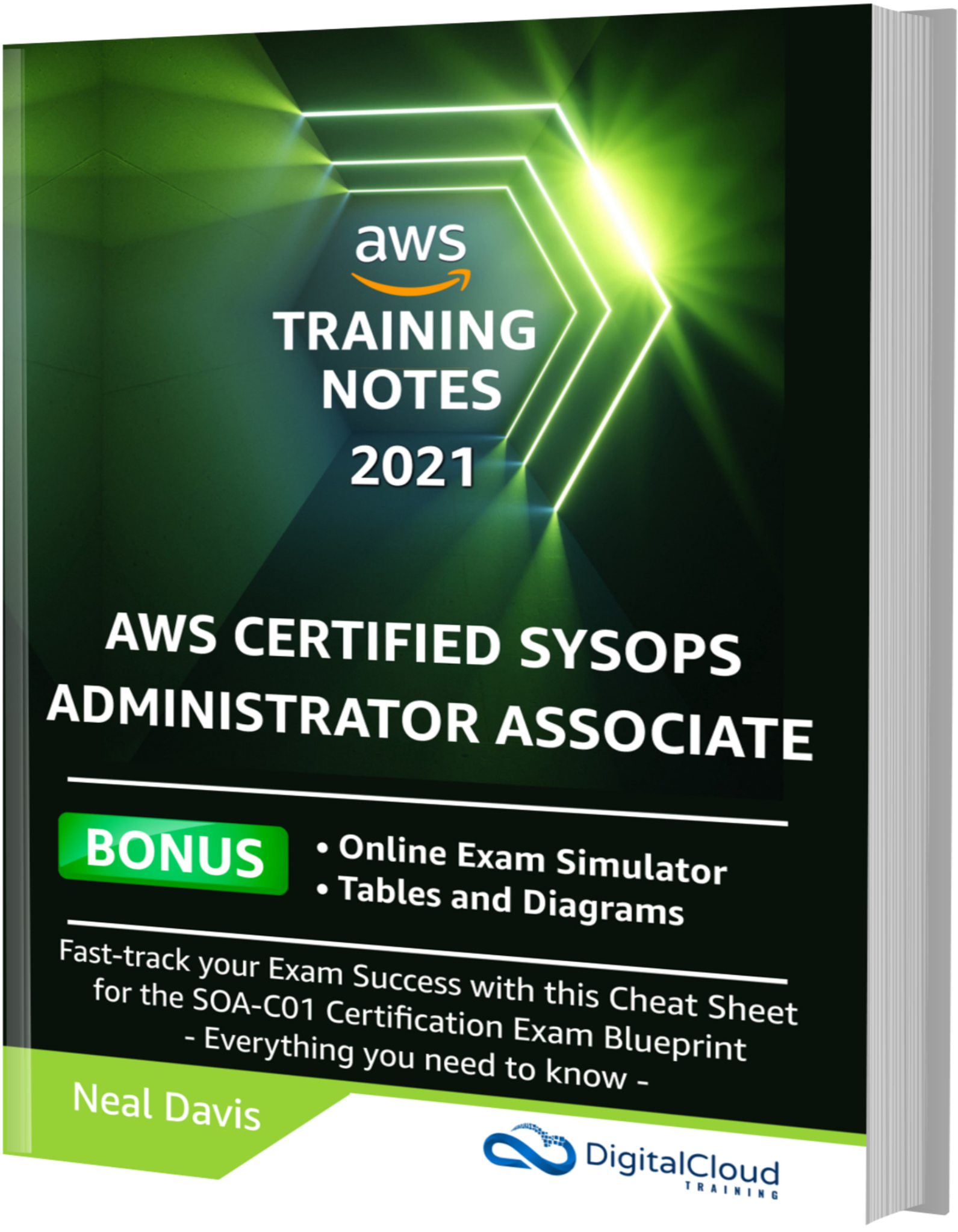
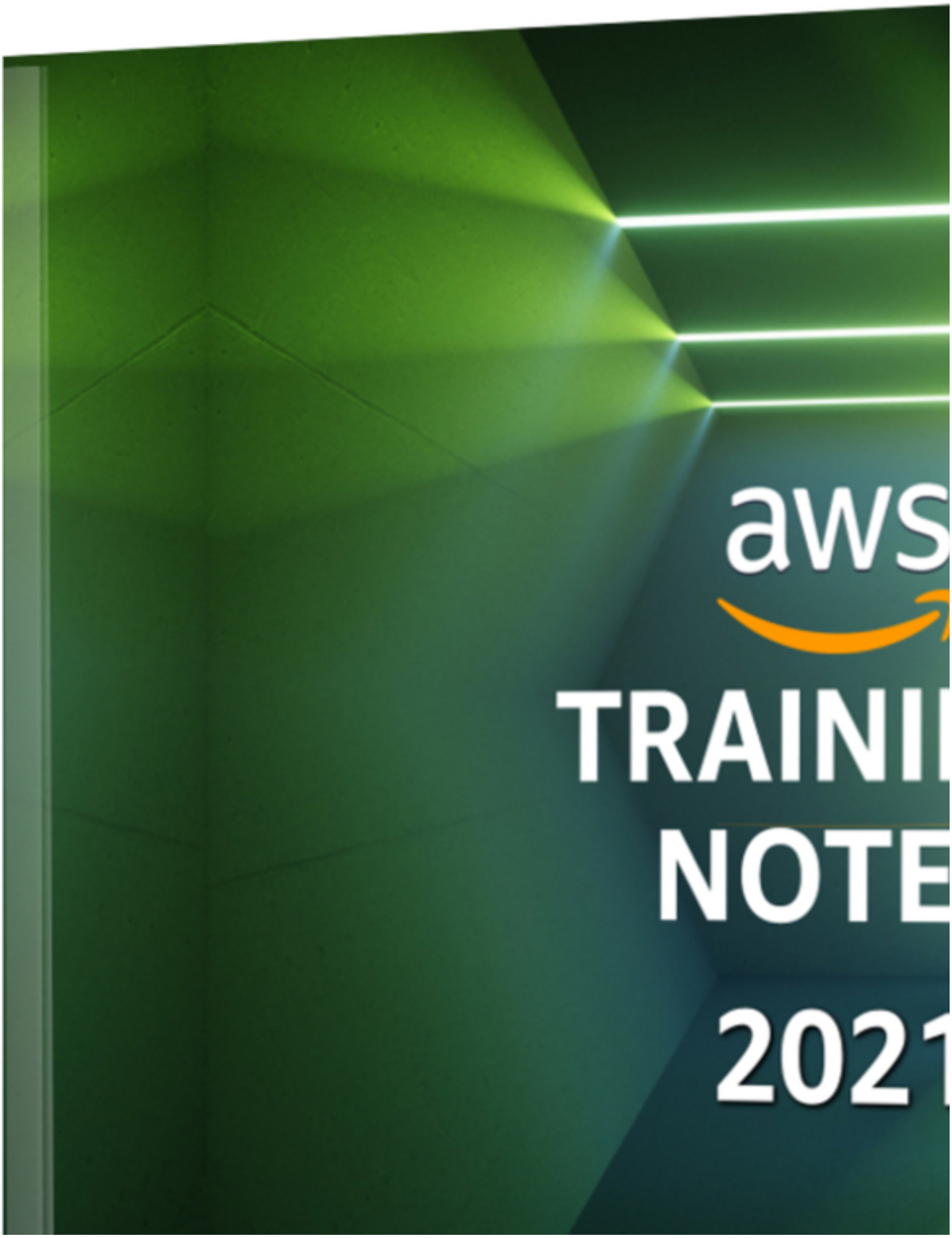

aws
TRAINING
NOTES
2021
AWS CERTIFIED SYSOPS
ADMINISTRATOR ASSOCIATE
BONUS
• Online Exam Simulator
• Tables and Diagrams
Fast-track your Exam Success with this Cheat Sheet
for the SOA-C01 Certification Exam Blueprint
- Everything you need to know -
Neal Davis
DigitalCloud
TRAINING


aws
TRAINI
NOTE
2021

WELCOME

Thanks for purchasing these training notes for the **AWS Certified SysOps Administrator Associate** exam from Digital Cloud Training. The information in this document relates to the latest SOA-C01 version of the exam blueprint.

The aim of putting all of the exam-specific information together into one document is to provide a centralized, detailed list of the facts you need to know before you sit the exam. This will shortcut your study time and maximize your chance of passing the AWS Certified SysOps exam first time.

I trust that you get great value from this popular resource that has been well received by our pool of over 250,000 students. Through diligent study of these learning materials, you will be in the perfect position to ace your AWS Certified SysOps Administrator Associate exam first time.

Wishing you all the best with your AWS Certification exam.

Neal Davis

Founder of Digital Cloud Training

TABLE OF CONTENTS

WELCOME

GETTING STARTED

- About these Training Notes
- Your Pathway to Success
- Limited Time Bonus Offer
- Contact, Support & Feedback
- Reviews Really Matter
- Join the AWS Community
- Connect with Neal on Social Media

HOW HARD IS THE AWS CERTIFIED SYSOPS ADMINISTRATOR ASSOCIATE?

- Is it worth getting?
- Are there any pre-requisites?
- What topics are covered?
- How much overlap is there with other associate level certifications?
- What are the exam questions like?
- How long will it take to prepare for the exam?

AMAZON EC2

- Amazon EC2 Instances
- Amazon Machine Images (AMIs)
- Deployment and Provisioning
- High Availability
- Monitoring and Reporting
- Logging and Auditing
- Authorization and Access Control

AMAZON EC2 PLACEMENT GROUPS

- Cluster Placement Group
- Spread Placement Group
- Partition Placement Group

ELASTIC LOAD BALANCING

- Application Load Balancer (ALB)

- Network Load Balancer (NLB)
- Classic Load Balancer (CLB)
- Deployment and Provisioning
- Monitoring and Reporting
- Logging and Auditing
- Authorization and Access Control

AMAZON EC2 AUTO SCALING

- Scheduled Scaling
- Dynamic Scaling
- Predictive Scaling
- Deployment and Provisioning
- High Availability
- Monitoring and Reporting
- Logging and Auditing
- Authorization and Access Control

AMAZON EBS

- EBS Volume Types
- Instance Store
- EBS vs Instance Store
- Snapshots
- Encryption
- Deployment and Provisioning
- Redundant Array of Independent Disks (RAID)
- Monitoring and Reporting
- Logging and Auditing

AMAZON EFS

- Amazon EFS Backups and Lifecycle Management
- Amazon EFS Performance
- Amazon EFS Encryption
- Amazon EFS Access Control
- Monitoring and Reporting
- Logging and Auditing

AWS STORAGE GATEWAY

- File Gateway

Volume Gateway
Gateway Virtual Tape Library
Managing AWS Storage Gateway
Monitoring AWS Storage Gateway

AWS SYSTEMS MANAGER

Systems Manager Components
Deployment and Provisioning
Monitoring and Reporting
Logging and Auditing
Authorization and Access Control

AWS OPSWORKS

AWS ELASTIC BEANSTALK

Elastic Beanstalk Layers
Deployment and Provisioning
High Availability
Monitoring and Reporting
Logging and Auditing
Authorization and Access Control

AWS CLOUDFORMATION

Deployment and Provisioning
Monitoring and Reporting
Authorization and Access Control

AMAZON VIRTUAL PRIVATE CLOUD (VPC)

Amazon Virtual Private Cloud (VPC) Overview
CIDR Blocks and IP Subnets for Amazon VPCs
NAT Gateways and NAT Instances
Security Groups
Network ACL’s
VPC Endpoints
VPC Peering
VPC Flow Logs
AWS Managed Virtual Private Network (VPN)
AWS Direct Connect

AMAZON ROUTE 53 (AWS ROUTE 53)

- Hosted Zones
- Health Checks
- Records
- Routing Policies
- Traffic Flow
- Route 53 Resolver
- Charges
- References

AMAZON S3

- Additional Capabilities
- Buckets
- Objects
- Subresources
- Cross-origin-resource-sharing (CORS)
- Storage Classes
- Transfer acceleration
- Static Websites
- Pre-Signed URLs
- MFA Delete
- Versioning
- Object Lifecycle Management
- Encryption
- Event Notifications
- Object Tags
- Cross Region Replication
- Same Region replication (SRR)
- S3 Analytics
- S3 Inventory
- Monitoring and Reporting
- Logging and Auditing
- Authorization and Access Control

AMAZON S3 GLACIER

- S3 Glacier
- S3 Glacier Deep Archive

S3 Glacier Vault

AMAZON CLOUDFRONT

Deployment and Provisioning
High Availability
Monitoring and Reporting
Logging and Auditing

AMAZON RDS

Encryption
DB Subnet Groups
Billing and Provisioning
Scalability
Performance
Multi-AZ and Read Replicas
High Availability Approaches for Databases
Monitoring, Logging and Reporting
Authorization and Access Control

AMAZON AURORA

Aurora Replicas
Cross-Region Read Replicas
Global Database
Multi-Master
Aurora Serverless
Fault-Tolerant and Self-Healing Storage
Aurora Auto Scaling
Automatic, Continuous, Incremental Backups and Point-in-Time Restore

AMAZON ELASTICACHE

Memcached
Redis
Caching strategies
Dealing with stale data – Time to Live (TTL)
Monitoring and Reporting
Logging and Auditing
Authorization and Access Control

AWS ORGANIZATIONS
- AWS Organizations Concepts
- Service Control Policies
- Resource Groups

AMAZON CLOUDWATCH
- Metrics
- Custom Metrics
- High-Resolution Metrics
- Namespace
- Dimensions
- Statistics
- CloudWatch Alarms
- CloudWatch Logs
- CloudWatch Logs Agent
- CloudWatch Events
- Useful API Actions

AWS CLOUDTRAIL
- References

AWS CONFIG
- AWS Config vs CloudTrail
- Config Rules
- Configuration Items
- Charges

AWS IAM – IDENTITY AND ACCESS MANAGEMENT
- Authentication Methods
- IAM Users
- Groups
- Roles
- Policies
- Inline Policies vs Managed Policies
- AWS Managed and Customer Managed Policies
- IAM Policy Evaluation Logic
- IAM Instance Profiles

AWS Security Token Service
Cross Account Access

AWS KMS AND AWS CLOUDHSM

AWS KMS
Customer Master Keys (CMK's)
Customer Managed CMK's
AWS Managed CMK's
AWS Owned CMK's
Data Encryption Keys
KMS Details
Key Management with KMS
Data Encryption Scenarios
Custom Key Store
Key deletion
AWS KMS API's
KMS Envelope Encryption
AWS CloudHSM

AWS WAF AND SHIELD

AWS Web Application Firewall (WAF)
AWS Shield
AWS Shield Standard
AWS Shield Advanced

EXAM SCENARIOS FOR AWS SYSOPS ADMINISTRATOR

Amazon EC2 and AWS Lambda
Elastic Load Balancing and Auto Scaling
Amazon EBS, EFS, and AWS Storage Gateway
AWS Systems Manager
AWS CloudFormation
Amazon Virtual Private Cloud (VPC)
Amazon Route 53
Amazon S3 and CloudFront
Amazon RDS and ElastiCache
Management, Governance and Billing
Security and Compliance

CONCLUSION
- Before taking the AWS Exam
- Reach out and Connect
- Bonus Offer
- Leave us a Review

OTHER BOOKS & COURSES BY NEAL DAVIS
- Courses for the AWS Certified Cloud Practitioner
- Courses for the AWS Certified Solutions Architect Associate
- Courses for the AWS Certified Developer Associate
- Courses for the AWS Certified SysOps Administrator Associate

ABOUT THE AUTHOR
- Connect with Neal on social media

GETTING STARTED

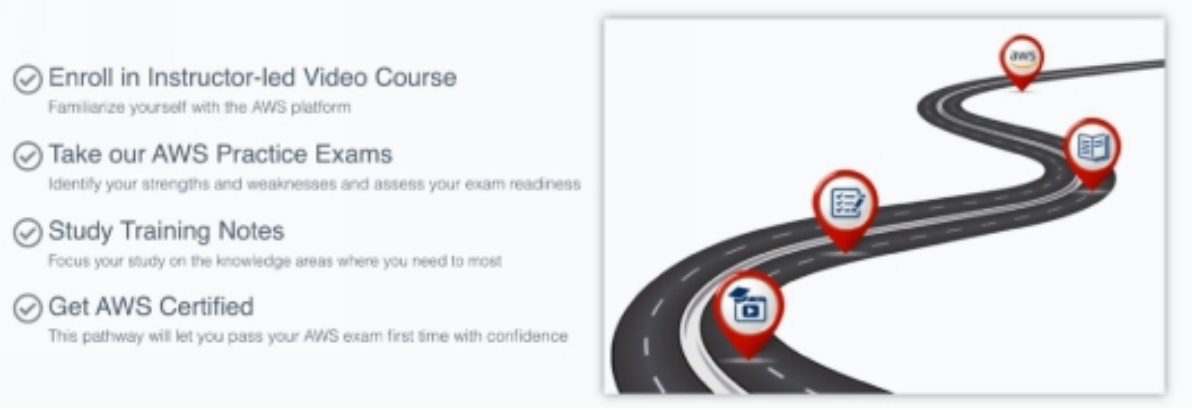
ABOUT THESE TRAINING NOTES

Please note that this document does not read like a book or instructional text. We provide a raw, point-to-point list of facts backed by tables and diagrams to help with understanding.

For easy navigation, the information on each AWS service in this document is organized into the same categories as they are in the AWS Management Console.

The scope of coverage of services, and what information is included for each service, is based on feedback from our pool of over 250,000 students who have taken the exam, as well as our own experience.

YOUR PATHWAY TO SUCCESS



HOW CAN YOU BEST PREPARE?

So, you're excited to get started with the AWS Certified SysOps Administrator Associate certification and wondering what resources are out there to help you. Let's start with the free options. Visit Digital Cloud Training for links to various free resources including **cheat sheets** , **practice questions** , **blog**

articles , **YouTube videos** , and **AWS documentation** : **https://digitalcloud.training/ amazon-aws-free-certification-training-sysops-administrator-associate/**

For the full training experience though, your best bet are the following training courses:

ON-DEMAND VIDEO COURSE

If you're new to AWS, we'd suggest first enrolling in the online instructor-led **AWS Certified SysOps Administrator Associate Hands-on Labs Video Course** from Digital Cloud Training to familiarize yourself with the AWS platform before returning to the Training Notes to get a more detailed understanding of the AWS services.

PRACTICE EXAM COURSE

To assess where you are at on your AWS journey, we recommend taking the **AWS Certified SysOps Administrator Associate Practice Exams** on the Digital Cloud Training website. The online exam simulator will help you identify your strengths and weaknesses. These practice tests are designed to reflect the difficulty of the AWS exam and are the closest to the real exam experience available.

Apply coupon code **AMZ20** at checkout for a 20% discount.

Our online Practice Exams are delivered in 4 different variations:

- **Exam Mode**

  In exam simulation mode, you complete one full-length practice exam and answer all 65 questions within the allotted time. You are then presented with a pass / fail score report showing your overall score and performance in each knowledge area to identify your strengths and weaknesses.

- **Training Mode**

  When taking the practice exam in training mode, you will be shown the answers and explanations for every question after clicking "check". Upon completion of the exam, the score report will show your overall score and performance in each knowledge area.

- **Knowledge Reviews**

Now that you have identified your strengths and weaknesses, you get to dive deep into specific areas with our knowledge reviews. You are presented with a series of questions focussed on a specific topic. There is no time limit and you can view the answer to each question as you go through them.

- **Final Exam Simulator**

The exam simulator randomly selects 65 questions from our pool of questions – mimicking the real AWS exam environment. The practice exam has the same format, style, time limit and passing score as the real AWS exam

STEP 3: TRAINING NOTES

As a final step, use these training notes to focus your study on the knowledge areas where you need to most. Get a detailed understanding of the AWS services and deep dive into the SOA-C01 exam objectives with detailed facts, tables and diagrams that will shortcut your time to success.

LIMITED TIME BONUS OFFER

To assess your AWS exam readiness, we have included one full-length practice exam from Digital Cloud Training. These 65 exam-difficulty practice questions are timed and scored and simulate the real AWS exam experience. To gain access to your free practice test on our interactive exam simulator online, simply navigate to the **CONCLUSION** at the back of this book where you'll find detailed instructions.

CONTACT, SUPPORT & FEEDBACK

We want you to get great value from these training resources. If for any reason you are not 100% satisfied, please contact us at **support@digitalcloud.training** . We promise to address all questions and concerns, typically within 24hrs. We really want you to have a 5-star learning experience!

The AWS platform is evolving quickly, and the exam tracks these changes with a typical lag of around 6 months. We are therefore reliant on student feedback to keep track of what is appearing in the exam. If there are any topics in your exam that weren't covered in our training resources, please provide us with feedback using this form **https://digitalcloud.training/student-feedback/** . We appreciate any feedback that will help us further improve our AWS training resources.

REVIEWS REALLY MATTER

If you enjoy reading reviews, please consider paying it forward. Reviews guide students and help us continuously improve our courses. We celebrate every honest review and truly appreciate it. We'd be thrilled if you could leave a rating at **amazon.com/ryp** or your local amazon store (e.g. amazon.co.uk/ryp).

JOIN THE AWS COMMUNITY

Our private **Facebook group** is a great place to ask questions and share knowledge and exam tips with the AWS community. Join the AWS Certification QA group on Facebook and share your exam feedback with the AWS community: **https://www.facebook.com/groups/awscertificationqa**

To join the discussion about all things related to Amazon Web Services on **Slack** , visit: **http://digitalcloud.training/slack** for instructions.

CONNECT WITH NEAL ON SOCIAL MEDIA

To learn more about the different ways of connecting with Neal, visit: **https://digitalcloud.training/neal-davis**

digitalcloud.training/neal-davis

youtube.com/c/digitalcloudtraining

facebook.com/digitalcloudtraining

Twitter @ **nealkdavis**

linkedin.com/in/nealkdavis

Instagram @digitalcloudtraining

HOW HARD IS THE AWS CERTIFIED SYSOPS ADMINISTRATOR ASSOCIATE?

The AWS Certified SysOps Administrator Associate certification has a reputation for being the hardest of the associate level certifications in Amazon Web Services' certification programs. But how difficult is it really?

The AWS SysOps Administrator exam focusses on exam scenarios that cover deployment and operational aspects of AWS services. This means it can be a bit challenging for those who don't have on-the-job experience. There is also a lot more coverage of monitoring, auditing and managing resources, as well as troubleshooting issues.

If you work with AWS services regularly then that will make the exam significantly easier.

IS IT WORTH GETTING?

Some people skip the SysOps Associate as they don't work in a systems operations role so they assume it is not that useful. However, one may argue that it is a valuable certification to gain for a couple of reasons:

1. Firstly, it's considered to be harder and fewer people have it on their resumes, so it does provide a bit of differentiation.
2. Secondly, if you're already doing the other associate level certifications, due to the amount of overlap you'll already be most of the way there – so it's definitely worth a bit of extra effort.

ARE THERE ANY PRE-REQUISITES?

There are no pre-requisites for taking the exam. However, we recommend taking at least one of the other associate level certification first or having equivalent industry experience. This will provide you with a broader understanding of AWS services and more hands-on experience (assuming you follow along with the labs).

WHAT TOPICS ARE COVERED?

There are seven domains in the AWS Certified SysOps Administrator Associate exam guide:

Domain 1: Monitoring and Reporting

- 1.1 Create and maintain metrics and alarms utilizing AWS monitoring services
- 1.2 Recognize and differentiate performance and availability metrics
- 1.3 Perform the steps necessary to remediate based on performance and availability metrics

Domain 2: High Availability

- 2.1 Implement scalability and elasticity based on use case
- 2.2 Recognize and differentiate highly available and resilient environments on AWS

Domain 3: Deployment and Provisioning

- 3.1 Identify and execute steps required to provision cloud resources
- 3.2 Identify and remediate deployment issues

Domain 4: Storage and Data Management

- 4.1 Create and manage data retention

- 4.2 Identify and implement data protection, encryption, and capacity planning needs

Domain 5: Security and Compliance

- 5.1 Implement and manage security policies on AWS
- 5.2 Implement access controls when using AWS
- 5.3 Differentiate between the roles and responsibility within the shared responsibility model

Domain 6: Networking

- 6.1 Apply AWS networking features
- 6.2 Implement connectivity services of AWS
- 6.3 Gather and interpret relevant information for network troubleshooting

Domain 7: Automation and Optimization

- 7.1 Use AWS services and features to manage and assess resource utilization
- 7.2 Employ cost-optimization strategies for efficient resource utilization
- 7.3 Automate manual or repeatable process to minimize management overhead

HOW MUCH OVERLAP IS THERE WITH OTHER ASSOCIATE LEVEL CERTIFICATIONS?

There is a lot of overlap between all of the associate level certifications. The core set of services are very similar to the AWS Solutions Architect Associate and AWS Developer Associate. However, there are some

differences in the objectives of the AWS Certified SysOps Administrator being more geared towards deployment, management and operational activities.

In addition to different objectives, there are some services covered in more detail, including:

- AWS Systems Manager (lots of coverage of inventory, patching, software management, etc.)
- AWS CloudFormation (at least compared to the AWS Solutions Architect Associate)
- Amazon ElastiCache (more focus on deployment types)
- AWS Config (for compliance)
- Amazon CloudWatch (lots of coverage of performance monitoring, events, logs, etc.)
- AWS IAM (more geared towards understanding IAM policies)
- AWS Organizations (know your SCPs!)
- AWS Cost management tools (fairly light coverage but more than the other associates: Cost Explorer, Cost and Usage Report, AWS Budgets)

WHAT ARE THE EXAM QUESTIONS LIKE?

Here are some resources you can use to check out the format of the exam questions:

1. You can find **20 FREE AWS Certified SysOps Administrator questions** here: **https://digitalcloud.training/quizzes/free-aws-certified-sysops-administrator-practice-exam/**
2. At the end of this document, you'll find some example **exam scenarios** with solutions

The exam scenarios are extracted from many common scenarios that students have seen come up on the exam so they are a really useful resource to gain an understanding of the types of questions you will see on the exam. This can also really help you in knowing what you need to learn to prepare yourself.

HOW LONG WILL IT TAKE TO PREPARE FOR THE EXAM?

We recommend taking at least one of the other associate level certifications first. If you do that then you'll already be 60-80% of the way there. Generally speaking, you need to put aside an additional 2-3 weeks to prepare for the SOA-C01 exam.

Download your **FREE Study Plan** to successfully prepare for your AWS SysOps Administrator Associate in 21 days here: **https://digitalcloud.training/aws-certification-study-plan-sysops-administrator-associate/**

The AWS Certified SysOps Administrator Associate is a great certification to get on your CV so we hope you're excited to get started with your exam preparation!

AMAZON EC2

Amazon Elastic Compute Cloud (Amazon EC2) is a service you can use to deploy virtual servers known as “instances” on the AWS Cloud.

Amazon EC2 provides the following features:

- Virtual computing environments, known as instances.
- Preconfigured templates for your instances, known as Amazon Machine Images (AMIs), that package the bits you need for your server (including the operating system and additional software).
- Various configurations of CPU, memory, storage, and networking capacity for your instances, known as instance types.
- Secure login information for your instances using key pairs (see below).
- Storage volumes for temporary data that’s deleted when you stop or terminate your instance, known as instance store volumes (ephemeral / non-persistent data).
- Persistent storage volumes for your data using Amazon Elastic Block Store (Amazon EBS), known as **Amazon EBS volumes** .
- Multiple physical locations for your resources, such as instances and Amazon EBS volumes, known as Regions and Availability Zones.
- A firewall that enables you to specify the protocols, ports, and source IP ranges that can reach your instances using security groups.
- Static IPv4 addresses for dynamic cloud computing, known as Elastic IP addresses.

- Metadata, known as tags, that you can create and assign to your Amazon EC2 resources.

AMAZON EC2 INSTANCES

An instance is a virtual server in the cloud. Its configuration at launch is a copy of the AMI that you specified when you launched the instance.

You can launch different types of instances from a single AMI.

An *instance type* essentially determines the hardware of the host computer used for your instance.

Each instance type offers different compute, memory, and storage capabilities.

Select an instance type based on the amount of memory and computing power that you need for the application or software that you plan to run on the instance.

There are many different families and types of instance available with varying combinations of CPU, memory, storage, and networking capacity.

The following tables shows some examples of the categories and families available:

Category	Families	Purpose/Design
General Purpose	A1, T3, T3a, T2, M5, M5a, M4	General purpose instances provide a balance of compute, memory and networking resources, and can be used for a variety of diverse workloads
Compute Optimized	C5, C5n, C4	Compute Optimized instances are ideal for compute bound applications that benefit from high performance processors
Memory Optimized	R5, R5a, R4, X1e, X1, High Memory, z1d	Memory optimized instances are designed to deliver fast performance for workloads that process large data sets in memory
Accelerated Computing	P3, P2, G4, G3, F1	Accelerated computing instances use hardware accelerators, or co-processors, to perform functions, such as floating-point number calculations, graphics processing, or data pattern matching
Storage Optimized	I3, I3en, D2, H1	This instance family provides Non-Volatile Memory Express (NVMe) SSD-backed instance storage optimized for low latency, very high random I/O performance, high sequential read throughput and provide high IOPS at a low cost

Burstable instances:

- T3, T3a, and T2 instances, are designed to provide a baseline level of CPU performance with the ability to burst to a higher level when required.
- Burstable performance instances are the only instance types that use credits for CPU usage.
- A CPU credit` provides for 100% utilization of a full CPU core for one minute.
- Each burstable performance instance continuously earns (at a millisecond-level resolution) a set rate of CPU credits per hour, **depending on the instance size** .

T2/T3 unlimited instances:

- **T2/T3 instances** are a low-cost, general purpose instance type that provides a baseline level of CPU performance with the ability to burst above the baseline when needed.
- T2/T3 Unlimited instances can sustain high CPU performance for as long as a workload needs it.
- The baseline performance and ability to burst are governed by CPU Credits.
- T2/T3 instances accumulate CPU Credits when they are idle, and consume CPU Credits when they are active.

Amazon EC2 instances can be placed in an **Amazon EC2 placement group** .

Amazon EC2 instances can be managed through **AWS Systems Manager** .

You can also use **AWS OpsWorks** to manage your instances using Chef and Puppet.

AWS Config can be used to record configuration items about Amazon EC2 instances and track changes.

AMAZON MACHINE IMAGES (AMIS)

An Amazon Machine Image (AMI) is a template that contains a software configuration (for example, an operating system, an application server, and applications).

From an AMI, you launch an instance, which is a copy of the AMI running as a virtual server in the cloud. You can launch multiple instances of an AMI.

An Amazon Machine Image (AMI) provides the information required to launch an instance.

You must specify an AMI when you launch an instance.

You can launch multiple instances from a single AMI when you need multiple instances with the same configuration.

You can use different AMIs to launch instances when you need instances with different configurations.

AMIs are specific to a region (you must select the AMI within each region separately).

An AMI includes the following:

- One or more **EBS snapshots** , or, for instance-store-backed AMIs, a template for the root volume of the instance (for example, an operating system, an application server, and applications).
- Launch permissions that control which AWS accounts can use the AMI to launch instances.
- A block device mapping that specifies the volumes to attach to the instance when it's launched.

You can create your own custom AMIs (also known as golden images) which can included pre-installed software and configuration settings.

AMIs can be public or private.

AMIs are private by default.

Public AMIs that are shared by others can be found on the Amazon Marketplace.

Public AMIs may come at an increased hourly cost, and:

- May contain preconfigured software (and can include licensing).
- Can be configured for you so you can get started quickly.

AMIs are stored on Amazon S3 however you cannot see them in the S3 console.

You get charged at S3 rates for the quantity of data stored in the AMI.

SHARING AN AMI ACROSS ACCOUNTS

You can share an AMI with specific AWS accounts without making the AMI public.

All you need is the AWS account IDs.

You can only share AMIs that have unencrypted volumes and volumes that are encrypted with a customer managed CMK.

You cannot share an AMI that has volumes that are encrypted with an AWS managed CMK.

If you share an AMI with encrypted volumes, you must also share any CMKs used to encrypt them.

You cannot copy an AMI with an associated billingProduct code that was shared with you from another AWS account. To get around this, launch an instance from the AMI, and then create an AMI from the instance.

DEPLOYMENT AND PROVISIONING

Amazon EC2 instances can be launched from within the EC2 management console.

You can also launch instances using the **RunInstances** API.

You can change instance types for EBS backed instances only.

To change the instance type you must stop the instance, change the instance type, and then start it up again.

When you stop and start and EC2 instance, it will generally be moved to different underlying hardware.

> *Exam tip: You can stop and start an EC2 instance to move it to a different physical host if EC2 status checks are failing or there is planned maintenance on the current physical host.*

You can stop and start your instance if it has an Amazon EBS volume as its root device.

When you stop a running instance, the following happens:

- The instance performs a normal shutdown and stops running; its status changes to stopping and then stopped.
- Any Amazon EBS volumes remain attached to the instance, and their data persists.
- Any data stored in the RAM of the host computer or the instance store volumes of the host computer is gone.
- In most cases, the instance is migrated to a new underlying host computer when it’s started.
- The instance retains its private IPv4 addresses and any IPv6 addresses when stopped and started (public addresses are released).
- The instance retains its associated Elastic IP addresses (you’re charged for any Elastic IP addresses associated with a stopped instance).

You can modify the following attributes of an instance only when it is stopped:

- Instance type.
- User data.
- Kernel.
- RAM disk.

You can change the instance initiated shutdown behavior so it terminates rather than stops the instance.

Configure by modifying the InstanceInitiatedShutdownBehavior.

Termination protection protects against accidental deletion.

The DisableApiTermination attribute controls whether the instance can be terminated using the console, CLI, or API.

Use Amazon **Elastic File System** (EFS) for mounting a shared filesystem to multiple EC2 instances.

CONNECTING TO YOUR AMAZON EC2 INSTANCE

Key pairs are used to securely connect to Amazon EC2 instances:

- A key pair consists of a **public key** that AWS stores, and a **private key file** that you store.
- For Windows AMIs, the private key file is required to obtain the password used to log into your instance.
- For Linux AMIs, the private key file allows you to securely SSH (secure shell) into your instance.

Common errors:

- Error “Unprotected private key file”. Private key files must have the permissions set to 400 (use CHMOD on Linux).
- Error “Host key not found”. User name must be supplied correctly when logging in.

Connection timeout issues:

- Security group may not be configured correctly (allow TCP port 22 inbound from your computer/network or 0.0.0.0/0).
- The CPU load on the EC2 instance could be high preventing the connection completing.

EC2 LAUNCH ISSUES

InstanceLimitExceeded error – the maximum number of instances in the region has been reached.

Increase your limit or launch in another region.

Limits are based on the instance type and number of vCPUs. See the **user guide** for more info.

InsufficientInstanceCapacity error – AWS does not have enough on-demand capacity to service the request in the specific Availability Zone.

Resolution options:

- Wait a few minutes and try again.
- Try a different instance type.
- Try launching a smaller number at a time.
- Try purchasing reserved instances instead.
- Submit a new request without specifying the Availability Zone.

Instance terminates immediately (goes from pending to terminated).

This is caused by one of the following:

- The limit for EBS volumes has been reached.
- An EBS snapshot is corrupt.
- The root EBS volume is encrypted and you don't have permission to access the KMS key to decrypt the data.
- The instance-store backed AMI that was used to launch the instance is missing a required part.

The EC2 console reports the exact reason in the State Transition Reason description.

IP ADDRESSES

There are three types of IP address that can be assigned to an Amazon EC2 instance

- Public – public address that is assigned automatically to instances in public subnets and reassigned if instance is stopped/started.
- Private – private address assigned automatically to all instances.
- Elastic IP – public address that is static.

Public IPv4 addresses are lost when the instance is stopped but private addresses (IPv4 and IPv6) are retained.

Public IPv4 addresses are retained if you restart the instance.

Elastic IPs are retained when the instance is stopped.

Elastic IP addresses are static public IP addresses that can be remapped (moved) between instances.

All accounts are limited to 5 elastic IP's per region by default.

AWS charge for elastic IP's when they're not being used.

Elastic IP address can mask the failure of an instance by remapping the address to another instance.

METADATA AND USER DATA

User data is data that is supplied by the user at instance launch in the form of a script.

Instance metadata is data about your instance that you can use to configure or manage the running instance.

User data is limited to 16KB.

User data and metadata are not encrypted.

Instance metadata is available at **http://169.254.169.254/latest/meta-data/** (the trailing "/" is required).

Instance user data is available at: **http://169.254.169.254/latest/user-data** .

The IP address 169.254.169.254 is a link-local address and is valid only from the instance.

On Linux you can use the curl command to view metadata and userdata, e.g.
"curl **http://169.254.169.254/latest/meta-data/** ".

The Instance Metadata Query tool allows you to query the instance metadata without having to type out the full URI or category names.

HIGH AVAILABILITY

High availability for Amazon EC2 instances is achieved by using **Amazon EC2 Auto Scaling** with Amazon Elastic Load Balancing – please check the relevant cheat sheets below for more information:

- **Amazon EC2 Auto Scaling Cheat Sheet**
- **Amazon Elastic Load Balancing Cheat Sheet**

Fault tolerance can be achieved by using RAID 1 with your Amazon EBS volumes – please check the relevant cheat sheet here for more information.

MONITORING AND REPORTING

Amazon EC2 sends metrics to Amazon CloudWatch and you can use the AWS Management Console, the AWS CLI, or an API to list the metrics that Amazon EC2 sends to CloudWatch.

CloudWatch stores data about a metric as a series of data points. Each data point has an associated time stamp. You can even publish an aggregated set of data points called a *statistic set* .

By default, each data point covers the 5 minutes that follow the start time of activity for the instance.

If you've enabled detailed monitoring (chargeable), each data point covers the next minute of activity from the start time.

You can find the list of standard metrics sent from Amazon EC2 to Amazon CloudWatch at the following link:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/viewing_metrics_with_cloudwatch.html

Custom metrics:

- You can publish your own metrics to CloudWatch using the AWS CLI or an API.
- You can view statistical graphs of your published metrics with the AWS Management Console.

- Can be used to gather system-level metrics including memory and disk utilization.

Each custom metric is one of the following:

- Standard resolution, with data having a one-minute granularity.
- High resolution, with data at a granularity of one second.
- Unified CloudWatch Agent

The unified **CloudWatch agent** enables you to do the following:

- Collect more system-level metrics from Amazon EC2 instances across operating systems. The metrics can include in-guest metrics, in addition to the metrics for EC2 instances. The additional metrics that can be collected are listed in **Metrics Collected by the CloudWatch Agent** .
- Collect system-level metrics from on-premises servers. These can include servers in a hybrid environment as well as servers not managed by AWS.
- Retrieve custom metrics from your applications or services using the StatsD and collectd protocols. StatsD is supported on both Linux servers and servers running Windows Server. collectd is supported only on Linux servers.
- Collect logs from Amazon EC2 instances and on-premises servers, running either Linux or Windows Server.

You can download and install the CloudWatch agent manually using the command line, or you can integrate it with SSM.

LOGGING AND AUDITING

Amazon EC2 and Amazon EBS are integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Amazon EC2 and Amazon EBS.

CloudTrail captures all API calls for Amazon EC2 and Amazon EBS as events, including calls from the console and from code calls to the APIs.

If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Amazon EC2 and Amazon EBS.

If you don't configure a trail, you can still view the most recent events in the CloudTrail console in Event history.

Using the information collected by CloudTrail, you can determine the request that was made to Amazon EC2 and Amazon EBS, the IP address from which the request was made, who made the request, when it was made, and additional details.

AUTHORIZATION AND ACCESS CONTROL

NETWORK ACCESS CONTROL:

A security group acts as a firewall that controls the traffic allowed to reach one or more instances. When you launch an instance, you assign it one or more security groups. You add rules to each security group that control traffic for the instance. You can modify the rules for a security group at any time; the new rules are automatically applied to all instances to which the security group is assigned.

AWS IDENTITY AND ACCESS MANAGEMENT (IAM):

Amazon EC2 supports identity-based policies. Identity-based policies are attached to an IAM user, group, or role. These policies let you specify what that identity can do (its permissions). For example, you can attach the policy to the IAM user named Paul, stating that he is allowed to perform the Amazon EC2 RunInstances action.

SHARING AMIS CROSS ACCOUNT:

Amazon EC2 enables you to specify additional AWS accounts that can use your Amazon Machine Images (AMIs) and Amazon EBS snapshots. These

permissions work at the AWS account level only; you can’t restrict permissions for specific users within the specified AWS account. All users in the AWS account that you’ve specified can use the AMI or snapshot. Each AMI has a LaunchPermission attribute that controls which AWS accounts can access the AMI.

AMAZON EC2 PLACEMENT GROUPS

Amazon EC2 placement groups are a logical grouping of instances in one of three configurations.

When you launch a new EC2 instance, the EC2 service attempts to place the instance in such a way that all of your instances are spread out across underlying hardware to minimize correlated failures.

You can use *placement groups* to influence the placement of a group of *interdependent* instances to meet the needs of your workload.

There are three placement strategies available with Amazon EC2 placement groups: cluster, spread, and partition.

The table below describes some key differences between clustered and spread placement groups:

	Clustered	Spread	Partition
What	Instances are placed into a low-latency group within a single AZ	Instances are spread across underlying hardware	Instances are grouped into logical segments called partitions which use distinct hardware
When	Need low network latency and/or high network throughput	Reduce the risk of simultaneous instance failure if underlying hardware fails	Need control and visibility into instance placement
Pros	Get the most out of enhanced networking Instances	Can span multiple AZs	Reduces likelihood of correlated failures for large workloads.
Cons	Finite capacity: recommend launching all you might need up front	Maximum of 7 instances running per group, per AZ	Partition placement groups are not supported for Dedicated Hosts

The following sub-sections provide more details on the three strategies for Amazon EC2 placement groups.

CLUSTER PLACEMENT GROUP

Clusters instances into a low-latency group in a single AZ:

- A cluster placement group is a logical grouping of instances within a single Availability Zone (cannot span AZs).
- A cluster placement group can span peered VPCs in the same Region.
- Instances in the same cluster placement group enjoy a higher per-flow throughput limit of up to 10 Gbps for TCP/IP traffic and are placed in the same high-bisection bandwidth segment of the network
- Cluster placement groups are recommended for applications that benefit from low network latency, high network throughput, or both, and if the majority of the network traffic is between the instances in the group.
- Must use a **supported instance type** .

A partition placement group supports a maximum of seven partitions per Availability Zone. The number of instances that you can launch in a partition placement group is limited only by your account limits.

When instances are launched into a partition placement group, Amazon EC2 tries to evenly distribute the instances across all partitions. Amazon EC2 doesn't guarantee an even distribution of instances across all partitions.

A partition placement group with Dedicated Instances can have a maximum of two partitions.

Partition placement groups are not supported for Dedicated Hosts.

AWS recommend that you launch your instances in the following ways:

- Use a single launch request to launch the number of instances that you need in the placement group.
- Use the same instance type for all instances in the placement group.

Troubleshooting cluster placement groups:

- If you try to add more instances to the placement group later, or if you try to launch more than one instance type in the placement group, you increase your chances of getting an insufficient capacity error.
- If you stop an instance in a placement group and then start it again, it still runs in the placement group. However, the start fails if there isn't enough capacity for the instance.
- If you receive a capacity error when launching an instance in a placement group that already has running instances, stop and start all of the instances in the placement group, and try the launch again. Starting the instances may migrate them to hardware that has capacity for all of the requested instances.

SPREAD PLACEMENT GROUP

Spreads instances across underlying hardware (can span AZs):

- A spread placement group is a group of instances that are each placed on distinct underlying hardware.
- Spread placement groups are recommended for applications that have a small number of critical instances that should be kept separate from each other.

Spread placement groups are recommended for applications that have a small number of critical instances that should be kept separate from each other.

Launching instances in a spread placement group reduces the risk of simultaneous failures that might occur when instances share the same racks.

Spread placement groups provide access to distinct racks, and are therefore suitable for mixing instance types or launching instances over time.

A spread placement group can span multiple Availability Zones in the same Region. You can have a maximum of seven running instances per Availability Zone per group.

Spread placement groups are not supported for Dedicated Instances or Dedicated Hosts.

Troubleshooting spread placement groups:

- If you start or launch an instance in a spread placement group and there is insufficient unique hardware to fulfill the request, the request fails. Amazon EC2 makes more distinct hardware available over time, so you can try your request again later.

PARTITION PLACEMENT GROUP

Divides each group into logical segments called partitions:

- Amazon EC2 ensures that each partition within a placement group has its own set of racks.
- Each rack has its own network and power source. No two partitions within a placement group share the same racks, allowing you to isolate the impact of hardware failure within your application.
- Partition placement groups can be used to deploy large distributed and replicated workloads, such as HDFS, HBase, and Cassandra, across distinct racks.

A partition placement group can have partitions in multiple Availability Zones in the same Region. A partition placement group can have a maximum of seven partitions per Availability Zone. The number of

instances that can be launched into a partition placement group is limited only by the limits of your account.

In addition, partition placement groups offer visibility into the partitions — you can see which instances are in which partitions. You can share this information with topology-aware applications, such as HDFS, HBase, and Cassandra. These applications use this information to make intelligent data replication decisions for increasing data availability and durability.

