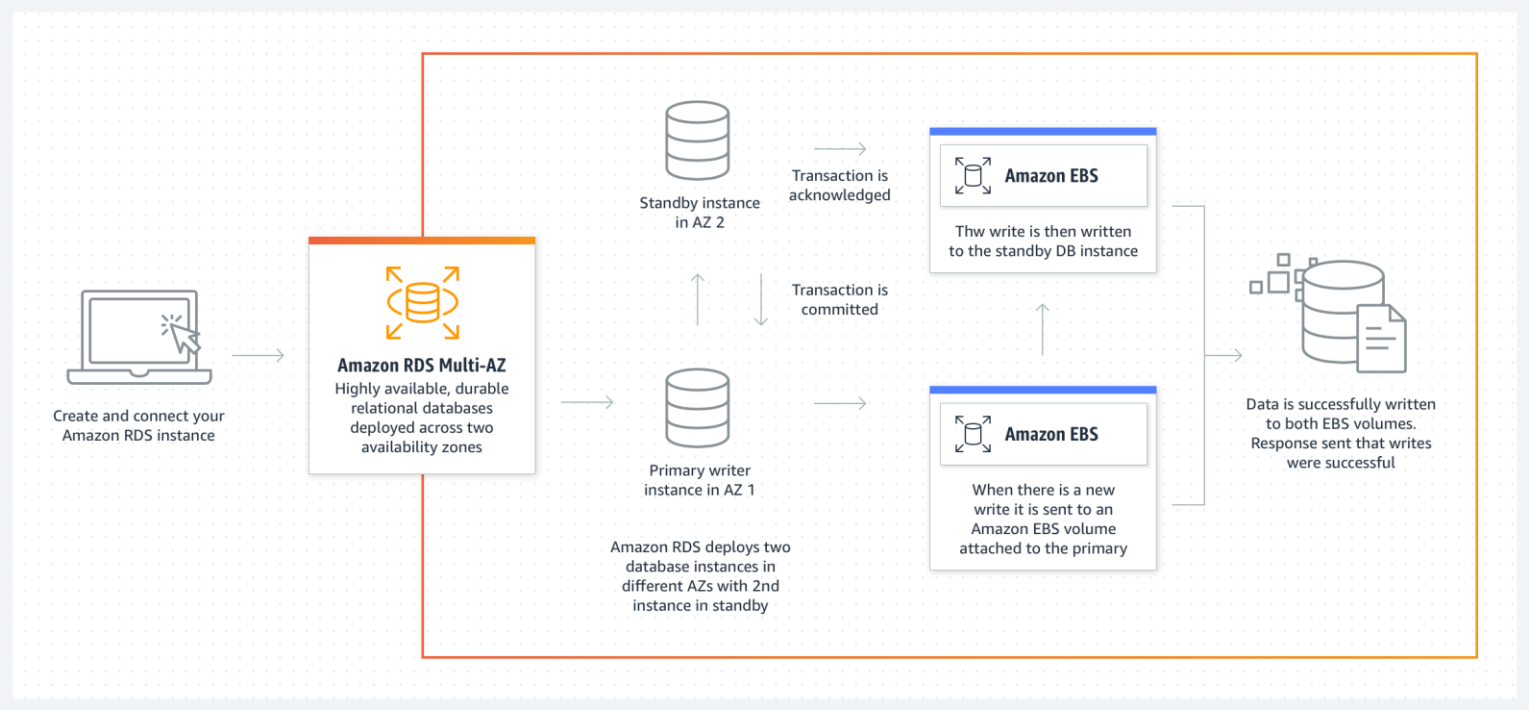
Multi-AZ with one standby

Capabilities

- Automatic fail over
- Protect database performance
- Enhance durability
- Increase availability

How it works

In an Amazon RDS Multi-AZ deployment, Amazon RDS automatically creates a primary database (DB) instance and synchronously replicates the data to an instance in a different AZ. When it detects a failure, Amazon RDS automatically fails over to a standby instance without manual intervention.



aws training and certification

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon RDS Multi-AZ (Cont'd)

Highly available, durable relational databases deployed across up to three Azs

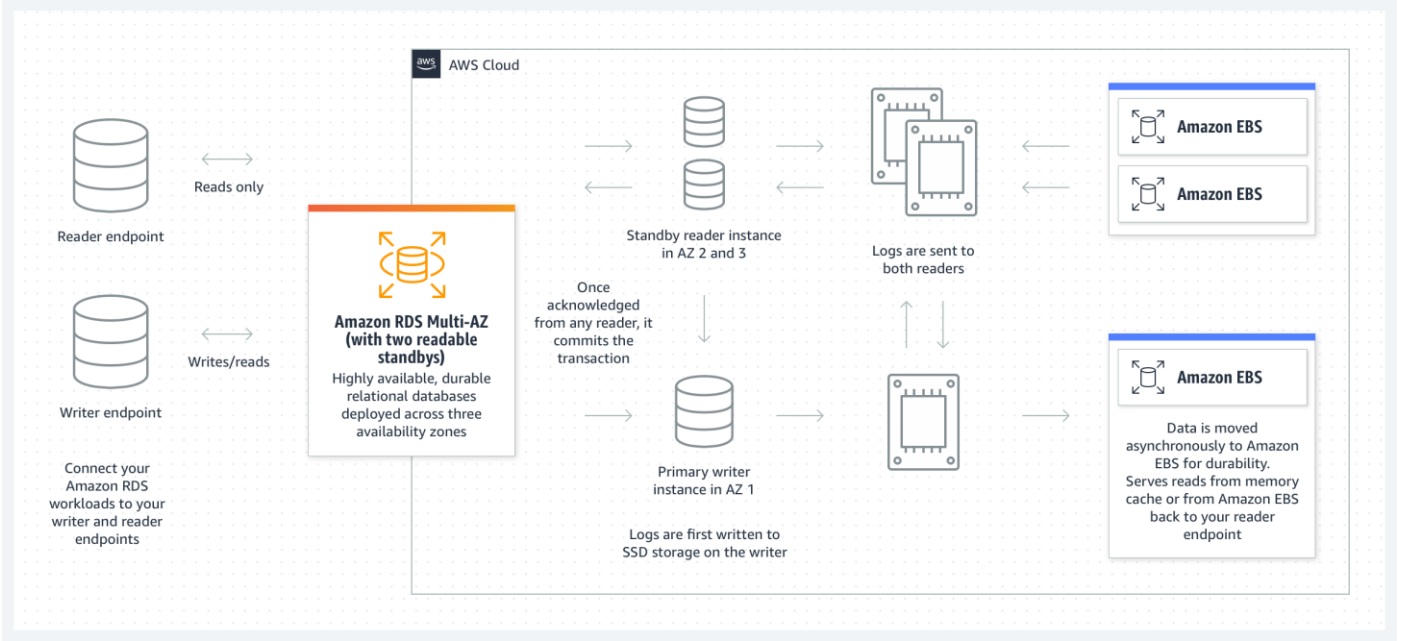
Multi-AZ with two readable standbys

Capabilities

- Automatic fail over typically under 35 seconds
- Maximize performance and scalability by splitting read and write
- Gain up to 2x faster transaction commit latency
- Increase read capacity

How it works

Deploy highly available, durable MySQL or PostgreSQL databases in three AZs. Gain automatic failovers in typically under 35 seconds, up to 2x faster transaction commit latency compared to Amazon RDS Multi-AZ with one standby, additional read capacity, and storage backed by AWS Graviton2–based instances.



aws training and certification

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon RDS Read Replicas

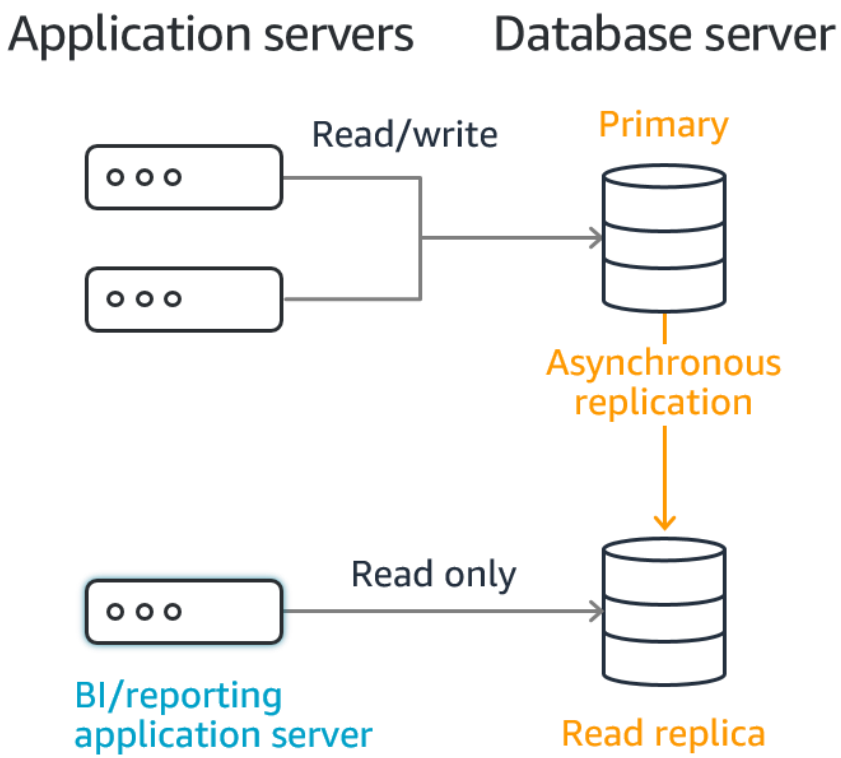
Highly available, durable relational databases deployed across up to three AZs

