AWS CERTIFIED

SOLUTIONS ARCHITECT ASSOCIATE

TRAINING NOTES

Fast-track your exam success with this cheat sheet for the SAA-C01 exam

Over 240 pages of detailed and regularly updated facts

- Everything you need to know to pass the exam in a concise, easy-to-read format
- Master the new exam pattern with our exam-difficulty, scenario-based questions
- Detailed explanations to prepare you for the AWS SAA-C01 exam

Neal Davis

DigitalCloud TRAINING

GETTING STARTED

Welcome ☺

Thanks for purchasing these training notes for the **AWS Certified Solutions Architect Associate** exam from Digital Cloud Training. The information in this document relates to the latest version of the SAA-C01 exam that was released in February 2018 and is regularly updated.

The SAA-C01 exam covers a broad set of AWS services and the aim of putting this information together 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 exam first time.

I hope you get great value from this resource and wish you the best of luck with your AWS Certified Solutions Architect Associate exam.

Neal Davis

AWS Solutions Architect & Founder of Digital Cloud Training

How to best prepare for your exam

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.

If you're new to AWS, we'd suggest first taking an online instructor-led course to familiarize yourself with the AWS platform before returning to the Training Notes to get a more detailed understanding of the AWS services. We highly recommend our AWS Certified Solutions Architect Associate Hands-on Labs Video Course. For more information on the "**Best courses for the AWS Certified Solutions Architect Associate Training"** read our blog post.

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 20,000 students who've taken the exam, as well as our own experience - and may differ between AWS services.

To assess where you are at on your AWS journey, we recommend taking our practice exams (see below) to identify your strengths and weaknesses. As a next step, use these training notes to focus your study on the knowledge areas where you need to most.

For further information, read our blog article "**AWS Certification Training: Preparing for the AWS Solutions Architect Associate Exam".**

Test your knowledge with AWS Practice Exams

When you're feeling ready to test your knowledge, check out our AWS Certified Solutions Architect Associate Practice Exams on the Digital Cloud Training website. Our practice tests are designed to reflect the difficulty of the AWS exam and are the closest to the real exam experience available.

To accommodate different learning styles, there are multiple practice types available. Combined, these are powerful tools to prepare you for your exam.

1. **Exam simulation mode** where you get the full timed, scored, exam experience.
2. **Training mode** where you can check every answer as you go through the exam.
3. **Knowledge reviews** - once you've identified your strengths and weaknesses using the practice exams, knowledge reviews allow you to view questions from a specific knowledge area so you can focus your time where you need to most.

There are currently 6 practice exams of 65 questions each (390 questions), and a larger pool of questions available in the Knowledge Reviews (over 500 in total).

Click here to fast-track your AWS exam success!

Contact, Support & Sharing

We hope you get great value from these resources. If for any reason you are not 100% satisfied, please send us your feedback to feedback@digitalcloud.training.

Our private Facebook group is a great place to ask questions and share knowledge and exam tips with the AWS community. Please join the discussion by clicking the link below:

https://www.facebook.com/groups/awscertificationqa

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 so please post your exam feedback to our Facebook group.

For technical support, contact us at: support@digitalcloud.training.

TABLE OF CONTENTS

COMPUTE

Amazon EC2

General

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers

You are limited to running up to a total of 20 On-Demand instances across the instance family, purchasing 20 Reserved Instances, and requesting Spot Instances per your dynamic spot limit per region (by default)

Amazon EC2 currently supports a variety of operating systems including: Amazon Linux, Ubuntu, Windows Server, Red Hat Enterprise Linux, SUSE Linux Enterprise Server, Fedora, Debian, CentOS, Gentoo Linux, Oracle Linux, and FreeBSD

EC2 compute units (ECU) provide the relative measure of the integer processing power of an Amazon EC2 instance

With EC2 you have full control at the operating system layer

Key pairs are used to securely connect to 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 into your 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 Instance Metadata Query tool allows you to query the instance metadata without having to type out the full URI or category names

Billing and provisioning

On demand:

- Pay for hours used with no commitment
- Low cost and flexibility with no upfront cost
- Ideal for auto scaling groups and unpredictable workloads
- Good for dev/test

Spot:

- Excess EC2 capacity that AWS tries to sell on a market exchange basis
- Very low hourly compute cost
- Ideal for grid computing and HPC
- Flexible start and end times
- Customer creates a Spot Request and specifies AMI, desired instance types, and other key information
- Customer defines highest price they're willing to pay for an instance. If capacity is constrained and others are willing to pay more, your instance might get terminated or stopped
- Charged by the hour unless AWS terminates in which case the hour is free
- Spot request can be a "fill and kill", "maintain", or "duration-based"
- For "One-Time Request", instance is terminated, and ephemeral data is lost
- For "Request and Maintain", instance can be configured to Terminate, Stop, or Hibernate until the price point can be met again
- Cannot use encrypted volumes

Reserved:

- Purchase (or agree to purchase) usage of EC2 instances in advance for significant discounts over On-Demand pricing
- Provides a capacity reservation when used in a specific AZ
- AWS Billing automatically applies discounted rates when you launch an instance that matches your purchased RI
- Capacity is reserved for a term of 1 or 3 years
- EC2 has three RI types: Standard, Convertible, and Scheduled
- Standard = commitment of 1 or 3 years, charged whether it's on or off
- Scheduled = reserved for specific periods of time, accrue charges hourly, billed in monthly increments over the term (1 year)
- Scheduled RIs match your capacity reservation to a predictable recurring schedule
- For the differences between standard and convertible RIs, see the table below
- RIs are used for steady state workloads and predictable usage
- Ideal for applications that need reserved capacity
- Upfront payments can reduce the hourly rate
- Can switch AZ within the same region
- Can change the instance size within the same instance type
- Instance type modifications are supported for Linux only
- Cannot change the instance size of Windows RIs
- Billed whether running or not
- Can sell reservations on the AWS marketplace
- Can be used in Auto Scaling Groups
- Can be used in Placement Groups
- Can be shared across multiple accounts within Consolidated Billing
- If you don't need your RI's, you can try to sell them on the Reserved Instance Marketplace

	Standard	Convertible
Terms	1 year, 3 year	1 year, 3 year
Average discount off On-Demand price	40% - 60%	31% - 54%
Change AZ, instance size, networking type	Yes via ModifyReservedInstance API or console	Yes via ExchangeReservedInstance API or console
Change instance family, OS, tenancy, payment options	No	Yes
Benefit from price reductions	No	Yes

RI Attributes:

- Instance type – designates CPU, memory, networking capability
- Platform – Linux, SUSE Linux, RHEL, Microsoft Windows, Microsoft SQL Server
- Tenancy – Default (shared) tenancy, or Dedicated tenancy
- Availability Zone (optional) – if AZ is selected, RI is reserved, and discount applies to that AZ (Zonal RI). If no AZ is specified, no reservation is created but the discount is applied to any instance in the family in any AZ in the region (Regional RI)

Dedicated hosts:

- Physical servers dedicated just for your use
- You then have control over which instances are deployed on that host
- Available as On-Demand or with Dedicated Host Reservation
- Useful if you have server-bound software licences that use metrics like per-core, per-socket, or per-VM
- Each dedicated host can only run one EC2 instance size and type
- Good for regulatory compliance or licensing requirements
- Predictable performance
- Complete isolation
- Most expensive option
- Billing is per host

Dedicated instances:

- Virtualized instances on hardware just for you
- Also uses physically dedicated EC2 servers

- Does not provide the additional visibility and controls of dedicated hosts (e.g. how instances are placed on a server)
- Billing is per instance
- May share hardware with other non-dedicated instances in the same account
- Available as On-Demand, Reserved Instances, and Spot Instances
- Cost additional $2 per hour per region

The following table describes some of the differences between dedicates instances and dedicated hosts:

Characteristic	Dedicated Instances	Dedicated Hosts
Enables the use of dedicated physical servers	X	X
Per instance billing (subject to a $2 per region fee)	X	
Per host billing		X
Visibility of sockets, cores, host ID		X
Affinity between a host and instance		X
Targeted instance placement		X
Automatic instance placement	X	X
Add capacity using an allocation request		X

Partial instance-hours consumed are billed based on instance usage

Instances are billed when they're in a running state - need to stop or terminate to avoid paying

Charging by the hour or second (by the second with Linux instances only)

Data between instances in different regions is charged (in and out)

Regional Data Transfer rates apply if at least one of the following is true, but are only charged once for a given instance even if both are true:

- The other instance is in a different Availability Zone, regardless of which type of address is used
- Public or Elastic IP addresses are used, regardless of which Availability Zone the other instance is in