Troubleshooting partition placement groups:

- If you start or launch an instance in a partition placement group and there is insufficient unique hardware to fulfill the request, the request fails. Amazon EC2 makes more distinct hardware available over time, so you can try your request again later.

ELASTIC LOAD BALANCING

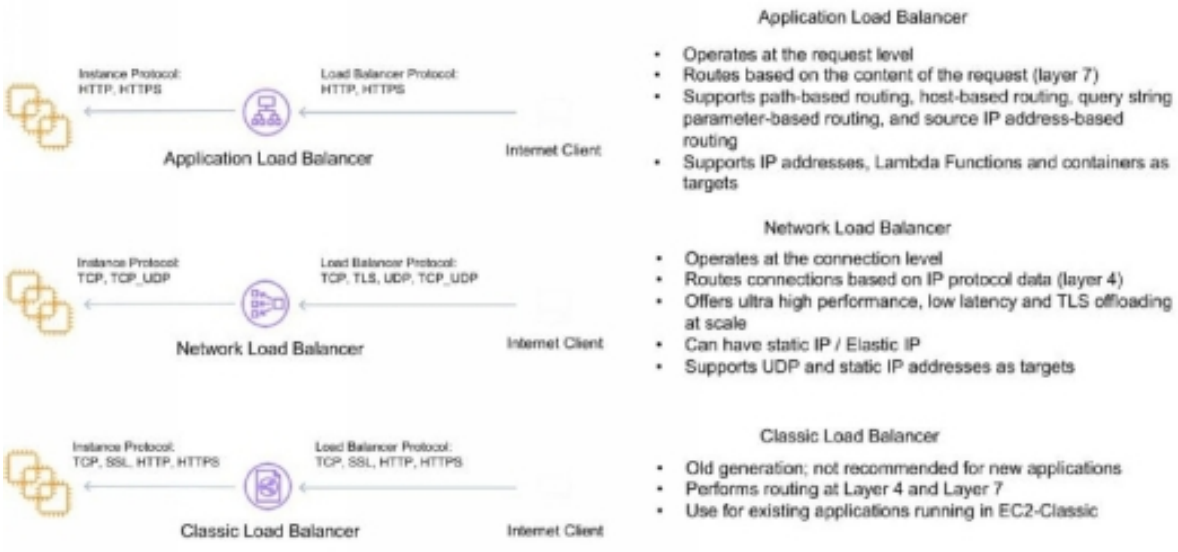
The Elastic Load Balancing (ELB) service on AWS distributes incoming connection requests to targets such as **Amazon EC2 instances** , containers, IP addresses, and AWS Lambda functions.

Traffic can be distributed across a single or multiple Availability Zones (AZs) within an AWS Region.

Elastic Load Balancing enables high availability, automatic scaling, and robust security necessary to make applications fault tolerant.

The Elastic Load Balancing service offers three different types of load balancer.

The following image provides an overview of some of the key differences between the three types of ELB:



APPLICATION LOAD BALANCER (ALB)

Application Load Balancer is best suited for load balancing of HTTP and HTTPS traffic and provides advanced request routing targeted at the

delivery of modern application architectures, including microservices and containers.

Operating at the individual request level (Layer 7), Application Load Balancer routes traffic to targets within Amazon Virtual Private Cloud (Amazon VPC) based on the content of the request.

Offers path-based routing and host-based routing to direct connections based on information in the request header.

Does not support a static IP address but does have a fixed DNS address.

You can put an NLB in front of an ALB to get a static IP address.

Provides SSL/TLS termination.

NETWORK LOAD BALANCER (NLB)

Network Load Balancer is best suited for load balancing of Transmission Control Protocol (TCP), User Datagram Protocol (UDP) and Transport Layer Security (TLS) traffic where extreme performance is required.

Operating at the connection level (Layer 4), Network Load Balancer routes traffic to targets within Amazon Virtual Private Cloud (Amazon VPC) and is capable of handling millions of requests per second while maintaining ultra-low latencies.

Network Load Balancer is also optimized to handle sudden and volatile traffic patterns.

NLB can be assigned a static / Elastic IP address (1 per subnet)

Also provides SSL/TLS termination.

CLASSIC LOAD BALANCER (CLB)

Classic Load Balancer provides basic load balancing across multiple Amazon EC2 instances and operates at both the request level and connection level. Classic Load Balancer is intended for applications that were built within the EC2-Classic network.

The CLB is the oldest ELB in AWS and is not covered much on the exam anymore and the remainder of this page covers concepts relating ONLY to the ALB and NLB.

DEPLOYMENT AND PROVISIONING

ELBs can be **Internet** facing or **internal-only** .

Internet facing ELB:

- ELB nodes have public IPs.
- Routes traffic to the private IP addresses of the EC2 instances.
- Need one public subnet in each AZ where the ELB is defined.
- ELB DNS name format: <name>-<id-number>.<region>.elb.amazonaws.com.

Internal only ELB:

- ELB nodes have private IPs.
- Routes traffic to the private IP addresses of the EC2 instances.
- ELB DNS name format: **internal** -<name>-<id-number>.<region>.elb.amazonaws.com.

ELB nodes use IP addresses within your subnets, ensure at least a /27 subnet and make sure there are at least 8 IP addresses available in order for the ELB to scale.

Deleting an ELB does not affect the instances registered against it (they won't be deleted, they just won't receive any more requests).

TARGET GROUPS

Target groups are a logical grouping of targets (EC2 instances or ECS).

Targets are the endpoints and can be EC2 instances, ECS containers, IP addresses, or AWS Lambda functions.

Target groups exist independently from the ELBTarget groups can have up to 1000 targets.

A single target can be in multiple target groups.

Only one protocol and one port can be defined per target group.

The target type in a target group can be an EC2 instance ID, IP address (must be a valid private IP from an existing subnet) or AWS Lambda Function (ALB only).

LISTENERS AND RULES

Listeners:

- Each ALB needs at least one listener and can have up to 10.
- Listeners define the port and protocol to listen on.
- Can add one or more listeners.
- Cannot have the same port in multiple listeners.

Listener rules:

- Rules determine how the load balancer routes requests to the targets in one or more target groups.
- Each rule consists of a priority, one or more actions, an optional host condition, and an optional path condition.
- Only one action can be configured per rule.
- One or more rules are required.
- Each listener has a default rule and you can optionally define additional rules.
- Up to 100 rules per ELB.
- Rules determine what action is taken when the rule matches the client request.
- Rules are defined on listeners.

- You can add rules that specify different target groups based on the content of the request (content-based routing).
- If no rules are found the default rule will be followed which directs traffic to the default target groups.

TROUBLESHOOTING YOUR AWS ELB

Common issues:

- A registered target is not in service:
  - **A security group does not allow traffic** – The security group associated with an instance must allow traffic from the load balancer using the health check port and health check protocol.
  - **A network access control list (ACL) does not allow traffic** – The network ACL associated with the subnets for your instances must allow inbound traffic on the health check port and outbound traffic on the ephemeral ports (1024-65535).
  - **The ping path does not exist** – Create a target page for the health check and specify its path as the ping path.
  - **The connection times out** – Verify that you can connect to the target directly from within the network using the private IP address of the target and the health check protocol.
  - **The target did not return a successful response code** – Confirm the success codes that the load balancer is expecting and that your application is configured to return these codes on success.
- Clients cannot connect to an internet-facing load balancer:
  - **Your Internet-facing load balancer is attached to a private subnet** – Verify that you specified public subnets for your load balancer.

- **A security group or network ACL does not allow traffic** – The security group for the load balancer and any network ACLs for the load balancer subnets must allow inbound traffic from the clients and outbound traffic to the clients on the listener ports.

- **The load balancer sends requests to unhealthy targets** – If there is at least one healthy target in a target group, the load balancer routes requests only to the healthy targets. If a target group contains only unhealthy targets, the load balancer routes requests to the unhealthy targets.

- **The load balancer sends a response code of 000** – With HTTP/2 connections, if the compressed length of any of the headers exceeds 8K, the load balancer sends a GOAWAY frame and closes the connection with a TCP FIN.

Some HTTP errors may be generated by the load balancer

The following HTTP errors are generated by the load balancer. The load balancer sends the HTTP code to the client, saves the request to the access log, and increments
the HTTPCode_ELB_4XX_Count or HTTPCode_ELB_5XX_Count metric
.

HTTP 400: Bad request – The client sent a malformed request that does not meet the HTTP specification. Or, the request header exceeded 16K per request line, 16K per single header, or 64K for the entire header.

HTTP 401: Unauthorized – You configured a listener rule to authenticate users. Either you configured OnUnauthenticatedRequest to deny unauthenticated users or the IdP denied access.

HTTP 403: Forbidden – You configured an AWS WAF web access control list (web ACL) to monitor requests to your Application Load Balancer and it blocked a request.

HTTP 405: Method not allowed – The client used the TRACE method, which is not supported by Application Load Balancers.

HTTP 408: Request timeout – The client did not send data before the idle timeout period expired. Sending a TCP keep-alive does not prevent this timeout. Send at least 1 byte of data before each idle timeout period elapses. Increase the length of the idle timeout period as needed.

HTTP 413: Payload too large – The target is a Lambda function and the request body exceeds 1 MB.

HTTP 414: URI too long – The request URL or query string parameters are too large.

HTTP 460 – The load balancer received a request from a client, but the client closed the connection with the load balancer before the idle timeout period elapsed.

HTTP 463 – The load balancer received an **X-Forwarded-For** request header with more than 30 IP addresses.

HTTP 500: Internal server error – Possible causes:

You configured an AWS WAF web access control list (web ACL) and there was an error executing the web ACL rules.

You configured a listener rule to authenticate users, but one of the following is true:

- The load balancer is unable to communicate with the IdP token endpoint or the IdP user info endpoint. Verify that the security groups for your load balancer and the network ACLs for your VPC allow outbound access to these endpoints. Verify that your VPC has internet access. If you have an internal-facing load balancer, use a NAT gateway to enable internet access.
- The size of the claims returned by the IdP exceeded the maximum size supported by the load balancer.
- A client submitted an HTTP/1.0 request without a host header, and the load balancer was unable to generate a redirect URL.
- A client submitted a request without an HTTP protocol, and the load balancer was unable to generate a redirect URL.

- The requested scope doesn’t return an ID token.

HTTP 501: Not implemented – The load balancer received a **Transfer-Encoding** header with an unsupported value. The supported values for **Transfer-Encoding** are chunked and identity. As an alternative, you can use the **Content-Encoding** header.

HTTP 502: Bad gateway – Possible causes:

- The load balancer received a TCP RST from the target when attempting to establish a connection.
- The load balancer received an unexpected response from the target, such as “ICMP Destination unreachable (Host unreachable)”, when attempting to establish a connection. Check whether traffic is allowed from the load balancer subnets to the targets on the target port.
- The target closed the connection with a TCP RST or a TCP FIN while the load balancer had an outstanding request to the target. Check whether the keep-alive duration of the target is shorter than the idle timeout value of the load balancer.
- The target response is malformed or contains HTTP headers that are not valid.
- The load balancer encountered an SSL handshake error or SSL handshake timeout (10 seconds) when connecting to a target.
- The deregistration delay period elapsed for a request being handled by a target that was deregistered. Increase the delay period so that lengthy operations can complete.
- The target is a Lambda function and the response body exceeds 1 MB.
- The target is a Lambda function that did not respond before its configured timeout was reached.

HTTP 503: Service unavailable – The target groups for the load balancer have no registered targets.

HTTP 504: Gateway timeout – Possible causes:

- The load balancer failed to establish a connection to the target before the connection timeout expired (10 seconds).
- The load balancer established a connection to the target but the target did not respond before the idle timeout period elapsed.
- The network ACL for the subnet did not allow traffic from the targets to the load balancer nodes on the ephemeral ports (1024-65535).
- The target returns a content-length header that is larger than the entity body. The load balancer timed out waiting for the missing bytes.
- The target is a Lambda function and the Lambda service did not respond before the connection timeout expired.

HTTP 561: Unauthorized – You configured a listener rule to authenticate users, but the IdP returned an error code when authenticating the user.

Targets can also generate HTTP errors.

The load balancer forwards valid HTTP responses from targets to the client, including HTTP errors. The HTTP errors generated by a target are recorded in
the HTTPCode_Target_4XX_Count and HTTPCode_Target_5XX_Count m etrics.

MONITORING AND REPORTING

Elastic Load Balancing publishes data points to **Amazon CloudWatch** for your load balancers and your targets.

CloudWatch enables you to retrieve statistics about those data points as an ordered set of time-series data, known as *metrics* .

Some of the key metrics reported for load balancers are:

- BackendConnectionErrors

- HealthyHostCount / UnhealthyHostCount
- HTTPCode_Backend_2XX – Successful request
- HTTPCode_Backend_3XX – Redirected request
- HTTPCode_ELB_4XX client error
- HTTPCode_ELB_5XX server error (generated by ELB)
- Latency
- RequestCount
- SurgeQueueLength – the total number of requests (HTTP listener) or connections (TCP listener) that are pending routing to a healthy instance. Can be used to scale out an Auto Scaling group. Max value is 1024.
- SpilloverCount – the total number of requests that were rejected because the surge queue is full.

For a detailed list of the metrics reported for load balancers and targets check the **AWS documentation here.**

LOGGING AND AUDITING

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in Elastic Load Balancing, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history** .

All Elastic Load Balancing actions for Application Load Balancers are logged by CloudTrail and are documented in the **Elastic Load Balancing API Reference version 2015-12-01** .

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or AWS Identity and Access Management (IAM) user credentials.

- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

ELB access logs can be stored on Amazon S3 and contain the following data:

- Time
- Client IP address
- Latencies
- Request paths
- Server response
- Trace ID

ELB and AWS X-Ray

- Elastic Load Balancing application load balancers add a trace ID to incoming HTTP requests in a header named X-Amzn-Trace-Id.
- Load balancers do not send data to X-Ray, and do not appear as a node on your service map.

AUTHORIZATION AND ACCESS CONTROL

Elastic Load Balancing supports **identity-based IAM policies** .

Elastic Load Balancing does not support resource-based policies.

Elastic Load Balancing has partial support for resource-level *permissions* .

For API actions that support resource-level permissions, you can control the resources that users are allowed to use with the action.

To specify a resource in a policy statement, you must use its Amazon Resource Name (ARN). When specifying an ARN, you can use the *

wildcard in your paths. For example, you can use the * wildcard when you do not want to specify the exact load balancer name.

AMAZON EC2 AUTO SCALING

Amazon EC2 Auto Scaling is an AWS service that automatically launches and terminates **Amazon Ec2** instances.

EC2 Auto Scaling is used for automatic, horizontal scaling of Amazon EC2 instances.

EC2 Auto Scaling uses scaling policies and metrics to determine when and how to scale out (launching instances) or in (terminating instances).

The Amazon EC2 Auto Scaling service performs the following three main functions:

- Monitors the health of running Amazon EC2 instances – uses health checks to identify if instances are healthy or unhealthy and whether they are able to receive traffic.
- Replaces impaired instances automatically – EC2 Auto Scaling automatically replaces instances that fail health checks.
- Balances capacity (number of instances) across Availability Zones (AZs) – when launching instances EC2 Auto Scaling tries to balance the instances across AZs.

SCHEDULED SCALING

Scaling based on a schedule allows you to scale your application ahead of known load changes.

For example, every week the traffic to your web application starts to increase on Wednesday, remains high on Thursday, and starts to decrease on Friday.

You can plan your scaling activities based on the known traffic patterns of your web application.

DYNAMIC SCALING

Amazon EC2 Auto Scaling enables you to follow the demand curve for your applications closely, reducing the need to manually provision Amazon EC2 capacity in advance.

For example, you can use target tracking scaling policies to select a load metric for your application, such as CPU utilization. Or, you could set a target value using the new “Request Count Per Target” metric from Application Load Balancer, a load balancing option for the Elastic Load Balancing service.

Amazon EC2 Auto Scaling will then automatically adjust the number of EC2 instances as needed to maintain your target.

PREDICTIVE SCALING

Predictive Scaling, a feature of AWS Auto Scaling uses machine learning to schedule the right number of EC2 instances in anticipation of approaching traffic changes.

Predictive Scaling predicts future traffic, including regularly-occurring spikes, and provisions the right number of EC2 instances in advance.

Predictive Scaling’s machine learning algorithms detect changes in daily and weekly patterns, automatically adjusting their forecasts.

This removes the need for manual adjustment of Auto Scaling parameters as cyclicality changes over time, making Auto Scaling simpler to configure.

Auto Scaling enhanced with Predictive Scaling delivers faster, simpler, and more accurate capacity provisioning resulting in lower cost and more responsive applications.

DEPLOYMENT AND PROVISIONING

EC2 Auto Scaling includes the following components:

GROUPS

EC2 instances are organized into *groups* so that they can be treated as a logical unit for the purposes of scaling and management. When you create a group, you can specify its minimum, maximum, and, desired number of EC2 instances.

CONFIGURATION TEMPLATES

Groups use a *launch template* or a *launch configuration* as a configuration template for its EC2 instances. You can specify information such as the AMI ID, instance type, key pair, security groups, and block device mapping for your instances.

SCALING OPTIONS

Amazon EC2 Auto Scaling provides several ways for you to scale your Auto Scaling groups. For example, you can configure a group to scale based on the occurrence of specified conditions (dynamic scaling) or on a schedule. For more information see “Scaling Options and Scaling Policies” below.

ACCESSING AMAZON EC2 AUTO SCALING

You can use the Amazon EC2 management console to manage EC2 Auto Scaling.

You can also access Amazon EC2 Auto Scaling using the **Amazon EC2 Auto Scaling API** .

Amazon EC2 Auto Scaling provides a Query API. These requests are HTTP or HTTPS requests that use the HTTP verbs GET or POST and a Query parameter named Action .

Can also use the AWS Command Line Interface (CLI) or the AWS Tools for Windows PowerShell.

SCALING OPTIONS AND SCALING POLICIES

The scaling options define the triggers and when instances should be provisioned/de-provisioned.

There are four scaling options:

- Maintain – keep a specific or minimum number of instances running.
- Manual – use maximum, minimum, or a specific number of instances.
- Scheduled – increase or decrease the number of instances based on a schedule.
- Dynamic – scale based on real-time system metrics (e.g. CloudWatch metrics).

The following table describes the scaling options that are available and when you should use them:

Scaling	What it is	When to use
Maintain	Ensures the required number of instances are running	Use when you always need a known number of instances running at all times
Manual	Manually change desired capacity via the console or CLI	Use when your needs change rarely enough that you're OK to make manual changes
Scheduled	Adjust min/max instances on specific dates/times or recurring time periods	Use when you know when your busy and quiet times are. Useful for ensuring enough instances are available *before* very busy times
Dynamic	Scale in response to system load or other triggers using metrics	Useful for changing capacity based on system utilization, e.g. CPU hits 80%

The scaling options are configured through Scaling Policies which determine when, if, and how the ASG scales and shrinks.

The following table describes the scaling policy types available for dynamic scaling policies and when to use them:

Scaling Policy	What it is	When to use
Target Tracking Policy	The scaling policy adds or removes capacity as required to keep the metric at, or close to, the specified target value	A use case is that you want to keep the aggregate CPU usage of your ASG at 70%
Simple Scaling Policy	Waits until health check and cool down period expires before re-evaluating	This is a more conservative way to add/remove instances. Useful when load is erratic. AWS recommend step scaling instead of simple in most cases
Step Scaling Policy	Increase or decrease the current capacity of your Auto Scaling group based on a set of scaling adjustments, known as step adjustments	Useful when you want to vary adjustments based on the size of the alarm breach

LAUNCH TEMPLATES VS LAUNCH CONFIGURATIONS

A launch template is similar to a **launch configuration** , in that it specifies instance configuration information. Included are the ID of the Amazon Machine Image (AMI), the instance type, a key pair, security groups, and the other parameters that you use to launch EC2 instances.

However, defining a launch template instead of a launch configuration allows you to have multiple versions of a template. With versioning, you can create a subset of the full set of parameters and then reuse it to create other templates or template versions. For example, you can create a default template that defines common configuration parameters and allow the other parameters to be specified as part of another version of the same template.

If you plan to continue to use launch configurations with Amazon EC2 Auto Scaling, be aware that not all Auto Scaling group features are available. For example, you cannot create an Auto Scaling group that launches both Spot and On-Demand Instances or that specifies multiple instance types.

EC2 AUTO SCALING LIFECYCLE HOOKS

Lifecycle hooks enable you to perform custom actions by pausing instances as an Auto Scaling group launches or terminates them.

When an instance is paused, it remains in a wait state either until you complete the lifecycle action using the complete-lifecycle-action command or the CompleteLifecycleAction operation, or until the timeout period ends (one hour by default).

Adding lifecycle hooks to your Auto Scaling group gives you greater control over how instances launch and terminate.

Instances can remain in a wait state for a finite period of time. The default is one hour (3600 seconds).

You can configure notifications for when an instance enters a wait state. You can use Amazon EventBridge, Amazon SNS, or Amazon SQS to receive the notifications.

HIGH AVAILABILITY

EC2 Auto Scaling can be used to implement high availability when you launch instances into at least two Availability Zones.

Use an Amazon Elastic Load Balancer or Amazon Route 53 to direct incoming connections to your EC2 instances.

EC2 Auto Scaling is a regional service so it cannot provide HA across multiple AWS Regions.

MONITORING AND REPORTING

When you enable Auto Scaling group metrics, your Auto Scaling group sends sampled data to **CloudWatch** every minute. There is no charge for enabling these metrics.

You can enable and disable Auto Scaling group metrics using the AWS Management Console, AWS CLI, or AWS SDKs.

The AWS/AutoScaling namespace includes the following metrics which are sent to CloudWatch every 1 minute:

- GroupMinSize
- GroupMaxSize
- GroupDesiredCapacity
- GroupInServiceInstances
- GroupPendingInstances
- GroupStandbyInstances
- GroupTerminatingInstances
- GroupTotalInstances

Metrics are also sent from the Amazon EC2 instances to Amazon CloudWatch:

- Basic monitoring sends EC2 metrics to CloudWatch about ASG instances every 5 minutes.
- Detailed can be enabled and sends metrics every 1 minute (chargeable).
- When the launch configuration is created from the console basic monitoring of EC2 instances is enabled by default.
- When the launch configuration is created from the CLI detailed monitoring of EC2 instances is enabled by default.

EC2 Auto Scaling uses health checks to ensure instances are healthy and available.

- By default Auto Scaling uses EC2 status checks.
- You can use ELB health checks and custom health checks in addition to the EC2 status checks.
- If any health check returns an unhealthy status the instance will be terminated.

- With ELB an instance is marked as unhealthy if ELB reports it as OutOfService.
- A healthy instance enters the InService state.
- If an instance is marked as unhealthy it will be scheduled for replacement.
- If connection draining is enabled, Auto Scaling waits for in-flight requests to complete or timeout before terminating instances.
- The health check grace period allows a period of time for a new instance to warm up before performing a health check (300 seconds by default).

Note : If using an ELB it is a best practice to enable ELB health checks as otherwise EC2 status checks may show an instance as being healthy that the ELB has determined is unhealthy. In this case the instance will be removed from service by the ELB but will not be terminated by Auto Scaling.

LOGGING AND AUDITING

Amazon CloudTrail captures all API calls for AWS Auto Scaling as events.

The calls captured include calls from the AWS Auto Scaling console and code calls to the AWS Auto Scaling API.

If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for AWS Auto Scaling.

If you don’t configure a trail, you can still view the most recent events in the CloudTrail console in **Event history** .

You can determine the requests that were made to AWS Auto Scaling, the IP address from which the requests were made, who made the requests, when they were made, and additional details.

AUTHORIZATION AND ACCESS CONTROL

EC2 Auto Scaling support **identity-based IAM policies** .

Amazon EC2 Auto Scaling does not support resource-based policies.

Amazon EC2 Auto Scaling uses **service-linked roles** for the permissions that it requires to call other AWS services on your behalf. A service-linked role is a unique type of IAM role that is linked directly to an AWS service.

There is a default service-linked role for your account, named **AWSServiceRoleForAutoScaling** . This role is automatically assigned to your Auto Scaling groups unless you specify a different service-linked role.

Amazon EC2 Auto Scaling also does not support Access Control Lists (ACLs).

You can apply tag-based, resource-level permissions in the identity-based policies that you create for Amazon EC2 Auto Scaling. This gives you better control over which resources a user can create, modify, use, or delete.

AMAZON EBS

Amazon Elastic Block Store volumes are network attached storage that can be attached to **EC2 instances** .

EBS volume data persists independently of the life of the instance.

EBS volumes do not need to be attached to an instance.

You can attach multiple EBS volumes to an instance.

EBS volume data is replicated across multiple servers in an Availability Zone (AZ).

EBS volumes must be in the same AZ as the instances they are attached to.

EBS VOLUME TYPES

SSD, General Purpose – GP2.

- Baseline of 3 IOPS per GiB with a minimum of 100 IOPS.
- Burst up to 3000 IOPS (for volumes >= 334GB).
- Up to 16,000 IOPS per volume.
- AWS designs gp2 volumes to deliver 90% of the provisioned performance 99% of the time. A gp2 volume can range in size from 1 GiB to 16 TiB.

SSD, Provisioned IOPS – I01.

- More than 16,000 IOPS.
- Up to 64,000 IOPS per volume.
- Up to 50 IOPS per GiB.

- Amazon EBS delivers the provisioned IOPS performance 99.9 percent of the time.

HDD, Throughput Optimized – (ST1):

- Frequently accessed, throughput intensive workloads with large datasets and large I/O sizes, such as MapReduce, Kafka, log processing, data warehouse, and ETL workloads.
- Throughput measured in MB/s, and includes the ability to burst up to 250 MB/s per TB, with a baseline throughput of 40 MB/s per TB and a maximum throughput of 500 MB/s per volume.
- Cannot be a boot volume.

HDD, Cold – (SC1):

- Lowest cost storage – cannot be a boot volume.
- Less frequently accessed workloads with large, cold datasets.
- These volumes can burst up to 80 MB/s per TB, with a baseline throughput of 12 MB/s per TB and a maximum throughput of 250 MB/s per volume.

HDD, Magnetic – Standard – cheap, infrequently accessed storage – lowest cost storage that can be a boot volume (AWS documentation does not reflect this but you CAN still choose magnetic when launching an instance).

EBS optimized instances:

- Dedicated capacity for Amazon EBS I/O.
- EBS-optimized instances are designed for use with all EBS volume types.
- Max bandwidth: 400 Mbps – 12000 Mbps.
- IOPS: 3000 – 65000.
- GP-SSD within 10% of baseline and burst performance 99.9% of the time.

- PIOPS within 10% of baseline and burst performance 99.9% of the time.
- Additional hourly fee.
- Available for select instance types.
- Some instance types have EBS-optimized enabled by default.

INSTANCE STORE

An instance store provides *temporary* (non-persistent) block-level storage for your instance.

This is different to EBS which provides persistent storage but is also a block storage service that can be a root or additional volume.

Instance store storage is located on disks that are physically attached to the host computer.

Instance store is ideal for temporary storage of information that changes frequently, such as buffers, caches, scratch data, and other temporary content, or for data that is replicated across a fleet of instances, such as a load-balanced pool of web servers.

EBS VS INSTANCE STORE

EBS-backed means the root volume is an EBS volume and storage is persistent.

Instance store-backed means the root volume is an instance store volume and storage is not persistent.

On an EBS-backed instance, the default action is for the root EBS volume to be deleted upon termination.

Instance store volumes are sometimes called Ephemeral storage (non-persistent).

Instance store backed instances cannot be stopped. If the underlying host fails the data will be lost.

Instance store volume root devices are created from AMI templates stored on S3.

EBS backed instances can be stopped. You will not lose the data on this instance if it is stopped (persistent).

EBS volumes can be detached and reattached to other EC2 instances.

EBS volume root devices are launched from AMI's that are backed by EBS snapshots.

Instance store volumes cannot be detached/reattached.

When rebooting the instances for both types data will not be lost.

By default, both root volumes will be deleted on termination unless you configured otherwise.

SNAPSHOTS

Snapshots capture a point-in-time state of an instance.

Cost-effective and easy backup strategy.

Share data sets with other users or accounts.

Can be used to migrate a system to a new AZ or region.

Can be used to convert an unencrypted volume to an encrypted volume.

Snapshots are stored on Amazon S3.

If you make periodic snapshots of a volume, the snapshots are incremental, which means that only the blocks on the device that have changed after your last snapshot are saved in the new snapshot.

Even though snapshots are saved incrementally, the snapshot deletion process is designed so that you need to retain only the most recent snapshot in order to restore the volume.

Snapshots can only be accessed through the EC2 APIs.

EBS volumes are AZ specific but snapshots are region specific.

ENCRYPTION

You can encrypt both the boot and data volumes of an **EC2 instance** . When you create an encrypted EBS volume and attach it to a supported instance type, the following types of data are encrypted:

- Data at rest inside the volume.
- All data moving between the volume and the instance.
- All snapshots created from the volume.
- All volumes created from those snapshots.

Expect the same IOPS performance on encrypted volumes as on unencrypted volumes.

EBS encrypts your volume with a data key using the industry-standard AES-256 algorithm.

Your data key is stored on-disk with your encrypted data, but not before EBS encrypts it with your CMK. Your data key never appears on disk in plaintext.

The same data key is shared by snapshots of the volume and any subsequent volumes created from those snapshots.

Snapshots of encrypted volumes are encrypted automatically.

EBS volumes restored from encrypted snapshots are encrypted automatically.

EBS volumes created from encrypted snapshots are also encrypted.

You can share snapshots, but if they’re encrypted it must be with a custom CMK key.

You can check the encryption status of your EBS volumes with **AWS Config** .

AMIS

An Amazon Machine Image (AMI) is a special type of virtual appliance that is used to create a virtual machine within the Amazon Elastic Compute Cloud ("EC2").

An AMI includes the following:

- A template for the root volume for the instance (for example, an operating system, an application server, and applications).
- Launch permissions that control which AWS accounts can use the AMI to launch instances.
- A block device mapping that specifies the volumes to attach to the instance when it's launched.

AMIs are either instance store-backed or EBS-backed.

Copying AMIs:

- You can copy an Amazon Machine Image (AMI) within or across an AWS region using the AWS Management Console, the AWS AWS Command Line Interface or SDKs, or the Amazon EC2 API, all of which support theCopyImage action.
- You can copy both Amazon EBS-backed AMIs and instance store-backed AMIs.
- You can copy encrypted AMIs and AMIs with encrypted snapshots.

DEPLOYMENT AND PROVISIONING

EBS VOLUMES

Termination protection is turned off by default and must be manually enabled (keeps the volume/data when the instance is terminated).

Root EBS volumes are deleted on termination by default.

Extra non-boot volumes are not deleted on termination by default.

The behaviour can be changed by altering the “DeleteOnTermination” attribute.

Volume sizes and types can be upgraded without downtime (except for magnetic standard).

Elastic Volumes allow you to increase volume size, adjust performance, or change the volume type while the volume is in use.

To migrate volumes between AZ’s create a snapshot then create a volume in another AZ from the snapshot (possible to change size and type).

EBS SNAPSHOTS

Volumes can be created from EBS snapshots that are the same size or larger.

Snapshots can be taken of non-root EBS volumes while running.

To take a consistent snapshots writes must be stopped (paused) until the snapshot is complete – if not possible the volume needs to be detached, or if it’s an EBS root volume the instance must be stopped.

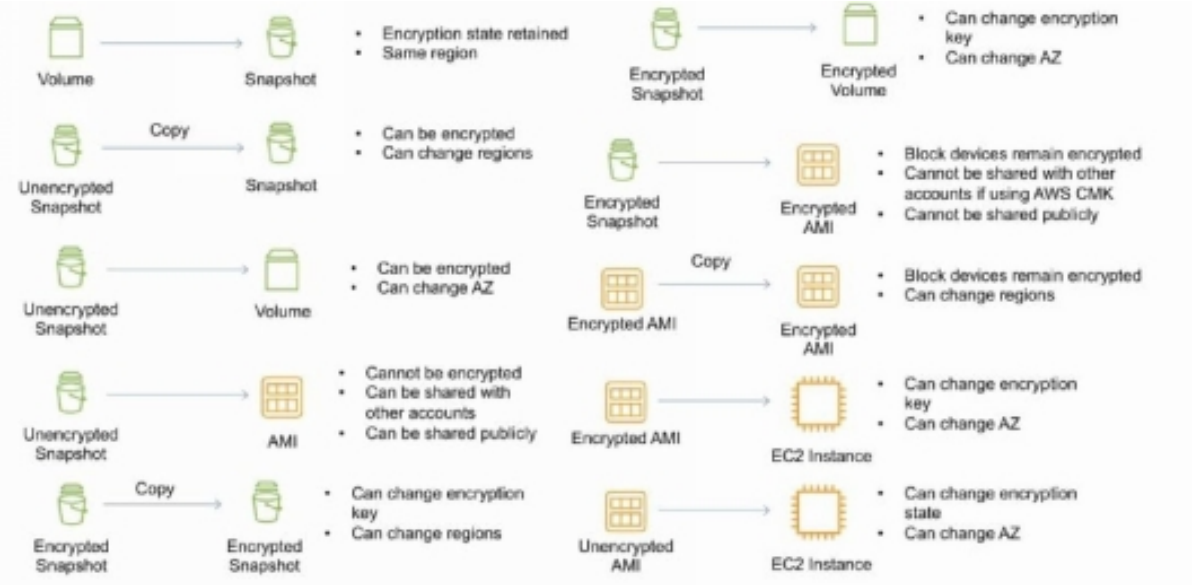
You can resize volumes through restoring snapshots with different sizes (configured when taking the snapshot).

Snapshots can be copied between regions (and be encrypted). Images are then created from the snapshot in the other region which creates an AMI that can be used to boot an instance.

You can create volumes from snapshots and choose the availability zone within the region.

EBS COPYING, SHARING AND ENCRYPTION METHODS

The following diagram depicts the various options for copying EBS volumes, sharing AMIs and snapshots and applying encryption:



REDUNDANT ARRAY OF INDEPENDENT DISKS (RAID)

RAID can be used to increase IOPS.

RAID 0 = 0 striping – data is written across multiple disks and increases performance but no redundancy.

- Use 2 or more disks.
- If one disk fails the entire RAID set fails.

RAID 1 = 1 mirroring – creates 2 copies of the data but does not increase performance, only redundancy.

- If one disk fails, the other is still working.
- Data gets sent to 2 EBS volumes at the same time.

RAID 0 and RAID 1 are potential options on EBS.

RAID 5 and RAID 6 are not recommended by AWS.

You can configure multiple striped gp2 or standard volumes (typically RAID 0).

You can configure multiple striped PIOPS volumes (typically RAID 0).

RAID is configured through the guest OS.

EBS optimized EC2 instances are another way of increasing performance.

Ensure the EC2 instance can handle the bandwidth required for the increased performance.

Use EBS optimized instances or instances with a 10 Gbps network interface.

Not recommended to use RAID for root/boot volumes.

MONITORING AND REPORTING

Amazon Elastic Block Store (Amazon EBS) sends data points to CloudWatch for **several metrics** .

There are two types of Amazon CloudWatch monitoring available for Amazon EBS volumes:

- Basic – Data is available automatically in 5-minute periods at no charge. This includes data for the root device volumes for EBS-backed instances.
- Detailed – Provisioned IOPS SSD (io1) volumes automatically send one-minute metrics to CloudWatch.

Amazon EBS General Purpose SSD (gp2), Throughput Optimized HDD (st1) , Cold HDD (sc1), and Magnetic (standard) volumes automatically send five-minute metrics to CloudWatch.

Provisioned IOPS SSD (io1) volumes automatically send one-minute metrics to CloudWatch. Data is only reported to CloudWatch when the volume is attached to an instance.

Volume **status checks** enable you to better understand, track, and manage potential inconsistencies in the data on an Amazon EBS volume.

Volume Status	I/O Enabled Status	I/O performance status (only available for Provisioned IOPS volumes)
ok	Enabled (I/O Enabled or I/O Auto-Enabled)	Normal (Volume performance is as expected)
warning	Enabled (I/O Enabled or I/O Auto-Enabled)	Degraded (Volume performance is below expectations) Severely Degraded (Volume performance is well below expectations)
impaired	Enabled (I/O Enabled or I/O Auto-Enabled) Disabled (Volume is offline and pending recovery, or is waiting for the user to enable I/O)	Stalled (Volume performance is severely impacted) Not Available (Unable to determine I/O performance because I/O is disabled)
insufficient-data	Enabled (I/O Enabled or I/O Auto-Enabled) Insufficient Data	Insufficient Data

LOGGING AND AUDITING

Amazon EC2 and Amazon EBS are integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Amazon EC2 and Amazon EBS.

CloudTrail captures all API calls for Amazon EC2 and Amazon EBS as events, including calls from the console and from code calls to the APIs.

AMAZON EFS

Amazon EFS is a fully-managed service for hosting Network File System (NFS) filesystems in the cloud.

It is an implementation of a NFS file share and is accessed using the NFS protocol.

It provides elastic storage capacity and pay for what you use (in contrast to **Amazon EBS** with which you pay for what you provision).

You can configure mount-points in one, or many, AZs.

You can mount an AWS EFS filesystem from on-premises systems ONLY if you are using AWS Direct Connect or a VPN connection.

Typical use cases include big data and analytics, media processing workflows, content management, web serving, home directories etc.

Uses a pay for what you use model with no pre-provisioning required.

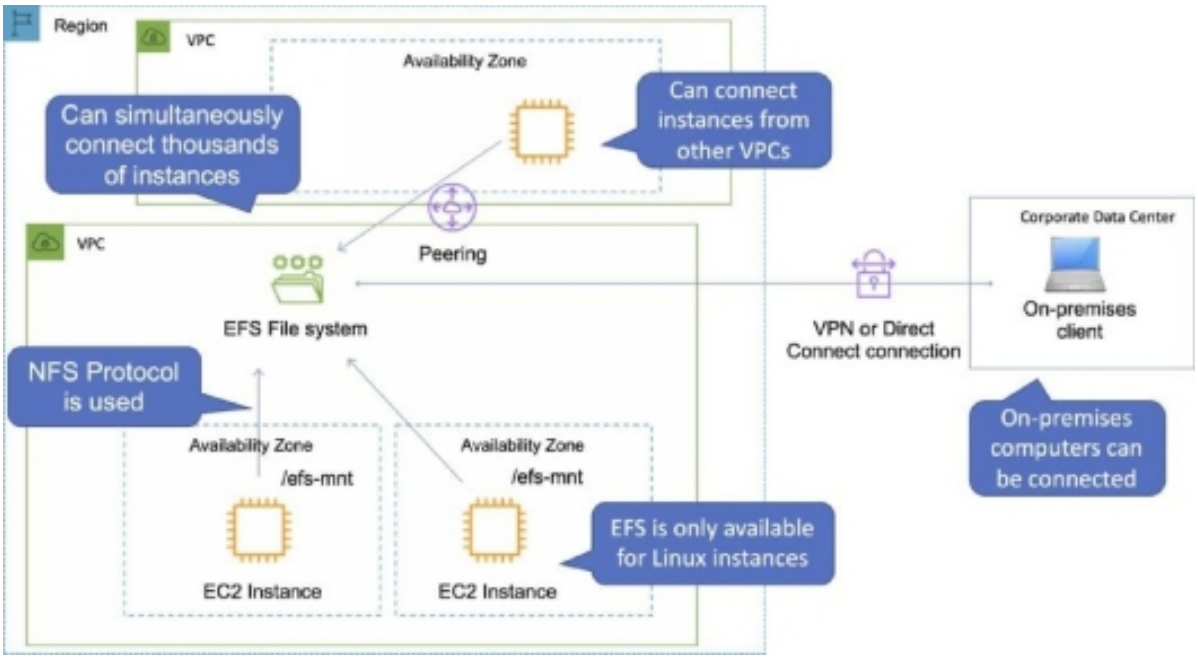
AWS EFS can scale up to petabytes.

AWS EFS is elastic and grows and shrinks as you add and remove data.

You can concurrently connect up to thousands of **Amazon EC2** instances, from multiple AZs.

A file system can be accessed concurrently from all AZs in the region where it is located.

The following diagram depicts the various options for mounting an EFS filesystem:



Access to AWS EFS file systems from on-premises servers can be enabled via AWS Direct Connect or AWS VPN.

You mount an AWS EFS file system on your on-premises Linux server using the standard Linux mount command for mounting a file system via the NFS protocol.

Can choose General Purpose or Max I/O (both SSD).

The Amazon VPC of the connecting instance must have DNS hostnames enabled.

EFS provides a file system interface, file system access semantics (such as strong consistency and file locking).

Data is stored across multiple AZ's within a region.

Read after write consistency.

Need to create mount targets and choose AZ's to include (recommended to include all AZ's).

Instances can be behind an **Elastic Load Balancer** (ELB).

EC2 Classic instances must mount via ClassicLink.

EFS is compatible with all Linux-based AMIs for Amazon EC2.

Using the EFS-to-EFS Backup solution, you can schedule automatic incremental backups of your Amazon EFS file system.

The following table provides a comparison of the storage characteristics of EFS vs EBS:

	Amazon EFS	Amazon EBS Provisioned IOPS
Availability and durability	Data is stored redundantly across multiple AZs.	Data is stored redundantly in a single AZ.
Access	Up to thousands of Amazon EC2 instances, from multiple AZs, can connect concurrently to a file system.	A single Amazon EC2 instance in a single AZ can connect to a file system.
Use cases	Big data and analytics, media processing workflows, content management, web serving, and home directories.	Boot volumes, transactional and NoSQL databases, data warehousing, and ETL.

AMAZON EFS BACKUPS AND LIFECYCLE MANAGEMENT

Automatic backups are enabled by default and use AWS Backup.

Lifecycle management moves files that have not been accessed for a period of time to the EFS Infrequent Access Storage class.

AMAZON EFS PERFORMANCE

There are two performance modes:

- “General Purpose” performance mode is appropriate for most file systems.
- “Max I/O” performance mode is optimized for applications where tens, hundreds, or thousands of EC2 instances are accessing the file system.

Amazon EFS is designed to burst to allow high throughput levels for periods of time.

There are two throughput modes:

- “Bursting” – throughput scales with file system size.
- “Provisioned” – Throughput is fixed at the specified amount.

Amazon EFS file systems are distributed across an unconstrained number of storage servers, enabling file systems to grow elastically to petabyte scale and allowing massively parallel access from Amazon EC2 instances to your data.

This distributed data storage design means that multithreaded applications and applications that concurrently access data from multiple Amazon EC2 instances can drive substantial levels of aggregate throughput and IOPS.

The table below compares high-level performance and storage characteristics for AWS’s file (EFS) and block (EBS) cloud storage offerings:

	Amazon EFS	Amazon EBS Provisioned IOPS
Per-operation latency	Low, consistent latency.	Lowest, consistent latency.
Throughput scale	10+ GB per second.	Up to 2 GB per second.

AMAZON EFS ENCRYPTION

EFS offers the ability to encrypt data at rest and in transit.

Encryption keys are managed by the AWS Key Management Service (KMS).

Data encryption in transit uses Transport Layer Security (TLS) 1.2 to encrypt data sent between your clients and EFS file systems.

Data encrypted at rest is transparently encrypted while being written, and transparently decrypted while being read.

Enable encryption at rest in the EFS console or by using the AWS CLI or SDKs.

Encryption of data at rest and of data in transit can be configured together or separately.

AMAZON EFS ACCESS CONTROL

When you create a file system, you create endpoints in your VPC called “mount targets”.

When mounting from an EC2 instance, your file system’s DNS name, which you provide in your mount command, resolves to a mount target’s IP address.

You can control who can administer your file system using IAM (user-based and resource-based policies)

You can control the NFS clients that can access your file systems (resource-based policies).

You can control access to files and directories with POSIX-compliant user and group-level permissions.

POSIX permissions allow you to restrict access from hosts by user and group.

EFS Security Groups act as a firewall, and the rules you add define the traffic flow.

MONITORING AND REPORTING

The Amazon EFS console shows the following monitoring information for your file systems:

- The current metered size.
- The number of mount targets.
- The lifecycle state.

Amazon EFS reports metrics for Amazon CloudWatch. A few useful metrics are:

- TotalIOBytes – use the daily Sum statistic to determine throughput.
- ClientConnections – use the daily Sum statistic to track the number of connections from EC2 instances.
- BurstCreditBalance – monitor the burst credit balance.

LOGGING AND AUDITING

Amazon EFS is integrated with AWS CloudTrail.

CloudTrail captures all API calls for Amazon EFS as events, including calls from the Amazon EFS console and from code calls to Amazon EFS API operations.

AWS STORAGE GATEWAY

The AWS Storage Gateway service enables hybrid storage between on-premises environments and the AWS Cloud.

It provides low-latency performance by caching frequently accessed data on premises, while storing data securely and durably in Amazon cloud storage services.

Implemented using a virtual machine that you run on-premises (VMware or Hyper-V virtual appliance).

Provides local storage resources backed by **Amazon S3** and **Glacier** .

Often used in disaster recovery preparedness to sync data to AWS.

AWS Storage Gateway supports three storage interfaces: file, volume, and tape.