Capabilities

- Enhanced performance
- Increased availability
- Designed for security

How it works

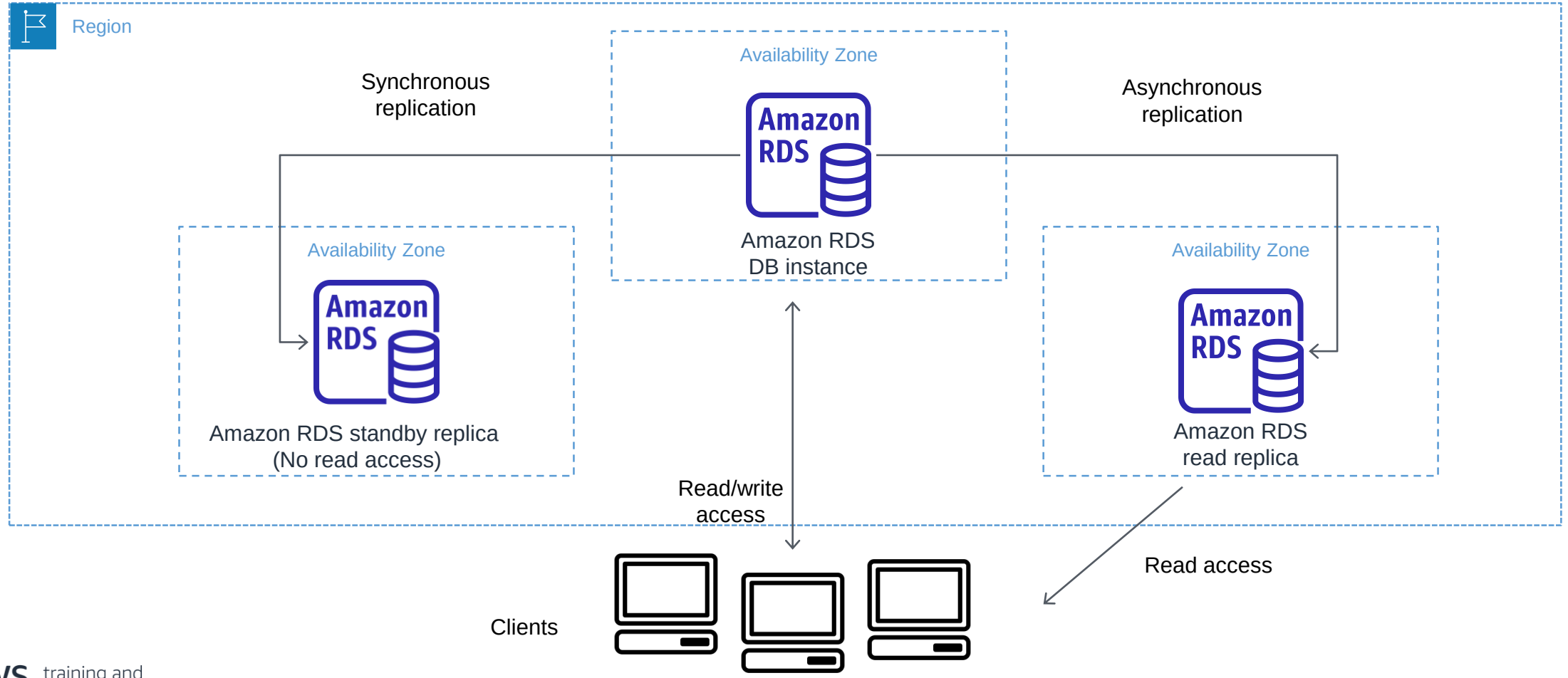
Easy to elastically scale out beyond capacity constraints of a single DB instance for read-heavy database workloads. Create one or more replicas of a given source DB Instance and serve high-volume application read traffic from multiple copies of your data, thereby increasing aggregate read throughput. Add read replicas to existing DB Instances directly from the console.



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Read Replicas + Multi-AZ

Combining services for high availability



aws training and certification

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon Aurora

MySQL and PostgreSQL compatible relational database – built for the cloud

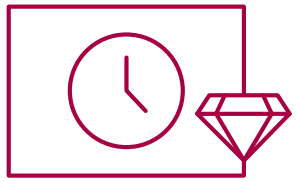
Aurora Resiliency

Aurora maintains six copies of data, two copies in each Availability Zone (3 AZ's). This protects against AZ +1 failures.

Primary and replicas all point to the same underlying storage

Exam Tip

Questions that speak to migrations of MySQL / PostgreSQL databases to the cloud that require high availability, resiliency, or multiple AZ's → Aurora should be your first thought!

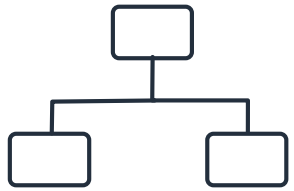


Availability and durability

Fault-tolerant, self-healing storage; six copies of data across three Availability Zones; continuous backup to Amazon S3

Performance and scalability

5x throughput of standard MySQL and 3x of standard PostgreSQL; scale-out up to 15 read replicas



Fully managed

Managed by RDS: no server provisioning, software patching, setup, configuration, or backups

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon Aurora High Availability

Amazon Aurora architecture involves separation of storage and compute.

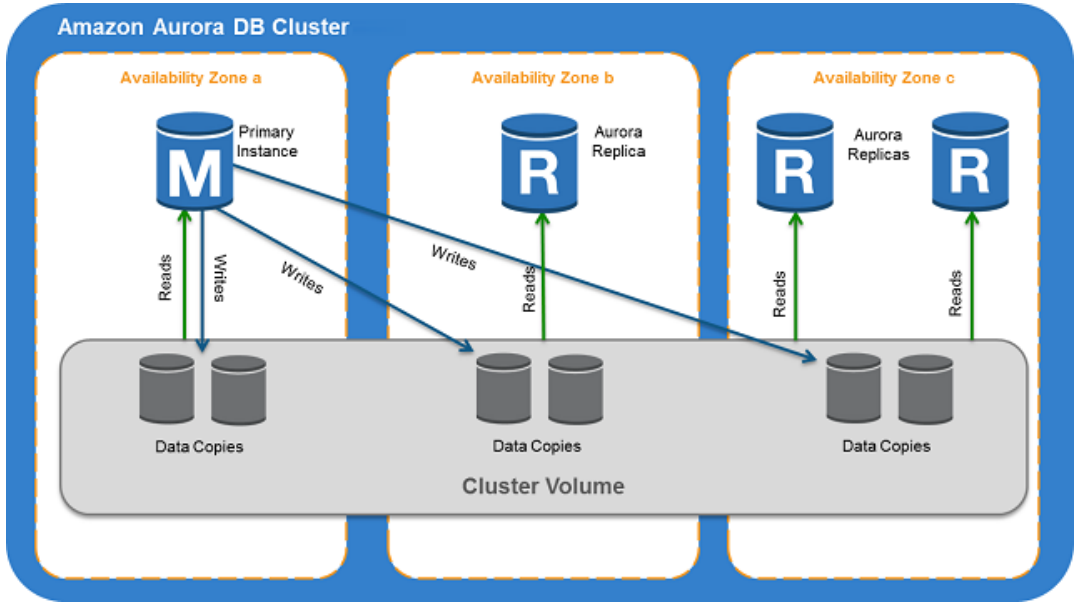
Aurora includes some high availability features that apply to data in your DB cluster

HA for Aurora Data

Aurora stores copies of the data in a DB cluster across multiple Availability Zones in a single AWS Region. Aurora stores these copies regardless of whether the instances in the DB cluster span multiple Availability Zones. For more information on Aurora, see Managing an Amazon Aurora DB cluster.

HA for Aurora DB Instances

For a cluster using single-master replication, after you create the primary instance, you can create up to 15 read-only Aurora Replicas. The Aurora Replicas are also known as reader instances.



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon Aurora Global Databases

Amazon Aurora global databases span multiple AWS Regions

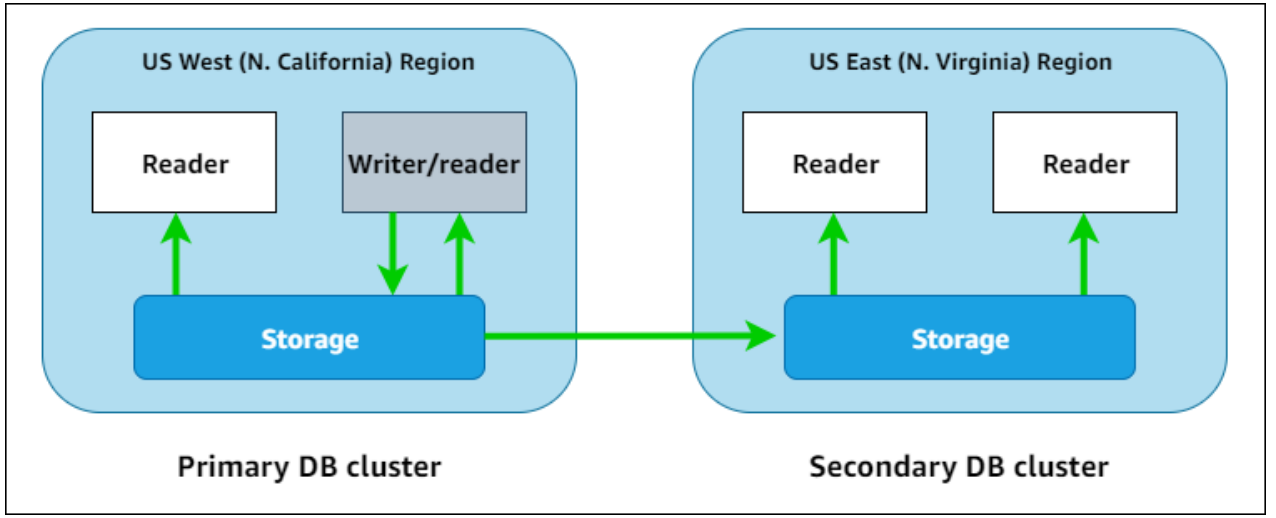
Aurora includes some high availability features that apply to data in your DB cluster

Overview

By using an Amazon Aurora global database, you can run your globally distributed applications using a single Aurora database that spans multiple AWS Regions.

Advantages

- Global reads with local latency
- Scalable secondary Aurora DB clusters
- Fast replication from primary to secondary Aurora DB clusters
- Recovery from Region-wide outages



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon DynamoDB

Fast and Flexible **Key-Value** database service for any scale

DynamoDB

A fully managed – multiple master, multiple region database solution for high performance, globally distributed applications

Disaster proof solution with multi-region redundancy

Exam Tip

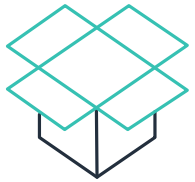
Key-Value is the dead giveaway for DynamoDB. Any question that references this on the exam is cluing you towards a solution that is built around DynamoDB

Serverless architecture

No hardware provisioning, software patching, or upgrades; scales up or down automatically; continuously backs up your data

Enterprise security

Encrypts all data by default and fully integrates with AWS Identity and Access Management for robust security



Global replication

Build global applications with fast access to local data by easily replicating tables across multiple AWS Regions

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon DynamoDB Standard-IA

Reduce costs by up to 60% for tables storing infrequently accessed data

Benefits

Class helps you reduce your DynamoDB costs by up to 60% for tables that store data that is infrequently accessed. You can switch between DynamoDB Standard and DynamoDB Standard-IA table classes with no impact on table performance, durability, or availability and without changing your application code.