Instance types

Family	Hint	Purpose/Design
D	DATA	Heavy data usage (e.g. file servers, DWs)
R	RAM	Memory optimized
M	MAIN	General purpose (e.g. app servers)
C	COMPUTE	Compute optimized
G	GRAPHICS	Graphics intensive workloads
I	IOPS	Storage I/O optimised (e.g. NoSQL, DWs)
F	FAST	FPGA hardware acceleration for applications
T	CHEAP (think T2)	Lowest cost (e.g. T2-micro)
P	GPU	GPU requirements
X	EXTREME RAM	Heavy memory usage (e.g. SAP HANA, Apache Spark)
U	HIGH MEMORY	High memory and bare metal performance - use for in memory DBs including SAP HANA
Z	HGH COMPUTE & MEMORY	Fast CPU, high memory and NVMe-based SSDs - use when high overall performance is required
H	HIGH DISK THROUGHPUT	Up to 16 TB of HDD-based local storage

Creating Instances

Option to request a spot instance and specify the maximum bid price

Choose whether to auto-assign a public IP - default is to use the subnet setting

Can add an instance to a placement group

Instances can be assigned to IAM roles which configures them with credentials to access AWS resources

Termination protection can be enabled and prevents you from terminating an instance

Basic monitoring is enabled by default (5-minute periods), detailed monitoring can be enabled (1-minute periods, chargeable)

Can define shared or dedicated tenancy

T2 unlimited allows applications to burst past CPU performance baselines as required (chargeable)

Can add a script to run on start-up (user data)

Can join to a directory (Windows instances only)

There is an option to enable an Elastic GPU (Windows instances only)

Storage options include adding additional volumes and choosing the volume type

Non-root volumes can be encrypted

Root volumes can be encrypted if the instance is launched from an encrypted AMI

There is an option to create tags (or can be done later)

You can select an existing security group or create a new one

You must create or use an existing key pair - this is required

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

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 regional. You can only launch an AMI from the region in which it is stored. However, you can copy AMI's to other regions using the console, command line, or the API

Networking

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

Elastic IPs are retained when the instance is stopped

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

An Elastic IP address is for use in a specific region only

You can assign custom tags to your Elastic IP addresses to categorize them

By default, EC2 instances come with a private IP

Public IP addresses are assigned for instances in public subnets (VPC)

Public IP addresses are always assigned for instances in EC2-Classic

DNS records for elastic IP's can be configured by filling out a form

Secondary IP addresses can be useful for hosting multiple websites on a server or redirecting traffic to a standby EC2 instance for HA

You can choose whether secondary IP addresses can be reassigned

You can associate a single private IPv4 address with a single Elastic IP address and vice versa

When reassigned the IPv4 to Elastic IP association is maintained

When a secondary private address is unassigned from an interface, the associated Elastic IP address is disassociated

You can assign or remove IP addresses from EC2 instances while they are running or stopped

All IP addresses (IPv4 and IPv6) remain attached to the network interface when detached or reassigned to another instance

You can attach a network interface to an instance in a different subnet as long as it's within the same AZ

You cannot team by adding ENIs to an instance

Eth0 is the primary network interface and cannot be moved or detached

By default, Eth0 is the only Elastic Network Interface (ENI) created with an EC2 instance when launched

You can add additional interfaces to EC2 instances (number dependent on instances family/type)

An ENI is bound to an AZ and you can specify which subnet/AZ you want the ENI to be added in

You can specify which IP address within the subnet to configure or leave it be auto-assigned

You can only add one extra ENI when launching but more can be attached later

ENIs can be "hot attached" to running instances

ENIs can be "warm-attached" when the instance is stopped

ENIs can be "cold-attached" when the instance is launched

If you add a second interface AWS will not assign a public IP address to eth0 (you would need to add an Elastic IP)

Default interfaces are terminated with instance termination

Manually added interfaces are not terminated by default

You can change the termination behaviour

An ENI can have:

- One primary IPv4 address
- One or more secondary IPv4 addresses
- One Elastic IP address corresponding to each IPv4 address (via NAT)
- One public IPv4 address
- One or more IPv6 addresses
- Up to 5 security groups

Enhanced Networking

Enhanced networking provides higher bandwidth, higher packet-per-second (PPS) performance, and consistently lower inter-instance latencies

If your packets-per-second rate appears to have reached its ceiling, you should consider moving to enhanced networking because you have likely reached the upper thresholds of the VIF driver

AWS currently supports enhanced networking capabilities using SR-IOV

SR-IOV provides direct access to network adapters, provides higher performance (packets-per-second) and lower latency

Must launch an HVM AMI with the appropriate drivers

Only available for certain instance types

Only supported in VPC

Networking Limits (per region or as specified)

Name	Default Limit
EC2-Classic Elastic IPs	5
EC2-VPC Elastic IPs	5
VPCs	5
Subnets per VPC	200
Security groups per VPC	500
Rules per VPC security group	50
VPC security groups per elastic network interface	5
Network interfaces	350
Network ACLs per VPC	200
Rules per network ACL	20
Route tables per VPC	200
Entries per route table	50
Active VPC peering connections	50
Outstanding VPC peering connection requests	25
Expiry time for an unaccepted VPC peering connection	168

Placement Groups

Placement groups are a logical grouping of instances in one of the following configurations:

Cluster—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

- 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

Spread—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

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

	Clustered	Spread
What	Instances are placed into a low-latency group within a single AZ	Instances are spread across underlying hardware
When	Need low network latency and/or high network throughput	Reduce the risk of simultaneous instance failure if underlying hardware fails
Pros	Get the most out of enhanced networking Instances	Can span multiple AZs
Cons	Finite capacity: recommend launching all you might need up front	Maximum of 7 instances running per group, per AZ

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

Recommended for applications that benefit from low latency and high bandwidth

Recommended to use an instance type that supports enhanced networking

Instances within a placement group can communicate with each other using private or public IP addresses

Best performance is achieved when using private IP addresses

Using public IP addresses, the performance is limited to 5Gbps or less

There is a maximum of 7 running instances per spread placement group within an AZ

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

Low-latency 10 Gbps or 25 Gbps network

Names must be unique within the account for the region

Limited instance types can be used

Recommended to keep instance types homogenous within a placement group

Need to provision the number of instances you need at one time

Cannot merge placement groups

Cannot move existing instances into a placement group

For existing instances, create an AMI then launch from the AMI into the placement group

Placement groups can work across VPC peers but not across AZs (and you won't get full bandwidth)

Can use reserved instances at an instance level but cannot reserve capacity for the placement group

IAM Roles

IAM roles are more secure than storing access keys and secret access keys on EC2 instances

IAM roles are easier to manage

You can attach an IAM role to an instance at launch time or at any time after by using the AWS CLI, SDK, or the EC2 console

IAM roles can be attached, modified, or replaced at any time

Only one IAM role can be attached to an EC2 instance at a time

IAM roles are universal and can be used in any region

Bastion/Jump Hosts

You can configure EC2 instances as bastion hosts (aka jump boxes) in order to access your VPC instances for management

Can use the SSH or RDP protocols

Need to configure a security group with the relevant permissions

Can use auto-assigned public IPs or Elastic IPs

Can use security groups to restrict the IP addresses/CIDRs that can access the bastion host

Use auto-scaling groups for HA (set to 1 to just replace)

Best practice is to deploy Linux bastion hosts in two AZs, use auto-scaling and Elastic IP

EC2 Migration

VM Import/Export is a tool for migrating VMware, Microsoft, XEN VMs to the Cloud

Can also be used to convert EC2 instances to VMware, Microsoft or XEN VMs

Supported for:

- Windows and Linux
- VMware ESX VMDKs and (OVA images for export only)
- Citrix XEN VHD
- Microsoft Hyper-V VHD

Can only be used via the API or CLI (not the console)

Stop the VM before generating VMDK or VHD images

AWS has a VM connector plugin for vCenter:

- Allows migration of VMs to S3
- Then converts into an EC2 AMI
- Progress can be tracked in vCenter

Monitoring

EC2 status checks are performed every minute and each returns a pass or a fail status

If all checks pass, the overall status of the instance is **OK**

If one or more checks fail, the overall status is **impaired**

System status checks detect (StatusCheckFailed_System) problems with your instance that require **AWS** involvement to repair

Instance status checks (StatusCheckFailed_Instance) detect problems that require **your** involvement to repair

Status checks are built into Amazon EC2, so they cannot be disabled or deleted

You can, however create or delete alarms that are triggered based on the result of the status checks

You can create Amazon CloudWatch alarms that monitor Amazon EC2 instances and automatically perform an action if the status check fails

Actions can include:

- Recover the instance (only supported on specific instance types and can be used only with StatusCheckFailed_System)
- Stop the instance (only applicable to EBS-backed volumes)
- Terminate the instance (cannot terminate if termination protection is enabled)
- Reboot the instance

It is a best practice to use EC2 to reboot instance rather than the OS (create a CloudWatch record)

CloudWatch Monitoring frequency:

- Standard monitoring = 5 mins
- Detailed monitoring = 1 min (chargeable)