The table below shows the different gateways available and the interfaces and use cases:

Name	Interface	Use Case
File Gateway	NFS, SMB	Allow on-prem or EC2 instances to store objects in S3 via NFS or SMB mount points
Volume Gateway Stored Mode	iSCSI	Asynchronous replication of on-premises data to S3
Volume Gateway Cached Mode	iSCSI	Primary data stored in S3 with frequently accessed data cached locally (on-premises)
Tape Gateway	iSCSI	Virtual media changer and tape library for use with existing backup software

Each gateway you have can provide one type of interface.

All data transferred between any type of gateway appliance and AWS storage is encrypted using SSL.

By default, all data stored by AWS Storage Gateway in S3 is encrypted server-side with Amazon S3-Managed Encryption Keys (SSE-S3).

When using the file gateway, you can optionally configure each file share to have your objects encrypted with AWS KMS-Managed Keys using SSE-KMS.

FILE GATEWAY

File gateway provides a virtual on-premises file server, which enables you to store and retrieve files as objects in Amazon S3.

Can be used for on-premises applications, and for Amazon EC2-resident applications that need file storage in S3 for object based workloads.

Used for flat files only, stored directly on S3.

File gateway offers SMB or NFS-based access to data in Amazon S3 with local caching.

File gateway supports Amazon S3 Standard, S3 Standard – Infrequent Access (S3 Standard – IA) and S3 One Zone – IA.

File gateway supports clients connecting to the gateway using NFS v3 and v4.1.

Microsoft Windows clients that support SMB can connect to file gateway.

The maximum size of an individual file is 5 TB.

VOLUME GATEWAY

The volume gateway represents the family of gateways that support block-based volumes, previously referred to as gateway-cached and gateway-stored modes.

Block storage – iSCSI based.

Cached Volume mode – the entire dataset is stored on S3 and a cache of the most frequently accessed data is cached on-site.

Stored Volume mode – the entire dataset is stored on-site and is asynchronously backed up to S3 (EBS point-in-time snapshots). Snapshots are incremental and compressed.

Each volume gateway can support up to 32 volumes.

In cached mode, each volume can be up to 32 TB for a maximum of 1 PB of data per gateway (32 volumes, each 32 TB in size).

In stored mode, each volume can be up to 16 TB for a maximum of 512 TB of data per gateway (32 volumes, each 16 TB in size).

GATEWAY VIRTUAL TAPE LIBRARY

Used for backup with popular backup software.

Each gateway is preconfigured with a media changer and tape drives. Supported by NetBackup, Backup Exec, Veeam etc.

When creating virtual tapes, you select one of the following sizes: 100 GB, 200 GB, 400 GB, 800 GB, 1.5 TB, and 2.5 TB.

A tape gateway can have up to 1,500 virtual tapes with a maximum aggregate capacity of 1 PB.

MANAGING AWS STORAGE GATEWAY

You might need to shutdown or reboot your VM for maintenance, such as when applying a patch to your hypervisor. Before you shutdown the VM, you must first stop the gateway.

- For file gateway, you just shutdown your VM.
- For volume and tape gateways, stop the gateway, reboot the VM, then start the gateway.

MONITORING AWS STORAGE GATEWAY

The following metrics are useful when **monitoring** cache usage for file, cached-volume, and tape gateways.

Metric	Description	Applies to
CacheHitPercent	Percent of application reads served from the cache. The sample is taken at the end of the reporting period. Unit: Percent	File, cached-volume, and tape gateways.
CacheUsed	The total number of bytes being used in the gateway's cache storage. The sample is taken at the end of the reporting period. Unit: Bytes	File, cached-volume, and tape gateways.

AWS SYSTEMS MANAGER

AWS Systems Manager is an AWS service that provides visibility and control of infrastructure on AWS.

AWS Systems Manager provides a unified interface through which you can view operational data from multiple AWS services.

AWS Systems Manager allows you to automate operational tasks across your AWS resources.

With Systems Manager, you can group resources, like **Amazon EC2** instances, Amazon S3 buckets, or Amazon RDS instances, by application, view operational data for monitoring and troubleshooting, and take action on your groups of resources.

You can create logical groups of resources such as applications, different layers of an application stack, or production versus development environments.

With Systems Manager, you can select a resource group and view its recent API activity, resource configuration changes, related notifications, operational alerts, software inventory, and patch compliance status.

You can also take action on each resource group depending on your operational needs.

Systems Manager simplifies resource and application management, shortens the time to detect and resolve operational problems, and makes it easy to operate and manage your infrastructure securely at scale.

SYSTEMS MANAGER COMPONENTS

SYSTEMS MANAGER INVENTORY

AWS Systems Manager collects information about your instances and the software installed on them, helping you to understand your system configurations and installed applications.

You can collect data about applications, files, network configurations, Windows services, registries, server roles, updates, and any other system properties.

The gathered data enables you to manage application assets, track licenses, monitor file integrity, discover applications not installed by a traditional installer, and more.

CONFIGURATION COMPLIANCE

AWS Systems Manager lets you scan your managed instances for patch compliance and configuration inconsistencies.

You can collect and aggregate data from multiple AWS accounts and Regions, and then drill down into specific resources that aren't compliant.

By default, AWS Systems Manager displays data about patching and associations. You can also customize the service and create your own compliance types based on your requirements.

AUTOMATION

AWS Systems Manager allows you to safely automate common and repetitive IT operations and management tasks across AWS resources.

With Systems Manager, you can create JSON/YAML documents that specify a specific list of tasks or use community published documents.

These documents can be executed directly through the AWS Management Console, CLIs, and SDKs, scheduled in a maintenance window, or triggered based on changes to AWS resources through Amazon CloudWatch Events.

You can track the execution of each step in the documents as well as require approvals for each step.

You can also incrementally roll out changes and automatically halt when errors occur.

RUN COMMAND

Use Systems Manager Run Command to remotely and securely manage the configuration of your managed instances at scale. Use Run Command to perform on-demand changes like updating applications or running Linux shell scripts and Windows PowerShell commands on a target set of dozens or hundreds of instances.

Run command requires the SSM agent to be installed on all managed instances.

Example tasks include: stopping, restarting, terminating, and resizing instances. Attaching and detaching EBS volumes, creating snapshots etc.

Often used to apply patches and updates.

Commands can be applied to a group of systems based on AWS instance tags or manual selection.

The commands and parameters are defined in a Systems Manager document.

Commands can be issued using the AWS Console, AWS CLI, AWS Tools for Windows PowerShell, the Systems Manager API, or Amazon SDKs.

SESSION MANAGER

AWS Systems Manager provides you safe, secure remote management of your instances at scale without logging into your servers, replacing the need for bastion hosts, SSH, or remote PowerShell.

It provides a simple way of automating common administrative tasks across groups of instances such as registry edits, user management, and software and patch installations.

Provides a command terminal for Linux instances and Windows PowerShell terminal for Windows instances.

Through integration with AWS Identity and Access Management (IAM), you can apply granular permissions to control the actions users can perform on instances.

All actions taken with Systems Manager are recorded by AWS CloudTrail, allowing you to audit changes throughout your environment.

Requires IAM permissions for EC2 instance to access SSM, S3, and CloudWatch Logs.

CloudTrail can intercept StartSession events using Session Manager.

Compared to SSH:

- Do not need to open port 22.
- Do not need bastion hosts for management.
- All commands are logged to S3 / CloudWatch.
- Secure shell access is authenticated using IAM user accounts, not key pairs.

PATCH MANAGER

AWS Systems Manager helps you select and deploy operating system and software patches automatically across large groups of Amazon EC2 or on-premises instances.

Through patch baselines, you can set rules to auto-approve select categories of patches to be installed, such as operating system or high severity patches, and you can specify a list of patches that override these rules and are automatically approved or rejected.

You can also schedule maintenance windows for your patches so that they are only applied during preset times.

Systems Manager helps ensure that your software is up-to-date and meets your compliance policies.

MAINTENANCE WINDOWS

AWS Systems Manager lets you schedule windows of time to run administrative and maintenance tasks across your instances.

This ensures that you can select a convenient and safe time to install patches and updates or make other configuration changes, improving the

availability and reliability of your services and applications.

DISTRIBUTOR

Distributor is an AWS Systems Manager feature that enables you to securely store and distribute software packages in your organization.

You can use Distributor with existing Systems Manager features like Run Command and State Manager to control the lifecycle of the packages running on your instances.

STATE MANAGER

AWS Systems Manager provides configuration management, which helps you maintain consistent configuration of your Amazon EC2 or on-premises instances.

With Systems Manager, you can control configuration details such as server configurations, anti-virus definitions, firewall settings, and more.

You can define configuration policies for your servers through the AWS Management Console or use existing scripts, PowerShell modules, or Ansible playbooks directly from GitHub or Amazon S3 buckets.

Systems Manager automatically applies your configurations across your instances at a time and frequency that you define.

You can query Systems Manager at any time to view the status of your instance configurations, giving you on-demand visibility into your compliance status.

PARAMETER STORE

AWS Systems Manager provides a centralized store to manage your configuration data, whether plain-text data such as database strings or secrets such as passwords.

This allows you to separate your secrets and configuration data from your code. Parameters can be tagged and organized into hierarchies, helping you manage parameters more easily.

For example, you can use the same parameter name, “db-string”, with a different hierarchical path, “dev/db-string” or “prod/db-string”, to store different values.

Systems Manager is integrated with AWS Key Management Service (KMS), allowing you to automatically encrypt the data you store.

You can also control user and resource access to parameters using AWS Identity and Access Management (IAM). Parameters can be referenced through other AWS services, such as Amazon Elastic Container Service, AWS Lambda, and AWS CloudFormation.

DEPLOYMENT AND PROVISIONING

RESOURCE GROUPS

You can use *resource groups* to organize your AWS resources. Resource groups make it easier to manage, monitor, and automate tasks on large numbers of resources at one time.

AWS Resource Groups provides two general methods for defining a resource group. Both methods involve using a query to identify the members for a group.

The first method relies on tags applied to AWS resources to add resources to a group. Using this method, you apply the same key/value pair tags to resources of various types in your account and then use the AWS Resource Groups service to create a group based on that tag pair.

The second method is based on resources available in an individual AWS CloudFormation stack. Using this method, you choose an AWS CloudFormation stack, and then choose resource types in the stack that you want to be in the group.

Allows the creation of logical groups of resources that you can perform actions on (such as patching).

Resource groups are regional in scope.

SYSTEMS MANAGER DOCUMENT

An AWS Systems Manager document (SSM document) defines the actions that Systems Manager performs on your managed instances.

Systems Manager includes more than a dozen pre-configured documents that you can use by specifying parameters at runtime.

Documents use JavaScript Object Notation (JSON) or YAML, and they include steps and parameters that you specify.

MONITORING AND REPORTING

INSIGHTS DASHBOARD

AWS Systems Manager automatically aggregates and displays operational data for each resource group through a dashboard.

Systems Manager eliminates the need for you to navigate across multiple AWS consoles to view your operational data.

With Systems Manager you can view API call logs from **AWS CloudTrail** , resource configuration changes from **AWS Config** , software inventory, and patch compliance status by resource group.

You can also easily integrate your **AWS CloudWatch** Dashboards, **AWS Trusted Advisor** notifications, and **AWS Personal Health Dashboard** performance and availability alerts into your Systems Manager dashboard.

Systems Manager centralizes all relevant operational data, so you can have a clear view of your infrastructure compliance and performance.

AMAZON CLOUDWATCH

You can configure and use the Amazon CloudWatch agent to collect metrics and logs from your instances instead of using SSM Agent for these tasks. The CloudWatch agent enables you to gather more metrics on EC2 instances than are available using SSM Agent. In addition, you can gather metrics from on-premises servers using the CloudWatch agent.

LOGGING AND AUDITING

Systems Manager is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Systems Manager. CloudTrail captures all API calls for Systems Manager as events, including calls from the Systems Manager console and from code calls to the Systems Manager APIs.

SSM Agent writes information about executions, commands, scheduled actions, errors, and health statuses to log files on each instance. You can view log files by manually connecting to an instance, or you can automatically send logs to Amazon CloudWatch Logs.

AUTHORIZATION AND ACCESS CONTROL

AWS Systems Manager supports **identity-based policies** .

AWS Systems Manager does not support **resource-based policies** .

You can attach tags to Systems Manager resources or pass tags in a request to Systems Manager.

To control access based on tags, you provide tag information in the condition element of a policy using the ssm:resourceTag/key-name, aws:ResourceTag/key-name, aws:RequestTag/key-name, or aws:TagKeys condition keys.

AWS OPSWORKS

AWS OpsWorks is a configuration management service that provides managed instances of Chef and Puppet two very popular automation platforms.

AWS OpsWorks automates how applications are configured, deployed and managed.

Provides configuration management to deploy code, automate tasks, configure instances, perform upgrades etc.

OpsWorks lets you use Chef and Puppet to automate how servers are configured, deployed, and managed across your **Amazon EC2** instances or on-premises compute environments.

OpsWorks is an automation platform that transforms infrastructure into code.

OpsWorks consists of Stacks and Layers:

- Stack are collections of resources needed to support a service or application.
- Stacks are containers of resources (EC2, RDS etc.) that you want to manage collectively.
- Every Stack contains one or more Layers and Layers automate the deployment of packages.
- Stacks can be cloned – but only within the same region.
- Layers represent different components of the application delivery hierarchy.
- EC2 instances, RDS instances, and ELBS are examples of Layers.

OpsWorks is a global service. But when you create a stack, you must specify a region and that stack can only control resources in that region.

There are three offerings: OpsWorks for Chef Automate, OpsWorks for Puppet Enterprise, and OpsWorks Stacks.

AWS OpsWorks for Chef Automate

- A fully-managed configuration management service that hosts Chef Automate, a suite of automation tools from Chef for configuration management, compliance and security, and continuous deployment.
- Completely compatible with tooling and cookbooks from the Chef community and automatically registers new nodes with your Chef server.
- Chef server stores recipes and configuration data.
- Chef client (node) is installed on each server.

AWS OpsWorks for Puppet Enterprise

- A fully-managed configuration management service that hosts Puppet Enterprise, a set of automation tools from Puppet for infrastructure and application management.

AWS OpsWorks Stacks

- An application and server management service that allows you to model your application as a stack containing different layers, such as load balancing, database, and application server.
- OpsWorks Stacks is an AWS creation and uses and embedded Chef Solo client installed on EC2 instances to run Chef recipes.

OpsWorks Stacks supports EC2 instances and on-premise servers as well as an agent.

AWS ELASTIC BEANSTALK

AWS Elastic Beanstalk can be used to quickly deploy and manage applications in the AWS Cloud.

Developers upload applications and Elastic Beanstalk handles the deployment details of capacity provisioning, load balancing, auto-scaling, and application health monitoring.

AWS Elastic Beanstalk leverages Elastic Load Balancing and Auto Scaling to automatically scale your application in and out based on your application's specific needs.

In addition, multiple availability zones give you an option to improve application reliability and availability by running in more than one zone.

Considered a Platform as a Service (PaaS) solution.

Elastic Beanstalk has some similarities with **AWS CloudFormation** though they are also quite different as detailed in the table below:

CloudFormation	Elastic Beanstalk
"Template-driven provisioning"	"Web apps made easy"
Deploys infrastructure using code	Deploys applications on EC2 (PaaS)
Can be used to deploy almost any AWS service	Deploys web applications based on Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker
Uses JSON or YAML template files	Uses ZIP or WAR files
Similar to Terraform	Similar to Google App Engine

Supports Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker web applications.

Deploys to server platforms such as Apache Tomcat, Nginx, Passenger, Puma, and IIS.

Developers can focus on writing code and don’t need to worry about deploying infrastructure.

You maintain full control of the underlying resources.

You pay only for the resources provisioned, not for Elastic Beanstalk itself.

Elastic Beanstalk automatically scales your application up and down.

You can select the EC2 instance type that is optimal for your application.

Can retain full administrative control or have Elastic Beanstalk do it for you.

The Managed Platform Updates feature automatically applies updates for your operating system, Java, PHP, Node.js etc.

Elastic Beanstalk monitors and manages application health and information is viewable via a dashboard.

AWS CloudFormation is used by Elastic Beanstalk to deploy the resources.

Integrated with CloudWatch and X-Ray for performance data and metrics.

ELASTIC BEANSTALK LAYERS

There are several layers that make up Elastic Beanstalk and each layer is described below:

Application:

- Within Elastic Beanstalk, an application is a collection of different elements, such as environments, environment configurations, and application versions.
- You can have multiple application versions held within an application.

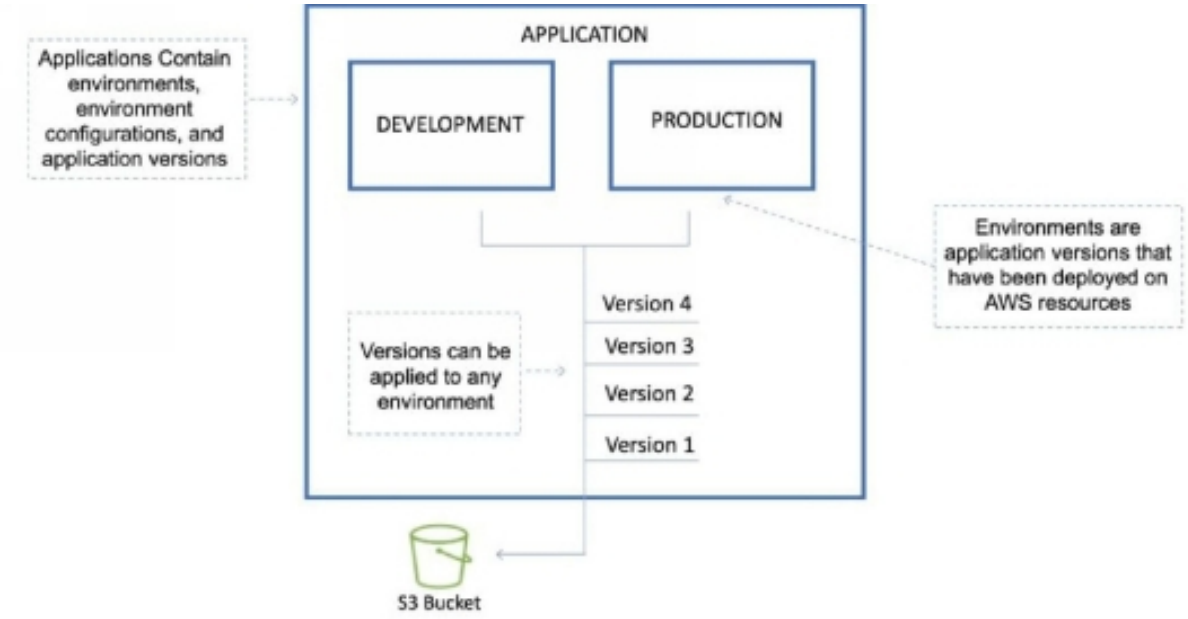
Application version:

- An application version is a very specific reference to a section of deployable code.

- The application version will point typically to an Amazon s3 bucket containing the code.

Environment:

- An environment refers to an application version that has been deployed on AWS resources.
- The resources are configured and provisioned by AWS Elastic Beanstalk.
- The environment is comprised of all the resources created by Elastic Beanstalk and not just an EC2 instance with your uploaded code.



Environment tier:

- Determines how Elastic Beanstalk provisions resources based on what the application is designed to do.
- **Web servers** are standard applications that listen for and then process HTTP requests, typically over port 80.
- **Workers** are specialized applications that have a background processing task that listens for messages on an Amazon SQS queue.

Environment configurations:

- An environment configuration is a collection of parameters and settings that dictate how an environment will have its resources provisioned by Elastic Beanstalk and how these resources will behave.

Configuration template:

- This is a template that provides the baseline for creating a new, unique environment configuration.

DEPLOYMENT AND PROVISIONING

AWS Elastic Beanstalk provides several options for how **deployments** are processed, including deployment policies and options that let you configure batch size and health check behavior during deployments.

DEPLOYMENT OPTIONS

Single instance: great for development.

High availability with load balancer: great for production.

DEPLOYMENT POLICIES

The deployment policies are: All at once, Rolling, Rolling with additional batch, and Immutable.

All at once:

- Deploys the new version to all instances simultaneously.
- All of your instances are out of service while the deployment takes place.
- Fastest deployment.
- Good for quick iterations in the development environment.
- You will experience an outage while the deployment is taking place – not ideal for mission-critical systems.

- If the update fails, you need to roll back the changes by re-deploying the original version to all of your instances.
- No additional cost.

Rolling:

- Update a few instances at a time (batch), and then move onto the next batch once the first batch is healthy (downtime for 1 batch at a time).
- The application is running both versions simultaneously.
- Each batch of instances is taken out of service while the deployment takes place.
- Your environment capacity will be reduced by the number of instances in a batch while the deployment takes place.
- Not ideal for performance-sensitive systems.
- If the update fails, you need to perform an additional rolling update to roll back the changes.
- No additional cost.
- Long deployment time.

Rolling with additional batch:

- Like Rolling but launches new instances in a batch ensuring that there is full availability.
- The application is running at capacity.
- You can set the bucket size.
- The application is running both versions simultaneously.
- Small additional cost.
- Additional batch is removed at the end of the deployment.
- Longer deployment.

- Good for production environments.

Immutable:

- Launches new instances in a new ASG and deploys the version update to these instances before swapping traffic to these instances once healthy.
- Zero downtime.
- New code is deployed to new instances using an ASG.
- High cost as double the number of instances running during updates.
- Longest deployment.
- Quick rollback in case of failures.
- Great for production environments.

Additionally, Elastic Beanstalk supports blue/green deployment.

Blue / Green deployment:

- This is not a feature within Elastic Beanstalk
- You create a new “staging” environment and deploy updates there.
- The new environment (green) can be validated independently and you can roll back if there are issues.
- Route 53 can be set up using weighted policies to redirect a percentage of traffic to the staging environment.
- Using Elastic Beanstalk, you can “swap URLs” when done with the environment test.
- Zero downtime.

The following tables summarizes the different deployment policies:

Deployment Policy	Deploy Time	Zero Downtime	Rollback	Extra Cost	Reduction in capacity
All at once	🕓	NO	Manual redeploy	NONE	YES (total)
Rolling	🕓🕓	YES	Manual redeploy	NONE	YES (batch size)
Rolling with additional batch	🕓🕓🕓	YES	Manual redeploy	YES (batch size)	NO
Immutable	🕓🕓🕓🕓	YES	Terminate new instances	YES (total)	NO
Blue/green	🕓🕓🕓🕓	YES	Swap URL	YES (varies)	NO

GOLDEN AMIS

When deploying code to Amazon EC2 using Beanstalk, Elastic Beanstalk must resolve application dependencies which can take a long time.

A golden AMI is a method of reducing this time by packaging all dependencies, configuration and software into the AMI before deploying.

ENCRYPTION IN-TRANSIT

Elastic Beanstalk works with HTTPS:

- Load the SSL certificate onto the load balancer.
- Can be performed from the console or in code (.ebextensions/securelistener-alb.config).
- SSL certificate can be provisioned using ACM or CLI.

For redirecting HTTP to HTTPS:

- Configure in the application.
- Configure the ALB with a rule.
- Ensure health checks are not redirected.

CUSTOM DOMAIN NAMES

If you're using AWS Elastic Beanstalk to deploy and manage applications in the AWS Cloud, you can use Amazon Route 53 to route DNS traffic for your domain, such as example.com, to a new or an existing Elastic Beanstalk environment.

You create either a *CNAME record* or an *alias record* , depending on whether the domain name for the environment includes the Region, such as **us-east-2** , in which you deployed the environment. New environments include the Region in the domain name; environments that were created before early 2016 do not.

If the domain name does NOT include the Region: create a CNAME record.

If the domain name DOES include the Region: create an Alias record.

TROUBLESHOOTING

If the environment health changes to RED:

- Review environment events.
- Check logs to view recent entries.
- Roll back to a previous working version of the application.

When accessing external resources make sure the security groups are correctly configured.

If commands timeout, increase the deployment timeout.

HIGH AVAILABILITY

You can add high availability to Elastic Beanstalk environments by choosing the option in the console.

When adding high availability Elastic Beanstalk will deploy an Auto Scaling group with multiple EC2 instances and a load balancer.

There is also an option to deploy with high availability using Spot and On-Demand instances.

MONITORING AND REPORTING

Elastic Beanstalk automatically uses **Amazon CloudWatch** to help you monitor your application and environment status. You can navigate to the Amazon CloudWatch console to see your dashboard and get an overview of all of your resources as well as your alarms. You can also choose to view more metrics or add custom metrics.

LOGGING AND AUDITING

With CloudWatch Logs, you can monitor and archive your Elastic Beanstalk application, system, and custom log files from Amazon EC2 instances of your environments.

You can also configure alarms that make it easier for you to react to specific log stream events that your metric filters extract.

The CloudWatch Logs agent installed on each Amazon EC2 instance in your environment publishes metric data points to the CloudWatch service for each log group you configure.

Each log group applies its own filter patterns to determine what log stream events to send to CloudWatch as data points.

Log streams that belong to the same log group share the same retention, monitoring, and access control settings.

In addition to instance logs, if you enable **enhanced health** for your environment, you can configure the environment to stream health information to CloudWatch Logs.

AUTHORIZATION AND ACCESS CONTROL

AWS Elastic Beanstalk supports **identity-based** policies.

AWS Elastic Beanstalk does not support **resource-based** policies.

AWS Elastic Beanstalk has partial support for **resource-level** permissions.

When you create an environment, AWS Elastic Beanstalk prompts you to provide two AWS Identity and Access Management (IAM) roles: a service role and an instance profile.

The **service role** is assumed by Elastic Beanstalk to use other AWS services on your behalf.

The **instance profile** is applied to the instances in your environment and allows them to retrieve **application versions** from Amazon Simple Storage Service (Amazon S3), upload logs to Amazon S3, and perform other tasks that vary depending on the environment type and platform.

You can also create **user policies** and apply them to IAM users and groups in your account to allow users to create and manage Elastic Beanstalk applications and environments. Elastic Beanstalk provides managed policies for full access and read-only access.

AWS CLOUDFORMATION

AWS CloudFormation is a service that allows you to manage, configure and provision your AWS infrastructure using code in a template.

AWS CloudFormation provides a common language for you to describe and provision all the infrastructure resources in your cloud environment.

Resources are defined using an AWS CloudFormation template.

CloudFormation interprets the template and makes the appropriate API calls to create the resources you have defined.

CloudFormation supports YAML or JSON.

CloudFormation can be used to provision a broad range of AWS resources.

Think of CloudFormation as deploying “infrastructure as code”.

CloudFormation has some similarities with **AWS Elastic Beanstalk** though they are also quite different as detailed in the table below:

CloudFormation	Elastic Beanstalk
“Template-driven provisioning”	“Web apps made easy”
Deploys infrastructure using code	Deploys applications on EC2 (PaaS)
Can be used to deploy almost any AWS service	Deploys web applications based on Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker
Uses JSON or YAML template files	Uses ZIP or WAR files
Similar to Terraform	Similar to Google App Engine

DEPLOYMENT AND PROVISIONING

KEY CONCEPTS

The following table describes the key concepts associated with AWS CloudFormation:

Component	Description
Templates	The JSON or YAML text file that contains the instructions for building out the AWS environment
Stacks	The entire environment described by the template and created, updated, and deleted as a single unit
StackSets	AWS CloudFormation StackSets extends the functionality of stacks by enabling you to create, update, or delete stacks across multiple accounts and regions with a single operation
Change Sets	A summary of proposed changes to your stack that will allow you to see how those changes might impact your existing resources before implementing them

TEMPLATES

A template is a YAML or JSON template used to describe the end-state of the infrastructure you are either provisioning or changing.

After creating the template, you upload it to CloudFormation directly or using Amazon S3.

CloudFormation reads the template and makes the API calls on your behalf.

The resulting resources are called a “Stack”.

Logical IDs are used to reference resources within the template.

Physical IDs identify resources outside of AWS CloudFormation templates, but only after the resources have been created.

TEMPLATE ELEMENTS

Mandatory:

- List of resources and associated configuration values.

Not mandatory:

- Template parameters (limited to 60).

- Output values (limited to 60).
- List of data tables.

TEMPLATE COMPONENTS

Resources – the *required* Resources section declares the AWS resources that you want to include in the stack, such as an **Amazon EC2** instance or an Amazon S3 bucket.

- Mandatory.
- Represent AWS components that will be created.
- Resources are declared and can reference each other.

The following example YAML code declares an EC2 instance as a resource:

```
Resources :
 MyEC2Instance :
   Type : "AWS::EC2::Instance"
   Properties :
     ImageId : "ami-0ff8a91507f77f867"
```

Parameters – use the *optional* Parameters section to customize your templates. Parameters enable you to input custom values to your template each time you create or update a stack.

- Provide inputs to your CloudFormation template.
- Useful for template reuse.

The following example declares a parameter named InstanceTypeParameter. This parameter lets you specify the Amazon EC2 instance type for the stack to use when you create or update the stack.

Note: the InstanceTypeParameter has a default value of t2.micro. This is the value that AWS CloudFormation uses to provision the stack unless another value is provided.

```
Parameters :
```

```
  InstanceTypeParameter :
    Type : String
    Default : t2 . micro
    AllowedValues :
      - t2 . micro
      - m1 . small
      - m1 . large
    Description : Enter t2 . micro , m1 . small , or m1 . large . Default
is t2 . micro .
```

Mappings – the *optional* Mappings section matches a key to a corresponding set of named values.

- Fixed variables.
- Good for differentiating between regions, environments, AMIs etc.
- Need to know the values in advance.
- For user-specific values use parameters instead.

The following example has region keys that are mapped to two sets of values: one named HVM64 and the other HVMG2.

```
RegionMap :
    us - east - 1 :
      HVM64 : ami - 0ff8a91507f77f867
      HVMG2 : ami - 0a584ac55a7631c0c
    us - west - 1 :
      HVM64 : ami - 0bdb828fd58c52235
      HVMG2 : ami - 066ee5fd4a9ef77f1
```

Exam tip: with mappings you can, for example, set values based on a region. You can create a mapping that uses the region name as a key and contains the values you want to specify for each specific region.

Outputs – the *optional* Outputs section declares output values that you can import into other stacks (to create cross-stack references), return in response (to describe stack calls), or view on the AWS CloudFormation console.

- Outputs can be imported into other stacks.
- Can view the outputs in the console or using the AWS CLI.
- Cannot delete a Stack if it's outputs are being referenced by another CloudFormation Stack.

In the following example YAML code, the output named StackVPC returns the ID of a VPC, and then exports the value for cross-stack referencing with the name VPCID appended to the stack's name

```
Outputs :
  StackVPC :
   Description : The ID of the VPC
    Value : ! Ref MyVPC
    Export :
      Name : ! Sub "${AWS::StackName}-VPCID"
```

Conditions – the *optional* Conditions section contains statements that define the circumstances under which entities are created or configured.

- Control the creation of resources based on a condition.
- Applied to resources and outputs.

In the sample YAML code below, resources are created only if the EnvType parameter is equal to prod:

```
Conditions :
 CreateProdResources : ! Equals [ ! Ref EnvType , prod ]
```

STACKS AND STACK SETS

Stacks:

- Deployed resources based on templates.

- Create, update and delete stacks using templates.
- Deployed through the Management Console, CLI or APIs.

Stack creation errors:

- Automatic rollback on error is enabled by default.
- You will be charged for resources provisioned even if there is an error.

Updating stacks:

- AWS CloudFormation provides two methods for updating stacks: direct update or creating and executing change sets.
- When you directly update a stack, you submit changes and AWS CloudFormation immediately deploys them.
- Use direct updates when you want to quickly deploy your updates.
- With change sets, you can preview the changes AWS CloudFormation will make to your stack, and then decide whether to apply those changes.

Stack Sets:

- AWS CloudFormation StackSets extends the functionality of stacks by enabling you to create, update, or delete stacks across multiple accounts and regions with a single operation.
- Using an administrator account, you define and manage an AWS CloudFormation template, and use the template as the basis for provisioning stacks into selected target accounts across specified regions.
- An administrator account is the AWS account in which you create stack sets.
- A stack set is managed by signing in to the AWS administrator account in which it was created.

- A target account is the account into which you create, update, or delete one or more stacks in your stack set.

Before you can use a stack set to create stacks in a target account, you must set up a trust relationship between the administrator and target accounts.

CloudFormation Nested Stacks:

- Nested stacks allow re-use of CloudFormation code for common use cases.
- For example standard configuration for a load balancer, web server, application server etc.
- Instead of copying out the code each time, create a standard template for each common use case and reference from within your CloudFormation template.

USER DATA WITH EC2

- User data can be included in CloudFormation.
- The script is passed into Fn::Base64
- The user data script logs are stored in /var/log/cloud-init-output.log
- Binary is available on Amazon EC2 at /opt/aws/bin/cfn-init

CLOUDFORMATION HELPER SCRIPTS

cfn-init:

- The cfn-init helper script reads template metadata from the AWS::CloudFormation::Init key and acts accordingly to:
- Fetch and parse metadata from AWS CloudFormation
- Install packages
- Write files to disk
- Enable/disable and start/stop services

- cfn-init does not require credentials, so you do not need to use the –access-key, –secret-key, –role, or –credential-file options.
- Logs go to /var/log/cfn-init.log

cfn-signal:

- The cfn-signal helper script signals AWS CloudFormation to indicate whether Amazon EC2 instances have been successfully created or updated.
- If you install and configure software applications on instances, you can signal AWS CloudFormation when those software applications are ready.
- You use the cfn-signal script in conjunction with a **CreationPolicy** or an Auto Scaling group with a **WaitOnResourceSignals** update policy.
- When AWS CloudFormation creates or updates resources with those policies, it suspends work on the stack until the resource receives the requisite number of signals or until the timeout period is exceeded.
- You can signal a creation policy (**CreationPolicy**) or a wait condition handle (**WaitOnResourceSignals**).

Troubleshooting errors:

- Make sure the AMI has the CloudFormation helper scripts included.
- Check that the cfn-init and cfn-signal commands have run successfully.
- Verify internet connectivity.

CREATION POLICIES AND WAIT CONDITIONS

- CreationPolicy attribute:

- Use the CreationPolicy attribute when you want to wait on resource configuration actions before stack creation proceeds.
- You can associate the CreationPolicy attribute with a resource to prevent its status from reaching create complete until AWS CloudFormation receives a specified number of success signals or the timeout period is exceeded.
- To signal a resource, you can use the **cfn-signal** helper script or **SignalResource** API.
- AWS CloudFormation publishes valid signals to the stack events so that you track the number of signals sent.

The following CloudFormation resources support creation policies:

- **AWS::AutoScaling::AutoScalingGroup**
- **AWS::EC2::Instance**
- **AWS::CloudFormation::WaitCondition**

DeletionPolicy attribute:

- With the DeletionPolicy attribute you can preserve or (in some cases) backup a resource when its stack is deleted.
- You specify a DeletionPolicy attribute for each resource that you want to control.
- If a resource has no DeletionPolicy attribute, AWS CloudFormation deletes the resource by default.

DependsOn attribute:

- With the DependsOn attribute you can specify that the creation of a specific resource follows another.
- When you add a DependsOn attribute to a resource, that resource is created only after the creation of the resource specified in the DependsOn attribute.

WaitCondition:

- Note: For Amazon EC2 and Auto Scaling resources, AWS recommends that you use a CreationPolicy attribute instead of wait conditions.
- You can use a wait condition for situations like the following:
- To coordinate stack resource creation with configuration actions that are external to the stack creation.
- To track the status of a configuration process.

UpdatePolicy Attribute (WaitOnResourceSignals)

Use the UpdatePolicy attribute to specify how AWS CloudFormation handles updates to the following resources:

- **AWS::AutoScaling::AutoScalingGroup** ,
- **AWS::ElastiCache::ReplicationGroup**
- **AWS::Elasticsearch::Domain**
- **AWS::Lambda::Alias**

UpdateReplacePolicy attribute:

- Use the UpdateReplacePolicy attribute to retain or (in some cases) backup the existing physical instance of a resource when it is replaced during a stack update operation.

ROLLBACKS AND CREATION FAILURES

Stack creation failures :

- By default everything will be deleted.
- You can optionally disable rollback (good for troubleshooting failures).

Stack update failures:

- The stack will automatically roll back to the previous known working state.

- The logs can assist with understanding what issue occurred.

MONITORING AND REPORTING

You can monitor the progress of a stack update by viewing the stack's events. The console's **Events** tab displays each major step in the creation and update of the stack sorted by the time of each event with latest events on top.

For resources created by CloudFormation, use AWS monitoring and reporting tools applicable to the service.

AUTHORIZATION AND ACCESS CONTROL

You can use IAM with AWS CloudFormation to control what users can do with AWS CloudFormation, such as whether they can view stack templates, create stacks, or delete stacks.

In addition to AWS CloudFormation actions, you can manage what AWS services and resources are available to each user.

That way, you can control which resources users can access when they use AWS CloudFormation.

For example, you can specify which users can create Amazon EC2 instances, terminate database instances, or update VPCs. Those same permissions are applied anytime they use AWS CloudFormation to do those actions.

AMAZON VIRTUAL PRIVATE CLOUD (VPC)

AMAZON VIRTUAL PRIVATE CLOUD (VPC) OVERVIEW

An Amazon Virtual Private Cloud (VPC) is a logically isolated space into which you can launch AWS resources.

You can create virtual networks called subnets within a VPC.

You have complete control over the virtual networking.

The components of a VPC include:

- **A Virtual Private Cloud** : A logically isolated virtual network in the AWS cloud. You define a VPC's IP address space from ranges you select.
- **Subnet** : A segment of a VPC's IP address range where you can place groups of isolated resources (maps to an AZ, 1:1).
- **Internet Gateway** : The Amazon VPC side of a connection to the public Internet.
- **NAT Gateway** : A highly available, managed Network Address Translation (NAT) service for your resources in a private subnet to access the Internet.
- **Hardware VPN Connection** : A hardware-based VPN connection between your Amazon VPC and your datacenter, home network, or co-location facility.
- **Virtual Private Gateway** : The Amazon VPC side of a VPN connection.

- **Customer Gateway** : Your side of a VPN connection.
- **Router** : Routers interconnect subnets and direct traffic between Internet gateways, virtual private gateways, NAT gateways, and subnets.
- **Peering Connection** : A peering connection enables you to route traffic via private IP addresses between two peered VPCs.
- **VPC Endpoints** : Enables private connectivity to services hosted in AWS, from within your VPC without using an an Internet Gateway, VPN, Network Address Translation (NAT) devices, or firewall proxies.
- **Egress-only Internet Gateway** : A stateful gateway to provide egress only access for IPv6 traffic from the VPC to the Internet.

NOTE : The SysOps Administrator Exam focusses on some specific operational concepts associated with VPCs. For a more in-depth background on Amazon VPC see the **cheats sheets for the Solutions Architect Associate here** .

CIDR BLOCKS AND IP SUBNETS FOR AMAZON VPCS

When you create a VPC, you must specify a range of IPv4 addresses for the VPC in the form of a Classless Inter-Domain Routing (CIDR) block.

You can then define ranges of IP addresses within the VPC CIDR that can be assigned to subnets. AWS resources obtain addresses from these IP ranges.

AWS recommend that CIDR blocks of /16 or smaller are used.

It is recommended these come from the private IP ranges specified in RFC 1918:

- 10.0.0.0 – 10.255.255.255 (10/8 prefix)
- 172.16.0.0 – 172.31.255.255 (172.16/12 prefix)

- 192.168.0.0 – 192.168.255.255 (192.168/16 prefix)

However, it is possible to create a VPC with publicly routable CIDR block.

The allowed block size is between a /28 netmask and /16 netmask.

The CIDR blocks of the subnets within a VPC cannot overlap.

The first four IP addresses and the last IP address in each subnet CIDR block are not available for you to use

For example, in a subnet with CIDR block 10.0.0.0/24, the following five IP addresses are reserved:

- 10.0.0.0: Network address.
- 10.0.0.1: Reserved by AWS for the VPC route.
- 10.0.0.2: Reserved by AWS.
- 10.0.0.3: Reserved by AWS for future use.
- 10.0.0.255: Network broadcast address (broadcast not supported).

For further information, check out this AWS **article** .

NAT GATEWAYS AND NAT INSTANCES

NAT GATEWAYS

NAT gateways are managed **for you** by AWS.

Fully-managed NAT service that replaces the need for NAT instances on EC2.

Must be created in a public subnet.

Uses an Elastic IP address for the public IP.

Private instances in private subnets must have a route to the NAT instance, usually the default route destination of 0.0.0.0/0.

Created in a specified AZ with redundancy in that zone.

For multi-AZ redundancy, create NAT Gateways in each AZ with routes for private subnets to use the local gateway.

Can’t use a NAT Gateway to access VPC peering, VPN or Direct Connect, so be sure to include specific routes to those in your route table.

NAT gateways are highly available in each AZ into which they are deployed.

Not associated with any security groups.

Automatically assigned a public IP address.

Remember to update route tables and point towards your gateway.

NAT INSTANCES

NAT instances are managed **by you** .

Used to enable private subnet instances to access the Internet.

NAT instance must live on a public subnet with a route to an Internet Gateway.

Private instances in private subnets must have a route to the NAT instance, usually the default route destination of 0.0.0.0/0.

When creating NAT instances always disable the source/destination check on the instance.

NAT instances must be in a single public subnet.

NAT instances need to be assigned to security groups.

Security groups for NAT instances must allow HTTP/HTTPS inbound from the private subnet and outbound to 0.0.0.0/0.

There needs to be a route from a private subnet to the NAT instance for it to work.

The amount of traffic a NAT instance can support is based on the instance type.

Using a NAT instance can lead to bottlenecks (not highly available).

Performance is dependent on instance size.

Can scale up instance size or use enhanced networking.

Can scale out by using multiple NATs in multiple subnets.

Can also use as a bastion (jump) host.

	NAT Gateway	NAT Instance
Availability	Highly available within an AZ	Not highly available (would require scripting)
Bandwidth	Up to 45 Gbps	Depends on the bandwidth of the EC2 instance type selected
Maintenance	Managed by AWS	Managed by you
Performance	Optimized for NAT	Amazon Linux AMI configured to perform NAT
Public IP	Elastic IP that cannot be detached	Elastic IP that can be detached
Security Groups	Cannot associate with a Security Group	Can associate with a Security Group
Bastion Host	Not supported	Can be used as a bastion host

SECURITY GROUPS

Security groups act like a firewall at the instance level.

Specifically security groups operate at the network interface level.

Can only assign permit rules in a security group, cannot assign deny rules.

There is an implicit deny rule at the end of the security group.

All rules are evaluated until a permit is encountered or continues until the implicit deny.

Can control ingress and egress traffic.

Security groups are stateful.

You can use security group names/IDs as the source or destination in other security groups.

You can use the security group name/ID as a source in its own inbound rules.

Security group members can be within any AZ or subnet within the VPC.

Security group membership can be changed whilst instances are running.

Any changes made will take effect immediately.

You cannot block specific IP addresses using security groups, use NACLs instead.

NETWORK ACL'S

Network ACL's function at the subnet level.

With NACLs you can have permit and deny rules.

Network ACLs contain a numbered list of rules that are evaluated in order from the lowest number until the explicit deny.

Network ACLs have separate inbound and outbound rules and each rule can allow or deny traffic.

Network ACLs are stateless so responses are subject to the rules for the direction of traffic.

NACLs only apply to traffic that is ingress or egress to the subnet not to traffic within the subnet.

A VPC automatically comes with a default network ACL which allows all inbound/outbound traffic.

A custom NACL denies all traffic both inbound and outbound by default.

All subnets must be associated with a network ACL.

You can create custom network ACL's. By default, each custom network ACL denies all inbound and outbound traffic until you add rules.

Each subnet in your VPC must be associated with a network ACL. If you don't do this manually it will be associated with the default network ACL.

You can associate a network ACL with multiple subnets; however a subnet can only be associated with one network ACL at a time.

Network ACLs do not filter traffic between instances in the same subnet.

NACLs are the preferred option for blocking specific IPs or ranges.

Security groups cannot be used to block specific ranges of IPs.

NACL is the first line of defence, the security group is the second line.

Changes to NACLs take effect immediately.

Security Group	Network ACL
Operates at the instance (interface) level	Operates at the subnet level
Supports allow rules only	Supports allow and deny rules
Stateful	Stateless
Evaluates all rules	Processes rules in order
Applies to an instance only if associated with a group	Automatically applies to all instances in the subnets its associated with

VPC ENDPOINTS

Enables private connectivity from a VPC to supported AWS services and **VPC endpoint** services powered by AWS PrivateLink.

Does not require an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection.

Endpoints are virtual devices.