Use Cases

Require long-term storage of data that is infrequently accessed, such as application logs, old social media posts, ecommerce order history, and past gaming achievements.

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon Quantum Ledger Database (Amazon QLDB)

Maintain an immutable, cryptographically verifiable log of data changes

How it Works

Amazon Quantum Ledger Database (QLDB) is a fully managed ledger database that provides a transparent, immutable, and cryptographically verifiable transaction log.

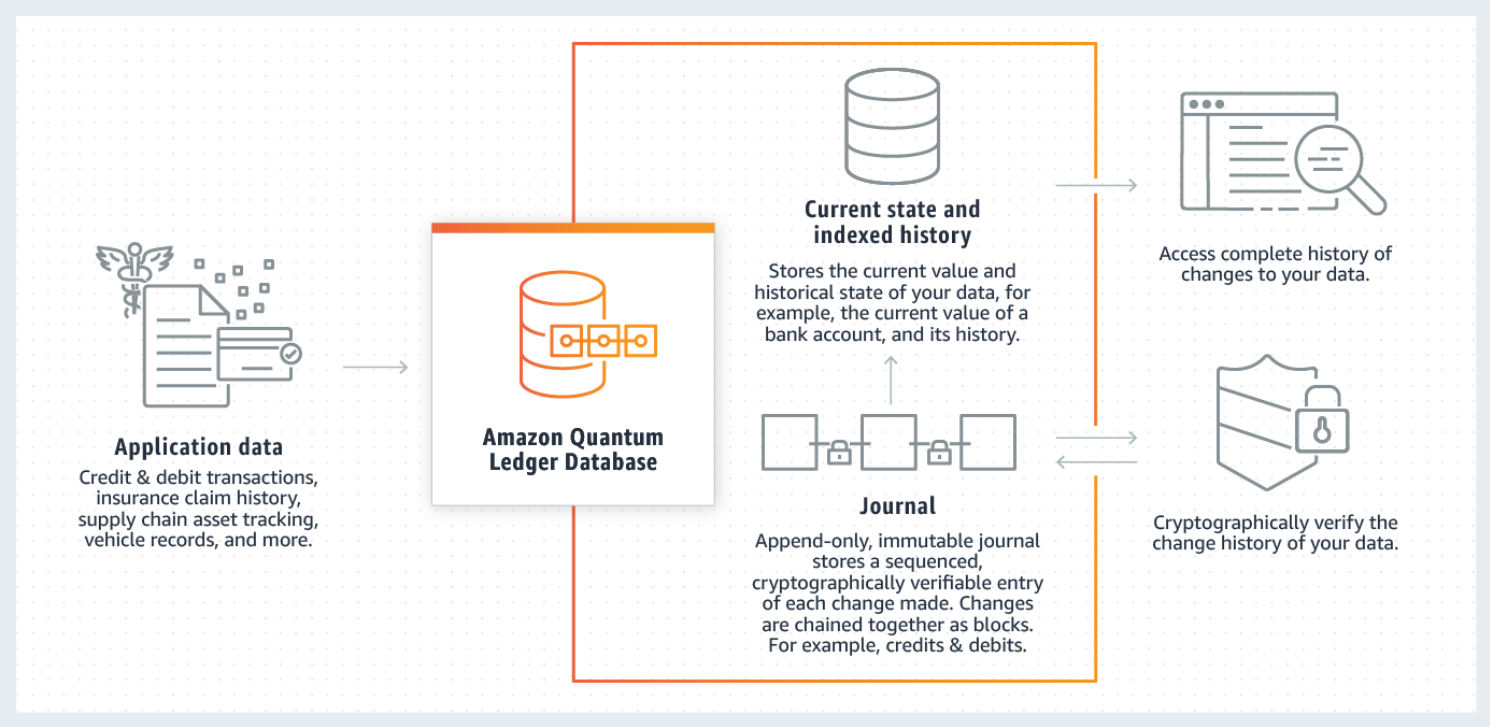
Use Cases

Store financial transactions

Reconcile supply chain systems

Maintain claims history

Centralize digital records



aws training and certification

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Comprehensive Example

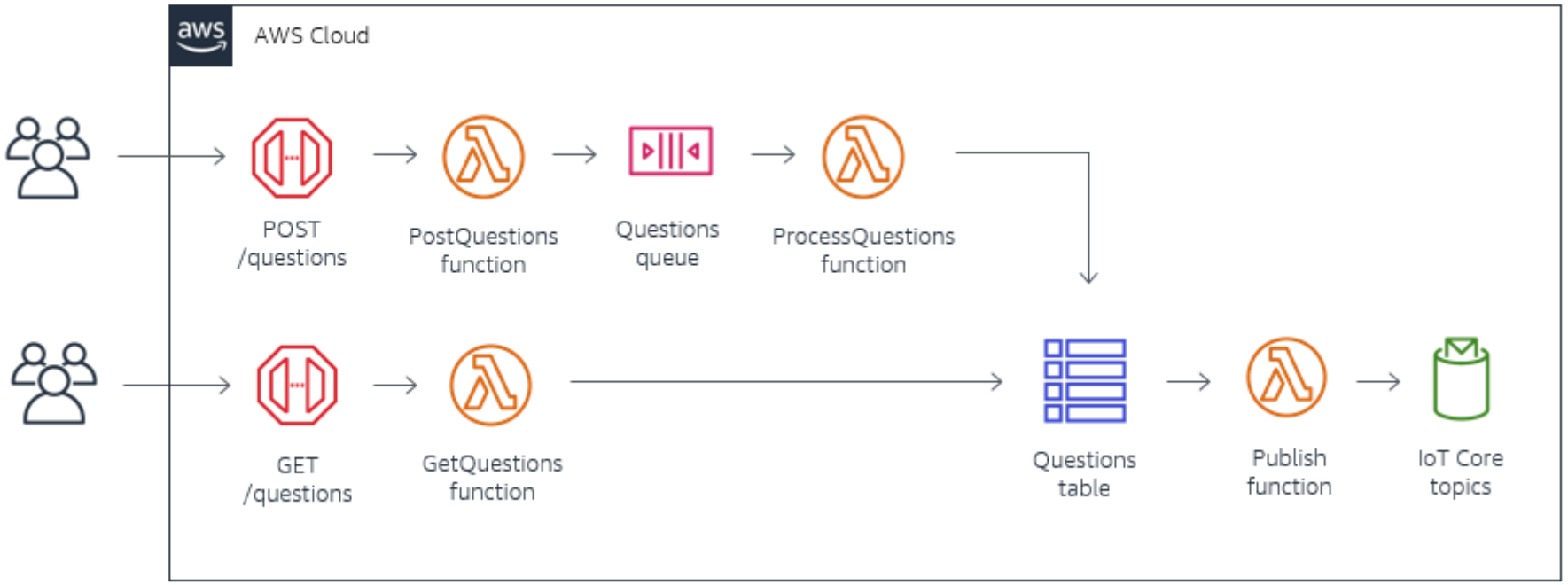
aws training and certification

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

Serverless Webapp

How it Works

Calls through API Gateway trigger Lambda functions. One function will queue questions in SQS for a processing function. Questions are loaded into a scalable DynamoDB table.



aws training and certification

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Security

aws training and certification

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

Amazon Macie

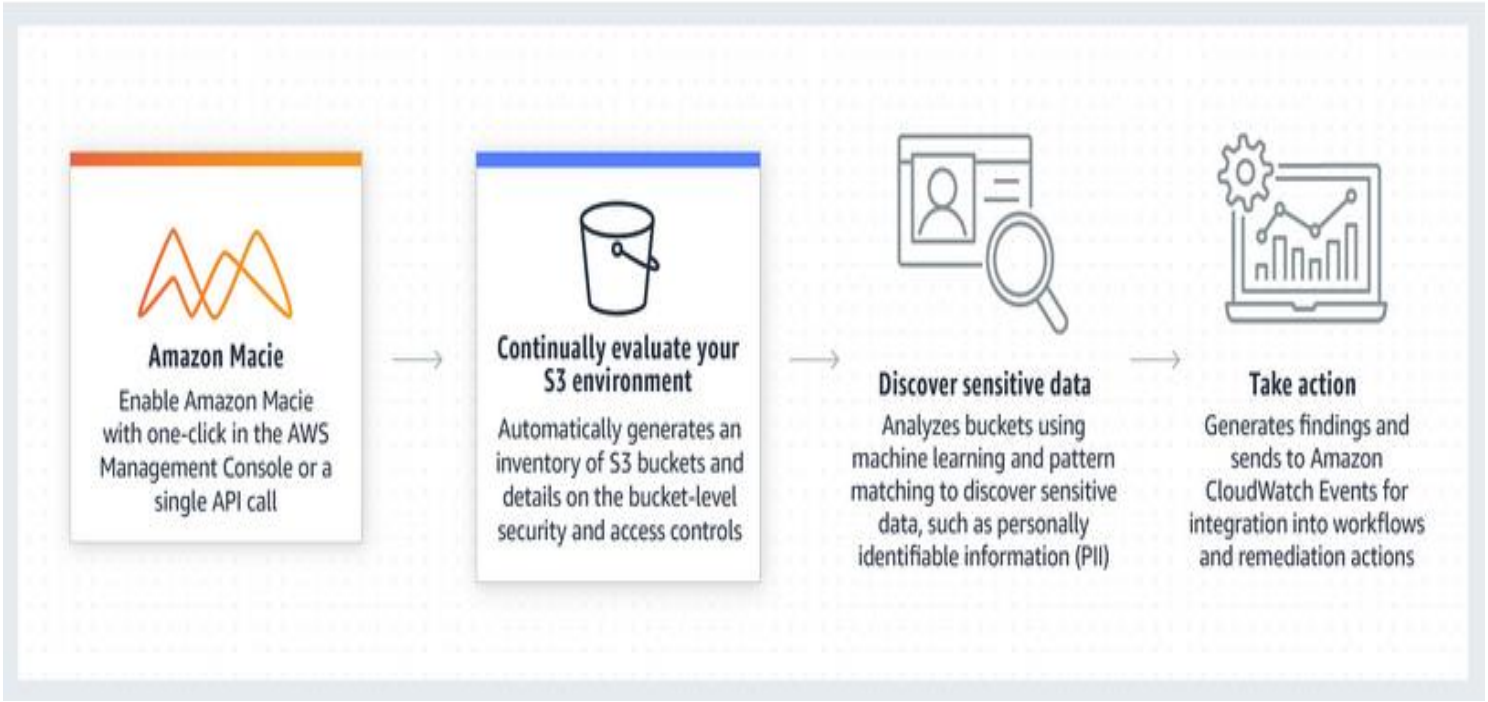
Amazon Macie continually **evaluates your Amazon S3 environment** and provides an S3 resource summary across all of your accounts.