Tags

A tag is a label that you assign to an AWS resource

Used to manage AWS assets

Tags are just arbitrary name/value pairs that you can assign to virtually all AWS assets to serve as metadata

Each tag consists of a key and an optional value, both of which you define

Tagging strategies can be used for cost allocation, security, automation, and many other uses. For example, you can use a tag in an IAM policy to implement access control

Enforcing standardized tagging can be done via AWS Config rules or custom scripts. For example, EC2 instances not properly tagged are stopped or terminated daily

Most resources can have up to 50 tag

Resource Groups

Resource groups are mappings of AWS assets defined by tags

Create custom consoles to consolidate metrics, alarms and config details around given tags

High Availability Approaches for Compute

Up-to-date AMIs are critical for rapid fail-over

AMIs can be copied to other regions for safety or DR staging

Horizontally scalable architectures are preferred because risk can be spread across multiple smaller machines versus one large machine

Reserved instances are the only way to guarantee that resources will be available when needed

Auto Scaling and Elastic Load Balancing work together to provide automated recovery by maintaining minimum instances

Route 53 health checks also provide “self-healing” redirection of traffic

Migration

AWS Server Migration Service (SMS) is an agent-less service which makes it easier and faster for you to migrate thousands of on-premises workloads to AWS

AWS SMS allows you to automate, schedule, and track incremental replications of live server volumes, making it easier for you to coordinate large-scale server migrations

Automates migration of on-premises VMware vSphere or Microsoft Hyper-V/SCVMM virtual machines to AWS

Replicates VMs to AWS, syncing volumes and creating periodic AMIs

Minimizes cutover downtime by syncing VMs incrementally

Supports Windows and Linux VMs only (just like AWS)

The Server Migration Connector is downloaded as a virtual appliance into your on-premises vSphere or Hyper-V environments

Amazon EBS

General

EBS is the Elastic Block Store

EBS 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

You cannot attach an EBS volume to multiple instances (use Elastic File Store instead)

EBS volume data is replicated across multiple servers in an AZ

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

EBS is designed for an annual failure rate of 0.1%-0.2% & an SLA of 99.95%

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

You can now create AMIs with encrypted root/boot volumes as well as data volumes (you can also use separate CMKs per volume)

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)

Auto-enable IO setting prevents the stopping of IO to a disk when AWS detects inconsistencies

The root device is created under /dev/sda1 or /dev/xvda

Magnetic EBS is for workloads that need throughput rather than IOPS

Throughput optimized EBS volumes cannot be a boot volume

Each instance that you launch has an associated root device volume, either an Amazon EBS volume or an instance store volume

You can use block device mapping to specify additional EBS volumes or instance store volumes to attach to an instance when it's launched

You can also attach additional EBS volumes to a running instance

You cannot decrease an EBS volume size

When changing volumes, the new volume must be at least the size of the current volume's snapshot

Images can be made public but not if they're encrypted

AMIs can be shared with other accounts

You can have up to 5,000 EBS volumes by default

You can have up to 10,000 snapshots by default

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

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 32,000 IOPS per volume (recent change in December 2018)

SSD, Provisioned IOPS - I01

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

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

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

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

Note: Amazon announced a 60% improvement in performance of General Purpose SSD (gp2) Volumes from 10,000 IOPS to 16,000 IOPS and from 160 MB/s to 250 MB/s of throughput per volume in December 2018, this is reflected in the table below:

	Solid State Drives (SSD)		Hard Disk Drives (HDD)	
Volume Type	EBS Provisioned IOPS SSD (io1)	EBS General Purpose SSD (gp2)	Throughput Optimized HDD (st1)	Cold HDD (sc1)
Short Description	Highest performance SSD volume designed for latency-sensitive transactional workloads	General Purpose SSD volume that balances price performance for a wide variety of transactional workloads	Low cost HDD volume designed for frequently accessed, throughput intensive workloads	Lowest cost HDD volume designed for less frequently accessed workloads
Use Cases	I/O-Intensive NoSQL and relational databases	Boot volumes, low-latency interactive apps, dev & test	Big data, data warehouses, log processing	Colder data requiring fewer scans per day
API Name	io1	gp2	st1	sc1
Volume Size	4GB - 16TB	1 GB - 16 TB	500 GB - 16 TB	500 GB - 16 TB
Max IOPS/Volume	64,000	16,000	500	250
Max Throughput/Volume	1,000 MB/s	250 MB/s	500 MB/s	250 MB/s
Max IOPS/Instance	80,000	80,000	80,000	80,000
Max Throughput/Instance	1,750 MB/s	1,750 MB/s	1,750 MB/s	1,750 MB/s

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 convert an unencrypted volume to an encrypted volume

Snapshots are stored on S3

Does not provide granular backup (not a replacement for backup software)

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

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 snapshot 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

To lower storage costs on S3 a full snapshot and subsequent incremental updates can be created

You are charged for data traffic to S3 and storage costs on S3

You are billed only for the changed blocks

Deleting a snapshot removes only the data not needed by any other snapshot

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

Encryption

All EBS types support encryption

All instance families now support encryption

Not all instance types support encryption

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

Data in transit between an instance and an encrypted volume is also encrypted

There is no direct way to change the encryption state of a volume

Either create an encrypted volume and copy data to it or take a snapshot, encrypt it, and create a new encrypted volume from the snapshot

To encrypt a volume or snapshot you need an encryption key, these are customer managed keys (CMK) and they are managed by the AWS Key Management Service (KMS)

A default CMK key is generated for the first encrypted volumes

Subsequent encrypted volumes will use their own unique key (AES 256 bit)

The CMK used to encrypt a volume is used by any snapshots and volumes created from snapshots

You cannot share encrypted volumes created using a default CMK key

You cannot change the CMK key that is used to encrypt a volume

You must create a copy of the snapshot and change encryption keys as part of the copy

This is required in order to be able to share the encrypted volume

By default, only the account owner can create volumes from snapshots

You can share unencrypted snapshots with the AWS community by making them public

You can also share unencrypted snapshots with other AWS accounts by making them private and selecting the accounts to share them with

You cannot make encrypted snapshots public

You can share encrypted snapshots with other AWS accounts using a non-default CMK key and configuring cross-account permissions to give the account access to the key, mark as private and configure the account to share with

The receiving account must copy the snapshot before they can then create volumes from the snapshot

It is recommended that the receiving account re-encrypt the shared and encrypted snapshot using their own CMK key

The following information applies to snapshots:

- Snapshots are created asynchronously and are incremental
- You can copy unencrypted snapshots (optionally encrypt)
- You can copy an encrypted snapshot (optionally re-encrypt with a different key)
- Snapshot copies receive a new unique ID
- You can copy within or between regions
- You cannot move snapshots, only copy them
- You cannot take a copy of a snapshot when it is in a "pending" state, it must be "complete"

- S3 Server-Side Encryption (SSE) protects data in transit while copying
- User defined tags are not copied
- You can have up to 5 snapshot copy requests running in a single destination per account
- You can copy Import/Export service, AWS Marketplace, and AWS Storage Gateway snapshots
- If you try to copy an encrypted snapshot without having access to the encryption keys it will fail silently (cross-account permissions are required)

Copying snapshots may be required for:

- Creating services in other regions
- DR - the ability to restore from snapshot in another region
- Migration to another region
- Applying encryption
- Data retention

To take application-consistent snapshots of RAID arrays:

- Stop the application from writing to disk
- Flush all caches to the disk
- Freeze the filesystem
- Unmount the RAID array
- Shut down the associated EC2 instance

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

Instance store-backed:

- Launch an EC2 instance from an AWS instance store-backed AMI
- Update the root volume as required
- Create the AMI which will upload to a user-specified S3 bucket (user bucket)
- Register the AMI with EC2 (creates another EC2 controlled S3 image)
- To make changes update the source then deregister and reregister
- Upon launch the image is copied to the EC2 host

- Deregister an image when the AMI is not needed anymore (does not affect existing instances created from the AMI)
- Instance store-backed volumes can only be created at launch time

EBS-backed:

- Must stop the instance to create a consistent image and then create the AMI
- AWS registers the AMIs manually
- During creation AWS creates snapshots of all attached volumes - there is no need to specify a bucket, but you will be charged for storage on S3
- You cannot delete the snapshot of the root volume as long as the AMI is registered (deregister and delete)
- You can now create AMIs with encrypted root/boot volumes as well as data volumes (can also use separate CMKs per volume)

Copying AMIs:

- You can copy an Amazon Machine Image (AMI) within or across an AWS region using the AWS Management Console, the 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

RAID

RAID can be used to increase IOPS

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

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

RAID 10 = 10 combination of RAID 1 and 2 resulting in increased performance and redundancy (at the cost of additional disks)

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

EBS Limits (per region)