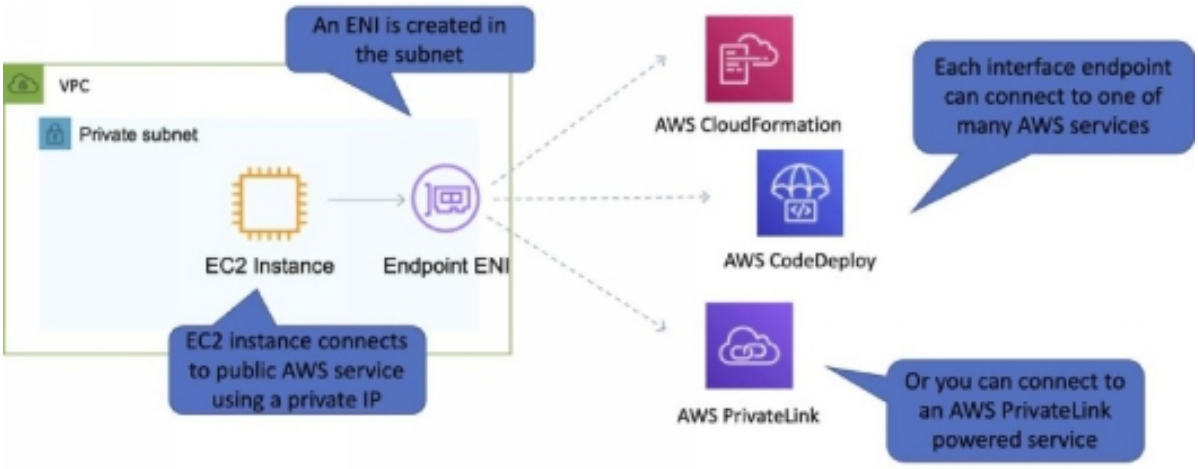
They are horizontally scaled, redundant, and highly available.

There are two types of VPC endpoints: interface endpoints and gateway endpoints.

INTERFACE ENDPOINTS

An interface endpoint is an elastic network interface with a private IP address that serves as an entry point for traffic destined to a supported service

With an interface endpoint you remove the need for an internet gateway, NAT device, or virtual private gateway.

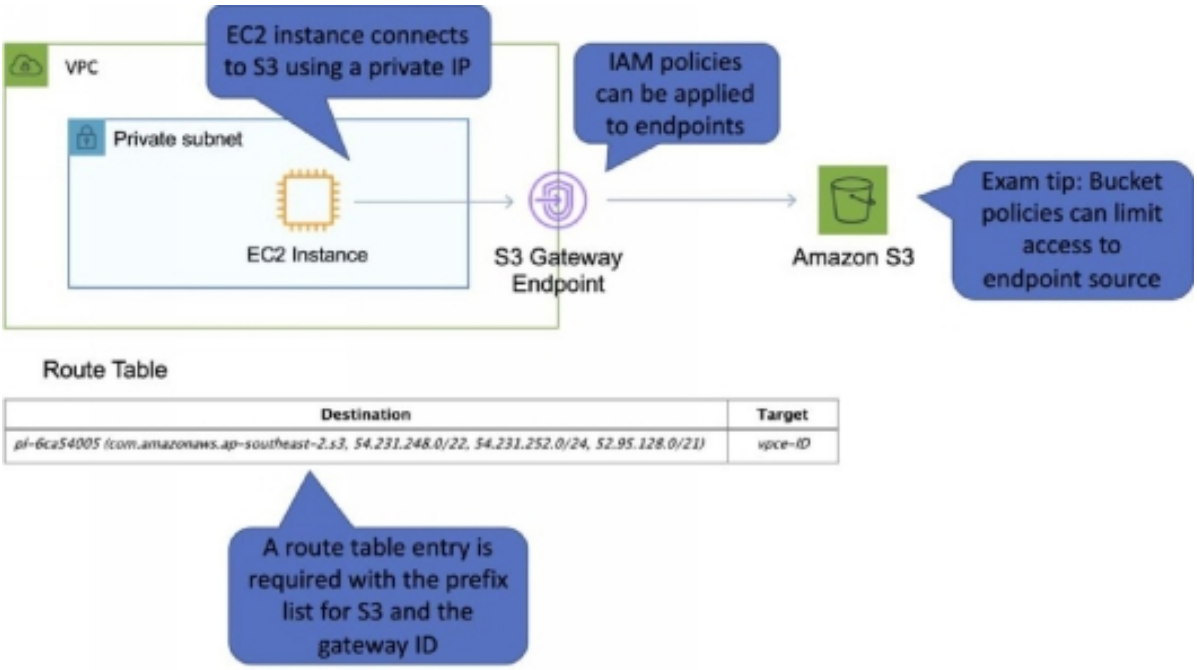


GATEWAY ENDPOINTS

A gateway endpoint is a gateway that you specify as a target for a route in your route table for traffic destined to a supported AWS service.

The following AWS services are supported:

- Amazon S3.
- DynamoDB.



VPC PEERING

A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them using private IPv4 addresses or IPv6 addresses.

Instances in either VPC can communicate with each other as if they are within the same network.

You can create a VPC peering connection between your own VPCs, or with a VPC in another AWS account.

The VPCs can be in different regions (also known as an inter-region VPC peering connection).

Data sent between VPCs in different regions is encrypted (traffic charges apply).

For inter-region VPC peering there are some limitations:

- You cannot create a security group rule that references a peer security group.
- Cannot enable DNS resolution.

- Maximum MTU is 1500 bytes (no jumbo frames support).
- Limited region support.
- AWS uses the existing infrastructure of a VPC to create a VPC peering connection.

It is neither a gateway nor a VPN connection, and does not rely on a separate piece of physical hardware.

There is no single point of failure for communication or a bandwidth bottleneck.

A VPC peering connection helps you to facilitate the transfer of data.

Can only have one peering connection between any two VPCs at a time.

Can peer with other accounts (within or between regions).

Cannot have overlapping CIDR ranges.

A VPC peering connection is a one to one relationship between two VPCs.

You can create multiple VPC peering connections for each VPC that you own, but transitive peering relationships are not supported.

You do not have any peering relationship with VPCs that your VPC is not directly peered with.

DNS is supported.

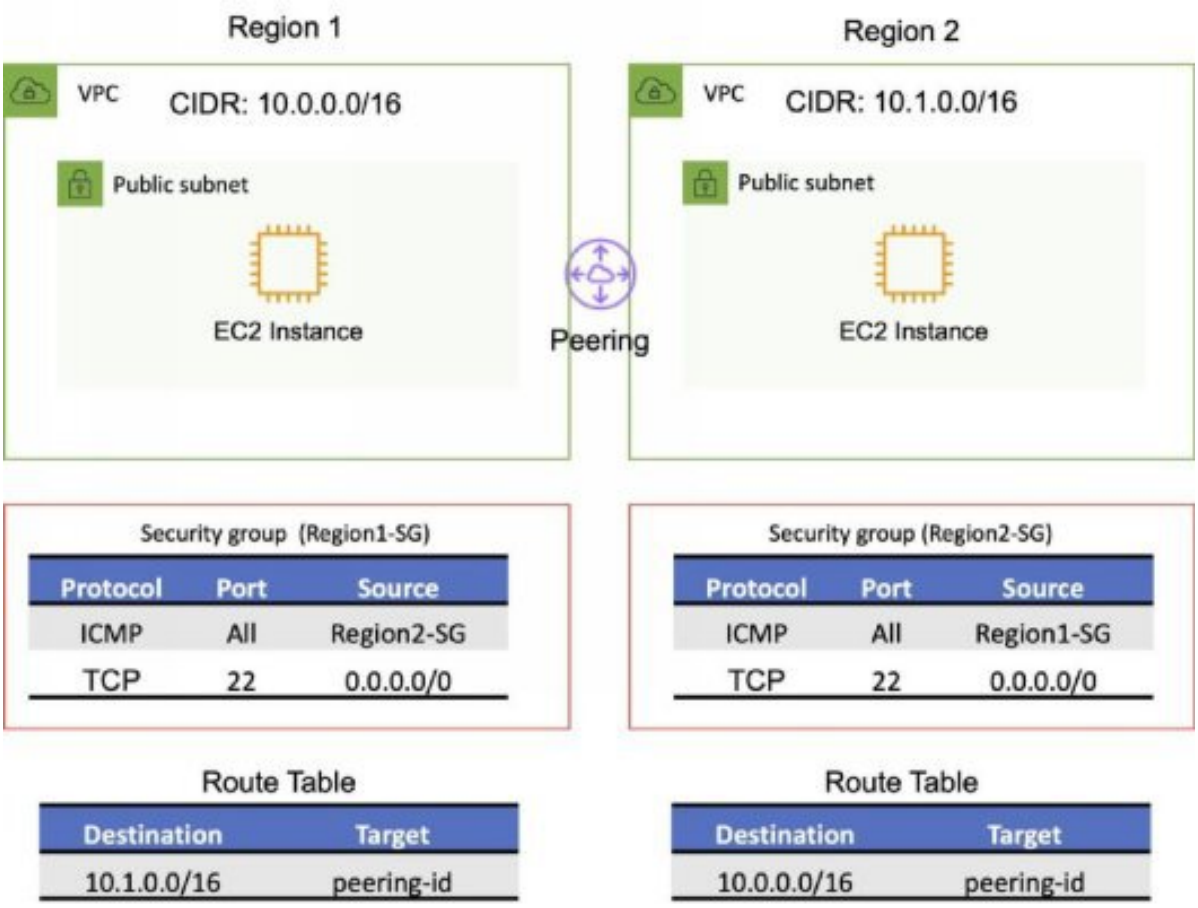
Must update route tables to configure routing.

Must update the inbound and outbound rules for VPC security group to reference security groups in the peered VPC.

When creating a VPC peering connection with another account you need to enter the account ID and VPC ID from the other account.

Need to accept the pending access request in the peered VPC.

The VPC peering connection can be added to route tables – shows as a target starting with “pcx-“.



VPC FLOW LOGS

Flow logs can be created at the following levels:

- VPC.
- Subnet.
- Network interface.

You can’t enable flow logs for VPC’s that are peered with your VPC unless the peer VPC is in your account.

You can’t tag a flow log.

You can’t change the configuration of a flow log after it’s been created.

After you've created a flow log, you cannot change its configuration (you need to delete and re-create).

Not all traffic is monitored, e.g. the following traffic is excluded:

- Traffic that goes to Route53.
- Traffic generated for Windows license activation.
- Traffic to and from 169.254.169.254 (instance metadata).
- Traffic to and from 169.254.169.123 for the Amazon Time Sync Service.
- DHCP traffic.
- Traffic to the reserved IP address for the default VPC router.

In the following example flow log connections to the SSH port (22) are being accepted and connections to the HTTP port (80) are being rejected:

version	account-id	interface-id	srcaddr	dstaddr	srcport	dstport	protocol	packets	bytes	start	end	action	log-status
2	55112233445	eni-0f5...	11.200.185.200	10.0.1.15	52933	22	6	1	401599...		1599...	ACCEPT	OK
2	55112233445	eni-0f5...	10.0.1.15	11.200.185.200	22	52933	6	1	401599...		1599...	ACCEPT	OK
2	55112233445	eni-0f5...	11.200.185.200	10.0.1.15	3624	80	6	1	441599...		1599...	REJECT	OK
2	55112233445	eni-0f5...	11.200.185.200	10.0.1.15	3624	80	6	1	441599...		1599...	REJECT	OK

It's important to understand the difference between the data that's included in a VPC Flow Log vs the data in an **Elastic Load Balancing** (ELB) Access Log. The following image shows a log entry from a ELB access log. Note that application level information (HTTP GET request):

```
http 2018-07-02T22:23:00.186641Z app/my-loadbalancer/50dc6c495c0c9188
192.168.131.39:2817 10.0.0.1:80 0.000 0.001 0.000 200 200 34 366
"GET http://www.example.com:80/ HTTP/1.1" "curl/7.46.0" - -
arn:aws:elasticloadbalancing:us-east-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
"Root=1-58337262-36d228ad5d99923122bbe354" "-" "-"
0 2018-07-02T22:22:48.364000Z "forward" "-" "-" 10.0.0.1:80 200 "-" "-"
```

AWS MANAGED VIRTUAL PRIVATE NETWORK (VPN)

VPNs are quick, easy to deploy, and cost effective.

A Virtual Private Gateway (VGW) is required on the AWS side.

A Customer Gateway is required on the customer side.

What	AWS Managed IPSec VPN Connection over your existing Internet
When	Quick and usually simple way to establish a secure tunnelled connection to a VPC; redundant link for Direct Connect or other VPC VPN
Pros	Supports static routes or BGP peering and routing
Cons	Dependent on your Internet connection
How	Create a Virtual Private Gateway (VPG) on AWS, and a Customer Gateway on the on-premises side

AWS DIRECT CONNECT

AWS Direct Connect makes it easy to establish a dedicated connection from an on-premises network to Amazon VPC.

Using AWS Direct Connect, you can establish private connectivity between AWS and your data center, office, or collocated environment.

This private connection can reduce network costs, increase bandwidth throughput, and provide a more consistent network experience than internet-based connections.

AWS Direct Connect lets you establish 1 Gbps or 10 Gbps dedicated network connections (or multiple connections) between AWS networks and one of the AWS Direct Connect locations.

It uses industry-standard VLANs to access Amazon Elastic Compute Cloud (Amazon EC2) instances running within an Amazon VPC using private IP addresses.

AWS Direct Connect does not encrypt your traffic that is in transit.

You can use the encryption options for the services that traverse AWS Direct Connect.

For more info on AWS Direct Connect check out the **article here** .

What	Dedicated network connection over private lines straight into the AWS backbone
When	Requires a large network link into AWS; lots of resources and services being provided on AWS to your corporate users
Pros	More predictable network performance; potential bandwidth cost reduction; up to 10 Gbps provisioned connections; supports BGP peering and routing
Cons	May require additional telecom and hosting provider relationships and/or network circuits; costly
How	Work with your existing data networking provider; create Virtual Interfaces (VIFs) to connect to VPCs (private VIFs) or other AWS services like S3 or Glacier (public VIFs)

AMAZON ROUTE 53 (AWS ROUTE 53)

Amazon Route 53 is a highly available and scalable Domain Name System (DNS) service.

Route 53 offers the following functions:

- Domain name registry.
- DNS resolution.
- Health checking of resources.

Route 53 can perform any combination of these functions.

Route 53 provides a worldwide distributed DNS service.

Route 53 is located alongside all edge locations.

Health checks verify Internet connected resources are reachable, available and functional.

Route 53 can be used to route Internet traffic for domains registered with another domain registrar (any domain).

When you register a domain with Route 53 it becomes the authoritative DNS server for that domain and creates a public hosted zone.

To make Route 53 the authoritative DNS for an existing domain without transferring the domain create a Route 53 public hosted zone and change the DNS Name Servers on the existing provider to the Route 53 Name Servers.

Changes to Name Servers may not take effect for up to 48 hours due to the DNS record Time To Live (TTL) values.

You can transfer domains to Route 53 only if the Top Level Domain (TLD) is supported.

You can transfer a domain from Route 53 to another registrar by contacting AWS support.

You can transfer a domain to another account in AWS however it does not migrate the hosted zone by default (optional).

It is possible to have the domain registered in one AWS account and the hosted zone in another AWS account.

Primarily uses UDP port 53 (can use TCP).

AWS offer a 100% uptime SLA for Route 53.

You can control management access to your Amazon Route 53 hosted zone by using IAM.

There is a default limit of 50 domain names but this can be increased by contacting support.

Private DNS is a Route 53 feature that lets you have authoritative DNS within your VPCs without exposing your DNS records (including the name of the resource and its IP address(es) to the Internet.

You can use the AWS Management Console or API to register new domain names with Route 53.

HOSTED ZONES

A hosted zone is a collection of records for a specified domain.

A hosted zone is analogous to a traditional DNS zone file; it represents a collection of records that can be managed together.

There are two types of zones:

- Public host zone – determines how traffic is routed on the Internet.

- Private hosted zone for VPC – determines how traffic is routed within VPC (resources are not accessible outside the VPC).

Amazon Route 53 automatically creates the Name Server (NS) and Start of Authority (SOA) records for the hosted zones.

Amazon Route 53 creates a set of 4 unique name servers (a delegation set) within each hosted zone.

You can create multiple hosted zones with the same name and different records.

NS servers are specified by Fully Qualified Domain Name (FQDN) but you can get the IP addresses from the command line (e.g. dig or nslookup).

For private hosted zones you can see a list of VPCs in each region and must select one.

For private hosted zones you must set the following VPC settings to “true”:

- enableDnsHostname.
- enableDnsSupport.

You also need to create a DHCP options set.

You can extend an on-premises DNS to VPC.

You cannot extend Route 53 to on-premises instances.

You cannot automatically register EC2 instances with private hosted zones (would need to be scripted).

HEALTH CHECKS

Health checks check the instance health by connecting to it.

Health checks can be pointed at:

- Endpoints.
- Status of other health checks.

- Status of a CloudWatch alarm.

Endpoints can be IP addresses or domain names.

You can create the following types of health checks:

- **HTTP** : Route 53 tries to establish a TCP connection. If successful, Route 53 submits an HTTP request and waits for an HTTP status code of 200 or greater and less than 400.
- **HTTPS** : Route 53 tries to establish a TCP connection. If successful, Route 53 submits an HTTPS request and waits for an HTTP status code of 200 or greater and less than 400.
- **HTTP_STR_MATCH** : Route 53 tries to establish a TCP connection. If successful, Route 53 submits an HTTP request and searches the first 5,120 bytes of the response body for the string that you specify in SearchString.
- **HTTPS_STR_MATCH** : Route 53 tries to establish a TCP connection. If successful, Route 53 submits an HTTPS request and searches the first 5,120 bytes of the response body for the string that you specify in SearchString.
- **TCP** : Route 53 tries to establish a TCP connection.
- **CLOUDWATCH_METRIC** : The health check is associated with a CloudWatch alarm. If the state of the alarm is OK, the health check is considered healthy. If the state is ALARM, the health check is considered unhealthy. If CloudWatch doesn’t have sufficient data to determine whether the state is OK or ALARM, the health check status depends on the setting for InsufficientDataHealthStatus: Healthy, Unhealthy, or LastKnownStatus.
- **CALCULATED** : For health checks that monitor the status of other health checks, Route 53 adds up the number of health checks that Route 53 health checkers consider to be healthy and compares that number with the value of HealthThreshold.

RECORDS

Amazon Route 53 currently supports the following DNS record types:

- A (address record).
- AAAA (IPv6 address record).
- CNAME (canonical name record).
- CAA (certification authority authorization).
- MX (mail exchange record).
- NAPTR (name authority pointer record).
- NS (name server record).
- PTR (pointer record).
- SOA (start of authority record).
- SPF (sender policy framework).
- SRV (service locator).
- TXT (text record).
- Alias (an Amazon Route 53-specific virtual record).

The Alias record is a Route 53 specific record type.

Alias records are used to map resource record sets in your hosted zone to Amazon Elastic Load Balancing load balancers, Amazon CloudFront distributions, AWS Elastic Beanstalk environments, or Amazon S3 buckets that are configured as websites.

You can use Alias records to map custom domain names (such as api.example.com) both to API Gateway custom regional APIs and edge-optimized APIs and to Amazon VPC interface endpoints.

The Alias is pointed to the DNS name of the service.

You cannot set the TTL for Alias records for ELB, S3, or Elastic Beanstalk environment (uses the service’s default).

Alias records work like a CNAME record in that you can map one DNS name (e.g. example.com) to another ‘target’ DNS name (e.g. elb1234.elb.amazonaws.com).

An Alias record can be used for resolving apex / naked domain names (e.g. example.com rather than sub.example.com).

A CNAME record can’t be used for resolving apex / naked domain names.

Generally use an Alias record where possible. The following table details the differences between Alias and CNAME records:

CNAME Records	Alias Records
Route 53 charges for CNAME queries	Route 53 doesn’t charge for alias queries to AWS resources
You can't create a CNAME record at the top node of a DNS namespace (zone apex)	You can create an alias record at the zone apex (however you can't route to a CNAME at the zone apex)
A CNAME record redirects queries for a domain name regardless of record type	Route 53 follows the pointer in an alias record only when the record type also matches
A CNAME can point to any DNS record that is hosted anywhere	An alias record can only point to a CloudFront distribution, Elastic Beanstalk environment, ELB, S3 bucket as a static website, or to another record in the same hosted zone that you’re creating the alias record in
A CNAME record is visible in the answer section of a reply from a Route 53 DNS server	An alias record is only visible in the Route 53 console or the Route 53 API
A CNAME record is followed by a recursive resolver	An alias record is only followed inside Route 53. This means that both the alias record and its target must exist in Route 53

Route 53 supports wildcard entries for all record types, except NS records.

ROUTING POLICIES

Routing policies determine how Route 53 responds to queries.

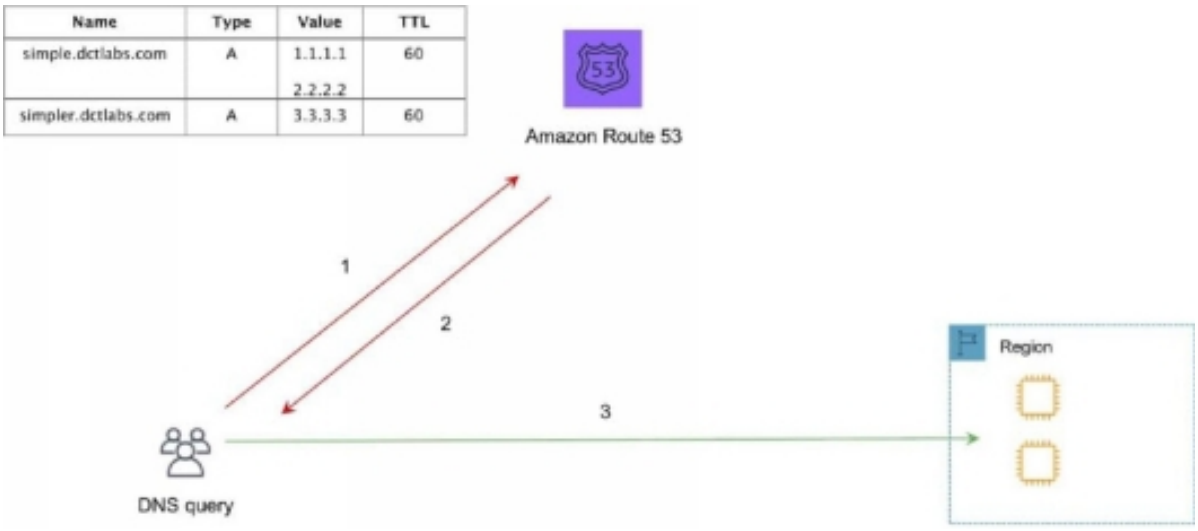
The following table highlights the key function of each type of routing policy:

Policy	What it Does
Simple	Simple DNS response providing the IP address associated with a name
Failover	If primary is down (based on health checks), routes to secondary destination
Geolocation	Uses geographic location you're in (e.g. Europe) to route you to the closest region
Geoproximity	Routes you to the closest region within a geographic area
Latency	Directs you based on the lowest latency route to resources
Multivalue answer	Returns several IP addresses and functions as a basic load balancer
Weighted	Uses the relative weights assigned to resources to determine which to route to

Simple:

- An A record is associated with one or more IP addresses.
- Uses round robin.
- Does not support health checks.

The following diagram depicts an Amazon Route 53 Simple routing policy configuration:



Failover:

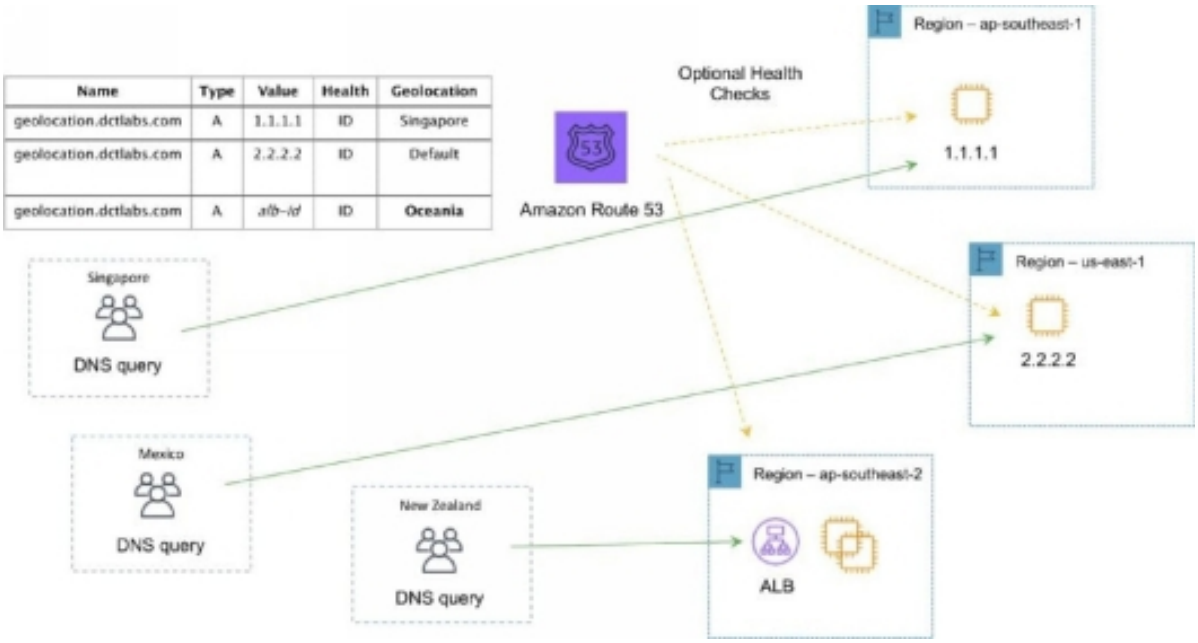
- Failover to a secondary IP address.
- Associated with a health check.
- Used for active-passive.
- Routes only when the resource is healthy.
- Can be used with ELB.
- When used with Alias records set Evaluate Target Health to “Yes” and do not use health checks.

The following diagram depicts an Amazon Route 53 Failover routing policy configuration:

Geo-location:

- Caters to different users in different countries and different languages.
- Contains users within a particular geography and offers them a customized version of the workload based on their specific needs.
- Geolocation can be used for localizing content and presenting some or all of your website in the language of your users.
- Can also protect distribution rights.
- Can be used for spreading load evenly between regions.
- If you have multiple records for overlapping regions, Route 53 will route to the smallest geographic region.
- You can create a default record for IP addresses that do not map to a geographic location.

The following diagram depicts an Amazon Route 53 Geolocation routing policy configuration:



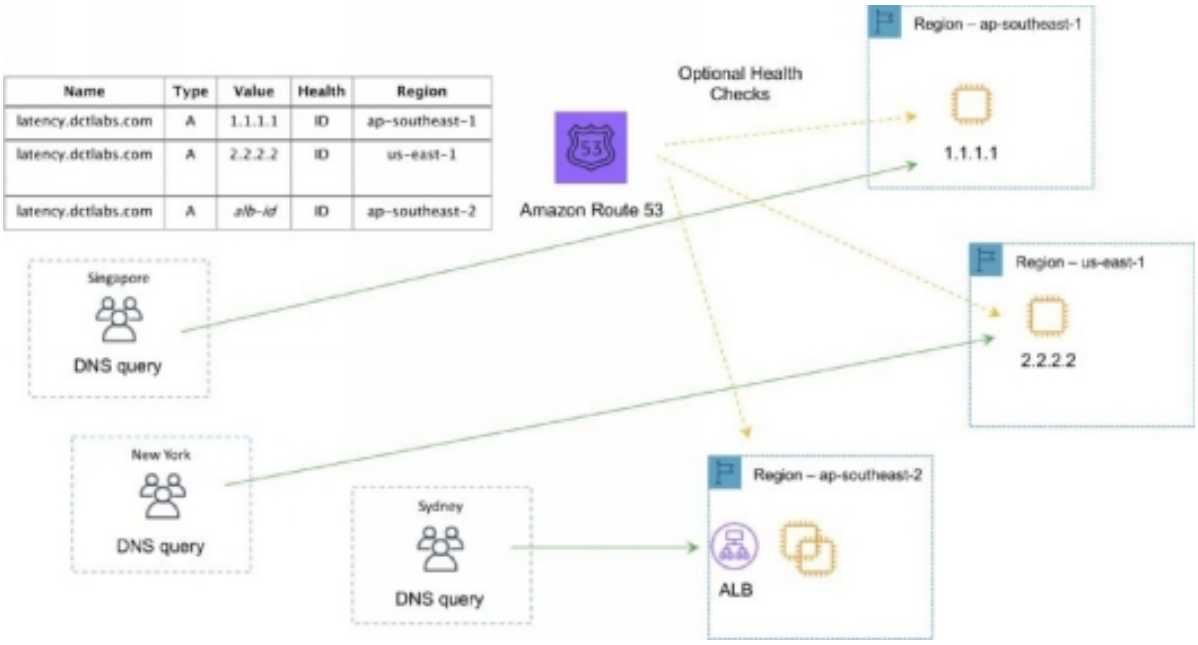
Geo-proximity routing policy (requires Route Flow):

- Use for routing traffic based on the location of resources and, optionally, shift traffic from resources in one location to resources in another.

Latency based routing:

- AWS maintains a database of latency from different parts of the world.
- Focussed on improving performance by routing to the region with the lowest latency.
- You create latency records for your resources in multiple EC2 locations.

The following diagram depicts an Amazon Route 53 Latency based routing policy configuration:



Multi-value answer routing policy:

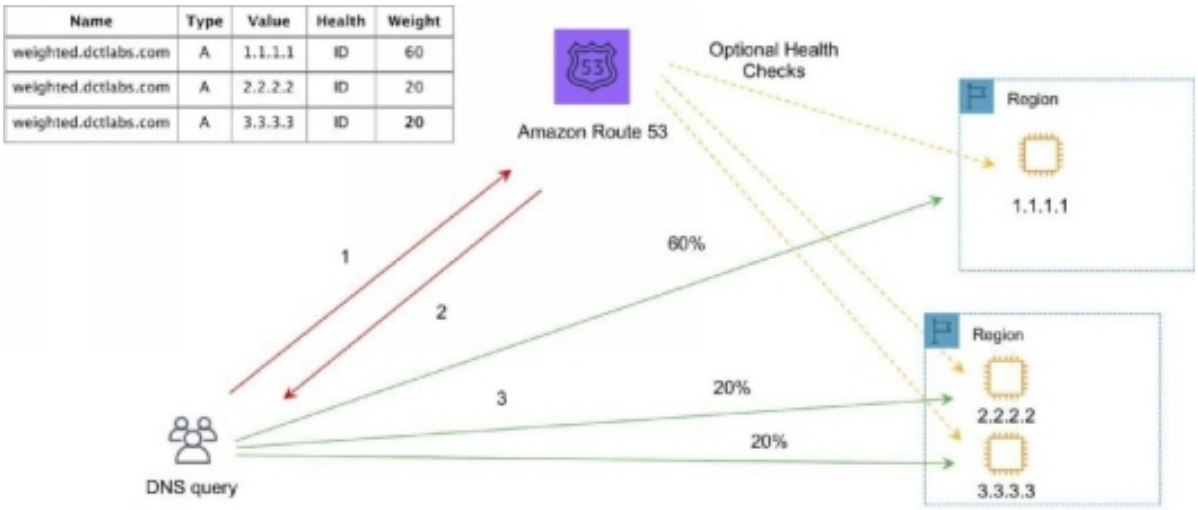
- Use for responding to DNS queries with up to eight healthy records selected at random.

The following diagram depicts an Amazon Route 53 Multivalue routing policy configuration:

Weighted:

- Similar to simple but you can specify a weight per IP address.
- You create records that have the same name and type and assign each record a relative weight.
- Numerical value that favours one IP over another.
- To stop sending traffic to a resource you can change the weight of the record to 0.

The following diagram depicts an Amazon Route 53 Weighted routing policy configuration:



TRAFFIC FLOW

Route 53 Traffic Flow provides Global Traffic Management (GTM) services.

Traffic flow policies allow you to create routing configurations for resources using routing types such as failover and geolocation.

Create policies that route traffic based on specific constraints, including latency, endpoint health, load, geo-proximity and geography.

Scenarios include:

- Adding a simple backup page in Amazon S3 for a website.
- Building sophisticated routing policies that consider an end user’s geographic location, proximity to an AWS region, and the health of each of your endpoints.

Amazon Route 53 Traffic Flow also includes a versioning feature that allows you to maintain a history of changes to your routing policies, and easily roll back to a previous policy version using the console or API.

ROUTE 53 RESOLVER

Route 53 Resolver is a set of features that enable bi-directional querying between on-premises and AWS over private connections.

Used for enabling DNS resolution for hybrid clouds.

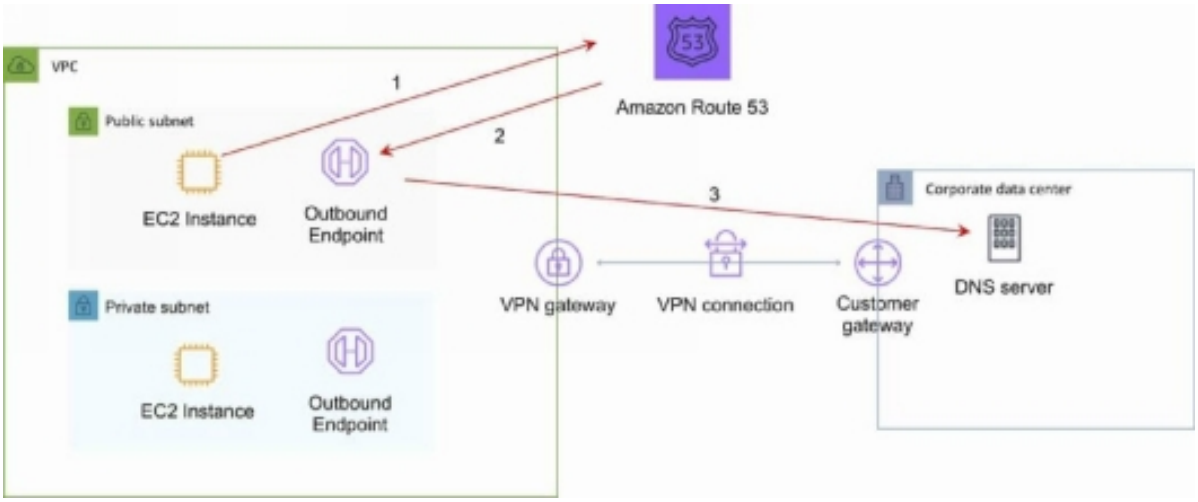
Route 53 Resolver Endpoints.

- Inbound query capability is provided by Route 53 Resolver Endpoints, allowing DNS queries that originate on-premises to resolve AWS hosted domains.
- Connectivity needs to be established between your on-premises DNS infrastructure and AWS through a Direct Connect (DX) or a Virtual Private Network (VPN).
- Endpoints are configured through IP address assignment in each subnet for which you would like to provide a resolver.

Conditional forwarding rules:

- Outbound DNS queries are enabled through the use of Conditional Forwarding Rules. .
- Domains hosted within your on-premises DNS infrastructure can be configured as forwarding rules in Route 53 Resolver.
- Rules will trigger when a query is made to one of those domains and will attempt to forward DNS requests to your DNS servers that were configured along with the rules.

- Like the inbound queries, this requires a private connection over DX or VPN.



CHARGES

You pay per hosted zone per month (no partial months).

A hosted zone deleted within 12 hours of creation is not charged (queries are charges).

Additional charges for:

- Queries.
- Traffic Flow.
- Health Checks.
- Route 53 Resolver ENIs + queries.
- Domain names.

Alias records are free of charge when the records are mapped to one of the following:

- Elastic Load Balancers.
- Amazon CloudFront distributions.

- AWS Elastic Beanstalk environments.
- Amazon S3 buckets that are configured as website endpoints.

Health checks are charged with different prices for AWS vs non-AWS endpoints.

You do not pay for the records that you add to your hosted zones.

Latency-based routing queries are more expensive.

Geo DNS and geo-proximity also have higher prices.

REFERENCES

https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/Welcome.html

https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/route-53-concepts.html

https://aws.amazon.com/route53/faqs/

https://aws.amazon.com/route53/pricing/

AMAZON S3

The Amazon Simple Storage Service (Amazon S3) is an object-based storage system.

Amazon S3 is built to store and retrieve any amount of data from anywhere on the Internet.

Amazon S3 stores objects and objects are stored in buckets.

It’s a simple storage service that offers an extremely durable, highly available, and infinitely scalable data storage infrastructure at very low costs.

Amazon S3 is a distributed architecture and objects are redundantly stored on multiple devices across multiple facilities (AZs) in an Amazon S3 region.

Amazon S3 is a simple key-based object store.

Amazon S3 data is made up of:

- Key (name).
- Value (data).
- Version ID.
- Metadata.
- Access Control Lists.

Keys can be any string, and they can be constructed to mimic hierarchical attributes.

Alternatively, you can use S3 Object Tagging to organize your data across all of your S3 buckets and/or prefixes.

Event notifications for specific actions, can send alerts or trigger actions.

Notifications can be sent to:

- SNS Topics.
- SQS Queue.
- Lambda functions.
- Need to configure SNS/SQS/Lambda before S3.
- No extra charges from S3 but you pay for SNS, SQS and Lambda.

S3 provides read after write consistency for PUTS of new objects.

S3 provides eventual consistency for overwrite PUTS and DELETES (takes time to propagate).

HTTP 200 code indicates a successful write to S3.

ADDITIONAL CAPABILITIES

Additional capabilities offered by Amazon S3 include:

Additional S3 Capability	How it Works
Transfer Acceleration	Speed up data uploads using CloudFront in reverse
Requester Pays	The requester rather than the bucket owner pays for requests and data transfer
Tags	Assign tags to objects to use in costing, billing, security etc.
Events	Trigger notifications to SNS, SQS, or Lambda when certain events happen in your bucket
Static Web Hosting	Simple and massively scalable static website hosting
BitTorrent	Use the BitTorrent protocol to retrieve any publicly available object by automatically generating a .torrent file

BUCKETS

Objects are stored in buckets:

- A bucket can be viewed as a container for objects.
- A bucket is a flat container of objects.
- It does not provide a hierarchy of objects.
- You can use an object key name (prefix) to mimic folders.

You can create folders in your buckets (only available through the Console).

You cannot create nested buckets.

Bucket names are part of the URL used to access the bucket.

An S3 bucket is region specific.

S3 is a universal namespace so names must be unique globally.

Can enable logging to a bucket.

Bucket naming:

- Bucket names must be at least 3 and no more than 63 character in length.
- Bucket names must start and end with a lowercase character or a number.
- Bucket names must be a series of one or more labels which are separated by a period.
- Bucket names can contain lowercase letters, numbers and hyphens.
- Bucket names cannot be formatted as an IP address.

For better performance, lower latency, and lower cost, create the bucket closer to your clients.

OBJECTS

Each object is stored and retrieved by a unique key (ID or name).

An object in S3 is uniquely identified and addressed through:

- Service end-point.
- Bucket name.
- Object key (name).
- Optionally, an object version.

Objects stored in a bucket will never leave the region in which they are stored unless you move them to another region or enable cross-region replication.

You can define permissions on objects when uploading and at any time afterwards using the AWS Management Console.

SUBRESOURCES

Sub-resources are subordinate to objects, they do not exist independently but are always associated with another entity such as an object or bucket.

Sub-resources (configuration containers) associated with buckets include:

- Lifecycle – define an object's lifecycle.
- Website – configuration for hosting static websites.
- Versioning – retain multiple versions of objects as they are changed.
- Access Control Lists (ACLs) – control permissions access to the bucket.
- Bucket Policies – control access to the bucket.
- Cross Origin Resource Sharing (CORS).
- Logging.

Sub-resources associated with objects include:

- ACLs – define permissions to access the object.
- Restore – restoring an archive.

CROSS-ORIGIN-RESOURCE-SHARING (CORS)

Used to allow requests to a different origin when connected to the main origin.

The request will fail unless the origin allows the requests using CORS headers (e.g. Access-Control-Allow-Origin).

Must enable the correct CORS headers.

Specify a CORS configuration on the S3 bucket.

STORAGE CLASSES

There are six S3 storage classes.

- S3 Standard (durable, immediately available, frequently accessed).
- S3 Intelligent-Tiering (automatically moves data to the most cost-effective tier).
- S3 Standard-IA (durable, immediately available, infrequently accessed).
- S3 One Zone-IA (lower cost for infrequently accessed data with less resilience).
- **S3 Glacier** (archived data, retrieval times in minutes or hours).
- S3 Glacier Deep Archive (lowest cost storage class for long term retention).

The table below provides the details of each Amazon S3 storage class:

	S3 Standard	S3 Intelligent-Tiering*	S3 Standard-IA	S3 One Zone-IA†	S3 Glacier	S3 Glacier Deep Archive
Designed for durability	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)
Designed for availability	99.99%	99.9%	99.9%	99.5%	99.99%	99.99%
Availability SLA	99.9%	99%	99%	99%	99.9%	99.9%
Availability Zones	≥3	≥3	≥3	1	≥3	≥3
Minimum capacity charge per object	N/A	N/A	128KB	128KB	40KB	40KB
Minimum storage duration charge	N/A	30 days	30 days	30 days	90 days	180 days
Retrieval fee	N/A	N/A	per GB retrieved	per GB retrieved	per GB retrieved	per GB retrieved
First byte latency	milliseconds	millseconds	milliseconds	milliseconds	select minutes or hours	select hours
Storage type	Object	Object	Object	Object	Object	Object
Lifecycle transitions	Yes	Yes	Yes	Yes	Yes	Yes

Objects stored in the S3 One Zone-IA storage class are stored redundantly within a single Availability Zone in the AWS Region you select.

TRANSFER ACCELERATION

Amazon S3 Transfer Acceleration enables fast, easy, and secure transfers of files over long distances between your client and your Amazon S3 bucket.

S3 Transfer Acceleration leverages **Amazon CloudFront** 's globally distributed AWS Edge Locations.

Used to accelerate object uploads to S3 over long distances (latency).

Transfer acceleration is as secure as a direct upload to S3.

You are charged only if there was a benefit in transfer times.

Need to enable transfer acceleration on the S3 bucket.

Cannot be disabled, can only be suspended.

May take up to 30 minutes to implement.

URL is: <bucketname>.s3-accelerate.amazonaws.com.

STATIC WEBSITES

S3 can be used to host static websites.

Cannot use dynamic content such as PHP, .Net etc.

Automatically scales.

You can use a custom domain name with S3 using a Route 53 Alias record.

When using a custom domain name the bucket name must be the same as the domain name.

Can enable redirection for the whole domain, pages or specific objects.

URL is: <bucketname>.s3-website-.amazonaws.com.

PRE-SIGNED URLS

Pre-signed URLs can be used to provide temporary access to a specific object to those who do not have AWS credentials.

By default all objects are private and can only be accessed by the owner.

To share an object you can either make it public or generate a pre-signed URL.

Expiration date and time must be configured.

These can be generated using SDKs for Java and .Net and AWS explorer for Visual Studio.

Can be used for downloading and uploading S3 objects.

MFA DELETE

MFA delete forces the user to generate a code on a device before performing operations on S3.

You must enable versioning on the bucket.

MFA delete can be required for the following operations:

- Permanently delete an object version.
- Suspend versioning on the bucket.

Only the bucket owner (root account) can enable / disable MFA-delete.

VERSIONING

Versioning stores all versions of an object (including all writes and even if an object is deleted).

Versioning protects against accidental object/data deletion or overwrites.

Enables “roll-back” and “un-delete” capabilities.

Versioning can also be used for data retention and archive.

Old versions count as billable size until they are permanently deleted.

Enabling versioning does not replicate existing objects.

Can be used for backup.

Once enabled versioning cannot be disabled only suspended.

Can be integrated with lifecycle rules.

Multi-factor authentication (MFA) delete can be enabled.

MFA delete can also be applied to changing versioning settings.

MFA delete applies to:

- Changing the bucket’s versioning state.
- Permanently deleting an object.

Cross Region Replication requires versioning to be enabled on the source and destination buckets.

Reverting to previous versions isn’t replicated.

By default a HTTP GET retrieves the most recent version.

Only the S3 bucket owner can permanently delete objects once versioning is enabled.

When you try to delete an object with versioning enabled a DELETE marker is placed on the object.

You can delete the DELETE marker and the object will be available again.

Deletion with versioning replicates the delete marker. But deleting the delete marker is not replicated.

Bucket versioning states:

- Enabled.
- Versioned.
- Un-versioned.

Objects that existed before enabling versioning will have a version ID of NULL.

Suspension:

- If you suspend versioning the existing objects remain as they are however new versions will not be created.
- While versioning is suspended new objects will have a version ID of NULL and uploaded objects of the same name will overwrite the existing object.

OBJECT LIFECYCLE MANAGEMENT

Used to optimize storage costs, adhere to data retention policies and to keep S3 volumes well-maintained.

A *lifecycle configuration* is a set of rules that define actions that Amazon S3 applies to a group of objects. There are two types of actions:

- **Transition actions** —Define when objects transition to another **storage class** . For example, you might choose to transition objects to the STANDARD_IA storage class 30 days after you created them, or archive objects to the GLACIER storage class one year after creating them.

There are costs associated with the lifecycle transition requests. For pricing information, see **Amazon S3 Pricing** .

- **Expiration actions** —Define when objects expire. Amazon S3 deletes expired objects on your behalf.

Lifecycle configuration is an XML file applied at the bucket level as a subresource.

Can be used in conjunction with versioning or independently.

Can be applied to current and previous versions.

Can be applied to specific objects within a bucket: objects with a specific tag or objects with a specific prefix.

ENCRYPTION

You can securely upload/download your data to Amazon S3 via SSL endpoints using the HTTPS protocol (In Transit – SSL/TLS).