Purpose

Amazon Macie uses machine learning and pattern matching to cost efficiently discover sensitive data at scale. Macie automatically detects a large and growing list of sensitive data types, including **personally identifiable information (PII)** such as names, addresses, and credit card numbers. It also gives you constant visibility of the data security and data privacy of **your data stored in Amazon S3.**

Easy to Deploy

With **one-click** in the AWS Management Console or a single API call, you can enable Amazon Macie in a single account. With a **few more clicks** in the console, you can enable Macie **across multiple accounts**.



Amazon Inspector

An automated vulnerability management service that continually scans **Amazon Elastic Compute Cloud (EC2) and container workloads** for software vulnerabilities and unintended network exposure.

Use Cases

Use up-to-date common vulnerabilities and exposures (CVE) information combined with factors such as network accessibility to create context-based risk scores that help you prioritize and address vulnerable resources.

Support compliance requirements and best practices for NIST CSF, PCI DSS, and other regulations with Amazon Inspector scans.

Identify zero-day vulnerabilities sooner

Key Concept

Amazon Inspector is an automated vulnerability management service that continually **scans Amazon Elastic Compute Cloud (EC2) and container workloads** for software vulnerabilities and unintended network exposure

AWS Certificate Manager (ACM)

Provision, manage, and deploy public and private SSL/TLS certificates

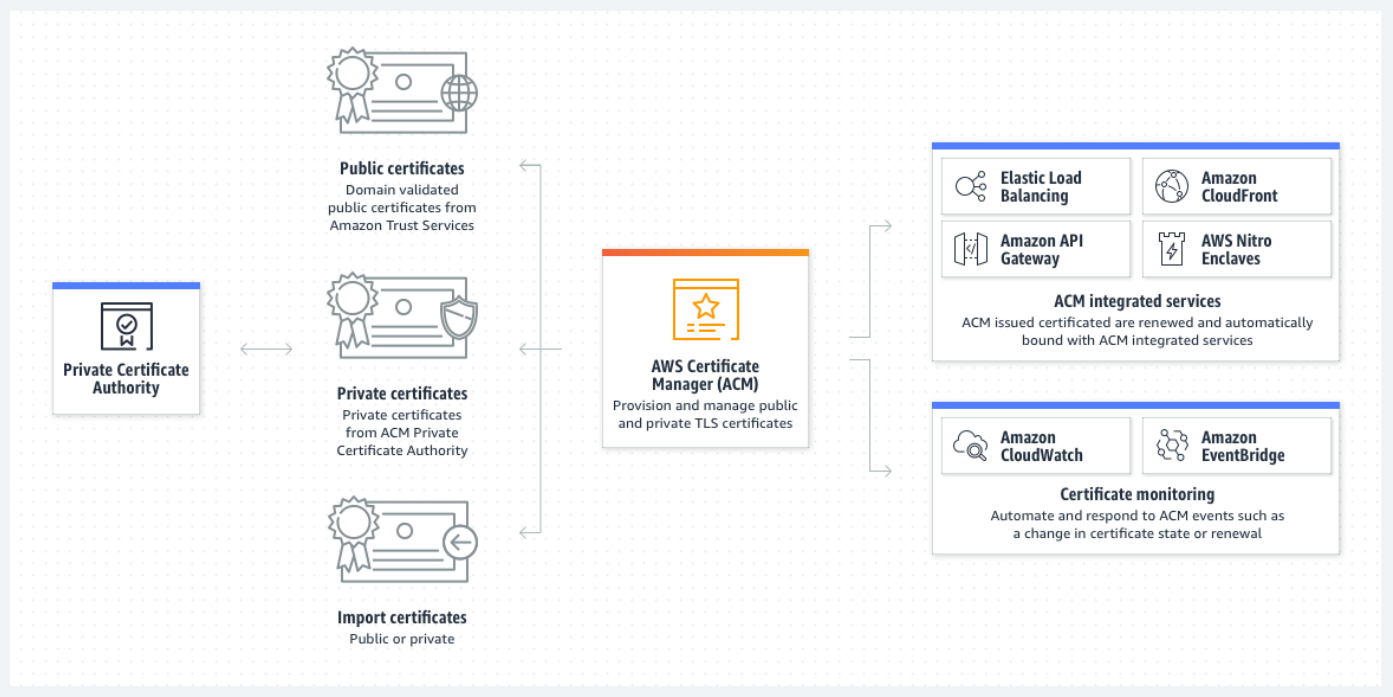
WS Certificate Manager removes the time-consuming manual process of purchasing, uploading, and renewing SSL/TLS certificates.

Manage

Request a certificate, deploy on an ACM-integrated AWS resource, and let ACM handle certificate renewals. Create private certificates for internal resources, and centrally manage certificate lifecycles.

Offload

ACM offloads the time consuming, error-prone process of acquiring an SSL/TLS certificate, and manages the renewal process. Public certificates for ACM-integrated services are free.



aws training and certification

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Directory Service

Managed Microsoft Active Directory in AWS

Enables your directory-aware workloads and AWS resources to use managed Active Directory (AD) in AWS

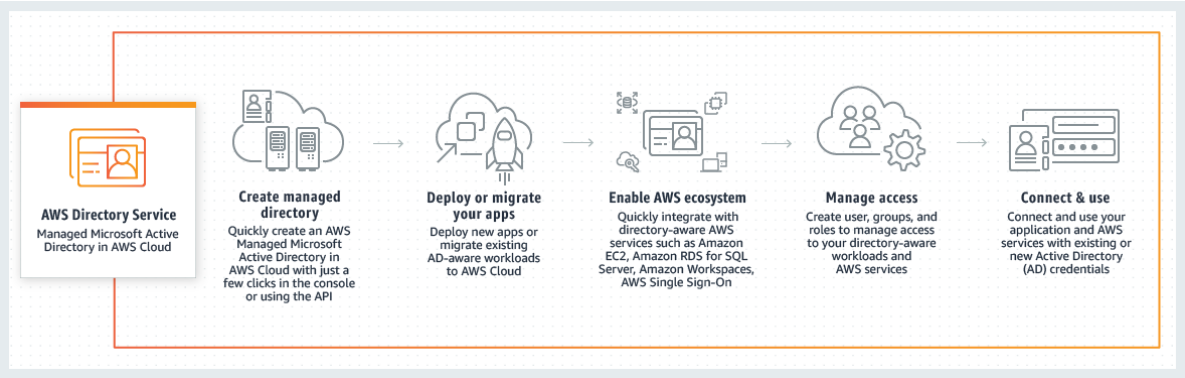
Managed Service

Fully managed service, provides directories w/ info about your organization. Reduce management tasks, without needing to build your own complex, highly-available directory.

Uses

Run directories in AWS cloud, or connect AWS resources to an existing on-prem Microsoft AD. Use to manage users, groups, SSO for applications / services, or create and apply join policies.

Use existing corporate credentials to administer AWS resources via AWS IAM role-based access.



aws training and certification

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon GuardDuty

Intelligent Threat Detection to Protect Your AWS Accounts and Workloads

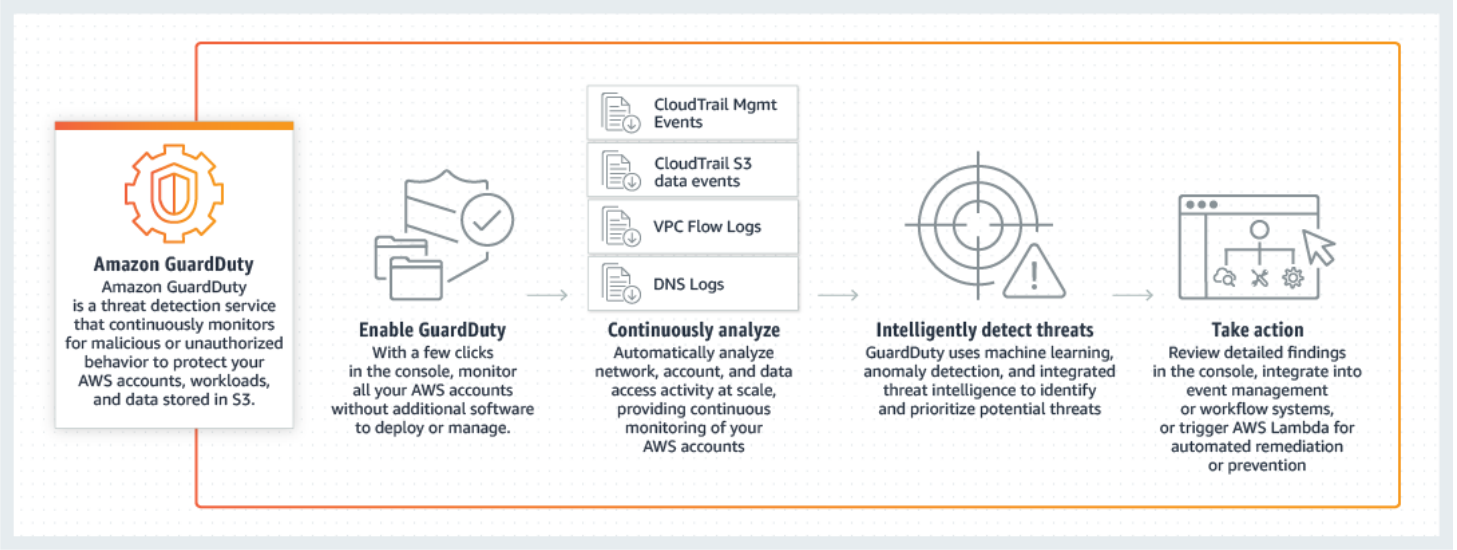
Threat detection service that continuously monitors for malicious activity and unauthorized behavior to protect your accounts, workloads, and S3 data

Threat Identification

A software agent that you can install on the EC2 instances that are included in the assessment target. The agent collects a wide set of configuration data (telemetry)

Automation

A rules package corresponds to a security goal that you might have. You can specify your security goal by selecting the appropriate rules package when you create an Amazon Inspector assessment template



aws training and certification

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon GuardDuty – Findings Format

ThreatPurpose:ResourceTypeAffected/ThreatFamilyName.DetectionMechanism!Artifact

ThreatPurpose - describes the primary purpose of a threat, an attack type or a stage of a potential attack.