Name	Default Limit
Provisioned IOPS	300,000
Provisioned IOPS (SSD) volume storage (TiB)	300
General Purpose (SSD) volume storage (TiB)	300
Magnetic volume storage (TiB)	300
Max Cold HDD (SC1) Storage in (TiB)	300
Max Throughput Optimized HDD (ST1) Storage (TiB)	300

Elastic Load Balancing

General ELB Concepts

Elastic Load Balancing automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, and IP addresses

There are three types of Elastic Load Balancer (ELB) on AWS:

- Classic Load Balancer (CLB) – this is the oldest of the three and provides basic load balancing at both layer 4 and layer 7
- Application Load Balancer (ALB) – layer 7 load balancer that routes connections based on the content of the request
- Network Load Balancer (NLB) – layer 4 load balancer that routes connections based on IP protocol data

Note: The Classic Load Balancer may be phased out over time and Amazon are promoting the ALB and NLB for most use cases within VPC

The following table provides a feature comparison:

Feature	Application Load Balancer	Network Load Balancer	Classic Load Balancer
Protocols	HTTP, HTTPS	TCP	TCP, SSL, HTTP, HTTPS
Platforms	VPC	VPC	EC2-Classic, VPC
Health Checks	✓	✓	✓
CloudWatch Metrics	✓	✓	✓
Logging	✓	✓	✓
Zonal fail-over	✓	✓	✓
Connection draining	✓	✓	✓
Load balancing to multiple ports on an instance	✓	✓	
WebSockets	✓	✓	
IP addresses as targets	✓	✓	
Lambda functions as targets	✓		
Load balancer deletion protection	✓	✓	
Path-based routing	✓		
Host-based routing	✓		
HTTP header-based routing	✓		
HTTP method-based routing	✓		
Query string parameter-based routing	✓		
Source IP address CIDR-based routing	✓		
Native HTTP/2	✓		
Configurable idle connection timeout	✓		✓

Feature	Application Load Balancer	Network Load Balancer	Classic Load Balancer
Cross-zone load balancing	✓	✓	✓
SSL offloading	✓	✓	✓
Server Name Indication (SNI)	✓		
Sticky sessions	✓		✓
Back-end server encryption	✓	✓	✓
Static IP		✓	
Elastic IP address		✓	
Preserve source IP address		✓	
Resource-based IAM permissions	✓	✓	✓
Tag-based IAM permissions	✓	✓	
Slow start	✓		
User authentication	✓		
Redirects	✓		
Fixed response	✓		
Custom security policies			✓

Elastic Load Balancing provides fault tolerance for applications by automatically balancing traffic across targets – Amazon EC2 instances, containers and IP addresses – and Availability Zones while ensuring only healthy targets receive traffic

An ELB can distribute incoming traffic across your Amazon EC2 instances in a single Availability Zone or multiple Availability Zones

Only 1 subnet per AZ can be enabled for each ELB

Route 53 can be used for region load balancing with ELB instances configured in each region

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

Internal-only load balancers do not need an Internet gateway

EC2 instances and containers can be registered against an ELB

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

An ELB forwards traffic to eth0 (primary IP address)

An ELB listener is the process that checks for connection requests

Listeners for CLB provide options for TCP and HTTP/HTTPS

Listeners for ALB only provide options for HTTP and HTTPS

Listeners for NLB only provide TCP as an option

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

For ALB at least 2 subnets must be specified

For NLB only one subnet must be specified (recommended to add at least 2)

For CLB you don't need to specify any subnets unless you have "Enable advanced VPC configuration" enabled in which case you must specify two

ELB uses a DNS record TTL of 60 seconds to ensure new ELB node IP addresses are used to service clients

By default, the ELB has an idle connection timeout of 60 seconds, set the idle timeout for applications to at least 60 seconds

Perfect Forward Secrecy (PFS) provides additional safeguards against the eavesdropping of encrypted data, through the use of a unique random session key

Server Order Preference lets you configure the load balancer to enforce cipher ordering, providing more control over the level of security used by clients to connect with your load balancer

ELB does not support client certificate authentication (API Gateway does support this)

ELB Security Groups

Security groups control the ports and protocols that can reach the front-end listener

In non-default VPCs you can choose which security group to assign

You must assign a security group for the ports and protocols on the front-end listener

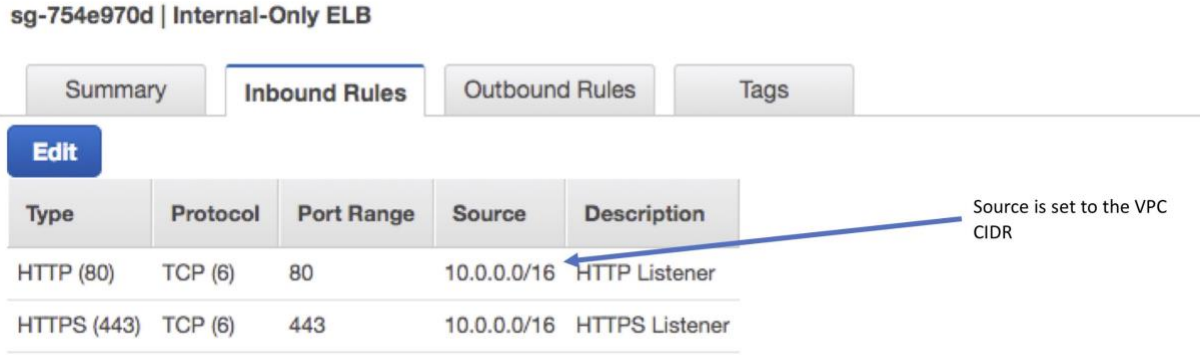
You need to also allow the ports and protocols for the health check ports and back-end listeners

Security group configuration for ELB:

Inbound to ELB (allow)

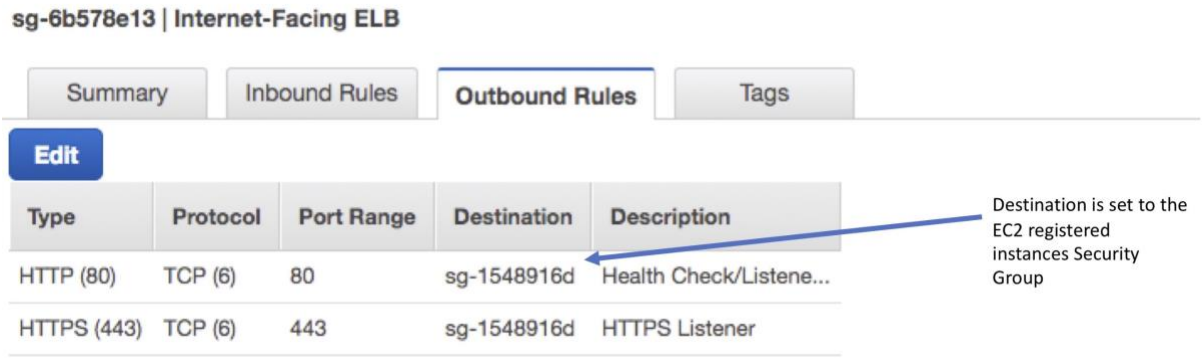
- Internet-facing ELB:
  - Source: 0.0.0.0/0
  - Protocol: TCP
  - Port: ELB listener ports

- Internal-only ELB:
  - Source: VPC CIDR
  - Protocol: TCP
  - Port: ELB Listener ports



Outbound (allow, either type of ELB):

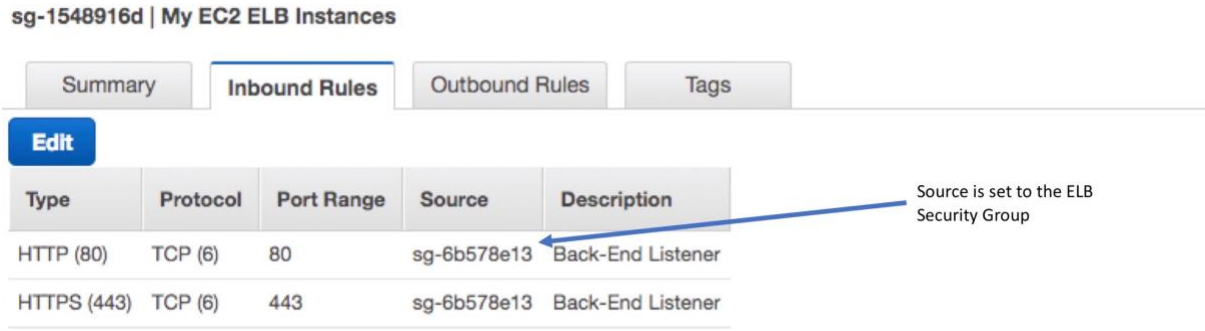
- Destination: EC2 registered instances security group
- Protocol: TCP
- Port: Health Check/Listener



Security group configuration for registered instances:

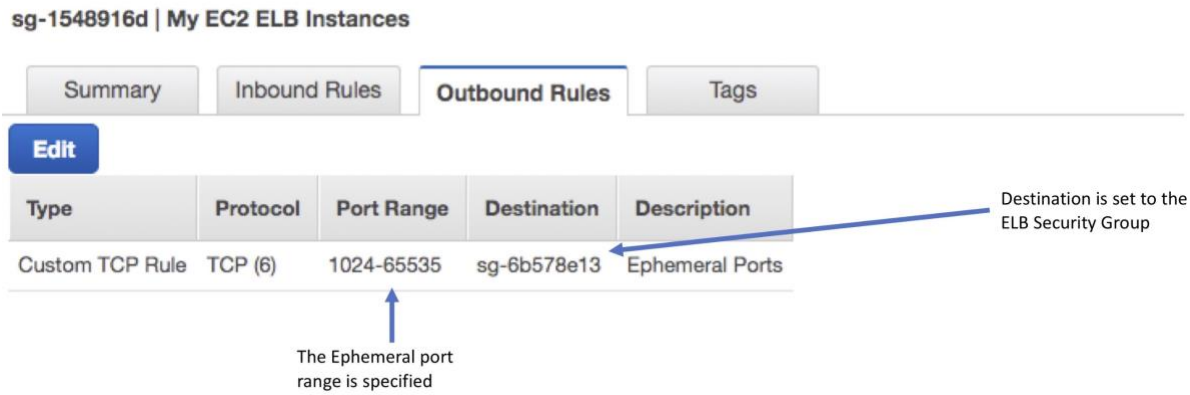
Inbound to registered instances (Allow, either type of ELB)

- Source: ELB Security Group
- Protocol: TCP
- Port: Health Check/Listener



Outbound (Allow, for both types of ELB)

- Destination: ELB Security Group
- Protocol: TCP
- Port: Ephemeral



It is also important to ensure NACL settings are set correctly

Distributed Denial of Service (DDoS) protection:

- ELB automatically distributes incoming application traffic across multiple targets, such as Amazon Elastic Compute Cloud (Amazon EC2) instances, containers, and IP addresses, and multiple Availability Zones, which minimizes the risk of overloading a single resource
- ELB, like CloudFront, only supports valid TCP requests, so DDoS attacks such as UDP and SYN floods are not able to reach EC2 instances
- ELB also offers a single point of management and can serve as a line of defence between the internet and your backend, private EC2 instances

ELB Monitoring

Monitoring takes place using:

- CloudWatch - every 1 minute
  - ELB service only sends information when requests are active
  - Can be used to trigger SNS notifications
- Access Logs
  - Disabled by default
  - Includes information about the clients (not included in CloudWatch metrics)
  - Can identify requester, IP, request type etc.
  - Can be optionally stored and retained in S3
- CloudTrail
  - Can be used to capture API calls to the ELB
  - Can be stored in an S3 bucket

Limits

Name	Default Limit
Application Load Balancers	20
Network Load Balancers	20
Target Groups	3000
Classic Load Balancers	20

Classic Load Balancer (CLB)

The Classic Load Balancer provides basic load balancing across multiple Amazon EC2 instances and operates at both the request level and connection level

Operates at layer 4 and layer 7

Supported protocols: TCP, SSL, HTTP, HTTPS

CLB does not support HTTP/2

Load balancers can listen on the following ports:

- [EC2-VPC] 1-65535
- [EC2-Classic] 25, 80, 443, 465, 587, 1024-65535

CLB's do not have pre-defined IPv4 addresses but are resolved using a DNS name

Does not support Elastic IPs

Supports IPv4 and IPv6

Within a VPC only IPv4 is supported

Provides SSL termination and processing

Sticky Sessions:

- Cookie-based sticky sessions are supported
- Session stickiness uses cookies and ensures a client is bound to an individual back-end instance for the duration of the cookie lifetime
- Cookies can be inserted by the application or by the load balancer when configured
- After cookies expire new requests will be routed by the load balancer normally and a new cookie will be inserted and bind subsequent sessions to the same back-end instance
- With application-inserted cookies if the back-end instance becomes unhealthy, new requests will be routed by the load balancer normally and a new cookie will be inserted and bind subsequent sessions to the same back-end instance
- With CLB-inserted cookies if the back-end instance becomes unhealthy, new requests will be routed by the load balancer normally BUT the session will no longer be sticky

Must have multiple CLBs for multiple SSL certs

Integrates with Auto Scaling, CloudWatch, CloudTrail and Route 53

Instances monitored by CLB are reported as InService or OutofService

Supports domain zone apex records, e.g. example.com

Wildcard certificates are supported

Health checks:

- Can be configured for HTTP, TCP, HTTPS, SSL
- Ping port specifies the port for the health check
- Ping path specifies the path to check, e.g. /index.html
- Can define timeout, interval, unhealthy threshold, healthy threshold

For fault tolerance it is recommended to distribute registered instances across multiple AZs (ideally evenly)

Cross-zone load balancing is enabled by default

When enabled cross-zone load balancing distributes traffic evenly between EC2 instances across AZs

Without cross-zone load balancing CLB sends traffic equally to each AZ configured regardless of the number of hosts in each AZ

Connection draining is enabled by default and provides a period of time for existing connections to close cleanly

When connection draining is in action a CLB will be in the status "InService: Instance deregistration currently in progress"

CLB can take 1 to 7 minutes to detect an increase in load and scale

If you're anticipating a fast increase in load you can contact AWS and instruct them to pre-warm (provision) additional CLB nodes

Listeners:

- A CLB listener is the process that checks for connection requests
- You can configure the protocol/port on which your CLB listener listens
- Front-end listeners check for traffic from clients to the CLB
- Back-end listeners are configured with the protocol/port to check for traffic from the CLB to the EC2 instances
- Front-end and back-end listeners can listen on ports 1-65535
- Front-end and back-end listeners must be at the same layer (e.g. layer 4 or layer 7)
- There is a 1:1 mapping between front-end and back-end listeners
- Up to 100 listeners can be configured
- Supports L4 (TCP, SSL) and L7 (HTTP, HTTPS) listeners

With packet interception the source IP/port will be from the ELB

Proxy protocol for TCP/SSL carries the source (client) IP/port information

The Proxy Protocol header helps you identify the IP address of a client when you have a load balancer that uses TCP for back-end connections

Ensure the client doesn't go through a proxy or there will be multiple proxy headers

Also need to ensure the EC2 instance's TCP stack can process the extra information

X-forwarded-for for HTTP/HTTPS carries the source IP/port information

To use an HTTPS listener the CLB must have an X.509 SSL/TLS server certificate - this will allow the CLB to terminate the secure session from the client to the CLB

The session between the CLB and the EC2 instance can be re-encrypted

You can use a certificate generated by AWS Certificate Manager (ACM) or your own certificate

If you don’t want interception/offloading you can use TCP listeners with certificates on the EC2 instances (traffic is secured end-to-end)

Proxy protocol only applies to L4

X-forwarded-for only applies to L7

To filter by source IP use NACLs for proxy protocol (L4) / X-forwarded-for (L7) headers with the EC2 instance's application performing the filtering

Security

CLB supports a single X.509 certificate

Two-way authentication with client certificates is not supported on the CLB - you would need to pass through the session using the proxy protocol and have an application that supports client-side certificates

When using end-to-end encryption use TCP not SSL/HTTPS on the CLB (does not support Session Stickiness)

AWS ACM certificates include an RSA public key - ensure you include a set of ciphers that support RSA in the security policy

The latest predefined security policy does not include support for SSLv3

When choosing a custom security policy, you can select the ciphers and protocols (only for CLB)

SSL Security Policy includes:

- Protocol Versions (SSL/TLS)
  - Supports TLS 1.0, 1.1, 1.2, SSL 3.0
- SSL Ciphers
  - Encryption algorithms
  - SSL can use different ciphers to encrypt data
- Server Order Preference
  - When enabled the first match in the cipher list with the Client list is used
  - If disabled (default) the first match in the client cipher list with the CLB is used

Application Load Balancer (ALB)

The Application Load Balancer operates at the request level (layer 7), routing traffic to targets - EC2 instances, containers and IP addresses based on the content of the request

You can load balance HTTP/HTTPS applications and use layer 7-specific features, such as X-Forwarded-For headers

Supports HTTPS termination between the clients and the load balancer

Supports management of SSL certificates through AWS IAM and AWS Certificate Manager for pre-defined security policies

Server Name Indication (SNI) supports multiple secure websites using a single secure listener

With Server Name Indication a client indicates the hostname to connect to

IP addresses as targets allows load balancing any application hosted in AWS or on-premises using IP addresses of the application back-ends as targets

Need at least 2 availability zones and you can distribute incoming traffic across your targets in multiple Availability Zones

Automatically scales its request handling capacity in response to incoming application traffic

Can configure an Application Load Balancer to be Internet facing or create a load balancer without public IP addresses to serve as an internal (non-Internet-facing) load balancer