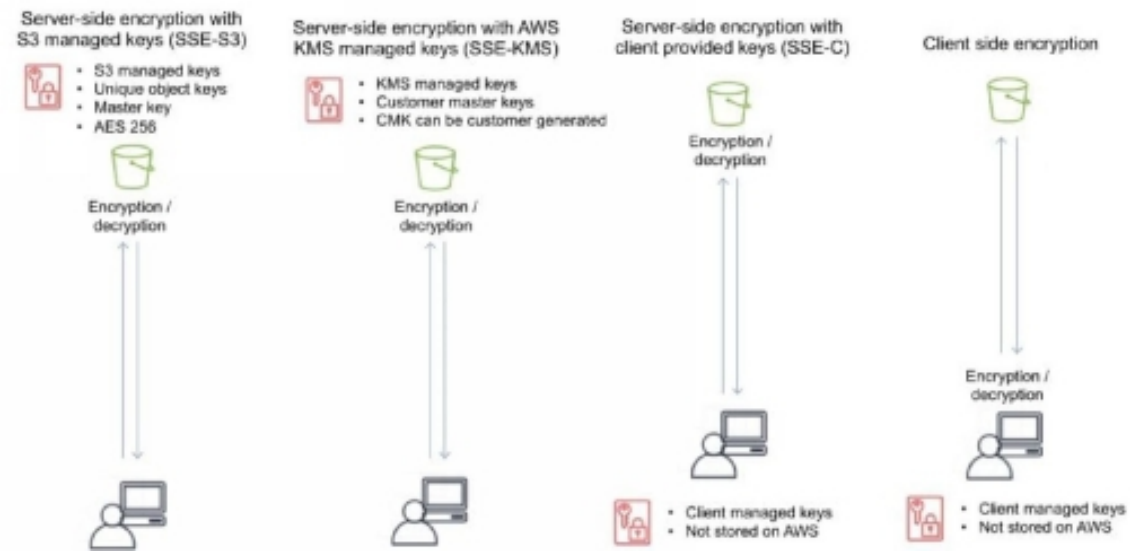
Encryption options:

Encryption Option	How it Works
SSE-S3	Use S3's existing encryption key for AES-256
SSE-C	Upload your own AES-256 encryption key which S3 uses when it writes objects
SSE-KMS	Use a key generated and managed by AWS KMS
Client-Side	Encrypt objects using your own local encryption process before uploading to S3

Server side encryption options:

- SSE-S3 – Server Side Encryption with S3 managed keys.
  - Each object is encrypted with a unique key.
  - Encryption key is encrypted with a master key.
  - AWS regularly rotate the master key.
  - Uses AES 256.
- SSE-KMS – Server Side Encryption with AWS KMS keys.
  - KMS uses Customer Master Keys (CMKs) to encrypt.
  - Can use the automatically created CMK key.
  - OR you can select your own key (gives you control for management of keys).
  - An envelope key protects your keys.
  - Chargeable.
- SSE-C – Server Side Encryption with client provided keys.
  - Client manages the keys, S3 manages encryption.
  - AWS does not store the encryption keys.
  - If keys are lost data cannot be decrypted.

The following diagram depicts the options for enabling encryption and shows you where the encryption is applied and where the keys are managed:



EVENT NOTIFICATIONS

Amazon S3 event notifications can be sent in response to actions in Amazon S3 like PUTs, POSTs, COPYs, or DELETEs.

Amazon S3 event notifications enable you to run workflows, send alerts, or perform other actions in response to changes in your objects stored in S3.

To enable notifications, you must first add a notification configuration that identifies the events you want Amazon S3 to publish and the destinations where you want Amazon S3 to send the notifications.

You can configure notifications to be filtered by the prefix and suffix of the key name of objects.

Amazon S3 can publish notifications for the following events:

- New object created events.
- Object removal events.
- Restore object events.
- Reduced Redundancy Storage (RRS) object lost events.
- Replication events.

Amazon S3 can send event notification messages to the following destinations:

- Publish event messages to an Amazon Simple Notification Service (Amazon SNS) topic.
- Publish event messages to an Amazon Simple Queue Service (Amazon SQS) queue.
- Publish event messages to AWS Lambda by invoking a Lambda function and providing the event message as an argument.

Need to grant Amazon S3 permissions to post messages to an Amazon SNS topic or an Amazon SQS queue.

Need to also grant Amazon S3 permission to invoke an AWS Lambda function on your behalf. For information about granting these permissions.

OBJECT TAGS

S3 object tags are key-value pairs applied to S3 objects which can be created, updated or deleted at any time during the lifetime of the object.

Allow you to create Identity and Access Management (IAM) policies, setup S3 Lifecycle policies, and customize storage metrics.

Up to ten tags can be added to each S3 object and you can use either the AWS Management Console, the REST API, the AWS CLI, or the AWS SDKs to add object tags.

CROSS REGION REPLICATION

CRR is an Amazon S3 feature that automatically replicates data across AWS Regions.

With CRR, every object uploaded to an S3 bucket is automatically replicated to a destination bucket in a different AWS Region that you choose.

Provides automatic, asynchronous copying of objects between buckets in different regions.

CRR is configured at the S3 bucket level.

You enable a CRR configuration on your source bucket by specifying a destination bucket in a different Region for replication.

You can use either the AWS Management Console, the REST API, the AWS CLI, or the AWS SDKs to enable CRR.

Versioning must be enabled for both the source and destination buckets .

Source and destination buckets must be in different regions.

With CRR you can only replication between regions, not within a region (see SRR below for single region replication).

Replication is 1:1 (one source bucket, to one destination bucket).

You can configure separate S3 Lifecycle rules on the source and destination buckets.

You can replicate KMS-encrypted objects by providing a destination KMS key in your replication configuration.

Triggers for replication are:

- Uploading objects to the source bucket.
- DELETE of objects in the source bucket.
- Changes to the object, its metadata, or ACL.

What is replicated:

- New objects created after enabling replication.
- Changes to objects.
- Objects created using SSE-S3 using the AWS managed key.
- Object ACL updates.

What isn’t replicated:

- Objects that existed before enabling replication (can use the copy API).
- Objects created with SSE-C and SSE-KMS.
- Objects to which the bucket owner does not have permissions.
- Updates to bucket-level subresources.
- Actions from lifecycle rules are not replicated.
- Objects in the source bucket that are replicated from another region are not replicated.

Deletion behaviour:

- If a DELETE request is made without specifying an object version ID a delete marker will be added and replicated.
- If a DELETE request is made specifying an object version ID the object is deleted but the delete marker is not replicated.

SAME REGION REPLICATION (SRR)

As the name implies you can use SRR to replication objects to a destination bucket within the same region as the source bucket.

This feature was released in September 2018.

Replication is automatic and asynchronous.

New objects uploaded to an Amazon S3 bucket are configured for replication at the bucket, prefix, or object tag levels.

Replicated objects can be owned by the same AWS account as the original copy or by different accounts, to protect from accidental deletion.

Replication can be to any Amazon S3 storage class, including **S3 Glacier** and S3 Glacier Deep Archive to create backups and long-term archives.

When an S3 object is replicated using SRR, the metadata, Access Control Lists (ACL), and object tags associated with the object are also part of the replication.

Once SRR is configured on a source bucket, any changes to the object, metadata, ACLs, or object tags trigger a new replication to the destination bucket.

S3 ANALYTICS

Can run analytics on data stored on Amazon S3.

This includes data lakes, IoT streaming data, machine learning, and artificial intelligence.

The following strategies can be used:

S3 Analytics Strategies	Service Used
Data Lake Concept	Athena, RedShift Spectrum, QuickSight
IoT Streaming Data Repository	Kinesis Firehose
Machine Learning and AI Storage	Rekognition, Lex, MXNet
Storage Class Analysis	S3 Management Analytics

S3 INVENTORY

You can use S3 Inventory to audit and report on the replication and encryption status of your objects for business, compliance, and regulatory needs.

Amazon S3 inventory provides comma-separated values (CSV), **Apache optimized row columnar (ORC)** or **Apache Parquet (Parquet)** output files that list your objects and their corresponding metadata on a daily or weekly basis for an S3 bucket or a shared prefix (that is, objects that have names that begin with a common string).

MONITORING AND REPORTING

Amazon **CloudWatch** metrics for Amazon S3 can help you understand and improve the performance of applications that use Amazon S3. There are several ways that you can use CloudWatch with Amazon S3.

- **Daily storage metrics for buckets** - Monitor bucket storage using CloudWatch, which collects and processes storage data from Amazon S3 into readable, daily metrics. These storage metrics for Amazon S3 are reported once per day and are provided to all customers at no additional cost.
- **Request metrics** - Monitor Amazon S3 requests to quickly identify and act on operational issues. The metrics are available at 1-minute intervals after some latency to process. These CloudWatch metrics are billed at the same rate as the Amazon CloudWatch custom metrics.
- **Replication metrics** - Monitor the total number of S3 API operations that are pending replication, the total size of objects pending replication, and the maximum replication time to the destination Region. Only replication rules that have S3 Replication Time Control (S3 RTC) enabled will publish replication metrics.

LOGGING AND AUDITING

You can record the actions that are taken by users, roles, or AWS services on Amazon S3 resources and maintain log records for auditing and compliance purposes.

To do this, you can use **Amazon S3 server access logging** , **AWS CloudTrail logs** , or a combination of both.

AWS recommend that you use AWS CloudTrail for **logging bucket and object-level actions** for your Amazon S3 resources.

Server access logging provides detailed records for the requests that are made to a bucket. This information can be used for auditing. You must not

set the bucket being logged to be the destination for the logs as this creates a logging loop and the bucket will grow in size exponentially.

AUTHORIZATION AND ACCESS CONTROL

By default, only the resource owner can access buckets and objects. The resource owner refers to the AWS account that creates the resource.

Access policy describes who has access to what. You can associate an access policy with a resource (bucket and object) or a user.

You can categorize the available Amazon S3 access policies as follows:

- **Resource-based policies** – Bucket policies and access control lists (ACLs) are resource-based because you attach them to your Amazon S3 resources.
- **User policies** – You can use IAM to manage access to your Amazon S3 resources. You can create IAM users, groups, and roles in your account and attach access policies to them granting them access to AWS resources, including Amazon S3.

IAM policies (user policies) can be used to apply permissions (allow/deny) for specific API actions to users and groups. Resources to apply the policy to can be buckets and objects.

Bucket policies can be used to grant public access to the bucket, force objects to be encrypted at upload, and to grant access another AWS account (cross-account access).

ACLs can be applied at the object-level and bucket-level:

- Object ACL – apply permissions at the individual object level (finer level of granularity).
- Bucket ACL – apply permissions at the bucket level.

Access to Amazon S3 buckets is blocked by default (**block public access** feature).

An IAM principal can access an S3 object only if the user permissions allow it or the resource policy allows it (and there’s no explicit deny).

You can use Access Analyzer for S3 to review all buckets that have bucket access control lists (ACLs), bucket policies, or access point policies that grant public or shared access.

Access Analyzer for S3 alerts you to buckets that are configured to allow access to anyone on the internet or other AWS accounts, including AWS accounts outside of your organization.

AMAZON S3 GLACIER

Amazon S3 Glacier is an archiving storage solution for infrequently accessed data.

There are two **Amazon S3** storage tiers for using S3 Glacier:

S3 GLACIER

- Same low latency and high throughput performance of S3 Standard.
- Designed for durability of 99.999999999% of objects in a single Availability Zone†.
- Designed for 99.99% availability.
- Backed with the **Amazon S3 Service Level Agreement** for availability.
- Supports SSL for data in transit and encryption of data at rest.
- S3 Lifecycle management for automatic migration of objects to other S3 Storage Classes.

S3 GLACIER DEEP ARCHIVE

- Designed for durability of 99.999999999% of objects across multiple Availability Zones.
- Data is resilient in the event of one entire Availability Zone destruction.
- Supports SSL for data in transit and encryption of data at rest.
- Low-cost design is ideal for long-term archive.

- Configurable retrieval times, from minutes to hours.
- S3 PUT API for direct uploads to S3 Glacier, and S3 Lifecycle management for automatic migration of objects.

The key difference between the tiers is that Deep Archive is lower cost but retrieval times are much longer (12 hours).

The S3 Glacier tier has configurable retrieval times from minutes to hours (you pay accordingly).

Archived objects are not available for real time access and you need to submit a retrieval request.

Glacier must complete a job before you can get its output.

Requested archival data is copied to S3 One Zone-IA.

Following retrieval you have 24 hours to download your data.

You cannot specify Glacier as the storage class at the time you create an object.

S3 Glacier is designed to sustain the loss of two facilities.

S3 Glacier automatically encrypts data at rest using AES 256 symmetric keys and supports secure transfer of data over SSL.

S3 Glacier may not be available in all AWS regions.

S3 Glacier objects are visible through S3 only (not Glacier directly).

S3 Glacier does not archive object metadata, you need to maintain a client-side database to maintain this information.

Archives can be 1 bytes up to 40TB.

S3 Glacier file archives of 1 byte – 4 GB can be performed in a single operation.

S3 Glacier file archives from 100MB up to 40TB can be uploaded to Glacier using the multipart upload API.

Uploading archives is synchronous.

Downloading archives is asynchronous.

The contents of an archive that has been uploaded cannot be modified.

You can upload data to Glacier using the CLI, SDKs or APIs – you cannot use the AWS Console.

S3 Glacier adds 32-40KB (indexing and archive metadata) to each object when transitioning from other classes using lifecycle policies.

AWS recommends that if you have lots of small objects they are combined in an archive (e.g. zip file) before uploading.

A description can be added to archives, no other metadata can be added.

S3 Glacier archive IDs are added upon upload and are unique for each upload.

Archive retrieval:

- Expedited is 1-5 minutes retrieval (most expensive).
- Standard is 3.5 hours retrieval (cheaper, 10GB data retrieval free per month).
- Bulk retrieval is 5-12 hours (cheapest, use for large quantities of data).

You can retrieve parts of an archive.

When data is retrieved it is copied to S3 and the archive remains in Glacier and the storage class therefore does not change.

AWS SNS can send notifications when retrieval jobs are complete.

Retrieved data is available for 24 hours by default (can be changed).

To retrieve specific objects within an archive you can specify the byte range (Range) in the HTTP GET request (need to maintain a DB of byte ranges).

S3 GLACIER VAULT

A vault is a collection of archives.

Vault has:

- One vault access policy
- One vault lock policy

Vault policies are written in JSON.

A vault access policy is similar to a bucket policy.

A vault lock policy is a policy that is locked for regulatory and compliance requirements. This type of policy is immutable – it can never be changed.

AMAZON CLOUDFRONT

Amazon CloudFront is a web service that gives businesses and web application developers an easy and cost-effective way to distribute content with low latency and high data transfer speeds.

Amazon CloudFront is a good choice for distribution of frequently accessed static content that benefits from edge delivery—like popular website images, videos, media files or software downloads.

Used for dynamic, static, streaming, and interactive content.

Amazon CloudFront is a global service:

- Ingress to upload objects.
- Egress to distribute content.

Amazon CloudFront provides a simple API that lets you:

- Distribute content with low latency and high data transfer rates by serving requests using a network of edge locations around the world.
- Get started without negotiating contracts and minimum commitments.

DEPLOYMENT AND PROVISIONING

DOMAIN NAMES

CloudFront typically creates a domain name such as a232323.cloudfront.net.

You can use a zone apex name on CloudFront.

CloudFront supports wildcard CNAME.

Alternate domain names can be added using an alias record (Route 53).

For other service providers use a CNAME (cannot use the zone apex with CNAME).

Moving domain names between distributions:

- You can move subdomains yourself.
- For the root domain you need to use AWS support.

Supports wildcard SSL certificates, Dedicated IP, Custom SSL and SNI Custom SSL (cheaper).

Supports Perfect Forward Secrecy which creates a new private key for each SSL session.

EDGE LOCATIONS AND REGIONAL EDGE CACHES

An edge location is the location where content is cached (separate to AWS regions/AZs).

Requests are automatically routed to the nearest edge location.

Edge locations are not tied to Availability Zones or regions.

Regional Edge Caches are located between origin web servers and global edge locations and have a larger cache.

Regional Edge Caches have larger cache-width than any individual edge location, so your objects remain in cache longer at these locations.

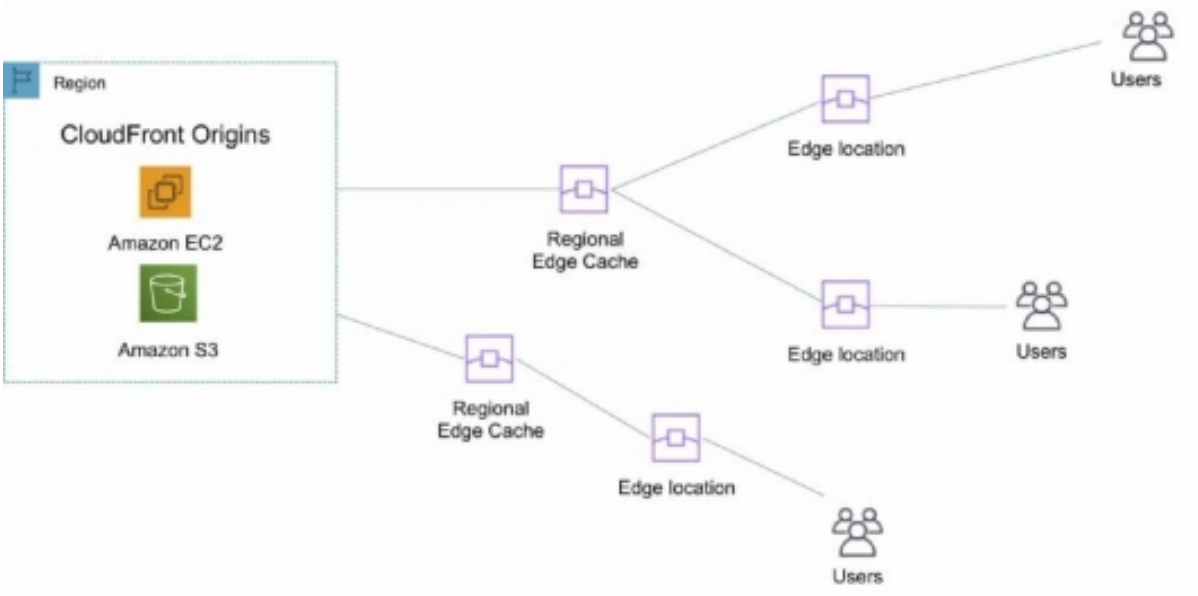
Regional Edge caches aim to get content closer to users.

Proxy methods PUT/POST/PATCH/OPTIONS/DELETE go directly to the origin from the edge locations and do not proxy through Regional Edge caches.

Dynamic content goes straight to the origin and does not flow through Regional Edge caches.

Edge locations are not just read only, you can write to them too.

The diagram below shows where Regional Edge Caches and Edge Locations are placed in relation to end users:



ORIGINS

An origin is the origin of the files that the CDN will distribute.

Origins can be either an **Amazon S3 bucket** , an **EC2 instance** , an **Elastic Load Balancer** , or Route 53 – can also be external (non-AWS).

When using Amazon S3 as an origin you place all of your objects within the bucket.

You can use an existing bucket and the bucket is not modified in any way.

By default all newly created buckets are private.

You can setup access control to your buckets using:

- Bucket policies.
- Access Control Lists.

You can make objects publicly available or use CloudFront signed URLs.

A custom origin server is a HTTP server which can be an **Amazon EC2** instance or an on-premise/non-AWS based web server.

When using an on-premise or non-AWS based web server you must specify the DNS name, ports and protocols that you want CloudFront to use when fetching objects from your origin.

Most CloudFront features are supported for custom origins except RTMP distributions (must be an Amazon S3 bucket).

When using EC2 for custom origins Amazon recommend:

- Use an AMI that automatically installs the software for a web server.
- Use ELB to handle traffic across multiple EC2 instances.
- Specify the URL of your load balancer as the domain name of the origin server.

Amazon S3 static website:

- Enter the S3 static website hosting endpoint for your bucket in the configuration.
- Example: http://<bucketname>.s3-website-<region>.amazonaws.com.

Objects are cached for 24 hours by default.

The expiration time is controlled through the TTL.

The minimum expiration time is 0.

Static websites on Amazon S3 are considered custom origins.

AWS origins are Amazon S3 buckets (not a static website).

CloudFront keeps persistent connections open with origin servers.

Files can also be uploaded to CloudFront.

High availability with Origin Failover:

- Can set up CloudFront with origin failover for scenarios that require high availability.

- Uses an origin group in which you designate a primary origin for CloudFront plus a second origin that CloudFront automatically switches to when the primary origin returns specific HTTP status code failure responses.
- For more info, check this **article** .
- Also works with Lambda@Edge functions.

DISTRIBUTIONS

To distribute content with CloudFront you need to create a distribution.

The distribution includes the configuration of the CDN including:

- Content origins.
- Access (public or restricted).
- Security (HTTP or HTTPS).
- Cookie or query-string forwarding.
- Geo-restrictions.
- Access logs (record viewer activity).

There are two types of distribution.

Web Distribution:

- Static and dynamic content including .html, .css, .php, and graphics files.
- Distributes files over HTTP and HTTPS.
- Add, update, or delete objects, and submit data from web forms.
- Use live streaming to stream an event in real time.

RTMP (deprecated but can still be deployed):

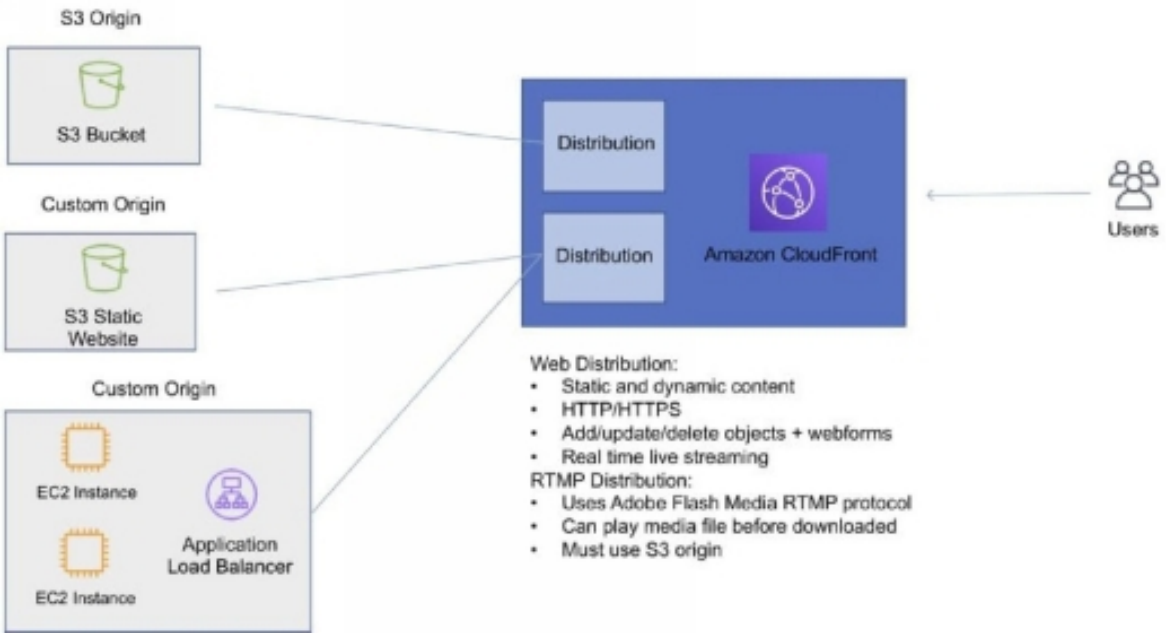
- Distribute streaming media files using Adobe Flash Media Server’s RTMP protocol.

- Allows an end user to begin playing a media file before the file has finished downloading from a CloudFront edge location.
- Files must be stored in an S3 bucket.

To use CloudFront live streaming, create a web distribution.

To delete a distribution it must first be disabled (can take up to 15 minutes).

The diagram below depicts Amazon CloudFront Distributions and Origins:



CACHE BEHAVIOR

Allows you to configure a variety of CloudFront functionality for a given URL path pattern.

For each cache behavior you can configure the following functionality:

- The path pattern (e.g. /images/*.jpg, /images*.php).
- The origin to forward requests to (if there are multiple origins).
- Whether to forward query strings.
- Whether to require signed URLs.
- Allowed HTTP methods.

- Minimum amount of time to retain the files in the CloudFront cache (regardless of the values of any cache-control headers).

The default cache behavior only allows a path pattern of /*.

Additional cache behaviors need to be defined to change the path pattern following creation of the distribution.

You can restrict access to content using the following methods:

- Restrict access to content using signed cookies or signed URLs.
- Restrict access to objects in your S3 bucket.

A special type of user called an Origin Access Identity (OAI) can be used to restrict access to content in an Amazon S3 bucket.

By using an OAI you can restrict users so they cannot access the content directly using the S3 URL, they must connect via CloudFront.

You can define the viewer protocol policy:

- HTTP and HTTPS.
- Redirect HTTP to HTTPS.
- HTTPS only.

You can define the Allowed HTTP Methods:

- GET, HEAD.
- GET, HEAD, OPTIONS.
- GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE.

For web distributions you can configure CloudFront to require that viewers use HTTPS.

Field-Level Encryption:

- Field-level encryption adds an additional layer of security on top of HTTPS that lets you protect specific data so that it is only visible to specific applications.

- Field-level encryption allows you to securely upload user-submitted sensitive information to your web servers.
- The sensitive information is encrypted at the edge closer to the user and remains encrypted throughout application processing.

Origin policy:

- HTTPS only.
- Match viewer – CloudFront matches the protocol with your custom origin.
- Use match viewer only if you specify Redirect HTTP to HTTPS or HTTPS only for the viewer protocol policy.
- CloudFront caches the object once even if viewers makes requests using HTTP and HTTPS.

Object invalidation:

- You can remove an object from the cache by invalidating the object.
- You cannot cancel an invalidation after submission.
- You cannot invalidate media files in the Microsoft Smooth Streaming format when you have enabled Smooth Streaming for the corresponding cache behavior.

Objects are cached for the TTL (always recorded in seconds, default is 24 hours, default max is 1 year).

Only caches for GET requests (not PUT, POST, PATCH, DELETE).

Dynamic content is cached.

Consider how often your files change when setting the TTL.

Invalidation can be used to immediately revoke cached objects – chargeable.

Deletions propagate.

RESTRICTIONS

Blacklists and whitelists can be used for geography – you can only use one at a time.

There are two options available for geo-restriction (geo-blocking):

- Use the CloudFront geo-restriction feature (use for restricting access to all files in a distribution and at the country level).
- Use a 3rd party geo-location service (use for restricting access to a subset of the files in a distribution and for finer granularity at the country level).

AWS WAF

AWS WAF is a web application firewall that lets you monitor HTTP and HTTPS requests that are forwarded to CloudFront and lets you control access to your content.

With AWS WAF you can shield access to content based on conditions in a web access control list (web ACL) such as:

- Origin IP address.
- Values in query strings.

CloudFront responds to requests with the requested content or an HTTP 403 status code (forbidden).

CloudFront can also be configured to deliver a custom error page.

Need to associate the relevant distribution with the web ACL.

SECURITY

PCI DSS compliant but recommended not to cache credit card information at edge locations.

HIPAA compliant as a HIPAA eligible service.

Distributed Denial of Service (DDoS) protection:

- CloudFront distributes traffic across multiple edge locations and filters requests to ensure that only valid HTTP(S) requests will be forwarded to backend hosts. CloudFront also supports geoblocking, which you can use to prevent requests from particular geographic locations from being served.

HIGH AVAILABILITY

CloudFront caches content at Edge Locations around the world. The more objects served by the cache, the fewer the requests to the origin. This reduces the load on your origin server and reduces latency.

You can set up CloudFront with **origin failover** for scenarios that require high availability.

To set up origin failover, you must have a distribution with at least two origins. Next, you create an origin group for your distribution that includes two origins, setting one as the primary. Finally, you create or update a cache behavior to use the origin group.

MONITORING AND REPORTING

You can view **operational metrics** about your CloudFront distributions and Lambda@Edge functions in the CloudFront console.

The following default metrics are included for all CloudFront distributions, at no additional cost:

Requests

The total number of viewer requests received by CloudFront, for all HTTP methods and for both HTTP and HTTPS requests.

Bytes downloaded

The total number of bytes downloaded by viewers for GET, HEAD, and OPTIONS requests.

Bytes uploaded

The total number of bytes that viewers uploaded to your origin with CloudFront, using POST and PUT requests.

4xx error rate

The percentage of all viewer requests for which the response's HTTP status code is 4xx.

5xx error rate

The percentage of all viewer requests for which the response's HTTP status code is 5xx.

Total error rate

The percentage of all viewer requests for which the response's HTTP status code is 4xx or 5xx.

In addition to the default metrics, you can enable additional metrics for an additional cost.

These additional metrics must be enabled for each distribution separately:

Cache hit rate

The percentage of all cacheable requests for which CloudFront served the content from its cache. HTTP POST and PUT requests, and errors, are not considered cacheable requests.

Origin latency

The total time spent from when CloudFront receives a request to when it starts providing a response to the network (not the viewer), for requests that are served from the origin, not the CloudFront cache. This is also known as *first byte latency* , or *time-to-first-byte* .

Error rate by status code

The percentage of all viewer requests for which the response's HTTP status code is a particular code in the 4xx or 5xx range. This metric is available for all of the following error codes: 401, 403, 404, 502, 503, and 504.

LOGGING AND AUDITING

S3 buckets can be configured to create access logs and cookie logs which log all requests made to the S3 bucket.

Amazon Athena can be used to analyze access logs.

CloudFront is integrated with CloudTrail.

CloudTrail saves logs to the S3 bucket you specify.

CloudTrail captures information about all requests whether they were made using the CloudFront console, the CloudFront API, the AWS SDKs, the CloudFront CLI, or another service.

CloudTrail can be used to determine which requests were made, the source IP address, who made the request etc.

To view CloudFront requests in CloudTrail logs you must update an existing trail to include global services.

AMAZON RDS

Amazon Relational Database Service (Amazon RDS) is a managed service that makes it easy to set up, operate, and scale a relational database in the cloud.

Amazon RDS is an Online Transaction Processing (OLTP) type of database.

The primary use case for Amazon RDS is a transactional database (rather than analytical).

It is best suited to structured, relational data store requirements.

It aims to be drop-in replacement for existing on-premise instances of the same databases.

Automated backups and patching are applied in customer-defined maintenance windows.

Allows push-button scaling, replication and redundancy.

Amazon RDS supports the following database engines:

- Amazon Aurora.
- MySQL.
- MariaDB.
- Oracle.
- SQL Server.
- PostgreSQL.

Amazon RDS is a managed service and you do not have access to the underlying **EC2 instance** (no root access).

The RDS service includes the following:

- Security and patching of the DB instances.

- Automated backup for the DB instances.
- Software updates for the DB engine.
- Easy scaling for storage and compute.
- Multi-AZ option with synchronous replication.
- Automatic failover for Multi-AZ option.
- Read replicas option for read heavy workloads.

A DB instance is a database environment in the cloud with the compute and storage resources you specify.

Database instances are accessed via endpoints.

Endpoints can be retrieved via the DB instance description in the AWS Management Console, **DescribeDBInstances** API or **describe-db-instances** command.

By default, customers are allowed to have up to a total of 40 Amazon RDS DB instances (only 10 of these can be Oracle or MS SQL unless you have your own licences).

Maintenance windows are configured to allow DB instances modifications to take place such as scaling and software patching (some operations require the DB instance to be taken offline briefly).

You can define the maintenance window or AWS will schedule a 30 minute window.

Windows integrated authentication for SQL only works with domains created using the AWS directory service – need to establish a trust with an on-premise AD directory.

Events and Notifications:

- Amazon RDS uses AWS SNS to send RDS events via SNS notifications.
- You can use API calls to the Amazon RDS service to list the RDS events in the last 14 days (DescribeEvents API).
- You can view events from the last 14 days using the CLI.

- Using the AWS Console you can only view RDS events for the last 1 day.

ENCRYPTION

You can encrypt your Amazon RDS instances and snapshots at rest by enabling the encryption option for your Amazon RDS DB instance.

Encryption at rest is supported for all DB types and uses AWS KMS.

When using encryption at rest the following elements are also encrypted:

- All DB snapshots.
- Backups.
- DB instance storage.
- Read Replicas.

You cannot encrypt an existing DB, you need to create a snapshot, copy it, encrypt the copy, then build an encrypted DB from the snapshot.

Data that is encrypted at rest includes the underlying storage for a DB instance, its automated backups, Read Replicas, and snapshots.

A Read Replica of an Amazon RDS encrypted instance is also encrypted using the same key as the master instance when both are in the same region.

If the master and Read Replica are in different regions, you encrypt using the encryption key for that region.

You can’t have an encrypted Read Replica of an unencrypted DB instance or an unencrypted Read Replica of an encrypted DB instance.

Encryption/decryption is handled transparently.

RDS supports SSL encryption between applications and RDS DB instances.

RDS generates a certificate for the instance.

DB SUBNET GROUPS

A DB subnet group is a collection of subnets (typically private) that you create in a VPC and that you then designate for your DB instances.

Each DB subnet group should have subnets in at least two Availability Zones in a given region.

It is recommended to configure a subnet group with subnets in each AZ (even for standalone instances).

During the creation of an RDS instance you can select the DB subnet group and the AZ within the group to place the RDS DB instance in.

You cannot pick the IP within the subnet that is allocated.

BILLING AND PROVISIONING

AWS Charge for:

- DB instance hours (partial hours are charged as full hours).
- Storage GB/month.
- I/O requests/month – for magnetic storage.
- Provisioned IOPS/month – for RDS provisioned IOPS SSD.
- Egress data transfer.
- Backup storage (DB backups and manual snapshots).

Backup storage for the automated RDS backup is free of charge up to the provisioned EBS volume size.

However, AWS replicate data across multiple AZs and so you are charged for the extra storage space on S3.

For multi-AZ you are charged for:

- Multi-AZ DB hours.
- Provisioned storage.
- Double write I/Os.

For multi-AZ you are not charged for DB data transfer during replication from primary to standby.

Oracle and Microsoft SQL licences are included or you can bring your own (BYO).

On-demand and reserved instance pricing available.

Reserved instances are defined based on the following attributes which must not be changed:

- DB engine.
- DB instance class.
- Deployment type (standalone, multi-AZ).
- License model.
- Region.

Reserved instances:

- Can be moved between AZs in the same region.
- Are available for multi-AZ deployments.
- Can be applied to Read Replicas if DB instance class and region are the same.
- Scaling is achieved through changing the instance class for compute, and modifying storage capacity for additional storage allocation.

SCALABILITY

You can only scale vertically for database writes (by changing the instance type).

You can scale horizontally for reads/queries by adding a read replica.

You cannot decrease the allocated storage for an RDS instance.

Scaling storage can happen while the RDS instance is running without outage however there may be performance degradation.

Scaling compute will cause downtime.

You can choose to have changes take effect immediately, however the default is within the maintenance window.

Scaling requests are applied during the the specified maintenance window unless “apply immediately” is used.

All RDS DB types support a maximum DB size of 64 TiB except for Microsoft SQL Server (16 TiB).

PERFORMANCE

Amazon RDS uses EBS volumes (never uses instance store) for DB and log storage.

There are three storage types available: General Purpose (SSD), Provisioned IOPS (SSD), and Magnetic.

General Purpose (SSD):

- Use for Database workloads with moderate I/O requirement.
- Cost effective.
- Also called gp2.
- 3 IOPS/GB.
- Burst up to 3000 IOPS.

Provisioned IOPS (SSD):

- Use for I/O intensive workloads.
- Low latency and consistent I/O.
- User specified IOPS (see table below).

For provisioned IOPS storage the table below shows the range of Provisioned IOPS and storage size range for each database engine.

Database Engine	Range of Provisioned IOPS	Range of Storage
MariaDB	1,000–40,000 IOPS	100 GiB–16 TiB
SQL Server, Enterprise and Standard editions	1000–32,000 IOPS	200 GiB–16 TiB
SQL Server, Web and Express editions	1000–32,000 IOPS	100 GiB–16 TiB
MySQL	1,000–40,000 IOPS	100 GiB–16 TiB
Oracle	1,000–40,000 IOPS	100 GiB–16 TiB
PostgreSQL	1,000–40,000 IOPS	100 GiB–16 TiB

Magnetic:

- Not recommended anymore, available for backwards compatibility.
- Doesn’t allow you to scale storage when using the SQL Server database engine.
- Doesn’t support elastic volumes.
- Limited to a maximum size of 4 TiB.
- Limited to a maximum of 1,000 IOPS.

MULTI-AZ AND READ REPLICAS

Multi-AZ and Read Replicas are used for high availability, fault tolerance and performance scaling.

The table below compares multi-AZ deployments to Read Replicas:

Multi-AZ Deployments	Read Replicas
Synchronous replication – highly durable	Asynchronous replication – highly scalable
Only database engine on primary instance is active	All read replicas are accessible and can be used for read scaling
Automated backups are taken from standby	No backups configured by default
Always span two Availability Zones within a single Region	Can be within an Availability Zone, Cross-AZ, or Cross-Region
Database engine version upgrades happen on primary	Database engine version upgrade is independent from source instance
Automatic failover to standby when a problem is detected	Can be manually promoted to a standalone database instance

MULTI-AZ

Multi-AZ RDS creates a replica in another AZ and synchronously replicates to it (DR only).

There is an option to choose multi-AZ during the launch wizard.

AWS recommends the use of provisioned IOPS storage for multi-AZ RDS DB instances.

Each AZ runs on its own physically distinct, independent infrastructure, and is engineered to be highly reliable.

You cannot choose which AZ in the region will be chosen to create the standby DB instance.

You can view which AZ the standby DB instance is created in.

A failover may be triggered in the following circumstances:

- Loss of primary AZ or primary DB instance failure.
- Loss of network connectivity on primary.
- Compute (EC2) unit failure on primary.
- Storage (EBS) unit failure on primary.
- The primary DB instance is changed.
- Patching of the OS on the primary DB instance.
- Manual failover (reboot with failover selected on primary).

During failover RDS automatically updates configuration (including DNS endpoint) to use the second node.

Depending on the instance class it can take 1 to a few minutes to failover to a standby DB instance.

It is recommended to implement DB connection retries in your application.

Recommended to use the endpoint rather than the IP address to point applications to the RDS DB.

The method to initiate a manual RDS DB instance failover is to reboot selecting the option to failover.

A DB instance reboot is required for changes to take effect when you change the DB parameter group or when you change a static DB parameter.

The DB parameter group is a configuration container for the DB engine configuration.

You will be alerted by a DB instance event when a failover occurs.

The secondary DB in a multi-AZ configuration cannot be used as an independent read node (read or write).

There is no charge for data transfer between primary and secondary RDS instances.

Multi-AZ deployments for the MySQL, MariaDB, Oracle and PostgreSQL engines use Amazon's failover technology.

Multi-AZ deployments for the SQL Server engine use SQL Server Database Mirroring (DBM).

System upgrades like OS patching, DB Instance scaling and system upgrades, are applied first on the standby, before failing over and modifying the other DB Instance.

In multi-AZ configurations snapshots and automated backups are performed on the standby to avoid I/O suspension on the primary instance.

Read Replica Support for Multi-AZ:

- Amazon RDS Read Replicas for MySQL and MariaDB support Multi-AZ deployments.
- Combining Read Replicas with Multi-AZ enables you to build a resilient disaster recovery strategy and simplify your database engine upgrade process.
- A Read Replica in a different region than the source database can be used as a standby database and promoted to become the new production database in case of a regional disruption.

- This allows you to scale reads whilst also having multi-AZ for DR.

The process for implementing maintenance activities is as follows:

- Perform operations on standby.
- Promote standby to primary.
- Perform operations on new standby (demoted primary).

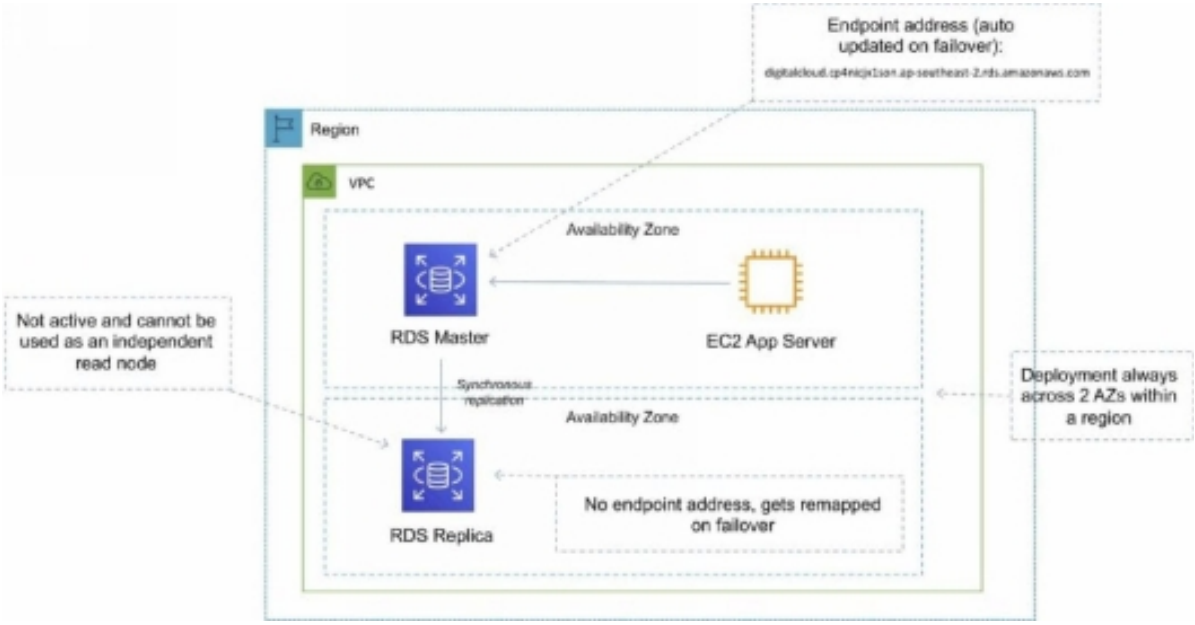
You can manually upgrade a DB instance to a supported DB engine version from the AWS Console.

By default upgrades will take effect during the next maintenance window.

You can optionally force an immediate upgrade.

In multi-AZ deployments version upgrades will be conducted on both the primary and standby at the same time causing an outage of both DB instance.

Ensure security groups and NACLs will allow your application servers to communicate with both the primary and standby instances.



READ REPLICAS

Read replicas are used for read heavy DBs and replication is asynchronous.

Read replicas are for workload sharing and offloading.

Read replicas provide read-only DR.

Read replicas are created from a snapshot of the master instance.

Must have automated backups enabled on the primary (retention period > 0).

Only supported for transactional database storage engines (InnoDB not MyISAM).

Read replicas are available for MySQL, PostgreSQL, MariaDB, Oracle and Aurora (not SQL Server).

For the MySQL, MariaDB, PostgreSQL, and Oracle database engines, Amazon RDS creates a second DB instance using a snapshot of the source DB instance.

It then uses the engines' native asynchronous replication to update the read replica whenever there is a change to the source DB instance.

Amazon Aurora employs an SSD-backed virtualized storage layer purpose-built for database workloads.

You can take snapshots of PostgreSQL read replicas but cannot enable automated backups.

You can enable automatic backups on MySQL and MariaDB read replicas.

You can enable writes to the MySQL and MariaDB Read Replicas.

You can have 5 read replicas of a production DB.

You cannot have more than four instances involved in a replication chain.

You can have read replicas of read replicas for MySQL and MariaDB but not for PostgreSQL.

Read replicas can be configured from the AWS Console or the API.

You can specify the AZ the read replica is deployed in.

The read replicas storage type and instance class can be different from the source but the compute should be at least the performance of the source.

You cannot change the DB engine.

In a multi-AZ failover the read replicas are switched to the new primary.

Read replicas must be explicitly deleted.

If a source DB instance is deleted without deleting the replicas each replica becomes a standalone single-AZ DB instance.

You can promote a read replica to primary.

Promotion of read replicas takes several minutes.

Promoted read replicas retain:

- Backup retention window.
- Backup window.
- DB parameter group.