ResourceTypeAffected - describes which AWS resource type is identified in this finding as the potential target of an adversary. Currently, GuardDuty can generate findings for EC2, S3, IAM, and EKS resources.

ThreatFamilyName - describes the overall threat or potential malicious activity that GuardDuty is detecting.

DetectionMechanism - describes the method in which GuardDuty detected the finding. This can be used to indicate a variation on a common finding type or a finding that GuardDuty used a specific mechanism to detect

.Custom indicates that GuardDuty detected the finding based on your custom threat lists

.Reputation indicates that GuardDuty detected the finding using a domain reputation score model.

Artifact - describes a specific resource that is owned by a tool that is used in the malicious activity.

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon GuardDuty – Threat Purposes

ThreatPurpose:ResourceTypeAffected/ThreatFamilyName.DetectionMechanism!Artifact

- Backdoor
- Behavior
- CredentialAccess
- Cryptocurrency
- DefenseEvasion
- Discovery
- Execution
- Exfiltration
- Impact
- InitialAccess
- Pentest
- Persistence
- Policy
- PrilegeEscalation
- Recon
- Stealth
- Trojan
- UnauthorizedAccess

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Key Management Service

Easily create and control the keys used to encrypt or digitally sign your data

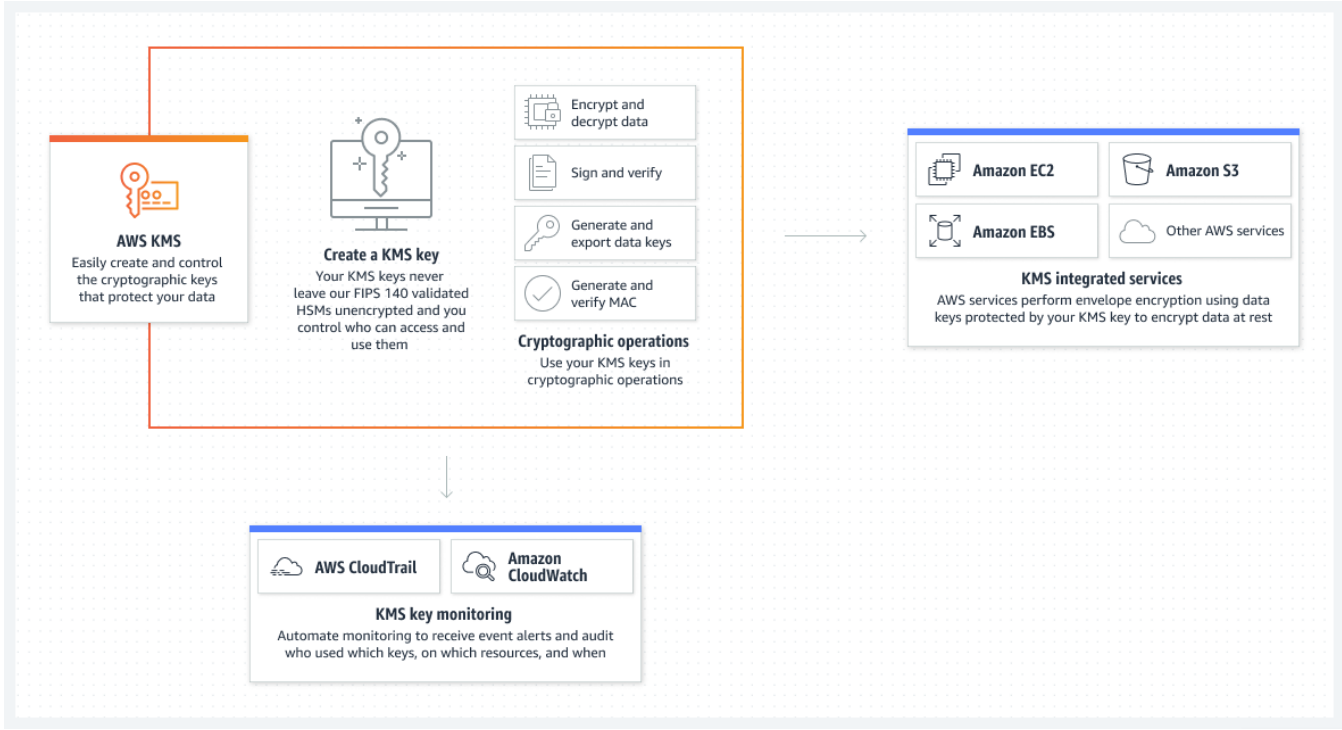
Secure and resilient service; uses hardware security modules to protect your keys.

Centralized

AWS KMS provides centralized control over lifecycle and permissions of your keys. Key can be generated by AWS KMS, imported from your own key management infrastructure, or those stored in your AWS CloudHSM cluster.

Scalable, Durable, HA

Fully managed service, automatically scales with use. Keys created cannot be exported from the service, so AWS takes responsibility for their durability (11 9's). Most AWS services rely on AWS KMS for encryption and decryption, so it is architected for a level of availability that can support the rest of AWS.



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Secrets Manager

Easily rotate, manage, and retrieve database credentials, API keys, and other secrets through their lifecycle

Summary

Secrets encrypted at rest for secure storage. Rotated on demand or by schedule. Secrets may be automatically replicated across multiple AWS Regions. Programmatic retrieval with audit and monitoring capabilities.

aws training and certification

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS WAF

Protect your web applications from common web exploits

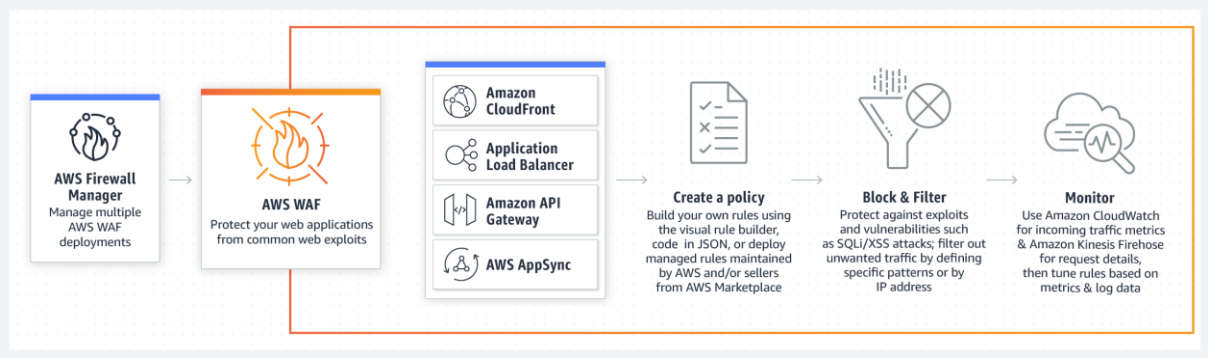
Helps protect web applications or APIs against common web exploits and bots that may affect availability, compromise security, or consume excessive resources.

Protection

Customize rules to filter out specific traffic patterns. Create security rules to block common attack patterns (SQL injection, XSS). Full-featured API to automate creation, deployment, and maintenance of security rules.

Deployment

- On Amazon CloudFront as part of CDN
- On Application Load Balancer fronting web or origin servers running on EC2
- Amazon API Gateway for REST APIs
- AWS AppSync for GraphQL APIs



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS WAF – Bot Control

Bot Control managed rule group blocks or rate-limits pervasive bots

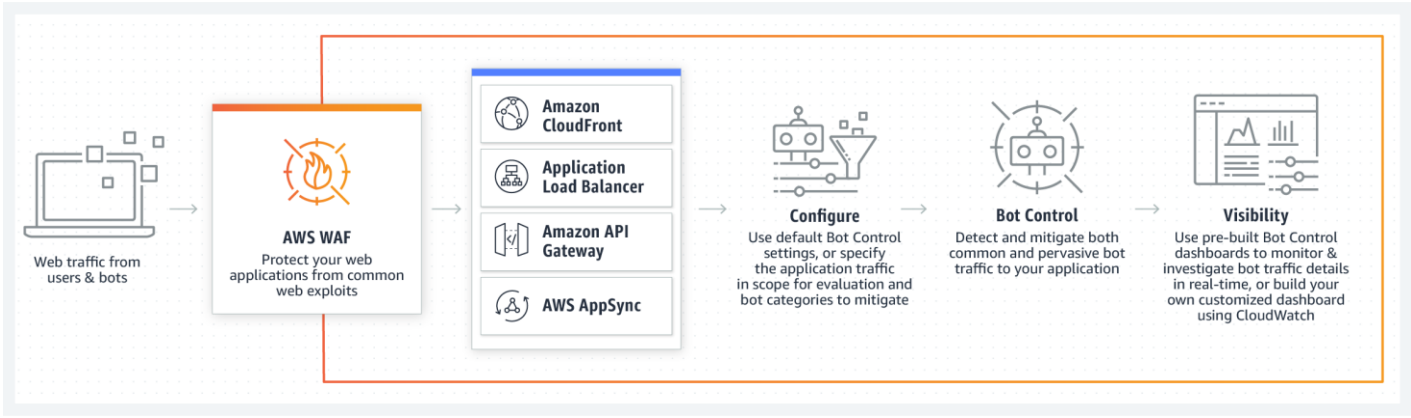
Visibility and control over common and pervasive bot traffic that can consume excess resources, skew metrics, cause downtime, or perform other undesired activities.

Monitor

Monitor bot traffic activity with dashboards providing detailed, real-time visibility into bot categories, identities, and other both traffic details.