Native IPv6 support

Internal only ALB only supports IPv4

Content-Based Routing allows the routing of requests to a service based on the content of the request:

- Host-based routing - route client requests based on the Host field of the HTTP header allowing you to route to multiple domains from the same load balancer
- Path-based routing - route a client request based on the URL path of the HTTP header (e.g. /images or /orders)

Provides support for micro-services and containers with load balancing across multiple ports on a single EC2 instance

Better performance for real-time streaming

Deletion protection can be enabled

Request tracing (allows you to track a request by its unique ID)

Better health checks and CloudWatch metrics

Integration with Amazon Cognito for user authentication

Uses a round-robin load balancing algorithm

Slow start mode allows targets to "warm up" with a ramp-up period

Health Checks:

- Can have custom response codes in health checks (200-399)
- There are more details provided in the API and management console for health check failures
- Reason codes are returned with failed health checks
- Health checks do not support WebSockets
- Fail open means if no AZ contains a healthy target, the load balancer nodes route requests to all targets

Detailed access log information is provided and saved to an S3 bucket every 5 or 6 minutes

ALB does not support back-end server authentication (CLB does)

ALB does not support EC2-Classic (CLB does)

Deletion protection is possible

Deregistration delay is similar to connection draining

Sticky Sessions:

- Session stickiness uses cookies and ensures a client is bound to an individual back-end instance for the duration of the cookie lifetime
- ALB supports load balancer-generated cookies only
- The name of the cookie is AWSALB
- The contents of these cookies are encrypted using a rotating key
- You cannot decrypt or modify load balancer-generated cookies
- Sticky sessions are enabled at the target group level
- You can also set the duration for the stickiness of the load balancer-generated cookie, in seconds

- WebSockets connections are inherently sticky (following the upgrade process)

Monitoring

CloudTrail can be used to capture API calls. Only pay for the S3 storage charges

CloudTrail records information on API calls only

To monitor other actions such as time the request was received, the client's IP address, request paths etc. use access logs

Access logging is optional and disabled by default

You are only charged for the S3 storage

ALB logs requests sent to the load balancer including requests that never made it to targets

ALB does not log health check requests

Logging of requests is best effort so shouldn't be relied on for auditing

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, or IP addresses

Target groups can exist independently from the ALB

Target 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 or IP address (must be a valid private IP from an existing subnet)

You cannot use public IP addresses as targets

You cannot use instance IDs and IP address targets within the same target group

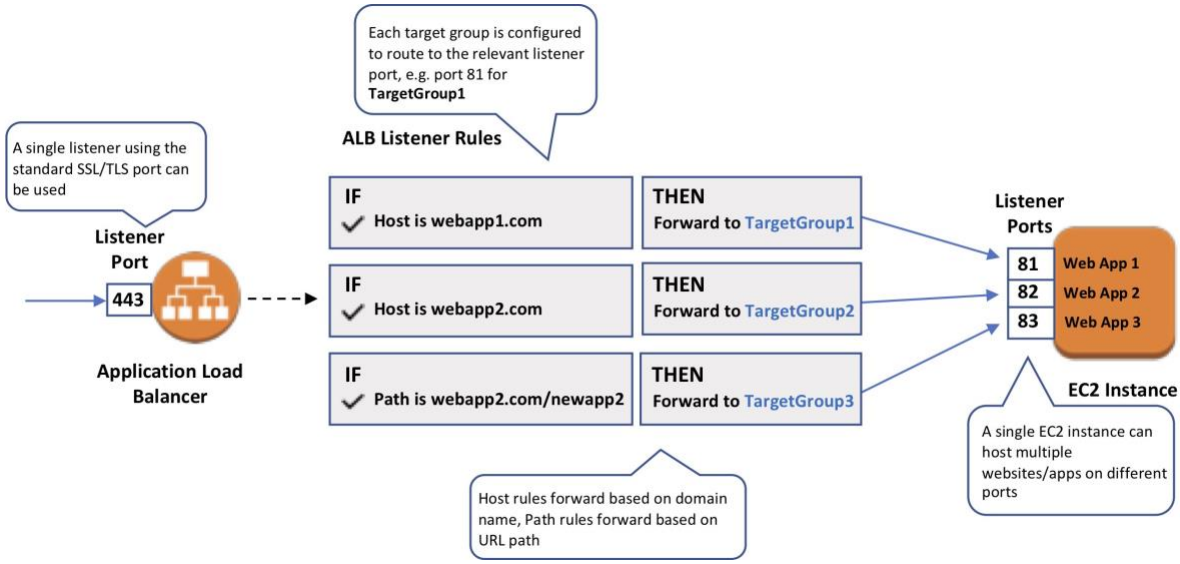
A target group can only be associated with one load balancer

The following diagram illustrates the basic components. Notice that each listener contains a default rule, and one listener contains another rule that routes requests to a different target group. One target is registered with two target groups

Target groups are used for registering instances against an ALB or NLB

Target groups are a regional construct

The following diagram shows how target groups can be used with host-based and target-based routing to route traffic to multiple websites, running on multiple ports, on a single EC2 instance:



The following attributes can be defined:

- Deregistration delay - the amount of time for Elastic Load Balancing to wait before deregistering a target
- Slow start duration - the time period, in seconds, during which the load balancer sends a newly registered target a linearly increasing share of the traffic to the target group
- Stickiness - indicates whether sticky sessions are enabled

The default settings for attributes are:

- Deregistration delay = 300 seconds
- Slow start duration = 0 seconds
- Stickiness = Not enabled

Auto Scaling groups can scale each target group individually

You can only use Auto Scaling with the load balancer if using instance IDs in your target group

Health checks are defined per target group

ALB can route to multiple target groups

You can register the same EC2 instance or IP address with the same target group multiple times using different ports (used for routing requests to micro-services)

If you register by instance ID the traffic is routed using the primary private IP address of the primary network interface

If you register by IP address you can route traffic to an instance using any private address from one or more network interfaces

You cannot mix different types within a target group (EC2, ECS, IP)

An EC2 instance can be registered with the same target group multiple times using multiple ports

IP addresses can be used to register:

- Instances in a peered VPC

- AWS resources that are addressable by IP address and port
- On-premises resources linked to AWS through Direct Connect or a VPN connection

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 ALB
- 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

Default rules:

- When you create a listener, you define an action for the default rule
- Default rules cannot have conditions
- You can delete the non-default rules for a listener at any time
- You cannot delete the default rule for a listener
- When you delete a listener all of its rules are deleted
- If no conditions for any of a listener's rules are met, the action for the default rule is taken

Rule priority:

- Each rule has a priority and they are evaluated in order of lowest to highest
- The default rule is evaluated last
- You can change the value of a non-default rule at any time
- You cannot change the value of the default rule

Rule action:

- Only one target group per action
- Each rule has a type and a target group

- The only supported action type is forward, which forwards requests to the target group
- You can change the target group for a rule at any time

Rule conditions:

- There are two types of rule condition: host and path
- When the conditions for a rule are met the action is taken
- Each rule can have up to 2 conditions, 1 path condition and 1 host condition
- Optional condition is the path pattern you want the ALB to evaluate in order for it to route requests

Request routing:

- After the load balancer receives a request it evaluates the listener rules in priority order to determine which rule to apply, and then selects a target from the target group for the rule action using the round robin routing algorithm
- Routing is performed independently for each target group even when a target is registered with multiple target groups
- You can configure listener rules to route requests to different target groups based on the content of the application traffic

Content-based routing:

- ALB can route requests based on the content of the request in the host field: host-based or path-based
- Host-based is domain name-based routing e.g. example.com or app1.example.com
- The host field contains the domain name and optionally the port number
- Path-based is URL based routing e.g. example.com/images, example.com/app1
- You can also create rules that combine host-based and path-based routing
- Anything that doesn't match content routing rules will be sent to a default target group

ALB and ECS

ECS service maintains the "desired count" of instances

Optionally a load balancer can distribute traffic across tasks

All containers in a single task definition are placed on a single EC2 container instance

You can put multiple containers in the same task definition behind a CLB

- Define multiple host ports in the service definition
- Define these listener ports as listeners on the CLB

ECS service can only use a single load balancer

If your task definition requires multiple ports per container you must use a CLB with multiple listeners

ALB cannot do multiple listeners on a single task definition

AWS does not recommend connecting multiple services to the same CLB

ALB allows containers to use dynamic host port mapping so that multiple tasks from the same service are allowed on the same container host

ALB supports path-based routing and priority rules

ALB integrates with EC2 container service using service load balancing

If a service uses multiple ports, then multiple task definitions will need to be created with multiple target groups

Federated authentication:

- ALB now supports authentication from OIDC compliant identity providers such as Google, Facebook and Amazon
- Implemented through an authentication action on a listener rule that integrates with Amazon Cognito to create user pools
- AWS SAM can also be used with Amazon Cognito