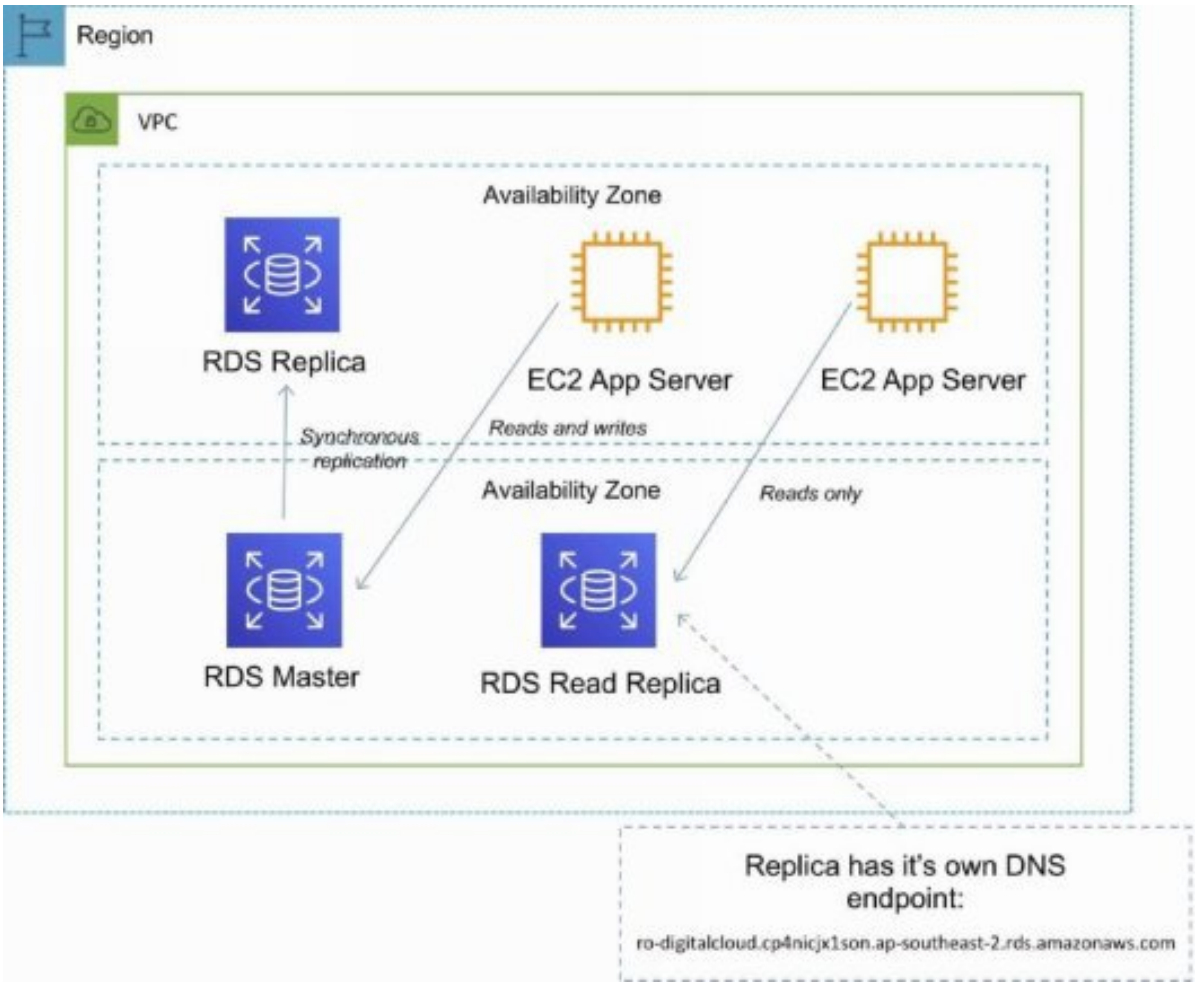
Existing read replicas continue to function as normal.

Each read replica has its own DNS endpoint.

Read replicas can have multi-AZ enabled and you can create read replicas of multi-AZ source DBs.

Read replicas can be in another region (uses asynchronous replication).

This configuration can be used for centralizing data from across different regions for analytics.



DB SNAPSHOTS

DB Snapshots are user-initiated and enable you to back up your DB instance in a known state as frequently as you wish, and then restore to that specific state.

Cannot be used for point-in-time recovery.

Snapshots are stored on S3.

Snapshots remain on S3 until manually deleted.

Backups are taken within a defined window.

I/O is briefly suspended while backups initialize and may increase latency (applicable to single-AZ RDS).

DB snapshots that are performed manually will be stored even after the RDS instance is deleted.

Restored DBs will always be a new RDS instance with a new DNS endpoint.

Can restore up to the last 5 minutes.

Only default DB parameters and security groups are restored – you must manually associate all other DB parameters and SGs.

It is recommended to take a final snapshot before deleting an RDS instance.

Snapshots can be shared with other AWS accounts.

HIGH AVAILABILITY APPROACHES FOR DATABASES

If possible, choose DynamoDB over RDS because of inherent fault tolerance.

If DynamoDB can't be used, choose Aurora because of redundancy and automatic recovery features.

If Aurora can't be used, choose Multi-AZ RDS.

Frequent RDS snapshots can protect against data corruption or failure and they won't impact performance of Multi-AZ deployment.

Regional replication is also an option, but will not be strongly consistent.

If the database runs on EC2, you have to design the HA yourself.

MONITORING, LOGGING AND REPORTING

You can use the following automated monitoring tools to watch Amazon RDS and report when something is wrong:

- **Amazon RDS Events** – Subscribe to Amazon RDS events to be notified when changes occur with a DB instance, DB snapshot,

DB parameter group, or DB security group.

- **Database log files** – View, download, or watch database log files using the Amazon RDS console or Amazon RDS API operations. You can also query some database log files that are loaded into database tables.
- **Amazon RDS Enhanced Monitoring** — Look at metrics in real time for the operating system.
- **Amazon RDS Performance Insights** — Assess the load on your database, and determine when and where to take action.
- **Amazon RDS Recommendations** — Look at automated recommendations for database resources, such as DB instances, read replicas, and DB parameter groups.

In addition, Amazon RDS integrates with Amazon CloudWatch, Amazon EventBridge, and AWS CloudTrail for additional monitoring capabilities:

- **Amazon CloudWatch Metrics** – Amazon RDS automatically sends metrics to CloudWatch every minute for each active database. You don’t get additional charges for Amazon RDS metrics in CloudWatch.
- **Amazon CloudWatch Alarms** – You can watch a single Amazon RDS metric over a specific time period. You can then perform one or more actions based on the value of the metric relative to a threshold that you set.
- **Amazon CloudWatch Logs** – Most DB engines enable you to monitor, store, and access your database log files in CloudWatch Logs.
- **Amazon CloudWatch Events and Amazon EventBridge** – You can automate AWS services and respond to system events such as application availability issues or resource changes. Events from AWS services are delivered to CloudWatch Events and EventBridge nearly in real time. You can write simple rules to indicate which events interest you and what automated actions to take when an event matches a rule

- **AWS CloudTrail** – You can view a record of actions taken by a user, role, or an AWS service in Amazon RDS. CloudTrail captures all API calls for Amazon RDS as events. These captures include calls from the Amazon RDS console and from code calls to the Amazon RDS API operations. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Amazon RDS. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history** .

AUTHORIZATION AND ACCESS CONTROL

Amazon RDS supports **identity-based policies** .

RDS does not support resource-based policies.

The following AWS managed policies, which you can attach to users in your account, are specific to Amazon RDS:

- **AmazonRDSReadOnlyAccess** – Grants read-only access to all Amazon RDS resources for the AWS account specified.
- **AmazonRDSFullAccess** – Grants full access to all Amazon RDS resources for the AWS account specified.

You can authenticate to your DB instance using AWS Identity and Access Management (IAM) database authentication. IAM database authentication works with MySQL and PostgreSQL. With this authentication method, you don't need to use a password when you connect to a DB instance. Instead, you use an authentication token.

IAM database authentication provides the following benefits:

- Network traffic to and from the database is encrypted using Secure Sockets Layer (SSL).
- You can use IAM to centrally manage access to your database resources, instead of managing access individually on each DB instance.

- For applications running on Amazon EC2, you can use profile credentials specific to your EC2 instance to access your database instead of a password, for greater security.

AMAZON AURORA

Amazon Aurora is a relational database service that combines the speed and availability of high-end commercial databases with the simplicity and cost-effectiveness of open source databases.

Amazon Aurora is an AWS proprietary database.

Fully managed service that is provisioned through the **Amazon RDS** console.

High performance, low price.

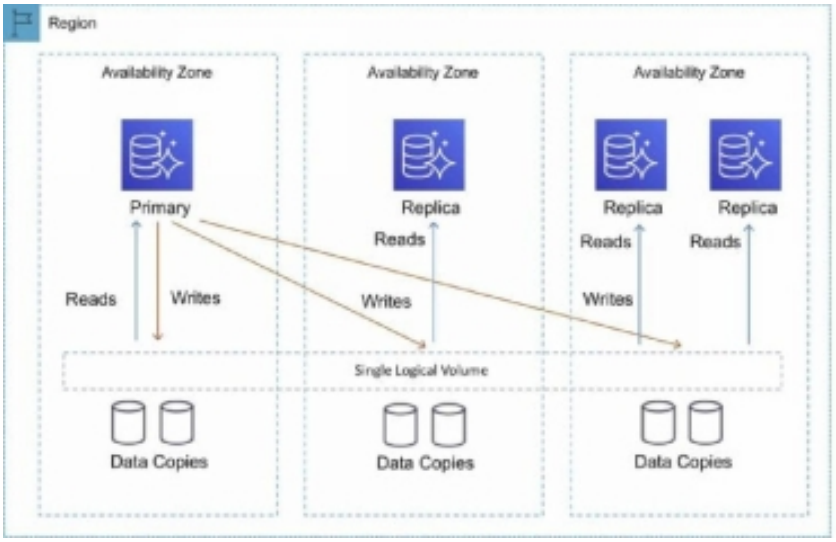
Scales in 10GB increments.

Scales up to 32vCPUs and 244GB RAM.

2 copies of data are kept in each AZ with a minimum of 3 AZ's (6 copies).

Can handle the loss of up to two copies of data without affecting DB write availability and up to three copies without affecting read availability.

The following digram depicts how Aurora Fault Tolerance and Replicas work:



Aurora Fault Tolerance

- Fault tolerance across 3 AZs
- Single logical volume
- Aurora Replicas scale-out read requests
- Up to 15 Aurora Replicas with sub-10ms replica lag
- Aurora Replicas are independent endpoints
- Can promote Aurora Replica to be a new primary or create new primary
- Set priority (tiers) on Aurora Replicas to control order of promotion
- Can use Auto Scaling to add replicas

AURORA REPLICAS

There are two types of replication: Aurora replica (up to 15), MySQL Read Replica (up to 5).

The table below describes the differences between the two replica options:

Feature	Aurora Replica	MySQL Replica
Number of replicas	Up to 15	Up to 5
Replication type	Asynchronous (milliseconds)	Asynchronous (seconds)
Performance impact on primary	Low	High
Replica location	In-region	Cross-region
Act as failover target	Yes (no data loss)	Yes (potentially minutes of data loss)
Automated failover	Yes	No
Support for user-defined replication delay	No	Yes
Support for different data or schema vs. primary	No	Yes
Support for different data or schema vs. primary	No	Yes

You can create read replicas for an AWS Aurora database in up to five AWS regions. This capability is available for AWS Aurora with MySQL compatibility.

CROSS-REGION READ REPLICAS

Cross-region read replicas allow you to improve your disaster recovery posture, scale read operations in regions closer to your application users, and easily migrate from one region to another.

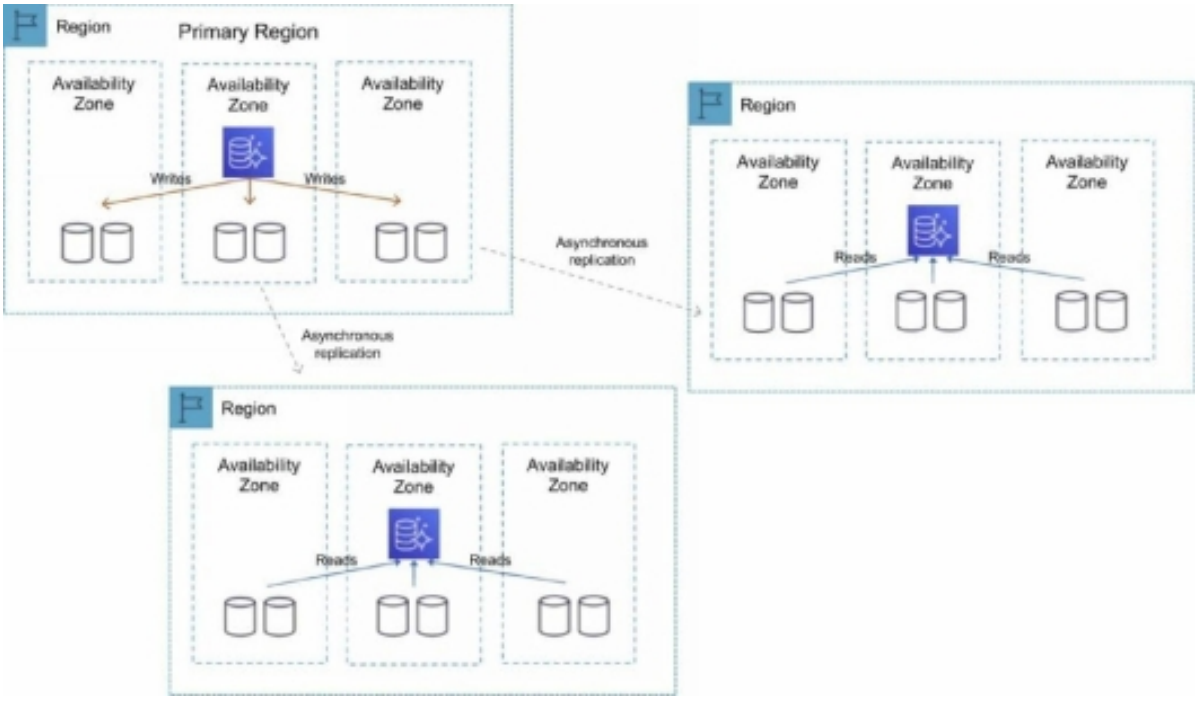
Cross-region replicas provide fast local reads to your users.

Each region can have an additional 15 Aurora replicas to further scale local reads.

You can choose between **Global Database** , which provides the best replication performance, and traditional binlog-based replication.

You can also set up your own binlog replication with external MySQL databases.

The following diagram depicts the Cross-Region Read Replica topology:



GLOBAL DATABASE

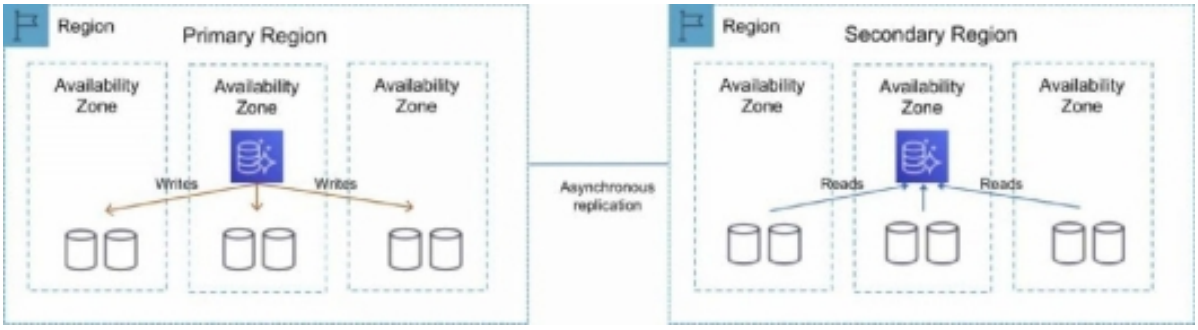
For globally distributed applications you can use **Global Database** , where a single Aurora database can span multiple AWS regions to enable fast local reads and quick disaster recovery.

Global Database uses storage-based replication to replicate a database across multiple AWS Regions, with typical latency of less than 1 second.

You can use a secondary region as a backup option in case you need to recover quickly from a regional degradation or outage.

A database in a secondary region can be promoted to full read/write capabilities in less than 1 minute.

The following table depicts the Aurora Global Database topology:



Watch this AWS Hands-On Labs tutorial to learn about Amazon Aurora databases and how to create a cross-region read replica with Aurora Global Database:

MULTI-MASTER

Amazon Aurora Multi-Master is a new feature of the Aurora MySQL-compatible edition that adds the ability to scale out write performance across multiple Availability Zones, allowing applications to direct read/write workloads to multiple instances in a database cluster and operate with higher availability.

Aurora Multi-Master is designed to achieve high availability and ACID transactions across a cluster of database nodes with configurable read after write consistency.

Architecture

- An Aurora cluster consists of a set of compute (database) nodes and a shared storage volume.
- The storage volume consists of six storage nodes placed in three Availability Zones for high availability and durability of user data.

- Every database node in the cluster is a writer node that can run read and write statements.

There is no single point of failure in the cluster.

Applications can use any writer node for their read/write and DDL needs.

A database change made by a writer node is written to six storage nodes in three Availability Zones, providing data durability and resiliency against storage node and Availability Zone failures.

The writer nodes are all functionally equal, and a failure of one writer node does not affect the availability of the other writer nodes in the cluster.

High Availability

Aurora Multi-Master improves upon the high availability of the single-master version of Amazon Aurora because all of the nodes in the cluster are read/write nodes.

With single-master Aurora, a failure of the single writer node requires the promotion of a read replica to be the new writer.

In the case of Aurora Multi-Master, the failure of a writer node merely requires the application using the writer to open connections to another writer.

AURORA SERVERLESS

Amazon Aurora Serverless is an on-demand, auto-scaling configuration for Amazon Aurora.

Available for MySQL-compatible and PostgreSQL-compatible editions.

The database automatically starts up, shuts down, and scales capacity up or down based on application needs.

It enables you to run a database in the cloud without managing any database instances. It’s a simple, cost-effective option for infrequent, intermittent, or unpredictable workloads.

You simply create a database endpoint and optionally specify the desired database capacity range and connect applications.

With Aurora Serverless, you only pay for database storage and the database capacity and I/O your database consumes while it is active.

Pay on a per-second basis for the database capacity you use when the database is active.

Can migrate between standard and serverless configurations with a few clicks in the Amazon RDS Management Console.

The table below provides a few example use cases for Amazon Aurora Serverless:

Use Case	Example
Infrequently-Used Applications	Application that is only used for a few minutes several times per day or week. Need a cost-effective database that only requires you to pay when it's active. With Aurora Serverless, you only pay for the database resources you consume.
New Applications	Deploying a new application and are unsure which instance size you need. With Aurora Serverless, you simply create an end-point and let the database auto-scale to the capacity requirements of your application.
Variable Workloads	Running a lightly-used application, with peaks of 30 minutes to several hours a few times each day or several times per year. Now you only pay for what the resources needed based on load – avoiding paying for unused resources or risking poor performance.
Unpredictable Workloads	Running workloads where there is database usage throughout the day, and also peaks of activity that are hard to predict. With Aurora Serverless, your database will auto-scale capacity to meet the needs of the application's peak load and scale back down when the surge of activity is over.
Development and Test Databases	Software development and QA teams are using databases during work hours, but don't need them on nights or weekends. With Aurora Serverless, your database automatically shuts down when not in use, and starts up much more quickly when work starts the next day.
Multitenant Applications	Web-based application with a database for each of your customers. Now you don't have to manage database capacity individually for each application in your fleet. Aurora manages individual database capacity for you, saving you valuable time.

FAULT-TOLERANT AND SELF-HEALING STORAGE

Each 10GB chunk of your database volume is replicated six ways, across three Availability Zones.

Amazon Aurora storage is fault-tolerant, transparently handling the loss of up to two copies of data without affecting database write availability and up to three copies without affecting read availability.

Amazon Aurora storage is also self-healing; data blocks and disks are continuously scanned for errors and replaced automatically.

AURORA AUTO SCALING

Aurora Auto Scaling dynamically adjusts the number of Aurora Replicas provisioned for an Aurora DB cluster using single-master replication.

Aurora Auto Scaling is available for both Aurora MySQL and Aurora PostgreSQL.

Aurora Auto Scaling enables your Aurora DB cluster to handle sudden increases in connectivity or workload.

When the connectivity or workload decreases, Aurora Auto Scaling removes unnecessary Aurora Replicas so that you don’t pay for unused provisioned DB instances.

AUTOMATIC, CONTINUOUS, INCREMENTAL BACKUPS AND POINT-IN-TIME RESTORE

Amazon Aurora’s backup capability enables point-in-time recovery for your instance.

This allows you to restore your database to any second during your retention period, up to the last five minutes.

Your automatic backup retention period can be configured up to thirty-five days.

Automated backups are stored in **Amazon S3** , which is designed for 99.999999999% durability. Amazon Aurora backups are automatic, incremental, and continuous and have no impact on database performance.

When automated backups are turned on for your DB Instance, Amazon RDS automatically performs a full daily snapshot of your data (during your preferred backup window) and captures transaction logs (as updates to your DB Instance are made).

Automated backups are enabled by default and data is stored on S3 and is equal to the size of the DB.

Amazon RDS retains backups of a DB Instance for a limited, user-specified period of time called the retention period, which by default is 7 days but can be up to 35 days.

There are two methods to backup and restore RDS DB instances:

- Amazon RDS automated backups.
- User initiated manual backups.

Both options back up the entire DB instance and not just the individual DBs.

Both options create a storage volume snapshot of the entire DB instance.

You can make copies of automated backups and manual snapshots.

Automated backups backup data to multiple AZs to provide for data durability.

Multi-AZ backups are taken from the standby instance (for MariaDB, MySQL, Oracle and PostgresSQL).

The DB instance must be in an Active state for automated backups to happen.

Only automated backups can be used for point-in-time DB instance recovery.

The granularity of point-in-time recovery is 5 minutes.

Amazon RDS creates a daily full storage volume snapshot and also captures transaction logs regularly.

You can choose the backup window.

There is no additional charge for backups but you will pay for storage costs on S3.

You can disable automated backups by setting the retention period to zero (0).

An outage occurs if you change the backup retention period from zero to a non-zero value or the other way around.

The retention period is the period AWS keeps the automated backups before deleting them.

Retention periods:

- By default the retention period is 7 days if configured from the console for all DB engines except Aurora.
- The default retention period is 1 day if configured from the API or CLI.
- The retention period for Aurora is 1 day regardless of how it is configured.
- You can increase the retention period up to 35 days.

During the backup window I/O may be suspended.

Automated backups are deleted when you delete the RDS DB instance.

Automated backups are only supported for InnoDB storage engine for MySQL (not for myISAM).

When you restore a DB instance the default DB parameters and security groups are applied – you must then apply the custom DB parameters and security groups.

You cannot restore from a DB snapshot into an existing DB instance.

Following a restore the new DB instance will have a new endpoint.

The storage type can be changed when restoring a snapshot.

AMAZON ELASTICACHE

Amazon ElastiCache is a fully managed implementations of two popular in-memory data stores – Redis and Memcached.

Amazon ElastiCache is a web service that makes it easy to deploy and run Memcached or Redis protocol-compliant server nodes in the cloud.

The in-memory caching provided by ElastiCache can be used to significantly improve latency and throughput for many read-heavy application workloads or compute-intensive workloads.

It can be put in front of databases such as RDS and DynamoDB – sits between the application and the database.

Good if your database is particularly read-heavy and the data does not change frequently.

Also good for compute-heavy workloads such as recommendation engines and it can be used to store the results of I/O intensive database queries of compute-intensive calculations.

Elasticache can be used for storing session state.

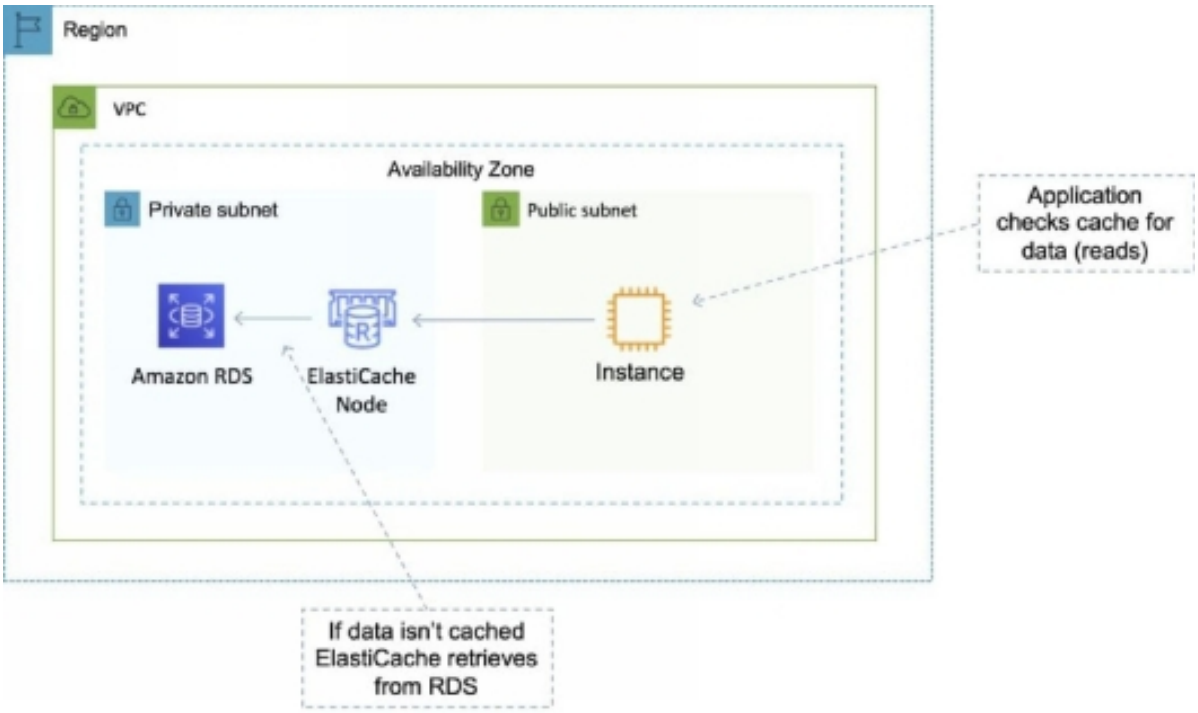
Push-button scalability for memory, writes and reads.

In-memory key/value store.

Billed by node size and hours of use.

Elasticache EC2 nodes cannot be accessed from the Internet, nor can they be accessed by EC2 instances in other VPCs.

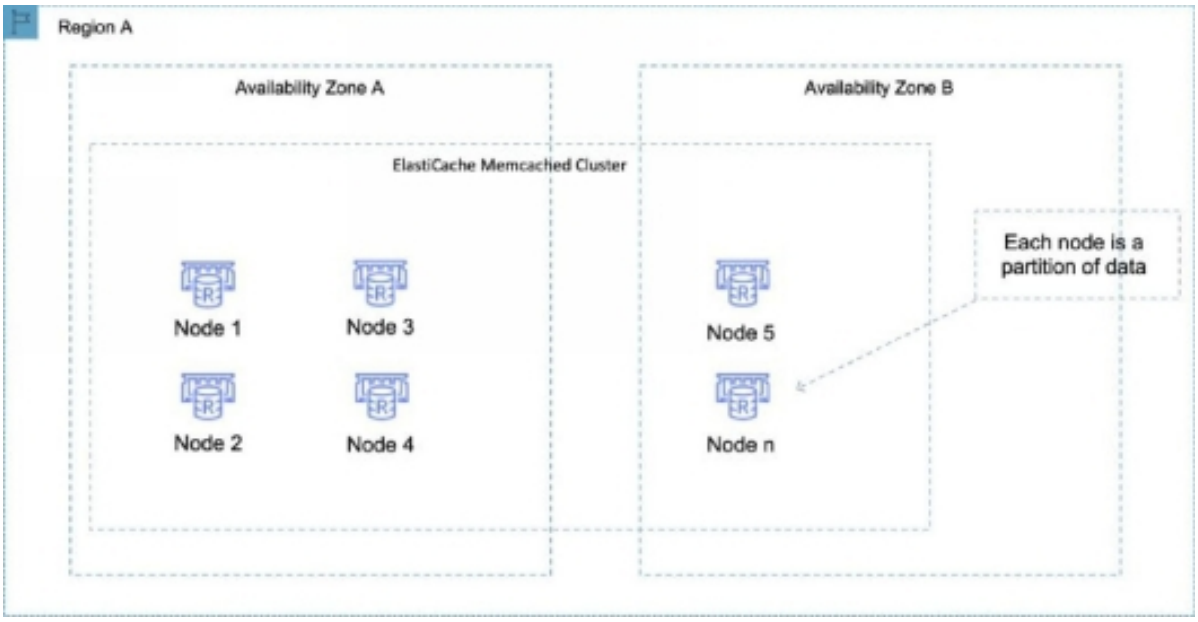
Exam tip: the key use cases for ElastiCache are offloading reads from a database, and storing the results of computations and session state. Also, remember that ElastiCache is an in-memory database and it's a managed service (so you can't run it on EC2).



There are two types of engine you can choose from: Memcached, Redis

MEMCACHED

- Simplest model and can run large nodes.
- It can be scaled in and out and cache objects such as DBs.
- Widely adopted memory object caching system.
- Multi-threaded.



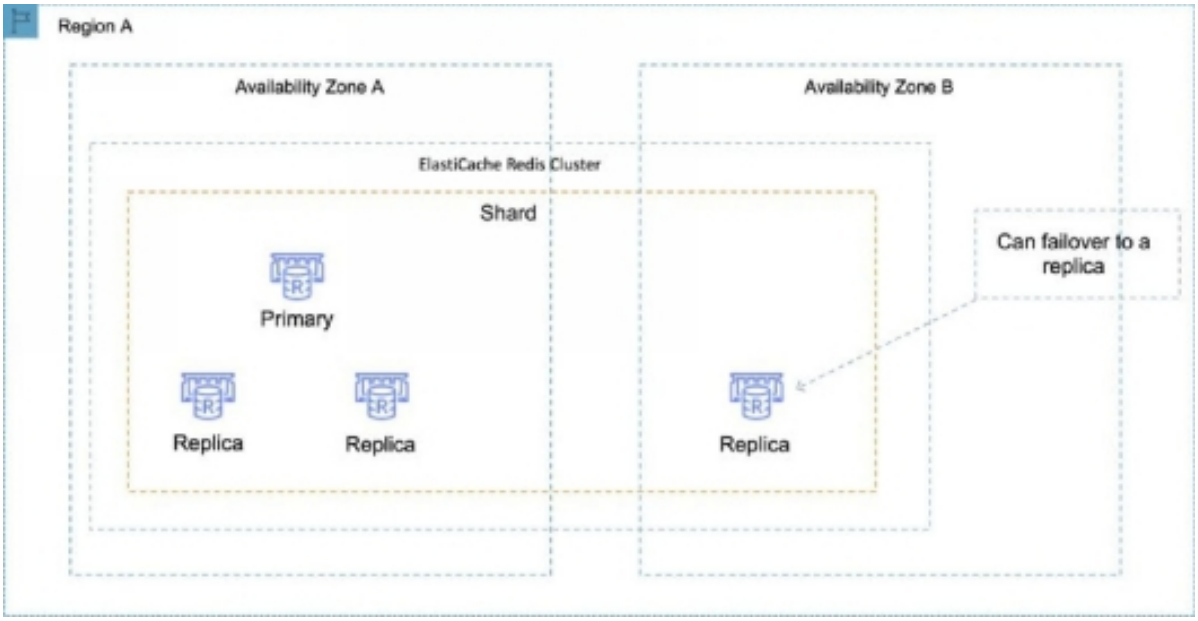
REDIS

- Open-source in-memory key-value store.
- Supports more complex data structures: sorted sets and lists.
- Supports master / slave replication and multi-AZ for cross-AZ redundancy.
- Supports automatic failover and backup/restore.

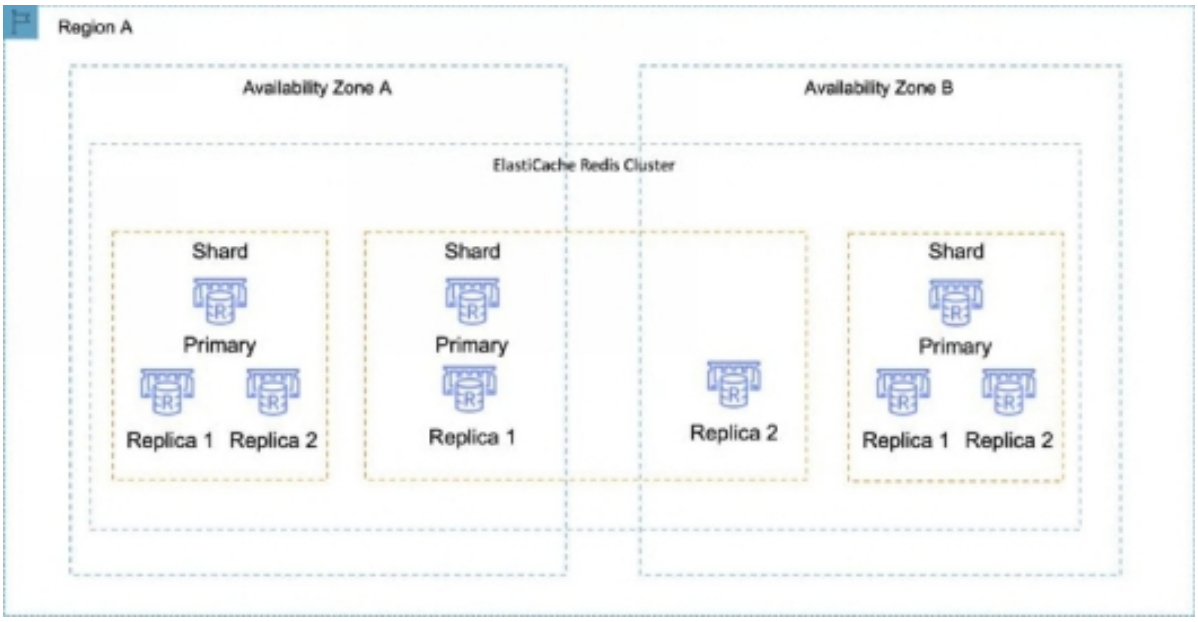
The following table provides a comparison of the different ElastiCache implementations:

Feature	Memcached	Redis (cluster mode disabled)	Redis (cluster mode enabled)
Data persistence	No	Yes	Yes
Data types	Simple	Complex	Complex
Data partitioning	Yes	No	Yes
Encryption	No	Yes	Yes
High availability (replication)	No	Yes	Yes
Multi-AZ	Yes, place nodes in multiple AZs. No failover or replication	Yes, with auto-failover. Uses read replicas (0-5 per shard)	Yes, with auto-failover. Uses read replicas (0-5 per shard)
Scaling	Up (node type); out (add nodes)	Single shard (can add replicas)	Add shards
Multithreaded	Yes	No	No
Backup and restore	No (and no snapshots)	Yes, automatic and manual snapshots	Yes, automatic and manual snapshots

The following diagram depicts Amazon ElastiCache Redis with Cluster Mode disabled:



The following diagram depicts Amazon ElastiCache Redis with Cluster Mode enabled:

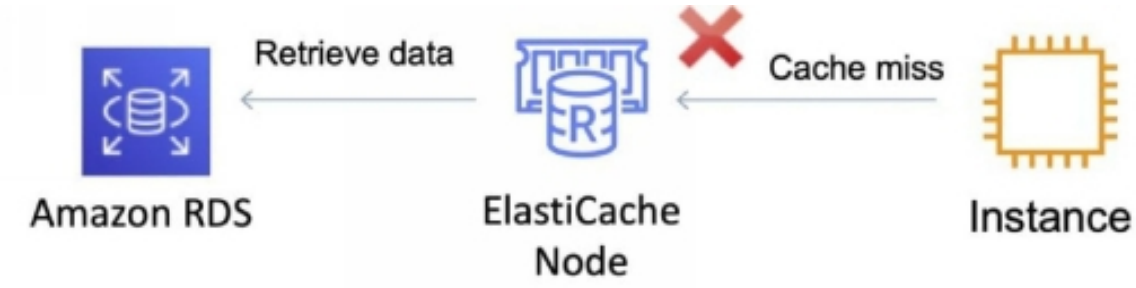


CACHING STRATEGIES

There are two caching strategies available: Lazy Loading and Write-Through:

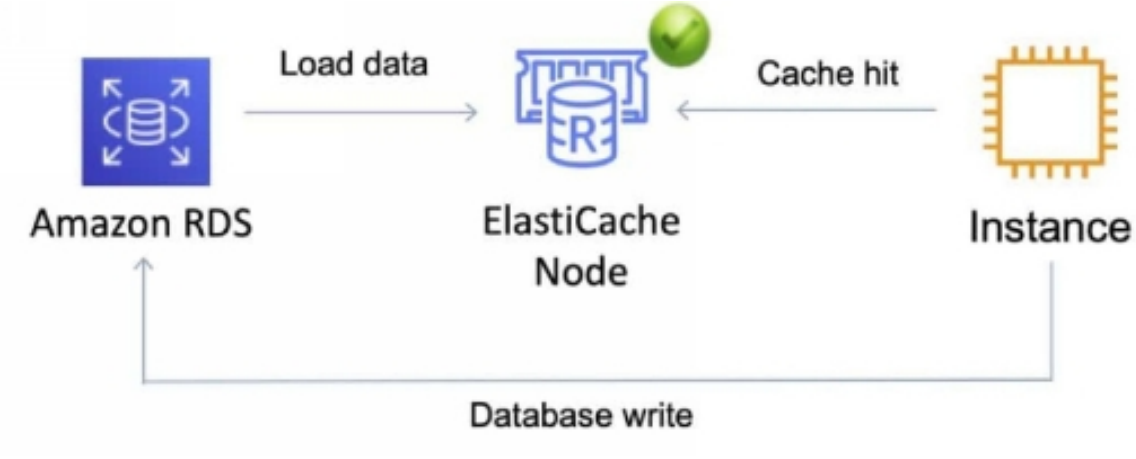
LAZY LOADING

- Loads the data into the cache only when necessary (if a cache miss occurs).
- Lazy loading avoids filling up the cache with data that won't be requested.
- If requested data is in the cache, ElastiCache returns the data to the application.
- If the data is not in the cache or has expired, ElastiCache returns a null.
- The application then fetches the data from the database and writes the data received into the cache so that it is available for next time.
- Data in the cache can become stale if Lazy Loading is implemented without other strategies (such as TTL).



WRITE THROUGH

- When using a write-through strategy, the cache is updated whenever a new write or update is made to the underlying database.
- Allows cache data to remain up-to-date.
- This can add wait time to write operations in your application.
- Without a TTL you can end up with a lot of cached data that is never read.



DEALING WITH STALE DATA – TIME TO LIVE (TTL)

- The drawbacks of lazy loading and write through techniques can be mitigated by a TTL.

- The TTL specifies the number of seconds until the key (data) expires to avoid keeping stale data in the cache.
- When reading an expired key, the application checks the value in the underlying database.
- Lazy Loading treats an expired key as a cache miss and causes the application to retrieve the data from the database and subsequently write the data into the cache with a new TTL.
- Depending on the frequency with which data changes this strategy may not eliminate stale data – but helps to avoid it.

Exam tip : Compared to DynamoDB Accelerator (DAX) remember that DAX is optimized for DymamoDB specifically and only supports the write-through caching strategy (does not use lazy loading).

MONITORING AND REPORTING

MEMCACHED METRICS

The following **CloudWatch** metrics offer good insight into ElastiCache Memcached performance:

CPUUtilization – This is a host-level metric reported as a percent.ecause Memcached is multi-threaded, this metric can be as high as 90%. If you exceed this threshold, scale your cache cluster up by using a larger cache node type, or scale out by adding more cache nodes.

SwapUsage – This is a host-level metric reported in bytes. This metric should not exceed 50 MB. If it does, we recommend that you increase the ConnectionOverhead parameter value.

Evictions – This is a cache engine metric. If you exceed your chosen threshold, scale your cluster up by using a larger node type, or scale out by adding more nodes.

CurrConnections – This is a cache engine metric. An increasing number of CurrConnections might indicate a problem with your application; you will need to investigate the application behavior to address this issue.

REDIS METRICS

The following CloudWatch metrics offer good insight into ElastiCache Redis performance:

EngineCPUUtilization – Provides CPU utilization of the Redis engine thread. Since Redis is single-threaded, you can use this metric to analyze the load of the Redis process itself.

MemoryFragmentationRatio – Indicates the efficiency in the allocation of memory of the Redis engine. Certain threshold will signify different behaviors. The recommended value is to have fragmentation above 1.0.

CacheHits – The number of successful read-only key lookups in the main dictionary.

CacheMisses – The number of unsuccessful read-only key lookups in the main dictionary.

CacheHitRate – Indicates the usage efficiency of the Redis instance. If the cache ratio is lower than ~0.8, it means that a significant amount of keys are evicted, expired or do not exist.

CurrConnections – The number of client connections, excluding connections from read replicas. ElastiCache uses two to four of the connections to monitor the cluster in each case.

LOGGING AND AUDITING

All Amazon ElastiCache actions are logged by AWS CloudTrail.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or IAM user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

AUTHORIZATION AND ACCESS CONTROL

Access to Amazon ElastiCache requires credentials that AWS can use to authenticate your requests. Those credentials must have permissions to access AWS resources, such as an ElastiCache cache cluster or an Amazon Elastic Compute Cloud (Amazon EC2) instance.

You can use **identity-based policies** with Amazon ElastiCache to provide the necessary access.

You can use **Redis Auth** to require a token with ElastiCache Redis.

The Redis authentication tokens enable Redis to require a token (password) before allowing clients to run commands, thereby improving data security.

AWS ORGANIZATIONS

AWS Organizations helps you centrally govern your environment as you grow and scale your workloads on AWS.

AWS Organizations helps you to centrally manage billing; control access, compliance, and security; and share resources across your AWS accounts.

Using AWS Organizations, you can automate account creation, create groups of accounts to reflect your business needs, and apply policies for these groups for governance.

You can also simplify billing by setting up a single payment method for all of your AWS accounts.

Through integrations with other AWS services, you can use Organizations to define central configurations and resource sharing across accounts in your organization.

AWS Organizations is available to all AWS customers at no additional charge.

The **AWS Organizations API** enables automation for account creation and management.

Available in two feature sets:

- Consolidated billing.
- All features.

By default, organizations support consolidated billing features.

Consolidated billing separates paying accounts and linked accounts.

You can use AWS Organizations to set up a single payment method for all the AWS accounts in your organization through consolidated billing.

With consolidated billing, you can see a combined view of charges incurred by all your accounts.

Can also take advantage of pricing benefits from aggregated usage, such as volume discounts for Amazon EC2 and Amazon S3.

Limit of 20 linked accounts for consolidated billing (default).

Policies can be assigned at different points in the hierarchy.

Can help with cost control through volume discounts.

Unused reserved EC2 instances are applied across the group.

Paying accounts should be used for billing purposes only.

Billing alerts can be setup at the paying account which shows billing for all linked accounts.

AWS ORGANIZATIONS CONCEPTS

Some of the core concepts you need to understand are listed here:

- **AWS Organization** – An organization is a collection of AWS accounts that you can organize into a hierarchy and manage centrally.
- **AWS Account** – An AWS account is a container for your AWS resources.
- **Master Account** – A master account is the AWS account you use to create your organization.
- **Member Account** – A member account is an AWS account, other than the master account, that is part of an organization.
- **Administrative Root** – An administrative root is the starting point for organizing your AWS accounts. The administrative root is the top-most container in your organization's hierarchy.
- **Organizational Unit (OU)** – An organizational unit (OU) is a group of AWS accounts within an organization. An OU can also contain other OUs enabling you to create a hierarchy.

- **Policy** – A policy is a “document” with one or more statements that define the controls that you want to apply to a group of AWS accounts.

SERVICE CONTROL POLICIES

Service control policies (SCPs) are a type of organization policy that you can use to manage permissions in your organization.

SCPs offer central control over the maximum available permissions for all accounts in your organization.

SCPs help you to ensure your accounts stay within your organization’s access control guidelines.

SCPs are available only in an organization that has all features enabled.

SCPs aren’t available if your organization has enabled only the consolidated billing features.

SCPs are similar to **AWS Identity and Access Management** (IAM) permission policies and use almost the same syntax.

However, an SCP never grants permissions. Instead, SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU).

You still need to attach identity-based or resource-based policies to principals or resources in your organization’s accounts to actually grant permissions to them.

The following example SCP restricts any instance launches that do not use the t2.micro instance type:

```
 .{
"Version": "2012-10-17",
"Statement": [
{
"Sid": "RequireMicroInstanceType",
"Effect": "Deny",
```

```
"Action": "ec2:RunInstances",
"Resource": "arn:aws:ec2:*:*:instance/*",
"Condition": {
"StringNotEquals":{
"ec2:InstanceType":"t2.micro"
}
}
}
]
}
```

More example SCPs can be found **here** .

RESOURCE GROUPS

You can use resource groups to organize your AWS resources.

In AWS, a resource is an entity that you can work with.

Resource groups make it easier to manage and automate tasks on large numbers of resources at one time.

Resource groups allow you to group resources and then tag them.

The Tag Editor assists with finding resources and adding tags.

You can access Resource Groups through any of the following entry points:

- On the navigation bar of the AWS Management Console.
- In the **AWS Systems Manager** console, from the left navigation pane entry for Resource Groups.
- By using the Resource Groups API, in AWS CLI commands or AWS SDK programming languages.

A resource group is a collection of AWS resources that are all in the same AWS region, and that match criteria provided in a query.

In Resource Groups, there are two types of queries on which you can build a group.

Both query types include resources that are specified in the format AWS::service::resource.