Use Cases

- Block unwanted traffic at the network edge
- Protect intellectual property
- Deliver alternate content in response to bot traffic



aws training and certification

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Config & CloudFormation

aws training and certification

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Config

Track resource inventory & changes

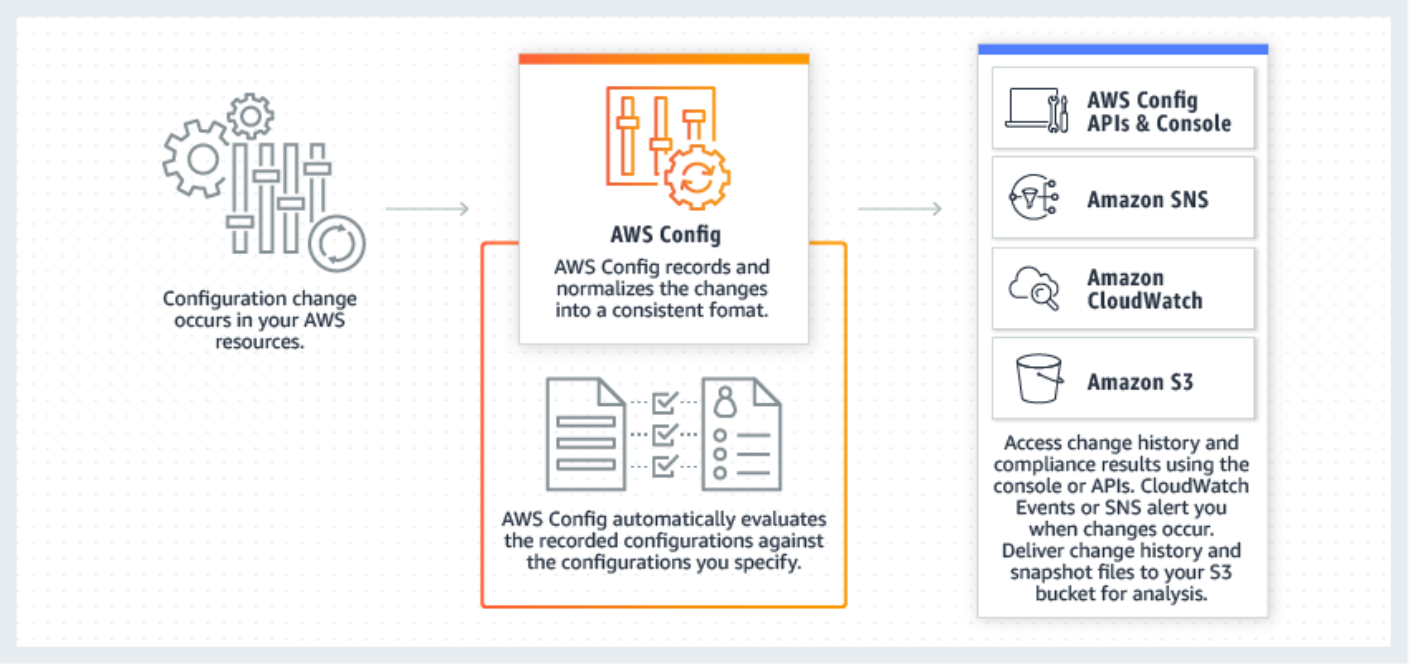
Enables you to assess, audit, and evaluate the configurations of your AWS resources. Continuously monitors and records your AWS resource configurations and allows you to automate the evaluation of recorded configurations against desired configurations

Discovery

AWS Config will discover resources that exist in your account, record their current configuration, and capture any changes to these configurations

Change Management

When your resources are created, updated, or deleted, AWS Config streams these configuration changes to Amazon Simple Notification Service (SNS), so that you are notified of all the configuration changes



aws training and certification

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS CloudFormation

Speed up cloud provisioning with infrastructure as code

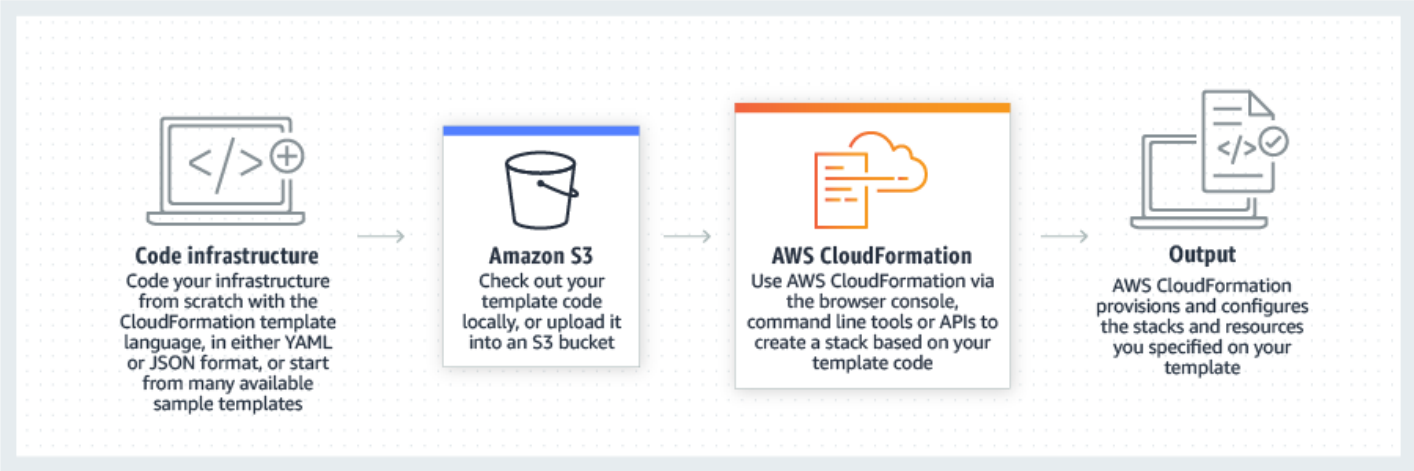
A CloudFormation template describes your desired resources and their dependencies so you can launch and configure them together as a stack

How it Works

AWS CloudFormation lets you model, provision, and manage AWS and third-party resources by treating infrastructure as code.

Use Cases

Common use cases for CloudFormation include managing infrastructure with DevOps through automated, test and deploy infrastructure templates and scaling production stacks at scale.



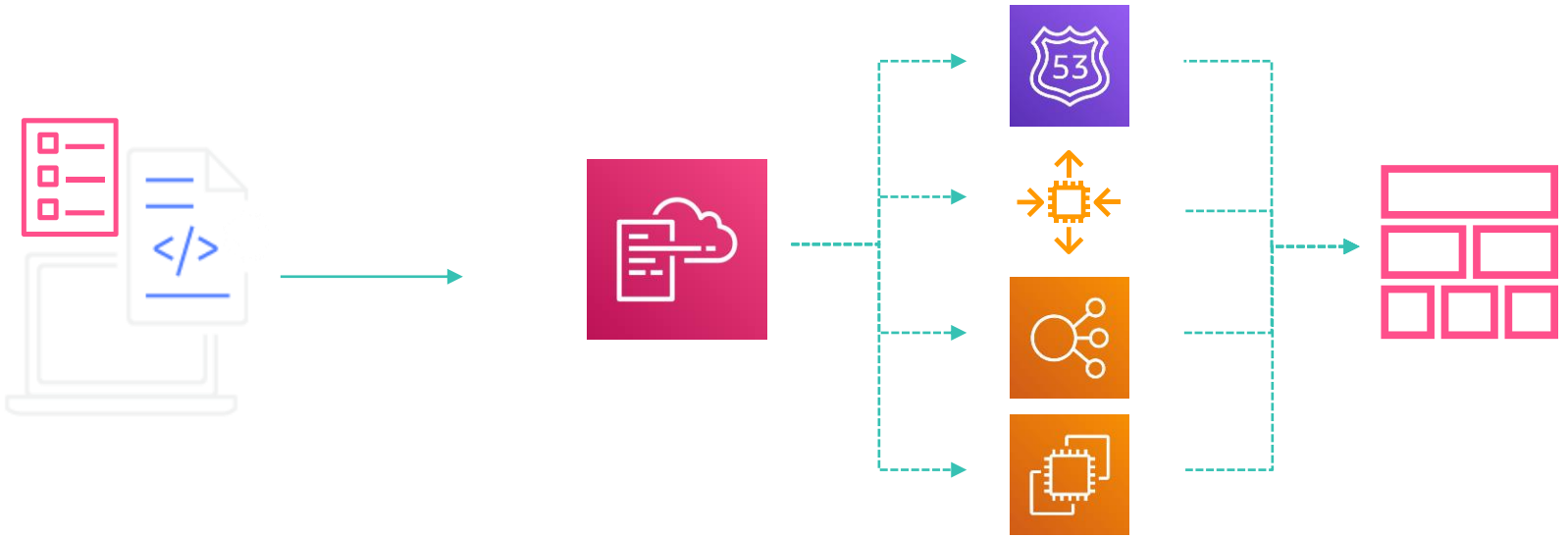
aws training and certification

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS CloudFormation

Service Overview

- FREE – you only pay for resources
- All regions
- APIs are called in parallel
- Manages **dependencies / relationships**



Code in YAML or JSON directly or use sample templates

Upload local files or from an S3 bucket

Create stack using API via AWS CloudFormation

Stacks and resources are provisioned as a running environment

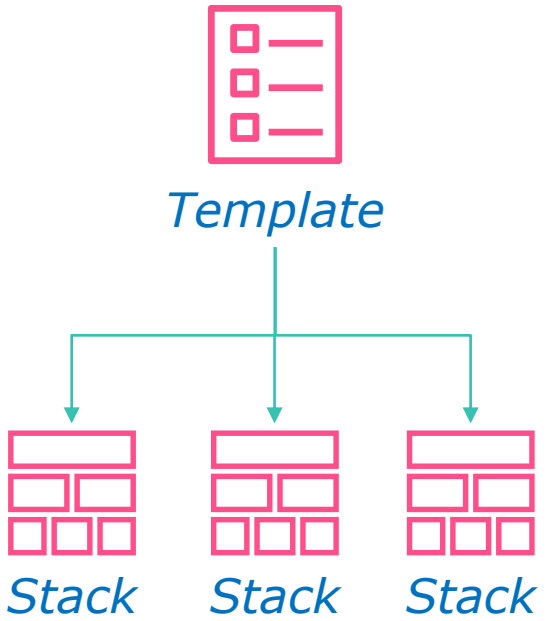
aws training and certification

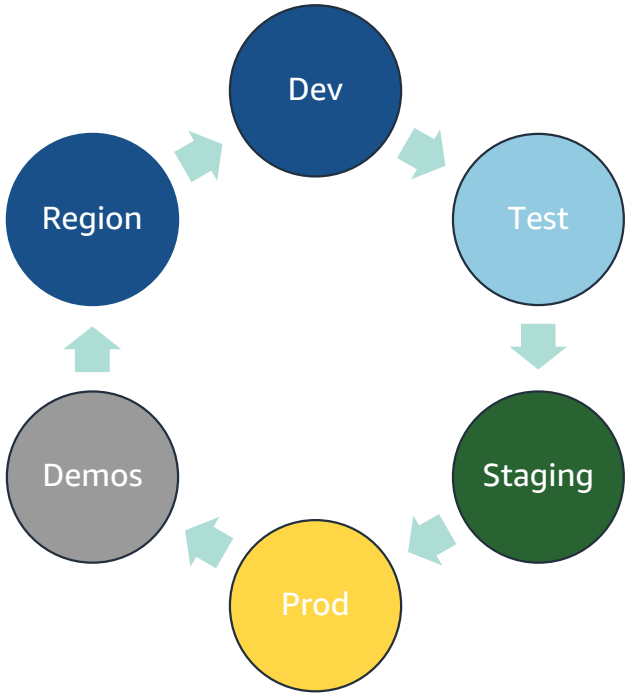
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Infrastructure as code