Network Load Balancer

Network Load Balancer operates at the connection level (Layer 4), routing connections to targets - Amazon EC2 instances, containers and IP addresses based on IP protocol data

It is architected to handle millions of requests/sec, sudden volatile traffic patterns and provides extremely low latencies

High throughput - designed to handle traffic as it grows and can load balance millions of requests/second

Extremely low latencies for latency-sensitive applications

Uses static IP addresses - each NLB provides a single IP address for each AZ

Can also assign an Elastic IP to the load balancer per AZ

The IP-per-AZ feature reduces latency with improved performance, improves availability through isolation and fault tolerance and makes the use of NLBs transparent to your client applications

Preserves the source IP of clients, and provides stable IP support and Zonal isolation

Can load balance any application hosted in AWS or on-premises using IP addresses of the application back-ends as targets

NLB supports connections from clients to IP-based targets in peered VPCs across different AWS Regions

Supports both network and application target health checks

Supports long-running/lived connections (ideal for WebSocket applications)

Supports failover between IP addresses within and across regions (uses Route 53 health checks)

Integration with Route 53 enables the removal of a failed load balancer IP address from service and subsequent redirection of traffic to an alternate Network Load Balancer in another region

Does not support SSL termination

Supports WebSockets

Supports cross-zone load balancing

Uses the same API as Application Load Balancer

Also uses Target Groups

CloudWatch reports Network Load Balancer metrics

Enhanced logging - can use the Flow Logs feature to record all requests sent to your load balancer

AWS Auto Scaling

General Auto Scaling Concepts

Amazon EC2 Auto Scaling helps you ensure that you have the correct number of Amazon EC2 instances available to handle the load for your application

You create collections of EC2 instances, called Auto Scaling groups

Automatically provides horizontal scaling (scale-out) for your instances

Triggered by an event of scaling action to either launch or terminate instances

Availability, cost, and system metrics can all factor into scaling

Auto Scaling is a region-specific service

Auto Scaling can span multiple AZs within the same AWS region

Auto Scaling can be configured from the Console, CLI, SDKs and APIs

There is no additional cost for Auto Scaling, you just pay for the resources (EC2 instances) provisioned

Auto Scaling works with ELB, CloudWatch and CloudTrail

You can determine which subnets Auto Scaling will launch new instances into

Auto Scaling will try to distribute EC2 instances evenly across AZs

Launch configuration is the template used to create new EC2 instances and includes parameters such as instance family, instance type, AMI, key pair and security groups

You cannot edit a launch configuration once defined

A launch configuration:

- Can be created from the AWS console or CLI
- You can create a new launch configuration, or
- You can use an existing running EC2 instance to create the launch configuration
    - The AMI must exist on EC2

    - EC2 instance tags and any additional block store volumes created after the instance launch will not be taken into account
- If you want to change your launch configurations you have to create a new one, make the required changes, and use that with your auto scaling groups

You can use a launch configuration with multiple Auto Scaling Groups (ASG)

An ASG is a logical grouping of EC2 instances managed by an Auto Scaling Policy

An ASG can be edited once defined

You can attach one or more classic ELBs to your existing ASG

You can attach one or more Target Groups to your ASG to include instances behind an ALB

The ELBs must be in the same region

Once you do this any EC2 instance existing or added by the ASG will be automatically registered with the ASG defined ELBs

If adding an instance to an ASG would result in exceeding the maximum capacity of the ASG the request will fail

You can add a running instance to an ASG if the following conditions are met:

- The instance is in a running state
- The AMI used to launch the instance still exists
- The instance is not part of another ASG
- The instance is in the same AZs for the ASG

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 type options available and when to use them:

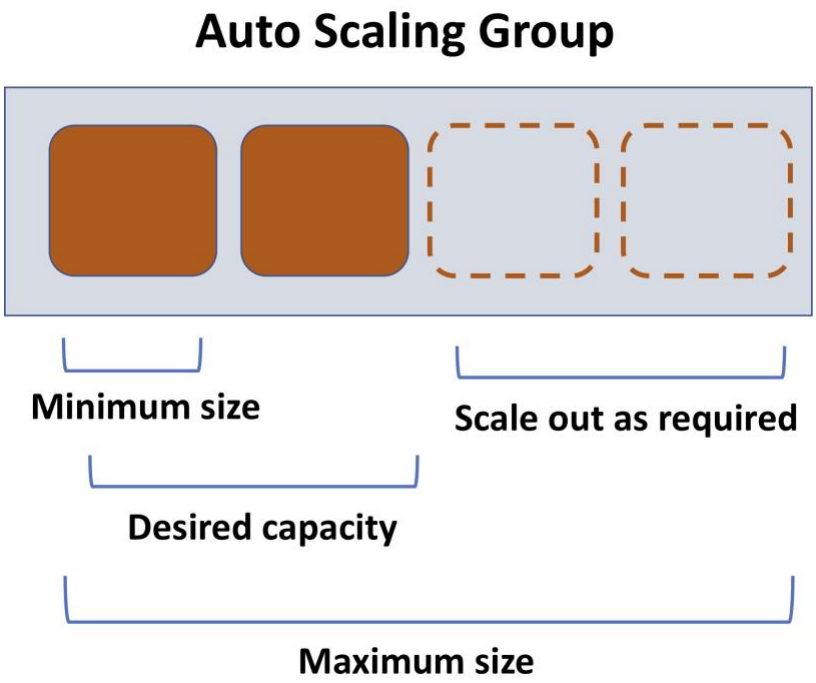
Scaling Type	What it is	When to use
Maintain	Ensures the required number of instances are running	Use when you always need a known number of instances running at all times
Manual	Manually change desired capacity via the console or CLI	Use when your needs change rarely enough that you're OK to make manual 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 policy options are configured through Scaling Policies which determines when, if, and how the ASG scales and shrinks

The following table describes the scaling policy options available and when to use them (and more detail further down the page):

Scaling	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

The diagram below depicts an Auto Scaling group with a Scaling policy set to a minimum size of 1 instance, a desired capacity of 2 instances, and a maximum size of 4 instances:



You can define Instance Protection which stops Auto Scaling from scaling in and terminating the instances

If Auto Scaling fails to launch instances in an AZ it will try other AZs until successful

The default health check grace period is 300 seconds

Scale-out is the process in which EC2 instances are launched by the scaling policy

Scale-in is the process in which EC2 instances are terminated by the scaling policy

It is recommended to create a scale-in event for each scale-out event created

Auto Scaling can perform rebalancing when it finds that the number of instances across AZs is not balanced

Auto Scaling rebalances by launching new EC2 instances in the AZs that have fewer instances first, only then will it start terminating instances in AZs that had more instances

Auto Scaling may go over the maximum number of instances by 10% temporarily for the purposes of rebalancing

An imbalance may occur due to:

- Manually removing AZs/subnets from the configuration
- Manually terminating EC2 instances
- EC2 capacity issues
- Spot price is reached

Health checks:

- By default uses EC2 status checks
- Can also use ELB health checks and custom health checks
- ELB health checks are 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)

If using an ELB it is best 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

Elastic IPs and EBS volumes are detached from terminated instances and will need to be manually reattached

Using custom health checks a CLI command can be issued to set the instance's status to unhealthy, e.g.:

aws autoscaling set-instance-health --instance-id i-123abc45d --health-status Unhealthy

Once in a terminating state an EC2 instance cannot be put back into service again

However, there is a short time period in which a CLI command can be run to change an instance to healthy

Unlike AZ rebalancing, termination of unhealthy instances happens first, then Auto Scaling attempts to launch new instances to replace terminated instances

You can manually remove (detach) instances from an ASG using the AWS Console or CLI

When detaching an instance, you can optionally decrement the ASG's desired capacity (so it doesn't launch another instance)

An instance can be attached to one ASG at a time

You can suspend and then resume one or more of the scaling processes for your Auto Scaling group

Suspending scaling processes can be useful when you want to investigate a configuration problem or other issue with your web application and then make changes to your application, without invoking the scaling processes

You can manually move an instance from an ASG and put it in the standby state

Instances in standby state are still managed by Auto Scaling, are charged as normal, and do not count towards available EC2 instance for workload/application use

Auto scaling does not perform health checks on instances in the standby state

Standby state can be used for performing updates/changes/troubleshooting etc. without health checks being performed or replacement instances being launched

When you delete an ASG the instances will be terminated

You can choose to use Spot instances in launch configurations and specify a bid price

Auto Scaling treats spot instances the same as on-demand instances

You cannot mix Spot instances with on-demand

If you want to change the bid price you need to create a new launch configuration

Auto Scaling can be configured to send an SNS email when:

- An instance is launched
- An instance is terminated
- An instance fails to launch
- An instance fails to terminate