- **Tag-based** – Tag-based queries include lists of resources and tags. Tags are keys that help identify and sort your resources within your organization. Optionally, tags include values for keys.
- **AWS CloudFormation stack-based** – In an **AWS CloudFormation** stack-based query, you choose an AWS CloudFormation stack in your account in the current region, and then choose resource types within the stack that you want to be in the group. You can base your query on only one AWS CloudFormation stack.

Resource groups can be nested; a resource group can contain existing resource groups in the same region.

AMAZON CLOUDWATCH

Amazon CloudWatch is a monitoring service for AWS cloud resources and the applications you run on AWS.

It is used to collect and track metrics, collect and monitor log files, and set alarms.

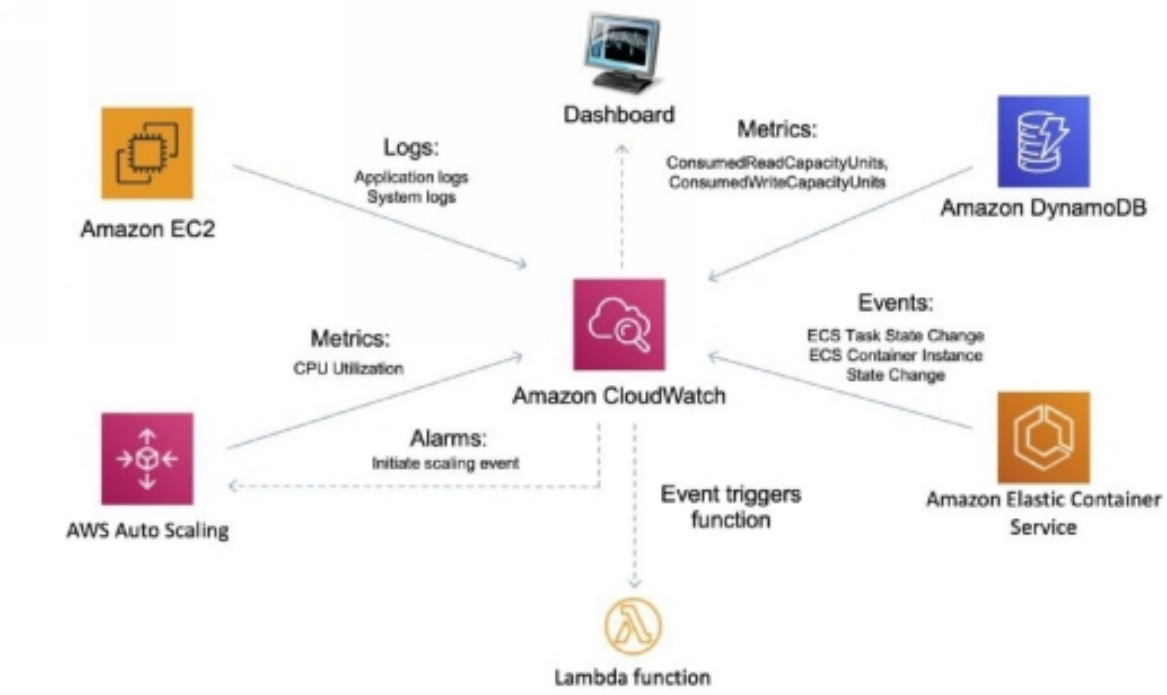
With CloudWatch you can:

- Gain system-wide visibility into resource utilization.
- Monitor application performance.
- Monitor operational health.

CloudWatch alarms monitor metrics and automatically initiate actions.

CloudWatch Logs centralizes logs from systems, applications, and AWS services.

CloudWatch Events delivers a stream of system events that describe changes in AWS resources.



CloudWatch is accessed via API, command-line interface, AWS SDKs, and the AWS Management Console.

CloudWatch integrates with IAM.

Can automatically react to changes in your AWS resources.

CloudWatch vs **CloudTrail** :

CloudWatch	CloudTrail
Performance monitoring	Auditing
Log events across AWS services – think operations	Log API activity across AWS services – think activities
Higher-level comprehensive monitoring and eventing	More low-level granular
Log from multiple accounts	Log from multiple accounts
Logs stored indefinitely	Logs stored to S3 or CloudWatch indefinitely
Alarms history for 14 days	No native alarming; can use CloudWatch alarms

Used to collect and track metrics, collect and monitor log files, and set alarms.

Automatically react to changes in your AWS resources.

With CloudWatch you can monitor resources such as:

- **EC2 instances** .
- DynamoDB tables.
- **RDS DB instances** .
- Custom metrics generated by applications and services.
- Any log files generated by your applications.

Gain system-wide visibility into resource utilization.

Monitor application performance.

Monitor operational health.

CloudWatch is accessed via API, command-line interface, AWS SDKs, and the AWS Management Console.

CloudWatch integrates with IAM.

METRICS

Metrics are the fundamental concept in CloudWatch.

A metric represents a time-ordered set of data points that are published to CloudWatch.

AWS services send metrics to CloudWatch.

You can also send your own custom metrics to CloudWatch.

Metrics exist within a region.

Metrics cannot be deleted but automatically expire after 15 months.

Metrics are uniquely defined by a name, a namespace, and zero or more dimensions.

CloudWatch retains metric data as follows:

- Data points with a period of less than 60 seconds are available for 3 hours. These data points are high-resolution custom metrics.
- Data points with a period of 60 seconds (1 minute) are available for 15 days.
- Data points with a period of 300 seconds (5 minute) are available for 63 days.
- Data points with a period of 3600 seconds (1 hour) are available for 455 days (15 months).

CUSTOM METRICS

You can publish your own metrics to CloudWatch using the AWS CLI or an API.

You can view statistical graphs of your published metrics with the AWS Management Console.

CloudWatch stores data about a metric as a series of data points.

Each data point has an associated time stamp.

You can even publish an aggregated set of data points called a statistic set.

HIGH-RESOLUTION METRICS

Each metric is one of the following:

- Standard resolution, with data having a one-minute granularity
- High resolution, with data at a granularity of one second

Metrics produced by AWS services are standard resolution by default.

When you publish a custom metric, you can define it as either standard resolution or high resolution.

When you publish a high-resolution metric, CloudWatch stores it with a resolution of 1 second, and you can read and retrieve it with a period of 1 second, 5 seconds, 10 seconds, 30 seconds, or any multiple of 60 seconds.

High-resolution metrics can give you more immediate insight into your application’s sub-minute activity.

Keep in mind that every PutMetricData call for a custom metric is charged, so calling PutMetricData more often on a high-resolution metric can lead to higher charges.

If you set an alarm on a high-resolution metric, you can specify a high-resolution alarm with a period of 10 seconds or 30 seconds, or you can set a regular alarm with a period of any multiple of 60 seconds.

There is a higher charge for high-resolution alarms with a period of 10 or 30 seconds.

NAMESPACE

A namespace is a container for CloudWatch metrics.

Metrics in different namespaces are isolated from each other, so that metrics from different applications are not mistakenly aggregated into the same statistics.

The following table provides some examples of namespaces for several AWS services:

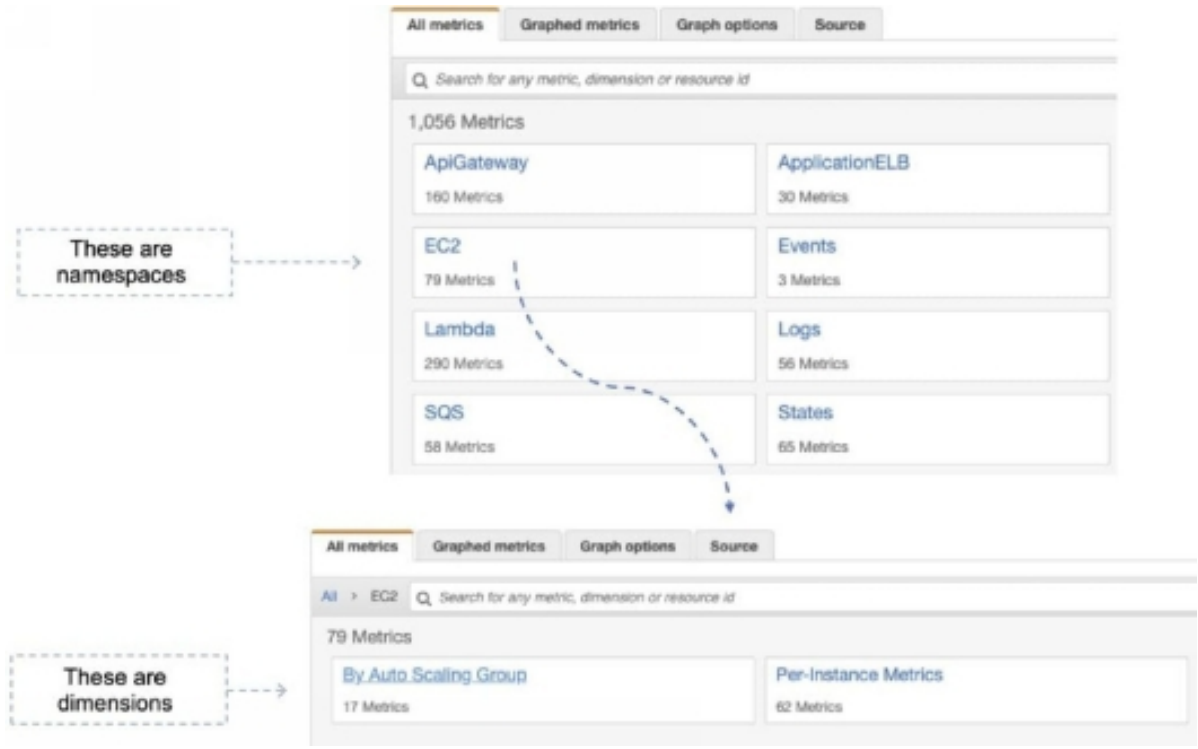
Service	Namespace
Amazon API Gateway	AWS/ApiGateway
Amazon CloudFront	**AWS/CloudFront**
AWS CloudHSM	AWS/CloudHSM
Amazon CloudWatch Logs	**AWS/Logs**
AWS CodeBuild	AWS/CodeBuild
Amazon Cognito	**AWS/Cognito**
Amazon DynamoDB	AWS/DynamoDB
Amazon EC2	**AWS/EC2**
AWS Elastic Beanstalk	AWS/ElasticBeanstalk

DIMENSIONS

In custom metrics, the –dimensions parameter is common.

A dimension further clarifies what the metric is and what data it stores.

You can have up to 10 dimensions in one metric, and each dimension is defined by a name and value pair.



How you specify a dimension is different when you use different commands.

With **put-metric-data** , you specify each dimension as MyName=MyValue, and with **get-metric-statistics** or **put-metric-alarm** you use the format Name=MyName, Value=MyValue.

For example, the following command publishes a Buffers metric with two dimensions named InstanceId and InstanceType.

```
aws cloudwatch put - metric - data -- metric - name Buffers -- namespace MyNameSpace -- unit Bytes -- value 231434333 - - dimensions InstanceId = 1 - 23456789 , InstanceType = m1 . small
```

This command retrieves statistics for that same metric. Separate the Name and Value parts of a single dimension with commas, but if you have multiple dimensions, use a space between one dimension and the next.

```
aws cloudwatch get - metric - statistics -- metric - name Buffers -- namespace MyNameSpace -- dimensions Name =
```

```
InstanceId , Value = 1 - 23456789 Name = InstanceType ,
Value = m1 . small -- start - time 2016 - 10 - 15T04 : 00 : 00Z
-- end - time 2016 - 10 - 19T07 : 00 : 00Z -- statistics Average
-- period 60
```

If a single metric includes multiple dimensions, you must specify a value for every defined dimension when you use **get-metric-statistics** . For example, the Amazon S3 metric BucketSizeBytes includes the dimensions BucketName and StorageType, so you must specify both dimensions with **get-metric-statistics** .

```
aws cloudwatch get - metric - statistics -- metric - name
BucketSizeBytes -- start - time 2017 - 01 - 23T14 : 23 : 00Z --
end - time 2017 - 01 - 26T19 : 30 : 00Z -- period 3600 --
namespace AWS / S3 -- statistics Maximum -- dimensions
Name = BucketName , Value = MyBucketName Name =
StorageType , Value = StandardStorage -- output table
```

PUBLISHING SINGLE DATA POINTS

To publish a single data point for a new or existing metric, use the **put-metric-data** command with one value and time stamp.

For example, the following actions each publish one data point.

```
aws cloudwatch put - metric - data -- metric - name
PageViewCount -- namespace MyService -- value 2 --
timestamp 2016 - 10 - 20T12 : 00 : 00.000Z
aws cloudwatch put - metric - data -- metric - name
PageViewCount -- namespace MyService -- value 4 --
timestamp 2016 - 10 - 20T12 : 00 : 01.000Z
aws cloudwatch put - metric - data -- metric - name
PageViewCount -- namespace MyService -- value 5 --
timestamp 2016 - 10 - 20T12 : 00 : 02.000Z
```

STATISTICS

Statistics are metric data aggregations over specified periods of time.

CloudWatch provides statistics based on the metric data points provided by your custom data or provided by other AWS services to CloudWatch.

Statistic	Description
Minimum	The lowest value observed during the specified period. You can use this value to determine low volumes of activity for your application.
Maximum	The highest value observed during the specified period. You can use this value to determine high volumes of activity for your application.
Sum	All values submitted for the matching metric added together. This statistic can be useful for determining the total volume of a metric.
Average	The value of Sum / SampleCount during the specified period. By comparing this statistic with the Minimum and Maximum, you can determine the full scope of a metric and how close the average use is to the Minimum and Maximum. This comparison helps you to know when to increase or decrease your resources as needed.
SampleCount	The count (number) of data points used for the statistical calculation.
pNN.NN	The value of the specified percentile. You can specify any percentile, using up to two decimal places (for example, p95.45). Percentile statistics are not available for metrics that include any negative values. For more information, see Percentiles.

CLOUDWATCH ALARMS

You can use an alarm to automatically initiate actions on your behalf.

An alarm watches a single metric over a specified time period, and performs one or more specified actions, based on the value of the metric relative to a threshold over time.

The action is a notification sent to an Amazon SNS topic or an Auto Scaling policy.

You can also add alarms to dashboards.

Alarms invoke actions for sustained state changes only.

CloudWatch alarms do not invoke actions simply because they are in a particular state.

The state must have changed and been maintained for a specified number of periods.

CLOUDWATCH LOGS

Amazon CloudWatch Logs lets you monitor and troubleshoot your systems and applications using your existing system, application and custom log files.

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

Features:

- **Monitor logs from Amazon EC2 instances** – monitors application and system logs and can trigger notifications.
- **Monitor CloudTrail Logged Events** – alarms can be created in CloudWatch based on API activity captured by CloudTrail.
- **Log retention** – by default, logs are retained indefinitely. Configurable per log group from 1 day to 10 years.

CloudWatch Logs can be used for real time application and system monitoring as well as long term log retention.

CloudWatch Logs keeps logs indefinitely by default.

CloudTrail logs can be sent to CloudWatch Logs for real-time monitoring.

CloudWatch Logs metric filters can evaluate CloudTrail logs for specific terms, phrases or values.

CLOUDWATCH LOGS AGENT

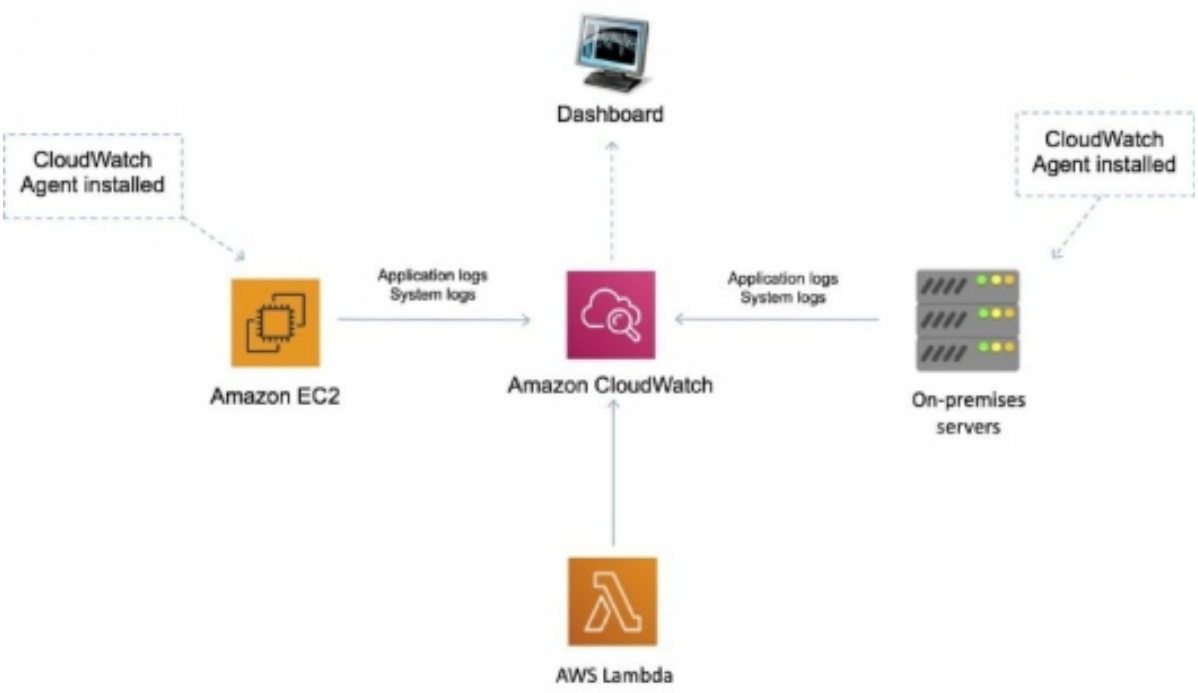
The CloudWatch Logs agent provides an automated way to send log data to CloudWatch Logs from Amazon EC2 instances.

There is now a unified CloudWatch agent that collects both logs and metrics.

The unified CloudWatch agent includes metrics such as memory and disk utilization.

The unified CloudWatch agent enables you to do the following:

- Collect more system-level metrics from Amazon EC2 instances across operating systems. The metrics can include in-guest metrics, in addition to the metrics for EC2 instances.
- Collect system-level metrics from on-premises servers. These can include servers in a hybrid environment as well as servers not managed by AWS.
- Retrieve custom metrics from your applications or services using the StatsD and collectd protocols.



CLOUDWATCH EVENTS

Amazon CloudWatch Events delivers a near real-time stream of system events that describe changes in AWS resources.

Can use CloudWatch Events to schedule automated actions that self-trigger at certain times using cron or rate expressions

Can match events and route them to one or more target functions or streams.

Targets include:

- Amazon EC2 instances.
- AWS Lambda functions.
- Streams in Amazon Kinesis Data Streams.
- Delivery streams in Amazon Kinesis Data Firehose.
- Log groups in Amazon CloudWatch Logs.
- Amazon ECS tasks.
- Systems Manager Run Command.
- Systems Manager Automation.
- AWS Batch jobs.
- Step Functions state machines.
- Pipelines in CodePipeline.
- CodeBuild projects.
- Amazon Inspector assessment templates.
- Amazon SNS topics.
- Amazon SQS queues.

In the following example, an EC2 instance changes state (terminated) and the event is sent to CloudWatch Events which forwards the event to the target (SQS queue).

USEFUL API ACTIONS

It is useful to understand the following API actions for the Developer Associate exam. You should check these out and other API actions on the AWS website as well prior to your exam.

GetMetricData

- Retrieve as many as 500 different metrics in a single request.

PutMetricData

- Publishes metric data points to Amazon CloudWatch.
- CloudWatch associates the data points with the specified metric.
- If the specified metric does not exist, CloudWatch creates the metric.

GetMetricStatistics

- Gets statistics for the specified metric.
- CloudWatch aggregates data points based on the length of the period that you specify.
- Maximum number of data points returned from a single call is 1,440.

PutMetricAlarm

- Creates or updates an alarm and associates it with the specified metric, metric math expression, or anomaly detection model.
- Alarms based on anomaly detection models cannot have Auto Scaling actions.

AWS CLOUDTRAIL

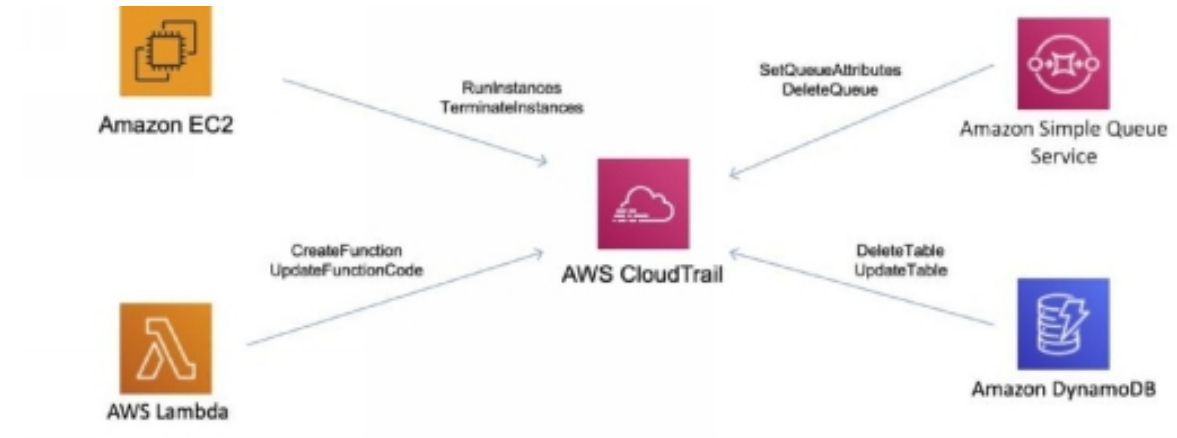
AWS CloudTrail is a web service that records activity made on your account.

A CloudTrail trail can be created which delivers log files to an Amazon S3 bucket.

CloudTrail is about logging and saves a history of API calls for your AWS account.

It enables governance, compliance, and operational and risk auditing of your AWS account.

Events include actions taken in the AWS Management Console, AWS Command Line Interface, and AWS SDKs and APIs.



CloudTrail provides visibility into user activity by recording actions taken on your account.

API history enables security analysis, resource change tracking, and compliance auditing.

Logs API calls made via:

- AWS Management Console.

- AWS SDKs.
- Command line tools.
- Higher-level AWS services (such as CloudFormation).

CloudTrail records account activity and service events from most AWS services and logs the following records:

- The identity of the API caller.
- The time of the API call.
- The source IP address of the API caller.
- The request parameters.
- The response elements returned by the AWS service.

CloudTrail is enabled on your AWS account when you create it.

CloudTrail is per AWS account.

You can create two types of trails for an AWS account:

- A trail that applies to all regions – records events in all regions and delivers to an S3 bucket.
- A trail that applies to a single region – records events in a single region and delivers to an S3 bucket. Additional single trails can use the same or a different S3 bucket.

Trails can be configured to log data events and management events:

- **Data events** : These events provide insight into the resource operations performed on or within a resource. These are also known as data plane operations.
- **Management events** : Management events provide insight into management operations that are performed on resources in your AWS account. These are also known as control plane operations. Management events can also include non-API events that occur in your account.

Example data events include:

- Amazon S3 object-level API activity (for example, GetObject, DeleteObject, and PutObject API operations).
- AWS Lambda function execution activity (the Invoke API).

Example management events include:

- Configuring security (for example, IAM AttachRolePolicy API operations).
- Registering devices (for example, Amazon EC2 CreateDefaultVpc API operations).
- Configuring rules for routing data (for example, Amazon EC2 CreateSubnet API operations).
- Setting up logging (for example, AWS CloudTrail CreateTrail API operations).
- CloudTrail log files are encrypted using S3 Server Side Encryption (SSE).

You can also enable encryption using SSE KMS for additional security.

A single KMS key can be used to encrypt log files for trails applied to all regions.

You can consolidate logs from multiple accounts using an S3 bucket:

1. Turn on CloudTrail in the paying account.
2. Create a bucket policy that allows cross-account access.
3. Turn on CloudTrail in the other accounts and use the bucket in the paying account.

You can integrate CloudTrail with CloudWatch Logs to deliver data events captured by CloudTrail to a CloudWatch Logs log stream.

CloudTrail log file integrity validation feature allows you to determine whether a CloudTrail log file was unchanged, deleted, or modified since

CloudTrail delivered it to the specified Amazon S3 bucket.

CloudWatch vs CloudTrail:

CloudWatch	CloudTrail
Performance monitoring	Auditing
Log events across AWS services – think operations	Log API activity across AWS services – think activities
Higher-level comprehensive monitoring and eventing	More low-level granular
Log from multiple accounts	Log from multiple accounts
Logs stored indefinitely	Logs stored to S3 or CloudWatch indefinitely
Alarms history for 14 days	No native alarming; can use CloudWatch alarms

REFERENCES

https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-getting-started.html

https://aws.amazon.com/cloudtrail/faqs/

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

https://aws.amazon.com/cloudtrail/pricing/

AWS CONFIG

AWS Config is a fully managed service that provides you with an AWS resource inventory, configuration history, and configuration change notifications to enable security and governance.

With AWS Config you can discover existing AWS resources, export a complete inventory of your AWS resources with all configuration details, and determine how a resource was configured at any point in time.

These capabilities enable compliance auditing, security analysis, resource change tracking, and troubleshooting.

Allow you to assess, audit and evaluate configurations of your AWS resources.

Very useful for Configuration Management as part of an ITIL program.

Creates a baseline of various configuration settings and files and can then track variations against that baseline.

AWS CONFIG VS CLOUDTRAIL

AWS CloudTrail records user API activity on your account and allows you to access information about this activity.

AWS Config records point-in-time configuration details for your AWS resources as Configuration Items (CIs).

You can use an AWS Config CI to answer “What did my AWS resource look like?” at a point in time.

You can use AWS CloudTrail to answer “Who made an API call to modify this resource?”.

CONFIG RULES

A Config Rule represents desired configurations for a resource and is evaluated against configuration changes on the relevant resources, as recorded by AWS Config.

AWS Config Rules can check resources for certain desired conditions and if violations are found the resources are flagged as “noncompliant”.

Examples of Config Rules:

- Is backup enabled on **Amazon RDS** ?
- Is CloudTrail enabled on the AWS account?
- Are **Amazon EBS** volumes encrypted.

CONFIGURATION ITEMS

A **Configuration Item (CI)** is the configuration of a resource at a given point-in-time. A CI consists of 5 sections:

1. Basic information about the resource that is common across different resource types (e.g., Amazon Resource Names, tags).
2. Configuration data specific to the resource (e.g., **Amazon EC2** instance type).
3. Map of relationships with other resources (e.g., EC2::Volume vol-3434df43 is “attached to instance” EC2 Instance i-3432ee3a).
4. AWS CloudTrail event IDs that are related to this state.
5. Metadata that helps you identify information about the CI, such as the version of this CI, and when this CI was captured.

CHARGES

With AWS Config, you are charged based on the number configuration items (CIs) recorded for supported resources in your AWS account.

AWS Config creates a configuration item whenever it detects a change to a resource type that it is recording.

AWS IAM – IDENTITY AND ACCESS MANAGEMENT

AWS IAM is used to securely control individual and group access to AWS resources.

IAM makes it easy to provide multiple users secure access to AWS resources.

IAM can be used to manage:

- Users.
- Groups.
- Access policies.
- Roles.
- User credentials.
- User password policies.
- Multi-factor authentication (MFA).
- API keys for programmatic access (CLI).

Provides centralized control of your AWS account.

Enables shared access to your AWS account.

By default new users are created with NO access to any AWS services – they can only login to the AWS console.

Permission must be explicitly granted to allow a user to access an AWS service.

IAM users are individuals who have been granted access to an AWS account.

Each IAM user has three main components:

- A user-name.
- A password.
- Permissions to access various resources.

You can apply granular permissions with IAM.

You can assign users individual security credentials such as access keys, passwords, and multi-factor authentication devices.

IAM is not used for application-level authentication.

Identity Federation (including AD, Facebook etc.) can be configured allowing secure access to resources in an AWS account without creating an IAM user account.

Multi-factor authentication (MFA) can be enabled/enforced for the AWS account and for individual users under the account.

MFA uses an authentication device that continually generates random, six-digit, single-use authentication codes.

You can authenticate using an MFA device in the following three ways:

- Through the **AWS Management Console** – the user is prompted for a user name, password and authentication code.
- Using the **AWS API** – restrictions are added to IAM policies and developers can request temporary security credentials and pass MFA parameters in their AWS STS API requests.
- Using the **AWS CLI** by obtaining temporary security credentials from STS (aws sts get-session-token).

It is a best practice to use MFA for all users and to use U2F or hardware MFA devices for all privileged users.

IAM is universal (global) and does not apply to regions.

The “root account” is the account created when you setup the AWS account. It has complete Admin access and is the only account that has this

access by default.

It is a best practice to not use the root account for anything other than billing.

AWS recommend that you use the AWS SDKs to make programmatic API calls to IAM.

However, you can also use the IAM Query API to make direct calls to the IAM web service.

AUTHENTICATION METHODS

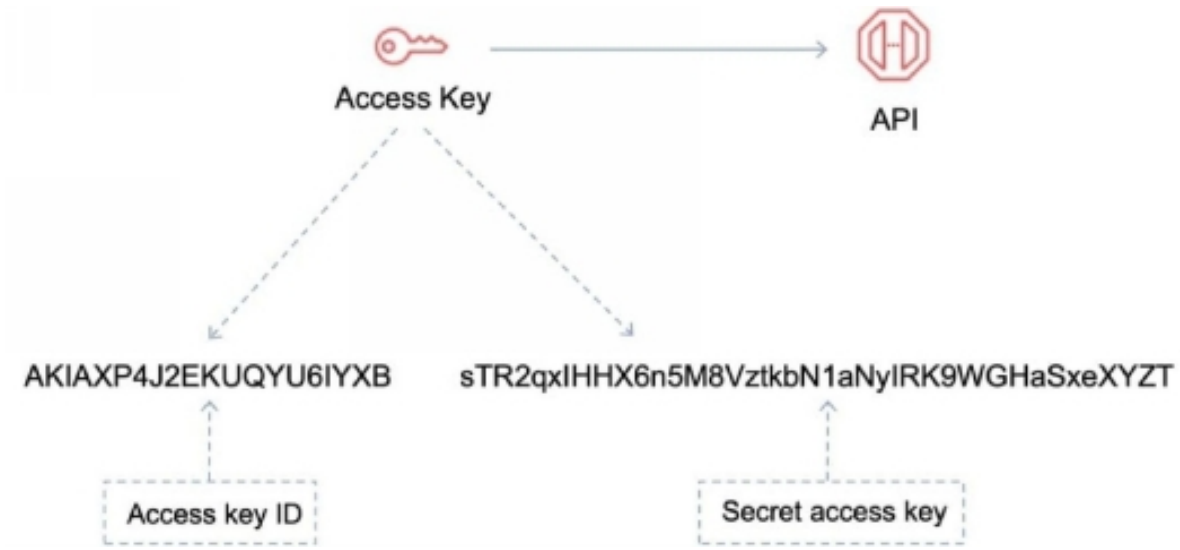
Console password:

- A password that the user can enter to sign into interactive sessions such as the AWS Management Console.
- You can allow users to change their own passwords.
- You can allow selected IAM users to change their passwords by disabling the option for all users and using an IAM policy to grant permissions for the selected users.

Access Keys:

- A combination of an **access key ID** and a **secret access key** .
- You can assign two active access keys to a user at a time.
- These can be used to make programmatic calls to AWS when using the **API** in program code or at a command prompt when using the **AWS CLI** or the **AWS PowerShell** tools.
- You can create, modify, view or rotate access keys.
- When created IAM returns the access key ID and secret access key.
- The secret access is returned only at creation time and if lost a new key must be created.

- Ensure access keys and secret access keys are stored securely.
- Users can be given access to change their own keys through IAM policy (not from the console).
- You can disable a user's access key which prevents it from being used for API calls.
- Access keys are stored in:
  - Linux: ~/.aws/credentials
  - Windows: %UserProfle%\.aws\credentials



Server certificates:

- SSL/TLS certificates that you can use to authenticate with some AWS services.
- AWS recommends that you use the AWS Certificate Manager (ACM) to provision, manage and deploy your server certificates.
- Use IAM only when you must support HTTPS connections in a region that is not supported by ACM.

The following diagram shows the different methods of authentication available with IAM:

IAM USERS

An IAM user is an entity that represents a person or service.

Can be assigned:

- An access key ID and secret access key for programmatic access to the AWS API, CLI, SDK, and other development tools.
- A password for access to the management console.

By default users cannot access anything in your account.

The account root user credentials are the email address used to create the account and a password.

The root account has full administrative permissions and these cannot be restricted.

Best practice for root accounts:

- Don’t use the root user credentials.
- Don’t share the root user credentials.
- Create an IAM user and assign administrative permissions as required.

- Enable MFA.

IAM users can be created to represent applications and these are known as “service accounts”.

You can have up to 5000 users per AWS account.

Each user account has a friendly name and an ARN which uniquely identifies the user across AWS.

A unique ID is also created which is returned only when you create the user using the API, Tools for Windows PowerShell or the AWS CLI.

You should create individual IAM accounts for users (best practice not to share accounts).

The Access Key ID and Secret Access Key are not the same as a password and cannot be used to login to the AWS console.

The Access Key ID and Secret Access Key can only be generated once and must be regenerated if lost.

A password policy can be defined for enforcing password length, complexity etc. (applies to all users).

You can allow or disallow the ability to change passwords using an IAM policy.

Access keys and passwords should be changed regularly.

GROUPS

Groups are collections of users and have policies attached to them.

A group is not an identity and cannot be identified as a principal in an IAM policy.

Use groups to assign permissions to users.

Use the principal of least privilege when assigning permissions.

You cannot nest groups (groups within groups).

ROLES

Roles are created and then “assumed” by trusted entities and define a set of permissions for making AWS service requests.

With IAM Roles you can delegate permissions to resources for users and services without using permanent credentials (e.g. user name and password).

IAM users or AWS services can assume a role to obtain temporary security credentials that can be used to make AWS API calls.

You can delegate using roles.

There are no credentials associated with a role (password or access keys).

IAM users can temporarily assume a role to take on permissions for a specific task.

A role can be assigned to a federated user who signs in using an external identity provider.

Temporary credentials are primarily used with IAM roles and automatically expire.

Roles can be assumed temporarily through the console or programmatically with the **AWS CLI** , **Tools for Windows PowerShell** or **API** .

IAM roles with **EC2** instances:

- IAM roles can be used for granting applications running on EC2 instances permissions to AWS API requests using instance profiles.
- Only one role can be assigned to an EC2 instance at a time.
- A role can be assigned at the **EC2 instance creation time or at any time afterwards** .
- When using the AWS CLI or API instance profiles must be created manually (it’s automatic and transparent through the console).

- Applications retrieve temporary security credentials from the instance metadata.

Role Delegation:

- Create an IAM role with two policies:
  - Permissions policy – grants the user of the role the required permissions on a resource.
  - Trust policy – specifies the trusted accounts that are allowed to assume the role.
- Wildcards (*) cannot be specified as a principal.
- A permissions policy must also be attached to the user in the trusted account.

POLICIES

Policies are documents that define permissions and can be applied to users, groups and roles.

Policy documents are written in JSON (key value pair that consists of an attribute and a value).

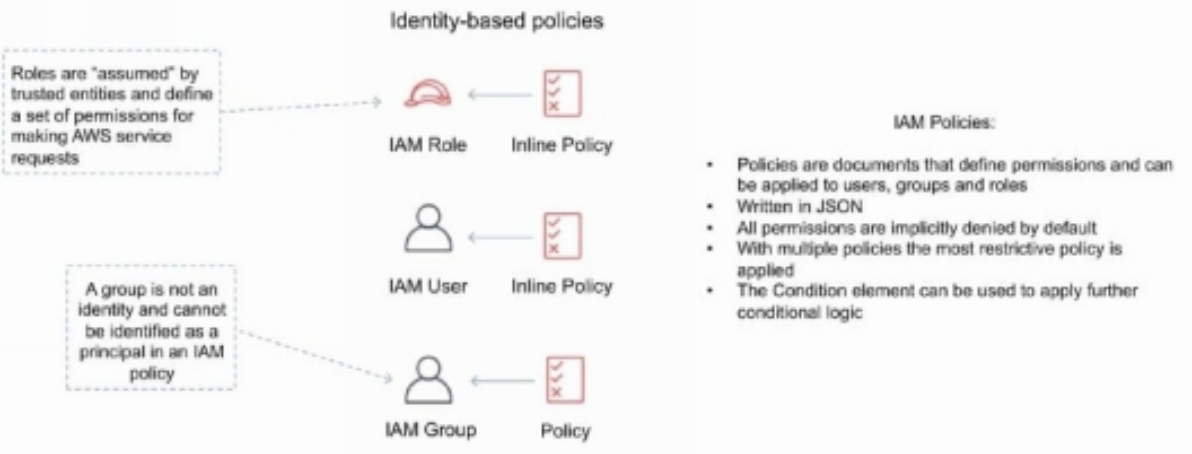
All permissions are implicitly denied by default.

The most restrictive policy is applied.

The IAM policy simulator is a tool to help you understand, test, and validate the effects of access control policies.

The Condition element can be used to apply further conditional logic.

The diagram below provides some more information on the relationship between IAM roles, users, groups and policies.



INLINE POLICIES VS MANAGED POLICIES

There are 3 types of policies:

- Managed policies.
- Customer managed policies.
- Inline policies.

Managed Policy:

- Created and administered by AWS.
- Used for common use cases based on job function.
- Save you having to create policies yourself.
- Can be attached to multiple users, groups, or roles within and across AWS accounts.
- Cannot change the permissions assigned.

Customer Managed Policy:

- Standalone policy that you create and administer in your own AWS account.
- Can be attached to multiple users, groups, and roles – but only within your own account.

- Can be created by copying an existing managed policy and then customizing it.
- Recommended for use cases where the existing AWS Managed Policies don’t meet the needs of your environment.

Inline Policy:

- Inline policies are embedded within the user, group or role to which it is applied.
- Strict 1:1 relationship between the entity and the policy.
- When you delete the user, group or role in which the inline policy is embedded, the policy will also be deleted.
- In most cases, AWS recommends using Managed Policies instead of inline policies.
- Inline policies are useful when you want to be sure that the permissions in a policy are not inadvertently assigned to any other user, group, or role.

AWS MANAGED AND CUSTOMER MANAGED POLICIES

An AWS managed policy is a standalone policy that is created and administered by AWS.

Standalone policy means that the policy has its own Amazon Resource Name (ARN) that includes the policy name.

AWS managed policies are designed to provide permissions for many common use cases.

You cannot change the permissions defined in AWS managed policies.

Some AWS managed policies are designed for specific job functions.

The job-specific AWS managed policies include:

- Administrator.

- Billing.
- Database Administrator.
- Data Scientist.
- Developer Power User.
- Network Administrator.
- Security Auditor.
- Support User.
- System Administrator.
- View-Only User.

You can create standalone policies that you administer in your own AWS account, which we refer to as customer managed policies.

You can then attach the policies to multiple principal entities in your AWS account.

When you attach a policy to a principal entity, you give the entity the permissions that are defined in the policy.

IAM POLICY EVALUATION LOGIC

By default, all requests are implicitly denied. (Alternatively, by default, the AWS account root user has full access.)

An explicit allow in an identity-based or resource-based policy overrides this default.

If a permissions boundary, Organizations SCP, or session policy is present, it might override the allow with an implicit deny.

An explicit deny in any policy overrides any allows.

A few concepts should be known to understand the logic:

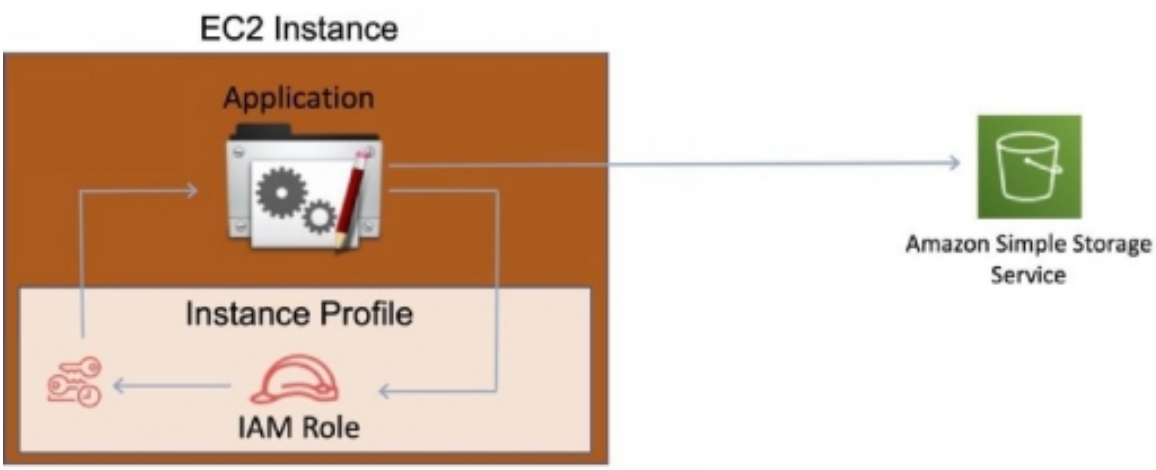
- **Identity-based policies** – Identity-based policies are attached to an IAM identity (user, group of users, or role) and grant permissions to IAM entities (users and roles).
- **Resource-based policies** – Resource-based policies grant permissions to the principal (account, user, role, or federated user) specified as the principal.
- **IAM permissions boundaries** – Permissions boundaries are an advanced feature that sets the maximum permissions that an identity-based policy can grant to an IAM entity (user or role).
- **AWS Organizations service control policies (SCPs)** – Organizations SCPs specify the maximum permissions for an organization or organizational unit (OU). Session policies – Session policies are advanced policies that you pass as parameters when you programmatically create a temporary session for a role or federated user.

The following flowchart details the IAM policy evaluation logic:

IAM INSTANCE PROFILES

An instance profile is a container for an IAM role that you can use to pass role information to an EC2 instance when the instance starts.

An instance profile can contain only one IAM role, although a role can be included in multiple instance profiles.



You can use the following AWS CLI commands to work with instance profiles in an AWS account:

- Create an instance profile: aws iam create-instance-profile
- Add a role to an instance profile: aws iam add-role-to-instance-profile
- List instance profiles: aws iam list-instance-profiles, aws iam list-instance-profiles-for-role
- Get information about an instance profile: aws iam get-instance-profile
- Remove a role from an instance profile: aws iam remove-role-from-instance-profile
- Delete an instance profile: aws iam delete-instance-profile

AWS SECURITY TOKEN SERVICE

The AWS Security Token Service (STS) is a web service that enables you to request temporary, limited-privilege credentials for IAM users or for users that you authenticate (federated users).

By default, AWS STS is available as a global service, and all AWS STS requests go to a single endpoint at **https://sts.amazonaws.com**

You can optionally send your AWS STS requests to endpoints in any region (can reduce latency).

All regions are enabled for STS by default but can be disabled.

The region in which temporary credentials are requested must be enabled.

Credentials will always work globally.

STS supports AWS CloudTrail, which records AWS calls for your AWS account and delivers log files to an **S3 bucket** .

Temporary security credentials work almost identically to long-term access key credentials that IAM users can use, with the following differences:

- Temporary security credentials are short-term.
- They can be configured to last anywhere from a few minutes to several hours.
- After the credentials expire, AWS no longer recognizes them or allows any kind of access to API requests made with them.
- Temporary security credentials are not stored with the user but are generated dynamically and provided to the user when requested.
- When (or even before) the temporary security credentials expire, the user can request new credentials, as long as the user requesting them still has permission to do so.

Advantages of STS are:

- You do not have to distribute or embed long-term AWS security credentials with an application.
- You can provide access to your AWS resources to users without having to define an AWS identity for them (temporary security credentials are the basis for IAM Roles and ID Federation).

- The temporary security credentials have a limited lifetime, so you do not have to rotate them or explicitly revoke them when they’re no longer needed.
- After temporary security credentials expire, they cannot be reused (you can specify how long the credentials are valid for, up to a maximum limit).

The AWS STS API action returns temporary security credentials that consist of:

- An access key which consists of an access key ID and a secret ID.
- A session token.
- Expiration or duration of validity.
- Users (or an application that the user runs) can use these credentials to access your resources.

With STS you can request a session token using one of the following APIs:

- AssumeRole – can only be used by IAM users (can be used for MFA).
- AssumeRoleWithSAML – can be used by any user who passes a SAML authentication response that indicates authentication from a known (trusted) identity provider.
- AssumeRoleWithWebIdentity – can be used by an user who passes a web identity token that indicates authentication from a known (trusted) identity provider.
- GetSessionToken – can be used by an IAM user or AWS account root user (can be used for MFA).
- GetFederationToken – can be used by an IAM user or AWS account root user.

STS AssumeRoleWithWebIdentity:

- Assume-role-with-web-identity is an API provided by STS (Security Token Service).
- Returns temporary security credentials for users authenticated by a mobile or web application or using a Web ID Provider like Amazon, Facebook or Google.
- For mobile applications, Cognito is recommended.
- Regular web applications can use the STS assume-role-with-web-identity API.

AWS recommends using Cognito for identity federation with Internet identity providers.