What it means

- Single source of truth to deploy the whole stack
- Infrastructure that you can replicate, re-deploy, and re-purpose
- Control versioning on your infrastructure and your application together
- Service rolls back to the last good state on failures
- Build your infrastructure and run it through your CI/CD pipeline





© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

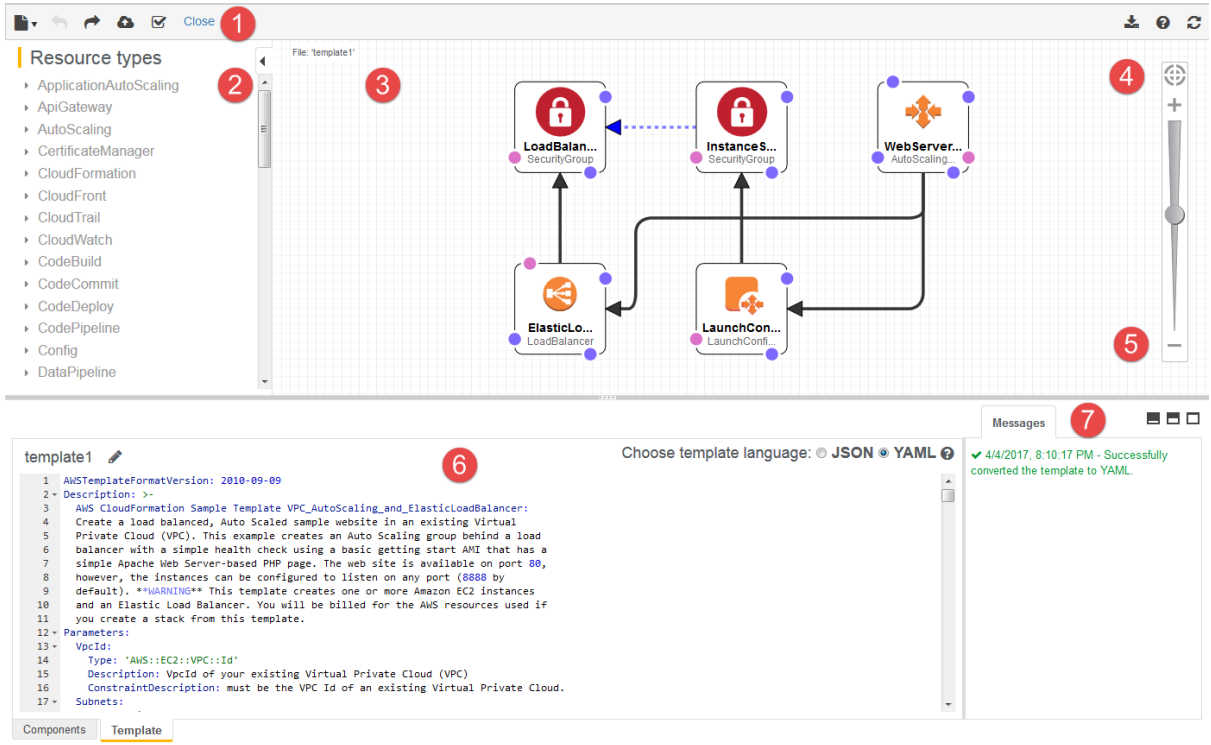
AWS CloudFormation Templates

Create templates to describe your AWS resources and their properties.

Templates

A CloudFormation template is a JSON or YAML formatted text file. CloudFormation uses these templates as blueprints for building your AWS resources. For example, in a template, you can describe an Amazon EC2 instance, such as the instance type, the AMI ID, block device mappings, and its Amazon EC2 key pair name.

1. An optional list of template parameters (input values supplied at stack creation time)
2. An optional list of output values (e.g., the complete URL to a web application)
3. An optional list of data tables used to look up static configuration values (e.g., AMI names)
4. The list of AWS resources and their configuration values
5. A template file format version number



aws training and certification

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS CloudFormation Stack

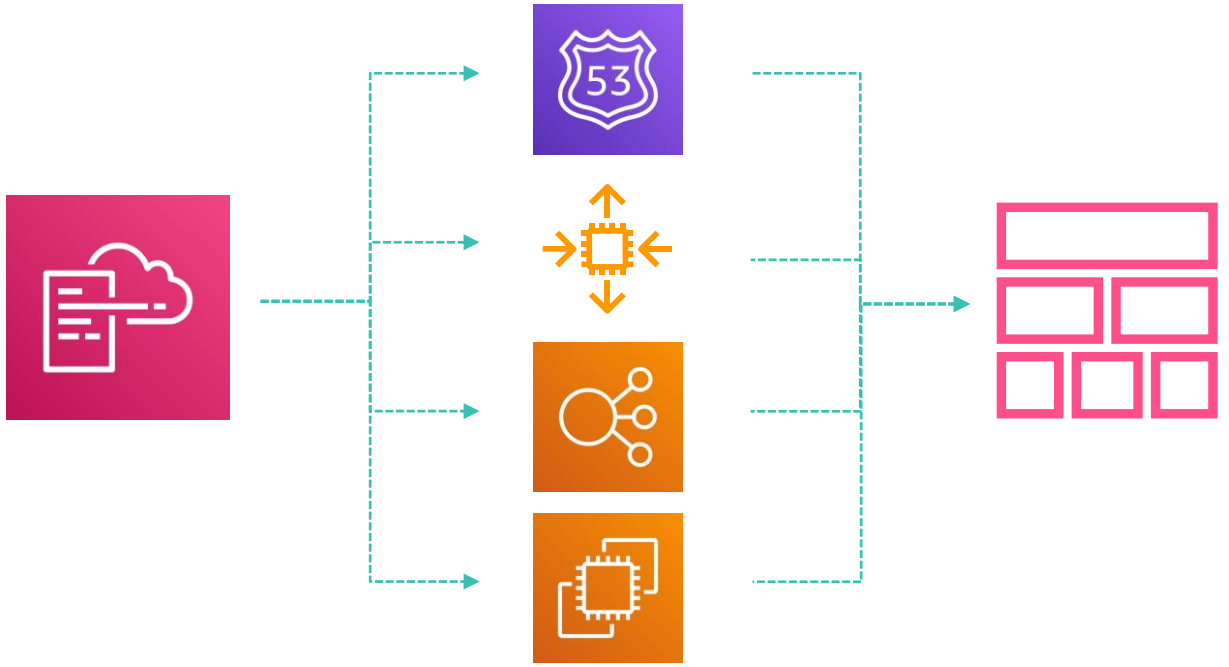
Stacks

A stack is a collection of AWS resources that you can manage as a single unit, or a template

AWS CloudFormation ensures all stack resources are created or deleted as appropriate

You can work with stacks by using the AWS CloudFormation console, API, or AWS CLI.

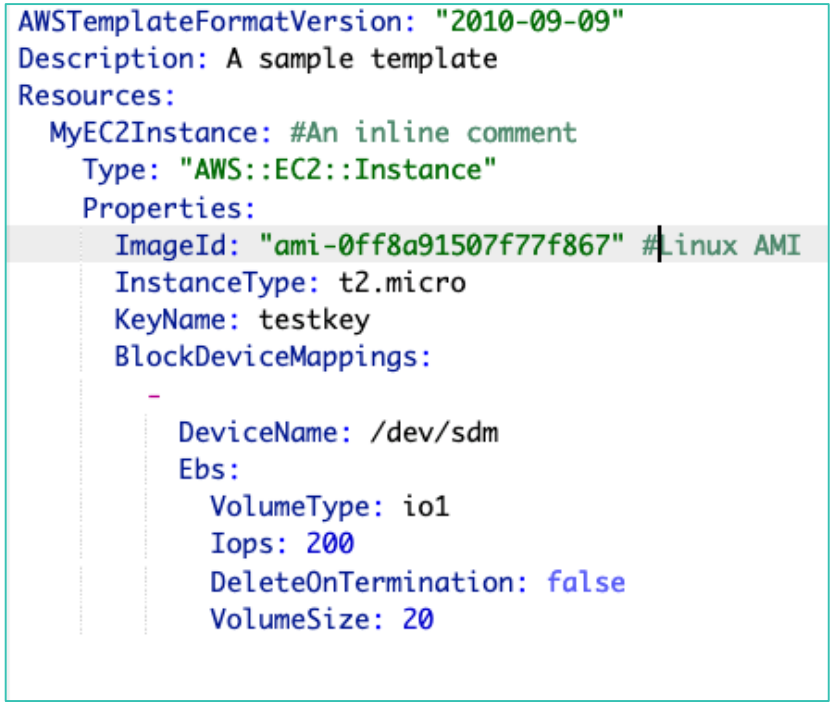
Nested stacks are stacks created as part of other stacks. You create a nested stack within another stack by using the AWS::CloudFormation::Stack resource



aws training and certification

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

CloudFormation Template Syntax



```yaml
AWSTemplateFormatVersion: "2010-09-09"
Description: A sample template
Resources:
  MyEC2Instance: #An inline comment
    Type: "AWS::EC2::Instance"
    Properties:
      ImageId: "ami-0ff8a91507f77f867" #Linux AMI
      InstanceType: t2.micro
      KeyName: testkey
      BlockDeviceMappings:
        -
          DeviceName: /dev/sdm
          Ebs:
            VolumeType: io1
            Iops: 200
            DeleteOnTermination: false
            VolumeSize: 20
```

```json
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "A sample template",
    "Resources": {
        "MyEC2Instance": {
            "Type": "AWS::EC2::Instance",
            "Properties": {
                "ImageId": "ami-0ff8a91507f77f867",
                "InstanceType": "t2.micro",
                "KeyName": "testkey",
                "BlockDeviceMappings": [
                    {
                        "DeviceName": "/dev/sdm",
                        "Ebs": {
                            "VolumeType": "io1",
                            "Iops": 200,
                            "DeleteOnTermination": false,
                            "VolumeSize": 20
                        }
                    }
                ]
            }
        }
    }
}
```

- YAML – Not a markup language
- YAML is a human friendly data serialization standard
- Comments – Use #

- JSON – JavaScript object notation
- Attribute-value pairs
- Similar to XML

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Template Anatomy

- Use templates to create and manage stacks
- JSON or YAML-formatted text files that describe your AWS infrastructure
- AWS CloudFormation JSON template structure and sections

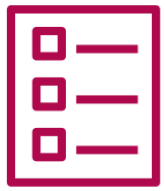
```
{
   "AWSTemplateFormatVersion” : "version date”,
   "Description”: "JSON string”,
   "Metadata”: {
      template metadata
   },
   "Parameters”: {
      set of parameters
   },
   "Mappings”: {
      set of mappings
   },
   "Conditions”: {
      set of conditions
   },
   "Transform”: {
      set of transforms
   },
   "Resources”: {
      set of resources
   },
   "Outputs”: {
      set of outputs
   }
}
```


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

CloudFormation Template Example

What will CloudFormation Provision?