Users can come from three sources:

Federation (typically AD)

- Uses SAML 2.0.
- Grants temporary access based on the users AD credentials.
- Does not need to be a user in IAM.
- Single sign-on allows users to login to the AWS console without assigning IAM credentials.

There are a couple of ways STS can be used

Scenario 1:

1. Develop an Identity Broker to communicate with LDAP and AWS STS.
2. Identity Broker always authenticates with LDAP first, then with AWS STS.
3. Application then gets temporary access to AWS resources.

Scenario 2:

1. Develop an Identity Broker to communicate with LDAP and AWS STS.

2. Identity Broker authenticates with LDAP first, then gets an IAM role associated with the user.
3. Application then authenticates with STS and assumes that IAM role.
4. Application uses that IAM role to interact with the service.

CROSS ACCOUNT ACCESS

Useful for situations where an AWS customer has separate AWS account – for example for development and production resources.

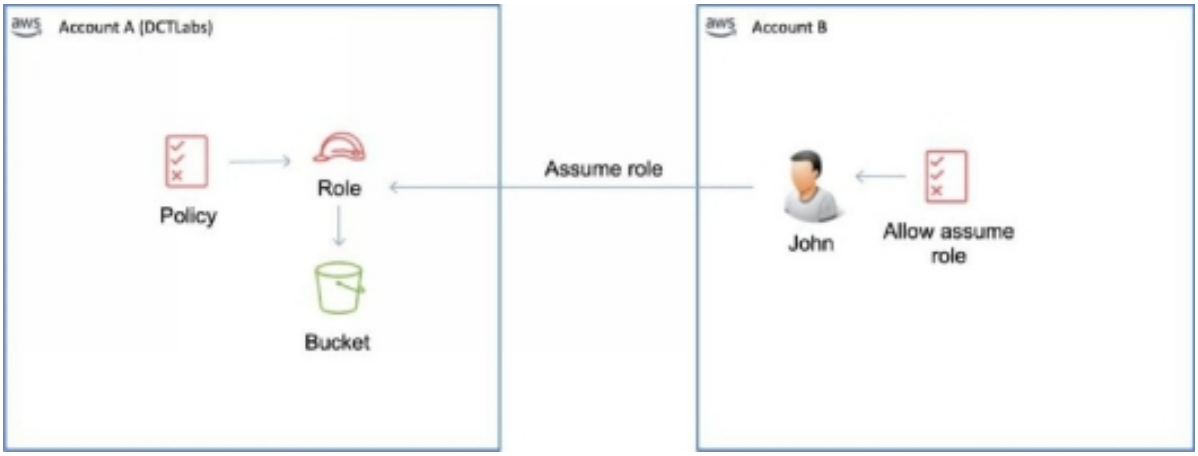
Cross Account Access makes is easier to work productively within a multi-account (or multi-role) AWS environment by making is easy to switch roles within the AWS Management Console.

Can sign-in to the console using your IAM user name and then switch the console to manage another account without having to enter another user name and password.

Lets users from one AWS account access resources in another.

To make a request in a different account the resource in that account must have an attached resource-based policy with the permissions you need.

Or you must assume a role (identity-based policy) within that account with the permissions you need.



IAM BEST PRACTICES

- Lock Away Your AWS Account Root User Access Keys.
- Create Individual IAM Users.
- Use Groups to Assign Permissions to IAM Users.
- Grant Least Privilege.
- Get Started Using Permissions with AWS Managed Policies.
- Use Customer Managed Policies Instead of Inline Policies.
- Use Access Levels to Review IAM Permissions.
- Configure a Strong Password Policy for Your Users.
- Enable MFA.
- Use Roles for Applications That Run on Amazon EC2 Instances.
- Use Roles to Delegate Permissions.
- Do Not Share Access Keys.
- Rotate Credentials Regularly.
- Remove Unnecessary Credentials.
- Use Policy Conditions for Extra Security.
- Monitor Activity in Your AWS Account.

AWS KMS AND AWS CLOUDHSM

AWS KMS

AWS Key Management Service (KMS) is a managed service that enables you to easily encrypt your data.

AWS KMS provides a highly available key storage, management, and auditing solution for you to encrypt data within your own applications and control the encryption of stored data across AWS services.

AWS KMS allows you to centrally manage and securely store your keys. These are known as customer master keys or CMKs.

CUSTOMER MASTER KEYS (CMK'S)

A Customer Master Key (CMK) consists of:

- Alias.
- Creation date.
- Description.
- Key state.
- Key material (either customer provided or AWS provided).

Customer master keys are the primary resources in AWS KMS.

The CMK includes metadata, such as the key ID, creation date, description, and key state.

The CMK also contains the key material used to encrypt and decrypt data.

AWS KMS supports symmetric and asymmetric CMKs.

CMKs are created in AWS KMS. Symmetric CMKs and the private keys of asymmetric CMKs never leave AWS KMS unencrypted.

By default, AWS KMS creates the key material for a CMK.

A CMK can encrypt data up to 4KB in size.

A CMK can generate, encrypt and decrypt Data Encryption Keys (DEKs).

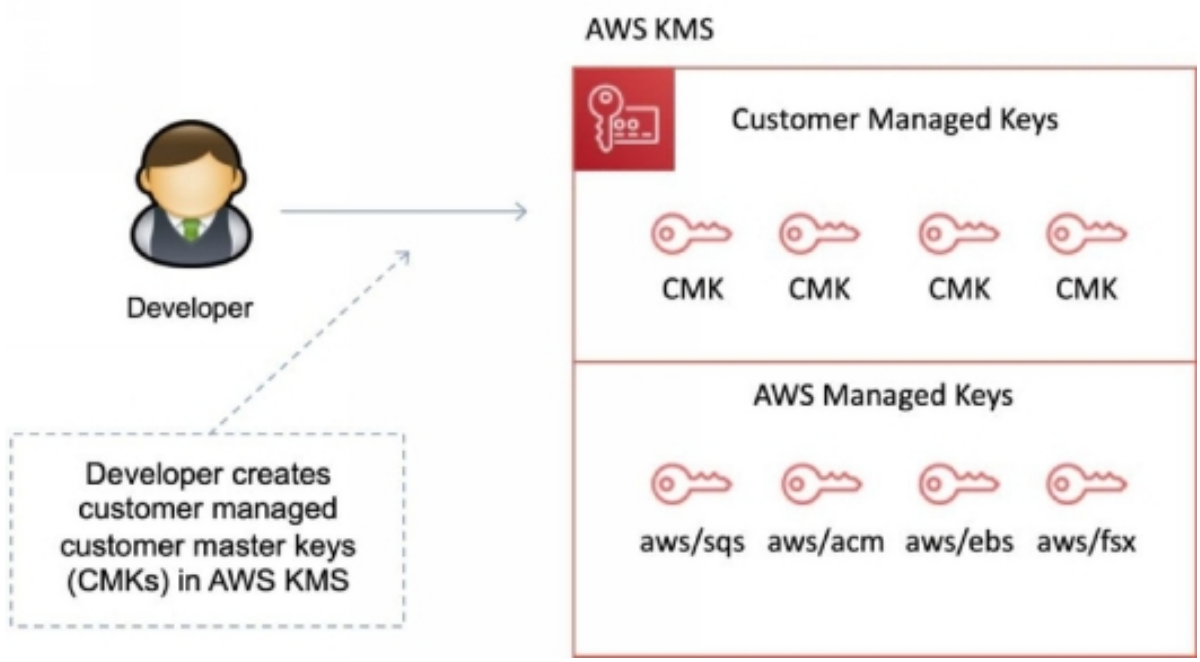
A CMK can never be exported from KMS (CloudHSM allows this).

AWS Managed CMKs:

- CMKs managed by AWS are used by AWS services that interact with KMS to encrypt data.
- They can only be used by the service that created them within a particular region.
- They are created on the first time you implement encryption using that service.

Customer managed CMKs:

- These provide the ability to implement greater flexibility.
- You can perform rotation, governing access and key policy configuration.
- You are able to enable and disable the key when it is no longer required.



CUSTOMER MANAGED CMK'S

Customer managed CMKs are CMKs in your AWS account that you create, own, and manage.

You have full control over these CMKs, including establishing and maintaining their key policies, IAM policies, and grants, enabling and disabling them, rotating their cryptographic material, adding tags, creating aliases that refer to the CMK, and scheduling the CMKs for deletion.

Customer managed CMKs incur a monthly fee and a fee for use in excess of the free tier.

AWS MANAGED CMK'S

AWS managed CMKs are CMKs in your account that are created, managed, and used on your behalf by an AWS service that is integrated with AWS KMS.

You cannot manage these CMKs, rotate them, or change their key policies.

You also cannot use AWS managed CMKs in cryptographic operations directly; the service that creates them uses them on your behalf.

You do not pay a monthly fee for AWS managed CMKs. They can be subject to fees for use in excess of the free tier, but some AWS services cover these costs for you.

AWS OWNED CMK'S

AWS owned CMKs are a collection of CMKs that an AWS service owns and manages for use in multiple AWS accounts.

Although AWS owned CMKs are not in your AWS account, an AWS service can use its AWS owned CMKs to protect the resources in your account.

You do not need to create or manage the AWS owned CMKs.

However, you cannot view, use, track, or audit them.

You are not charged a monthly fee or usage fee for AWS owned CMKs and they do not count against the AWS KMS quotas for your account.

DATA ENCRYPTION KEYS

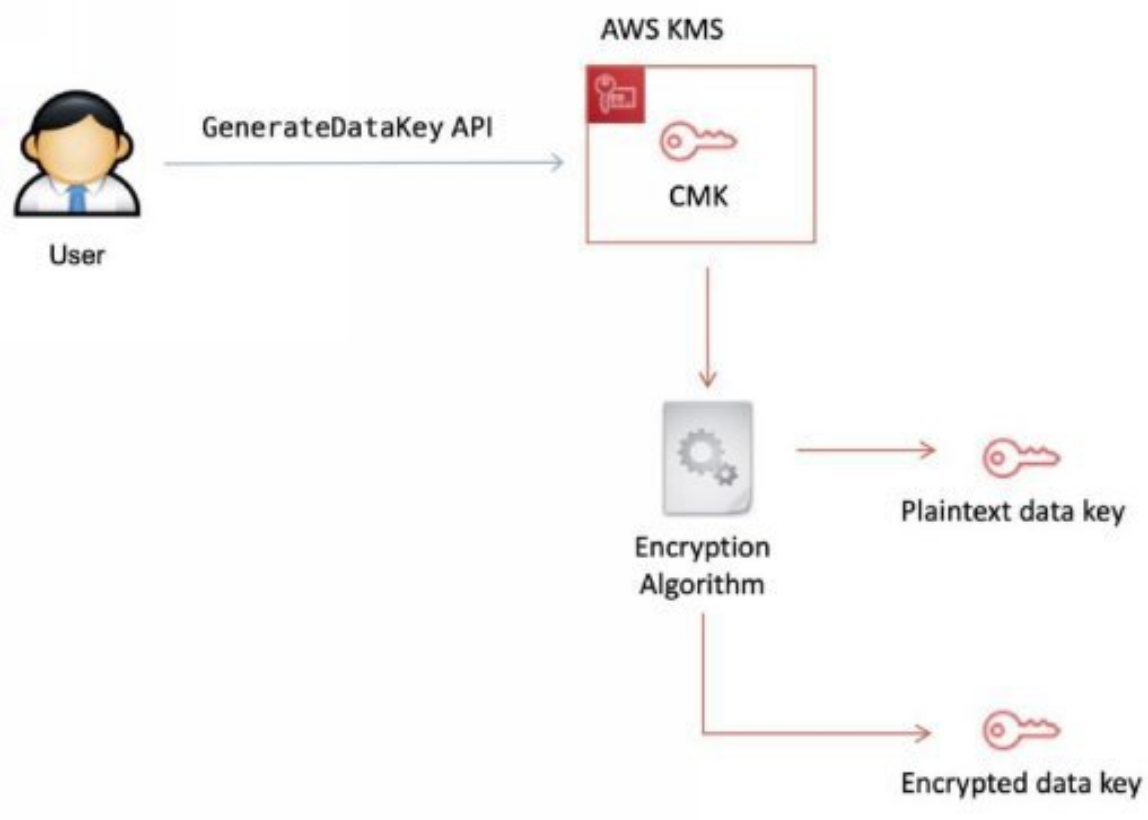
Data keys are encryption keys that you can use to encrypt data, including large amounts of data and other data encryption keys.

You can use AWS KMS customer master keys (CMKs) to generate, encrypt, and decrypt data keys.

AWS KMS does not store, manage, or track your data keys, or perform cryptographic operations with data keys.

You must use and manage data keys outside of AWS KMS.

The GenerateDataKey API can be used to create a data encryption key using a CMK:



KMS DETAILS

You set usage policies on the keys that determine which users can use them to encrypt and decrypt data and under which conditions.

Key material options:

- KMS generated.
- Import your own.

You can generate CMKs in KMS, in an AWS CloudHSM cluster, or import them from your own key management infrastructure.

These master keys are protected by hardware security modules (HSMs) and are only ever used within those modules.

You can submit data directly to KMS to be encrypted or decrypted using these master keys.

KMS now has the option for symmetric and asymmetric keys.

KMS is for encryption at rest only (not in transit, use SSL).

KMS is tightly integrated into many AWS services like Lambda, S3, EBS, EFS, DynamoDB, SQS etc.

Data keys are not retained or managed by KMS.

AWS services encrypt your data and store an encrypted copy of the data key along with the data it protects.

When a service needs to decrypt your data they request KMS to decrypt the data key using your master key.

If the user requesting data from the AWS service is authorized to decrypt under your master key policy, the service will receive the decrypted data key from KMS with which it can decrypt your data and return it in plaintext.

All requests to use your master keys are logged in AWS CloudTrail so you can understand who used which key under which context and when they used it.

You can control who manages and accesses keys via IAM users and roles.

You can audit the use of keys via CloudTrail.

KMS differs from Secrets Manager as its purpose-built for encryption key management.

KMS is validated by many compliance schemes (e.g. PCI DSS Level 1, FIPS 140-2 Level 2).

Exam tip : Encryption keys are regional.

KEY MANAGEMENT WITH KMS

You can perform the following key management functions in AWS KMS:

- Create keys with a unique alias and description.
- Import your own key material.

- Define which IAM users and roles can manage keys.
- Define which IAM users and roles can use keys to encrypt and decrypt data.
- Choose to have AWS KMS automatically rotate your keys on an annual basis.
- Temporarily disable keys so they cannot be used by anyone.
- Re-enable disabled keys.
- Delete keys that you no longer use.
- Audit use of keys by inspecting logs in AWS CloudTrail.
- Create custom key stores*.
- Connect and disconnect custom key stores*.
- Delete custom key stores*.

* The use of custom key stores requires CloudHSM resources to be available in your account.

DATA ENCRYPTION SCENARIOS

Typically, data is encrypted in one of the following three scenarios:

1. You can use KMS APIs directly to encrypt and decrypt data using your master keys stored in KMS.
2. You can choose to have AWS services encrypt your data using your master keys stored in KMS. In this case data is encrypted using data keys that are protected by your master keys in KMS.
3. You can use the AWS Encryption SDK that is integrated with AWS KMS to perform encryption within your own applications, whether they operate in AWS or not.

CUSTOM KEY STORE

The AWS KMS custom key store feature combines the controls provided by AWS CloudHSM with the integration and ease of use of AWS KMS.

You can configure your own CloudHSM cluster and authorize KMS to use it as a dedicated key store for your keys rather than the default KMS key store.

When you create keys in KMS you can chose to generate the key material in your CloudHSM cluster. Master keys that are generated in your custom key store never leave the HSMs in the CloudHSM cluster in plaintext and all KMS operations that use those keys are only performed in your HSMs.

In all other respects master keys stored in your custom key store are consistent with other KMS CMKs.

KEY DELETION

You can schedule a customer master key and associated metadata that you created in AWS KMS for deletion, with a configurable waiting period from 7 to 30 days.

This waiting period allows you to verify the impact of deleting a key on your applications and users that depend on it.

The default waiting period is 30 days.

You can cancel key deletion during the waiting period.

AWS KMS API'S

The following APIs are useful to know for the exam:

Encrypt (aws kms encrypt):

- Encrypts plaintext into ciphertext by using a customer master key (CMK).
- You can encrypt small amounts of arbitrary data, such as a personal identifier or database password, or other sensitive information.

- You can use the Encrypt operation to move encrypted data from one AWS region to another.

Decrypt (aws kms decrypt):

- Decrypts ciphertext that was encrypted by a AWS KMS customer master key (CMK) using any of the following operations:
  - Encrypt
  - GenerateDataKey
  - GenerateDataKeyPair
  - GenerateDataKeyWithoutPlaintext
  - GenerateDataKeyPairWithoutPlaintext

Re-encrypt (aws kms re-encrypt):

- Decrypts ciphertext and then re-encrypts it entirely within AWS KMS.
- You can use this operation to change the customer master key (CMK) under which data is encrypted, such as when you manually rotate a CMK or change the CMK that protects a ciphertext.
- You can also use it to re-encrypt ciphertext under the same CMK, such as to change the encryption context of a ciphertext.

Enable-key-rotation:

- Enables automatic rotation of the key material for the specified symmetric customer master key (CMK).
- You cannot perform this operation on a CMK in a different AWS account.

GenerateDataKey (aws kms generate-data-key):

- Enables automatic rotation of the key material for the specified symmetric customer master key (CMK).
- You cannot perform this operation on a CMK in a different AWS account.

GenerateDataKeyWithoutPlaintext (generate-data-key-without-plaintext):

- Generates a unique symmetric data key.
- This operation returns a data key that is encrypted under a customer master key (CMK) that you specify.
- To request an asymmetric data key pair, use the GenerateDataKeyPair or GenerateDataKeyPairWithoutPlaintext operations.

KMS ENVELOPE ENCRYPTION

AWS KMS is integrated with AWS services and client-side toolkits that use a method known as envelope encryption to encrypt your data.

Under this method, KMS generates data keys which are used to encrypt data and are themselves encrypted using your master keys in KMS:

- A CMK is used to encrypt the data key (envelope key).
- The envelope key is used to decrypt the data.

AWS CLOUDHSM

AWS CloudHSM is a cloud-based hardware security module (HSM) that enables you to easily generate and use your own encryption keys on the AWS Cloud.

With CloudHSM, you can manage your own encryption keys using FIPS 140-2 Level 3 validated HSMs.

CloudHSM offers you the flexibility to integrate with your applications using industry-standard APIs, such as PKCS#11, Java Cryptography

Extensions (JCE), and Microsoft CryptoNG (CNG) libraries.

CloudHSM is standards-compliant and enables you to export all of your keys to most other commercially-available HSMs, subject to your configurations.

It is a fully-managed service that automates time-consuming administrative tasks for you, such as hardware provisioning, software patching, high-availability, and backups.

CloudHSM also enables you to scale quickly by adding and removing HSM capacity on-demand, with no up-front costs.

CloudHSM runs in your VPC.

The following table helps to understand the key differences between AWS CloudHSM and AWS KMS:

	CloudHSM	AWS KMS
Tenancy	Single-tenant HSM	Multi-tenant AWS service
Availability	Customer-managed durability and available	Highly available and durable key storage and management
Root of Trust	Customer managed root of trust	AWS managed root of trust
FIPS 140-2	Level 3	Level 2 / Level 3 in some areas
3rd Party Support	Broad 3rd Party Support	Broad AWS service support

AWS WAF AND SHIELD

AWS WAF and AWS Shield help protect your AWS resources from web exploits and DDoS attacks.

AWS WAF is a web application firewall service that helps protect your web apps from common exploits that could affect app availability, compromise security, or consume excessive resources.

AWS Shield provides expanded DDoS attack protection for your AWS resources. Get 24/7 support from our DDoS response team and detailed visibility into DDoS events.

We'll now go into more detail on each service.

AWS WEB APPLICATION FIREWALL (WAF)

AWS WAF is a web application firewall that helps protect your web applications from common web exploits that could affect application availability, compromise security, or consume excessive resources.

AWS WAF helps protect web applications from attacks by allowing you to configure rules that allow, block, or monitor (count) web requests based on conditions that you define.

These conditions include IP addresses, HTTP headers, HTTP body, URI strings, SQL injection and cross-site scripting.

AWS WAF gives you control over which traffic to allow or block to your web applications by defining customizable web security rules.

New rules can be deployed within minutes, letting you respond quickly to changing traffic patterns.

When AWS services receive requests for web sites, the requests are forwarded to AWS WAF for inspection against defined rules.

Once a request meets a condition defined in the rules, AWS WAF instructs the underlying service to either block or allow the request based on the action you define.

With AWS WAF you pay only for what you use.

AWS WAF pricing is based on how many rules you deploy and how many web requests your web application receives.

There are no upfront commitments.

AWS WAF is tightly integrated with Amazon CloudFront and the Application Load Balancer (ALB), services.

When you use AWS WAF on Amazon CloudFront, rules run in all AWS Edge Locations, located around the world close to end users.

This means security doesn't come at the expense of performance.

Blocked requests are stopped before they reach your web servers.

When you use AWS WAF on an Application Load Balancer, your rules run in region and can be used to protect internet-facing as well as internal load balancers.

WEB TRAFFIC FILTERING

AWS WAF lets you create rules to filter web traffic based on conditions that include IP addresses, HTTP headers and body, or custom URIs.

This gives you an additional layer of protection from web attacks that attempt to exploit vulnerabilities in custom or third party web applications.

In addition, AWS WAF makes it easy to create rules that block common web exploits like SQL injection and cross site scripting.

AWS WAF allows you to create a centralized set of rules that you can deploy across multiple websites.

This means that in an environment with many websites and web applications you can create a single set of rules that you can reuse across applications rather than recreating that rule on every application you want to protect.

FULL FEATURE API

AWS WAF can be completely administered via APIs.

This provides organizations with the ability to create and maintain rules automatically and incorporate them into the development and design process.

For example, a developer who has detailed knowledge of the web application could create a security rule as part of the deployment process.

This capability to incorporate security into your development process avoids the need for complex handoffs between application and security teams to make sure rules are kept up to date.

AWS WAF can also be deployed and provisioned automatically with AWS CloudFormation sample templates that allow you to describe all security rules you would like to deploy for your web applications delivered by Amazon CloudFront.

AWS WAF is integrated with Amazon CloudFront, which supports custom origins outside of AWS – this means you can protect web sites not hosted in AWS.

Support for IPv6 allows the AWS WAF to inspect HTTP/S requests coming from both IPv6 and IPv4 addresses.

REAL-TIME VISIBILITY

AWS WAF provides real-time metrics and captures raw requests that include details about IP addresses, geo locations, URIs, User-Agent and Referers.

AWS WAF is fully integrated with Amazon CloudWatch, making it easy to setup custom alarms when thresholds are exceeded or particular attacks occur.

This information provides valuable intelligence that can be used to create new rules to better protect applications.

AWS SHIELD

AWS Shield is a managed Distributed Denial of Service (DDoS) protection service that safeguards applications running on AWS.

AWS Shield provides always-on detection and automatic inline mitigations that minimize application downtime and latency, so there is no need to engage AWS Support to benefit from DDoS protection.

There are two tiers of AWS Shield – Standard and Advanced.

AWS SHIELD STANDARD

All AWS customers benefit from the automatic protections of AWS Shield Standard, at no additional charge.

AWS Shield Standard defends against most common, frequently occurring network and transport layer DDoS attacks that target web sites or applications.

When using AWS Shield Standard with Amazon CloudFront and Amazon Route 53, you receive comprehensive availability protection against all known infrastructure (Layer 3 and 4) attacks.

AWS SHIELD ADVANCED

Provides higher levels of protection against attacks targeting applications running on Amazon Elastic Compute Cloud (EC2), Elastic Load Balancing (ELB), Amazon CloudFront, AWS Global Accelerator and Amazon Route 53 resources.

In addition to the network and transport layer protections that come with Standard, AWS Shield Advanced provides additional detection and mitigation against large and sophisticated DDoS attacks, near real-time visibility into attacks, and integration with AWS WAF, a web application firewall.

AWS Shield Advanced also gives you 24×7 access to the AWS DDoS Response Team (DRT) and protection against DDoS related spikes in your Amazon Elastic Compute Cloud (EC2), Elastic Load Balancing (ELB),

Amazon CloudFront, AWS Global Accelerator and Amazon Route 53 charges.

AWS Shield Advanced is available globally on all Amazon CloudFront, AWS Global Accelerator, and Amazon Route 53 edge locations.

Origin servers can be Amazon S3, Amazon Elastic Compute Cloud (EC2), Elastic Load Balancing (ELB), or a custom server outside of AWS.

AWS Shield Advanced includes DDoS cost protection, a safeguard from scaling charges as a result of a DDoS attack that causes usage spikes on protected Amazon EC2, Elastic Load Balancing (ELB), Amazon CloudFront, AWS Global Accelerator, or Amazon Route 53.

If any of the AWS Shield Advanced protected resources scale up in response to a DDoS attack, you can request credits via the regular AWS Support channel.

EXAM SCENARIOS FOR AWS SYSOPS ADMINISTRATOR

AMAZON EC2 AND AWS LAMBDA

Exam Scenario	Solution
Administrator needs to check if any **Amazon EC2** instances will be affected by scheduled hardware maintenance	Check the AWS Personal Health Dashboard
Scheduled hardware maintenance will affect a critical EC2 instance	Stop and start the instance to move it to different underlying hardware
When launching an EC2 instance the InsufficientInstanceCapacity error is experienced	This means AWS does not currently have enough capacity to service the request for that instance type. Try a different AZ or instance type
The error InstanceLimitExceeded is experienced when launching EC2 instances	EC2 instance limits have been reached, need to contact support to request an increased limit
System status checks are failing for an EC2 instance	Stop and start again to move to a new host

ELASTIC LOAD BALANCING AND AUTO SCALING

Exam Scenario	Solution
Design required for highly available and secure website on EC2 with **ALB** , and DB on EC2	Launch ALB in public subnets, web servers in private subnets and DB

	layer in private subnets – all layers across AZs
HealthyHostCount metrics for an ALB have dropped from 6 to 2. Need to determine the cause	The health checks on target EC2 instances are failing
An instance attached to an ALB exceeded the UnhealthyThresholdCount for consecutive health check failures. What will happen?	Health checks will continue and the ALB will take the instance out of service
Requirement to track the source IP of clients and the instance that processes the request	Check the ALB access logs for this information
503 and 504 errors experienced and instances have high CPU utilization	Use **EC2 Auto Scaling** to dynamically scale

AMAZON EBS, EFS, AND AWS STORAGE GATEWAY

Exam Scenario	Solution
User deleted some data in an **Amazon EBS** volume and there's a recent snapshot	Can create a new EBS volume from the snapshot and attach it to an instance and copy the delete file across
EBS volume runs out of space and need to prevent it happening again	Use CloudWatch agent on EC2 and monitor disk metrics with CloudWatch alarm
Low latency access required for image files in an office location with synchronized backup to offsite location. Local access required and disaster recovery	Use an **AWS Storage Gateway** volume gateway configured as a stored volume
EBS volume capacity is increased but cannot see the space	Need to extend the volume's file system to gain access to extra space

Need to replace user-shared drives. Must support POSIX permissions and NFS protocols and be accessible from on-premise servers and EC2	Use **Amazon EFS**

AWS SYSTEMS MANAGER

Exam Scenario	Solution
Application running on EC2 needs login credentials for a DB that are stored as secure strings in SSM Parameter Store	Create an IAM role for the instance and grant permission to read the parameters
Linux instances are patched with **Systems Manager Patch Manager** . Application slows down whilst updates are happening	Change maintenance window to patch 10% of instances in the patch group at a time
Custom Linux AMI used with AWS Systems Manager. Can't find instances in Session Manager console	Need to add permissions to instance profile and install the SSM agent on the instances
Multiple environments require authentication credentials for external service. Deployed using CloudFormation	Use an AWS Config rule to identify noncompliant keys. Create a custom AWS Systems Manager Automation document for remediatio
IAM access keys used to manage EC2 instances using the CLI. Company policy mandates that access keys are automatically disabled after 60 days	Use an AWS Config rule to identify noncompliant keys. Create a custom AWS Systems Manager Automation document for remediation

AWS CLOUDFORMATION

Exam Scenario	Solution
Need to review updates to an **AWS CloudFormation** stack before	Use change sets

deploying them in production	
Stack deployed and manual changes were made. Need to capture changes and update template	Use drift detection and use output to update template and redeploy the stack
Need to update new version of app on EC2 and ALB. Must avoid DNS changes and be able to rollback	Update template with AutoScalingReplacingUpdate policy and perform an update
Need to write a single template that can be deployed across several environments / Region	Use parameters to enter custom values and use Ref intrinsic function to reference the parameter
Tried to launch instance in a different region from a working template and it fails	Probably due to incorrect AMI ID

AMAZON VIRTUAL PRIVATE CLOUD (VPC)

Exam Scenario	**Solution**
Need to identify the instances that are generating the most traffic using a NAT gateway	Use **VPC flow logs** on the NAT gateway ENI and use CloudWatch insights to filter based on source IP address
Latency on a NAT instance has increased, need a solution that scales with demand cost-efficiently	Swap with a NAT gateway
NAT gateway is NOT highly available across AZs, only within an AZ	Use multiple NAT gateways for HA across AZs
NAT instance deployed but not working	Make sure to disable source/destination checks
Need to enable access to S3 without the instances using public IP addresses	Use a NAT gateway or VPC endpoint

AMAZON ROUTE 53

Exam Scenario	Solution
Use **Route 53** to direct based on health checks with (2xx) traffic to primary and other responses to secondary	Need to create an A record for each server and a HTTP (not TCP) health check
Route 53 health check uses string matching for "/html". Alert shows health check fails	The search string must appear entirely within the first 5,120 bytes of the response body
Need to make a website promotion visible to users from a specific country only	Use Route 53 geolocation routing policy
New website runs on EC2 behind ALB. Need to create record in Route 53 to point to the domain apex (e.g. example.com)	Use an alias record
Hosted zone in Account A and ALB in Account B. Need the most cost-effective and efficient solution for pointing to the ALB	Create an Alias record in Account A that points to ALB in Account B

AMAZON S3 AND CLOUDFRONT

Exam Scenario	Solution
Static website on **Amazon S3** with custom domain name	Requires that the bucket name matches the DNS name / record set name in Route 53
503 errors experienced with new site and thousands of user	Request rate is too high
Discrepancy with number of objects in bucket console vs CloudWatch	Use Amazon S3 Inventory to properly determine the number of objects in a bucket

Need to enforce encryption on all objects uploaded to bucket	Use a bucket policy with a “Condition”: { “Bool”: { “aws:SecureTransport”: “false” statement for PutObject and with the resource set to the bucket
Unauthorized users tried to connect to S3 buckets. Need to know which buckets are targeted and who is trying to get access	Use S3 server access logs and Athena to query for HTTP 403 errors and look for IAM user or role making requests

AMAZON RDS AND ELASTICACHE

Exam Scenario	Solution
Automated failover of a multi-AZ DB occurred	This may be due to storage failure on primary DB or the instance type could have been changed
Need to encrypt unencrypted RDS database	Take a snapshot, encrypt it, then restore a new encrypted instance from the snapshot
RDS DB query latency is high and CPU utilization is at 100%	Scale up with larger instance type
Need to share RDS DB snapshots across different accounts. Data must be encrypted	Use an AWS KMS key for encryption and update key policy to grant accounts with access then share snapshot
DB needs to be made HA to protect against failure and updates cannot impact users in business hours	Change to Multi-AZ outside of business hours

MANAGEMENT, GOVERNANCE AND BILLING

Exam Scenario	Solution
Audit requests to **AWS**	use CloudTrail and look for the

Organizations for creating new accounts by federated users	federated identity user name
Employees have created individual AWS accounts not under control. Security team need them in AWS Organizations	Send each account an invitation from the central organization
Need to restrict ability to launch specific instance types for a specific team/account	Use an organizations SCP to deny launches unless the instance type is T2, create an IAM group in the account granting access to T2 instances to the relevant users
Need to test notification settings for **CloudWatch** alarm with SNS	Use the set-alarm-state CLI command to test
Need to automatically disable access keys that are greater than 90 days old	Use an **AWS Config** rule to identify noncompliant keys and use Systems Manager Automation to remediate

SECURITY AND COMPLIANCE

Exam Scenario	**Solution**
Company wishes to force users to change their passwords regularly	Create an **IAM password policy** and enabled password expiration
Need to restrict access to a bucket based on source IP range	Use bucket policy with “Condition”: “NotIpAddress”: statement
Need to control access to group of EC2 instances with specific tags	Use an IAM policy with a condition element granting access based on the tag and attach an IAM policy to the user or groups that require access
IAM policy for SQS queue allows too much access. Who is responsible for correcting the issue?	According the AWS shared responsibility mode, this is a customer responsibility
Data is encrypted with **AWS KMS** customer-managed CMKs. Need to	Just enable key rotation in AWS KMS for the CMK (backing key is rotated, data key is not changed)

enable rotation ensuring the data remains readable	

CONCLUSION

We trust that these training notes have helped you to gain a complete understanding of the facts you need to know to pass the AWS Certified SysOps Administrator Associate exam first time.

The exam covers a broad set of technologies. It’s vital to ensure you are armed with the knowledge to answer whatever questions come up in your certification exam. We recommend reviewing these training notes until you’re confident in all areas.

BEFORE TAKING THE AWS EXAM

Get Hands-On experience with AWS

AWS certification exams such as the SysOps test your hands-on knowledge and experience with the AWS platform. It's therefore super important to have some practical experience before you sit the exam.

Our **AWS Certified SysOps Administrator Associate video course** provides a practical approach to learning. Through over 15 hours of on demand video you'll learn how deploy, manage, and operate scalable, highly available, and fault tolerant systems on AWS. Our mixture of in-depth theory, logical diagrams and hands-on training, will fully prepare you for the AWS SysOps Certification exam.

By the end of the course you will have developed a strong experience-based skillset. This is the best way to gain hands-on skills and will give you an edge on the day of your exam.

Assess your exam readiness with practice exams

The Digital Cloud Training practice questions are the closest to the actual exam and the only exam-difficulty questions on the market. If you can pass these mock exams, you’re well set to ace the real thing. To learn more, visit

https://digitalcloud.training/aws-certified-sysops-administrator-associate-exam-training/

REACH OUT AND CONNECT

We want you to have a 5-star learning experience. If anything is not 100% to your liking, please email us at **support@digitalcloud.training** . We promise to address all questions and concerns. We really want you to get great value from these training resources.

The AWS platform is evolving quickly, and the exam tracks these changes with a typical lag of around 6 months. We are therefore reliant on student feedback to keep track of what is appearing in the exam. If there are any topics in your exam that weren't covered in our training resources, please provide us with feedback using this form **https://digitalcloud.training/student-feedback/** . We appreciate your feedback that will help us further improve our AWS training resources.

To discuss any exam-specific questions you may have, please join the discussion on **Slack** . Visit **http://digitalcloud.training/slack** for instructions.

Also, remember to join our private Facebook group to ask questions and share your knowledge with the AWS community: **https://www.facebook.com/groups/awscertificationqa**

BONUS OFFER

To assess your AWS exam readiness, we have included one full-length practice exam from Digital Cloud Training. These 65 exam-difficulty practice questions are timed and scored and simulate the real AWS exam

experience. To gain access to your **free practice exam with 65 exam-difficulty questions** on the interactive online exam simulator, follow the steps below:

Step 1 : Visit **https://learn.digitalcloud.training/product/aws-sample-practice-exam-certified-sysops-administrator-associate-bonus/** or simply scan this QR code.

Step 2 : Click "Add to cart" and add coupon code "AMZBONUS" to reduce the price from $9.99 to $0.

Step 3 : Upon registration, **log in** to **http://learn.digitalcloud.training/login** and go to '**My Courses** ' to access your practice exam.

For those who have already purchased the full set of practice questions, please note that these 65 questions are included in the pool of questions.

LEAVE US A REVIEW

Your reviews help us improve our courses and help your fellow AWS students make the right choices. We celebrate every honest review and truly appreciate it. You can leave a review at any time by visiting **amazon.com/ryp** or your local amazon store (e.g. amazon.co.uk/ryp).

Best wishes for your AWS certification journey!


DigitalCloud
TRAINING

OTHER BOOKS & COURSES BY NEAL DAVIS

All of our courses are available on **digitalcloud.training/aws-training-courses**

Apply coupon code **AMZ20** for a 20% discount.

COURSES FOR THE AWS CERTIFIED CLOUD PRACTITIONER

Course	Description
AWS Certified Cloud Practitioner Instructor-led Video Course	**HIGHLY FLEXIBLE COURSE STRUCTURE:** You can move quickly through the course, focusing on the theory lectures. **GUIDED HANDS-ON EXERCISES:** To gain more practical experience with AWS services, you have the option to explore the guided hands-on exercises. **EXAM-CRAM LECTURES** : Get through the key exam facts in the shortest time possible with the exam-cram lectures that you'll find at the end of each section. **HIGH-QUALITY VISUALS** : We've spared no effort to create a highly visual training course with lots of table and graphs.
AWS Certified Cloud Practitioner (online) Practice Exams + Exam Simulator	Get access to the **Practice Exam course** from Digital Cloud Training: 6 sets of practice tests with 65 Questions each. All questions are unique, 100% scenario-based and conform to the latest CLF-C01 exam blueprint. Our AWS Practice Tests are delivered in 4 different modes:

	• Exam Mode • Training Mode • Knowledge Reviews • Final Exam Simulator (with 500 practice questions)
AWS Certified Cloud Practitioner (offline) Practice Tests (ebook)	There are **6 practice exams with 65 questions each** covering the five domains of the AWS CLF-C01 exam blueprint. Each set of questions is repeated once without answers and explanations, and once with answers and explanations, so you get to choose from two methods of preparation: 1: To simulate the exam experience and assess your exam readiness, use the "**PRACTICE QUESTIONS ONLY** " sets. 2: To use the practice questions as a learning tool, use the "**PRACTICE QUESTIONS, ANSWERS & EXPLANATIONS** " sets to view the answers and read the in-depth explanations as you move through the questions.
Training Notes for the AWS Certified Cloud Practitioner (cheat sheets)	This book is based on the CLF-C01 exam blueprint and provides a deep dive into the subject matter in a concise and easy-to-read format so you can fast-track your time to success. AWS Solutions Architect, Neal Davis, has consolidated the information you need to be successful.

COURSES FOR THE AWS CERTIFIED SOLUTIONS ARCHITECT ASSOCIATE

Course	Description
AWS Certified Solutions Architect Associate	This popular AWS Certified Solutions Architect Associate (SAA-C02) video course is delivered through guided Hands-On Labs exercises • 28 hours Video Lessons

Instructor-led Video Course	• Exam Cram Lectures • 90 Quiz Questions • High-Quality Visuals • Guided Hands-on Exercises • Build Applications on AWS
AWS Certified Solutions Architect Associate (online) Practice Tests	Get access to the **Practice Exam course** from Digital Cloud Training: 6 sets of practice tests with 65 Questions each. All questions are unique, 100% scenario-based and conform to the latest AWS SAA-C02 exam blueprint. Our AWS Practice Tests are delivered in 4 different modes: • Exam Mode • Training Mode • Knowledge Reviews • Final Exam Simulator (with 500 practice questions)
AWS Certified Solutions Architect Associate (offline) Practice Tests (ebook)	There are **6 practice exams with 65 questions each** covering the AWS SAA-C02 exam blueprint. Each set of questions is repeated once without answers and explanations, and once with answers and explanations. 1: To simulate the exam experience and assess your exam readiness, use the "**PRACTICE QUESTIONS ONLY** " sets. 2: To use the practice questions as a learning tool, use the "**PRACTICE QUESTIONS, ANSWERS & EXPLANATIONS** " sets to view the answers and read the in-depth explanations as you move through the questions.
Training Notes for the AWS Certified Solutions Architect Associate (cheat sheets)	Deep dive into the SAA-C02 exam objectives with over 300 pages of detailed facts, tables and diagrams. Save valuable time by getting straight to the facts you need to know to pass your AWS Certified Solutions Architect Associate exam first time! This book is based on the 2020 SAA-C02 exam blueprint and provides a deep dive into the subject

	matter in a concise and easy-to-read format so you can fast-track your time to success.

COURSES FOR THE AWS CERTIFIED DEVELOPER ASSOCIATE

Course	Description
AWS Certified Developer Associate Instructor led Video Course	This popular **AWS Certified Developer Associate Exam Training** for the **DVA-C01 certification exam** is packed with over 28 hours of comprehensive video lessons, hands-on labs, quizzes and exam-crams. With our mixture of in-depth theory, architectural diagrams and hands-on training, you'll learn how to architect and build applications on **Amazon Web Services** , fully preparing you for the **AWS Developer Certification** exam. With this complete **AWS Developer training** course, you have everything you need to comfortably pass the **AWS Developer Certification** exam at the first attempt.
AWS Certified Developer Associate (online) Practice Tests	Get access to the **Practice Exam Course** from Digital Cloud Training with 390 Questions in 6 sets of practice tests. All questions are unique and conform to the latest AWS DVA-C01 exam blueprint. Our AWS Practice Tests are delivered in 4 different modes: • Exam Mode • Training Mode • Knowledge Reviews • Final Exam Simulator
AWS Certified Developer Associate (offline) Practice Tests (ebook)	There are **6 practice exams with 65 questions each** covering all topics for the AWS DVA-C01 exam. Each set of questions is repeated once without answers and explanations, and once with answers and explanations, so you get to choose from two methods of preparation:

	1: To simulate the exam experience and assess your exam readiness, use the "**PRACTICE QUESTIONS ONLY** " sets. 2: To use the practice questions as a learning tool, use the "**PRACTICE QUESTIONS, ANSWERS & EXPLANATIONS** " sets to view the answers and read the in-depth explanations as you move through the questions.
Training Notes for the AWS Certified Developer Associate (cheat sheets)	With these in-depth AWS Training Notes for the Developer Associate, you'll learn everything you need to know to ace your exam! Fast-track your exam success with over 340 pages of exam-specific facts, tables and diagrams. AWS Solution Architect and founder of Digital Cloud Training, Neal Davis, has consolidated ALL of the key information into this essential cheat sheet. Based on the latest DVA-C01 certification exam, these Training Notes will shortcut your study time and maximize your chance of passing your exam first time.

COURSES FOR THE AWS CERTIFIED SYSOPS ADMINISTRATOR ASSOCIATE

Course	Description
AWS Certified SysOps Administrator Associate Instructor led Video Course	This popular **AWS Certified SysOps Administrator Exam Training** for the **SOA-C01 certification exam** is packed with 15 hours of comprehensive video lessons, exam scenarios and practical exercises. With our mixture of in-depth theory, logical diagrams and hands-on training, you'll learn how deploy, manage, and operate scalable, highly available, and fault tolerant systems on AWS, fully preparing you for the AWS SysOps Certification exam. With this complete **AWS SysOps training** course, you have everything

	you need to comfortably pass the **AWS SysOps Certification** exam at the first attempt.
AWS Certified SysOps Administrator Associate (online) Practice Tests	Get access to the **Practice Exam Course** from Digital Cloud Training with 195 Questions in 3 sets of practice tests. All questions are unique and conform to the latest AWS SOA-C01 exam blueprint. Our AWS Practice Tests are delivered in 4 different modes: • Exam Mode • Training Mode • Knowledge Reviews • Final Exam Simulator
Training Notes for the AWS Certified Developer Associate (cheat sheets)	With these in-depth AWS Training Notes for the SysOps Administrator, you'll learn everything you need to know to ace your exam! Fast-track your exam success with exam-specific facts, tables and diagrams. Founder of Digital Cloud Training, Neal Davis, has consolidated ALL of the key information into this essential cheat sheet. Based on the latest SOA-C01 certification exam, these Training Notes will shortcut your study time and maximize your chance of passing your exam first time.

ABOUT THE AUTHOR

Neal Davis is the founder of Digital Cloud Training, AWS Cloud Solutions Architect and successful IT instructor. With more than 20 years of experience in the tech industry, Neal is a true expert in virtualization and cloud computing. His passion is to help others achieve career success by offering in-depth AWS certification training resources.

Neal started **Digital Cloud Training** to provide a variety of training resources for Amazon Web Services (AWS) certifications that represent a higher standard of quality than is otherwise available in the market.

Through our hands-on AWS training courses, we help students build the knowledge and practical skill set they need to not only pass their AWS certification exams with flying colours but to also excel in their cloud career.

Our AWS training is delivered to suit many learning styles, using an effective combination of visual aids, hands-on training, online cheat sheets and high quality practice questions that reflect the difficulty and style of real AWS exam questions.

With all of these quality resources, learners have everything they need to confidently pass their exams. Students regularly report pass marks with average scores well above 85%.

We've built an active community around our cutting-edge training courses so you have the guidance and support you need every step of the way. Join the AWS Community of over 250,000 happy students that are currently enrolled in Digital Cloud Training courses.

CONNECT WITH NEAL ON SOCIAL MEDIA

All Links available on **https://digitalcloud.training/neal-davis**

digitalcloud.training/neal-davis

youtube.com/c/digitalcloudtraining

facebook.com/digitalcloudtraining

Twitter @ **nealkdavis**

linkedin.com/in/nealkdavis

Instagram @digitalcloudtraining