JSON



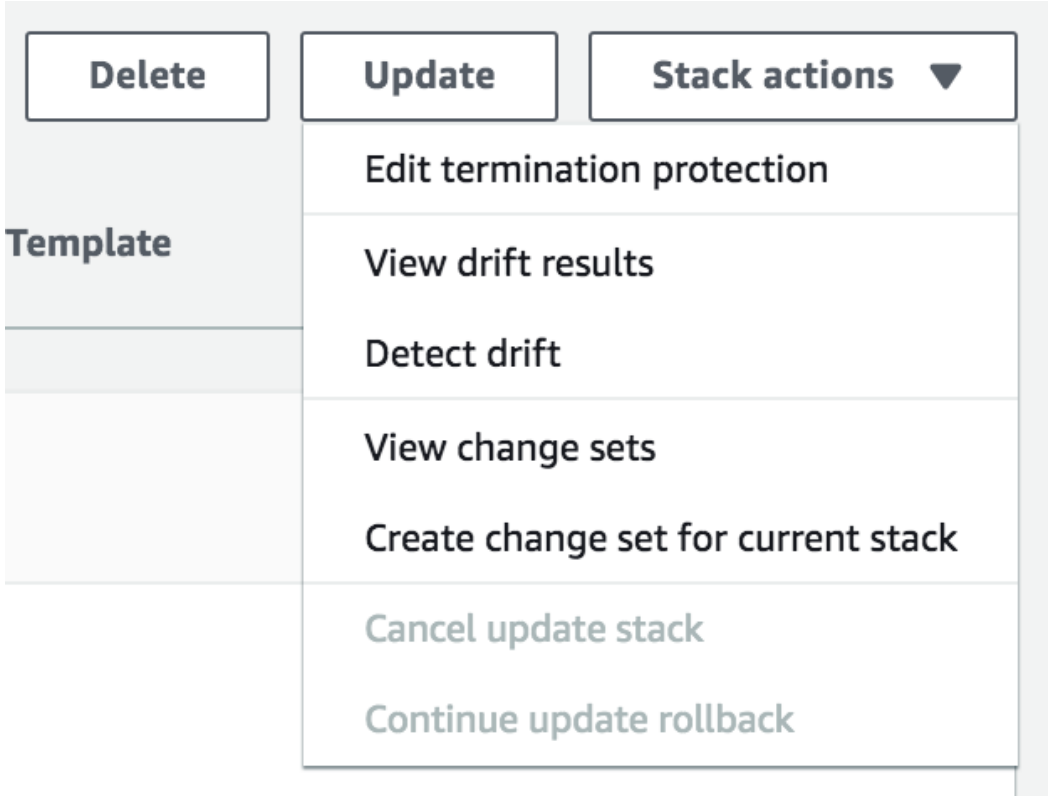
Cloud Formation Template

```
"AWSTemplateFormatVersion": "2010-09-09",
"Description": "A sample template",
"Resources": {
          "MyEC2Instance": {
                    "Type": "AWS::EC2::Instance",
                    "Properties": {
                              "ImageId": "ami-0ff8a91507f77f867",
                              "InstanceType": "t2.micro",
                              "KeyName": "testkey",
                              "BlockDeviceMappings": [
                                        {
                                                  "DeviceName": "/dev/sdm",
                                                  "Ebs": {
                                                  "VolumeType": "io1",
                                                  "Iops": 200,
                                                  "DeleteOnTermination": false,
                                                  "VolumeSize": 20
```


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Stack Updates

- Make changes to a stack's settings or change its resources by updating stack
- When you update a stack, you submit changes to AWS CloudFormation
- Two methods for updating stacks: *direct update* or *change sets* (you create and execute)

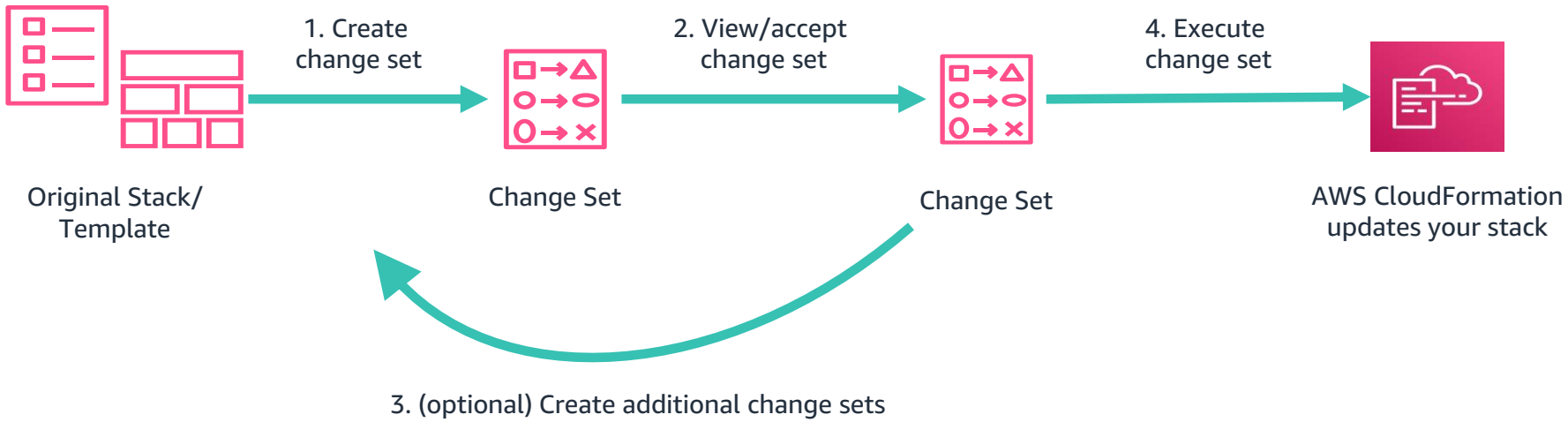



```
aws cloudformation update-stack --stack-name mystack --use-previous-template --notification-arns "arn:aws:sns:us-east-1:12345678912:mytopic" "arn:aws:sns:us-east-1:12345678912:mytopic2"
```


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Change Sets

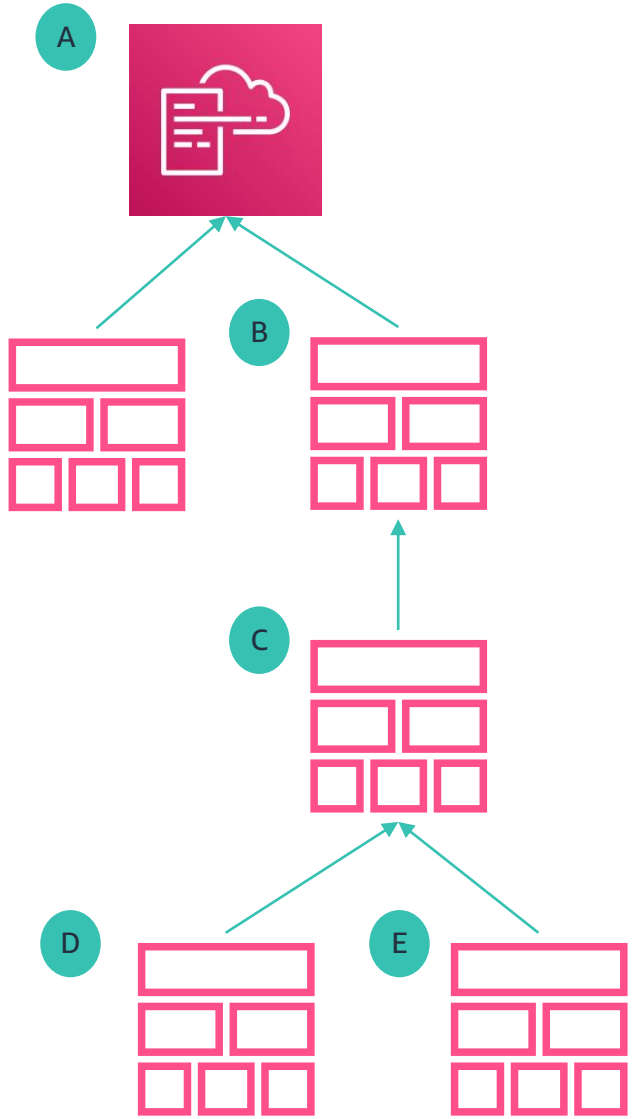
- Change sets enable you to preview how proposed changes to a stack might impact your running resources
- AWS CloudFormation makes the changes to your stack only when you decide to execute the change set



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Nested Stacks

- Monolithic to Modular: common patterns can emerge in which you declare the same components in multiple templates
- Root (A) AWS CloudFormation is the root stack for all the other, nested, stacks in the hierarchy
- Nested stack templates must be placed in Amazon S3
- Broad permissions required to create a stack
- Blast radius – Takes one parent stack to destroy them all
- Using nested stacks to declare common components is considered a best practice



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

aws training and certification

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