Merging ASGs

- You can merge multiple single AZ Auto Scaling Groups into a single multi-AZ ASG
- Merging can only be performed by using the CLI
- Process is to rezone one of the groups to cover/span the other AZs for the other ASGs
- Then delete the other ASGs
- Can be performed on ASGs with or without ELBs attached to them
- The resulting ASG must be one of the pre-existing ASGs

Cooldown Period:

- The cooldown period is a configurable setting for your Auto Scaling group that helps to ensure that it doesn't launch or terminate additional instances before the previous scaling activity takes effect
- The default cooldown period is applied when you create your Auto Scaling group
- The default value is 300 seconds
- You can configure the default cooldown period when you create the Auto Scaling group, using the AWS Management Console, the create-auto-scaling-group command (AWS CLI), or the CreateAutoScalingGroup API operation
- Automatically applies to dynamic scaling and optionally to manual scaling but not supported for scheduled scaling
- Can override the default cooldown via scaling-specific cooldown

Scaling Policies

An ASG can have multiple policies attached to it at any time

Simple scaling:

- Single adjustment (up or down) in response to an alarm
- Waits for a cooldown timer to expire before responding to more alarms

Scheduled:

- You cannot configure two scheduled activities at the same date/time
- Scheduled actions can be edited from the AWS Console or CLI
- Cooldown timer is not supported for scheduled or step on-demand scaling

Dynamic:

- An alarm is an object that watches over a single metric, e.g. CPU/memory/network utilisation
- You need to have a scale-out and a scale-in policy configured

Step scaling:

- Configure multiple steps/adjustments
- Does not support cool down timers
- Can respond to multiple alarms and initiate multiple scaling activities
- Supports a warm-up timer which is the time it will take a newly launched instance to be ready

The warm-up period is the period of time in which a newly created EC2 instance launched by ASG using step scaling is not considered toward the ASG metrics

For simple or step scaling a scaling adjustment cannot change the capacity of the group above the maximum group size or below the minimum group size

Target scaling is new and attempts to keep the utilization of instances at a target level based on CPU/memory/network etc.

Monitoring

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 CLI detailed monitoring of EC2 instances is enabled by default

When you enable Auto Scaling group metrics, Auto Scaling sends sampled data to CloudWatch every minute

Configure ASG and EC2 monitoring options so they use the same time period, e.g. detailed monitoring (EC2) and 60 seconds (ASG), or basic monitoring (EC2) and 300 seconds (ASG)

Limits

Name	Default Limit
Auto Scaling Groups	200
Launch Configurations	200

Amazon ECS

General ECS Concepts

Amazon Elastic Container Service (ECS) is a highly scalable, high performance container management service that supports Docker containers and allows you to easily run applications on a managed cluster of Amazon EC2 instances

Amazon ECS eliminates the need for you to install, operate, and scale your own cluster management infrastructure

Using API calls you can launch and stop container-enabled applications, query the complete state of clusters, and access many familiar features like security groups, Elastic Load Balancing, EBS volumes and IAM roles

Amazon ECS can be used to schedule the placement of containers across clusters based on resource needs and availability requirements

There is no additional charge for Amazon ECS. You pay for AWS resources (e.g. EC2 instances or EBS volumes) you create to store and run your application

Possible to use Elastic Beanstalk to handle the provisioning of an Amazon ECS cluster, balancing load, auto-scaling, monitoring, and placing your containers across your cluster

Alternatively use ECS directly for more fine-grained control for customer application architectures

It is possible to associate a service on Amazon ECS to an Application Load Balancer (ALB) for the Elastic Load Balancing (ELB) service

The ALB supports a target group that contains a set of instance ports. You can specify a dynamic port in the ECS task definition which gives the container an unused port when it is scheduled on the EC2 instance

ECS provides Blox, a collection of open source projects for container management and orchestration. Blox makes it easy to consume events from Amazon ECS, store the cluster state locally and query the local data store through APIs

You can use any AMI that meets the Amazon ECS AMI specification

ECS vs EKS

Amazon also provide the Elastic Container Service for Kubernetes (Amazon EKS) which can be used to deploy, manage, and scale containerized applications using Kubernetes on AWS

The table below describes some of the differences between these services to help you understand when you might choose one over the other:

Amazon ECS	Amazon EKS
Managed, highly available, highly scalable container platform	
AWS-specific platform that supports Docker containers	Compatible with upstream Kubernetes so it's easy to lift and shift from other Kubernetes deployments
Considered simpler to learn and use	Considered more feature-rich and complex with a steep learning curve
Leverages AWS services like Route 53, ALB, and CloudWatch	A hosted Kubernetes platform that handles many things internally
"Tasks" are instances of containers that are run on underlying compute but more or less isolated	"Pods" are containers collocated with one another and can have shared access to each other
Limited extensibility	Extensible via a wide variety of third-party and community add-ons

Launch Types

An Amazon ECS launch type determines the type of infrastructure on which your tasks and services are hosted

There are two launch types and the table below describes some of the differences between the two launch types:

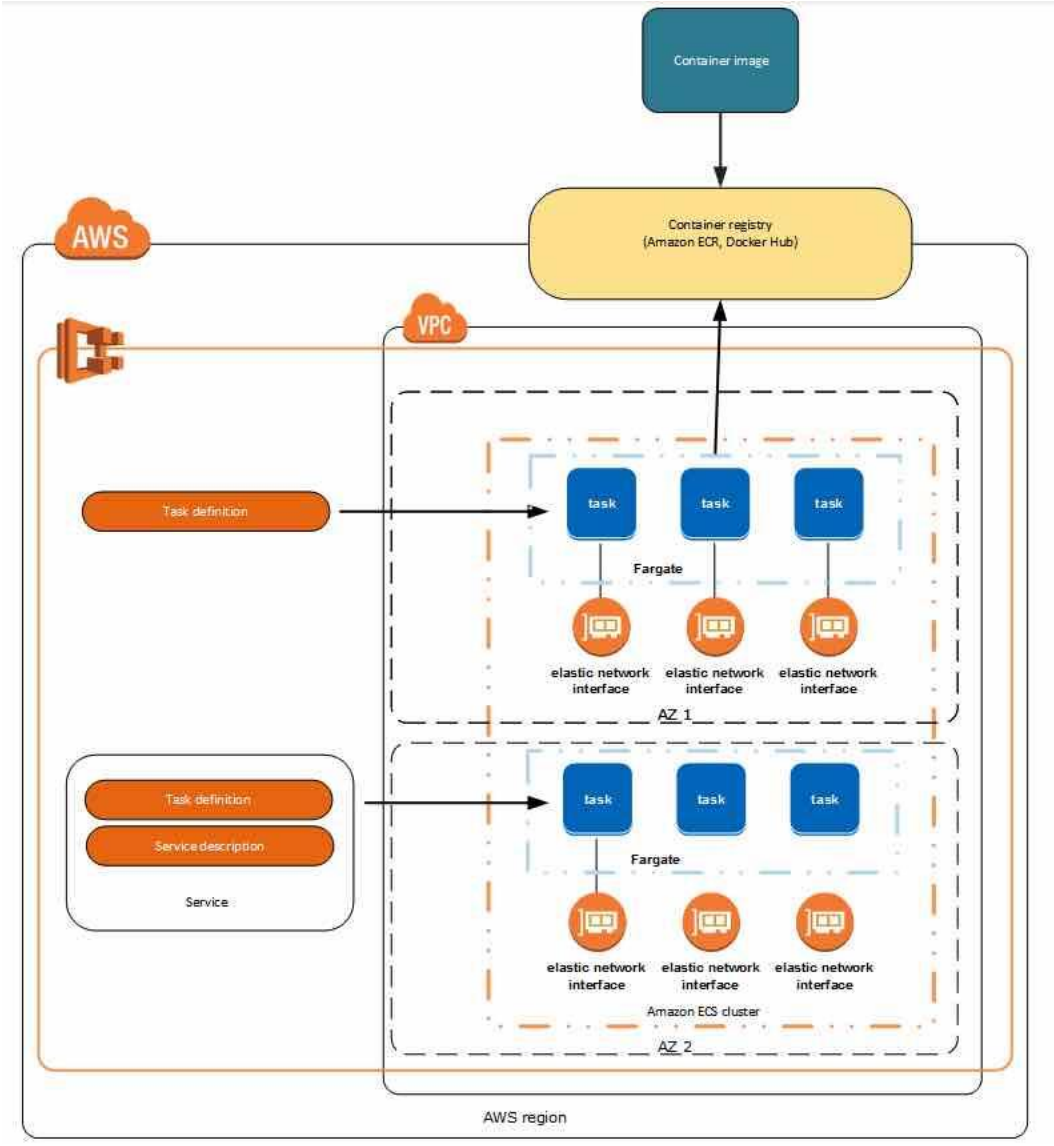
Amazon EC2	Amazon Fargate
You explicitly provision EC2 instances	The control plane asks for resources and Fargate automatically provisions
You're responsible for upgrading, patching, care of EC2 pool	Fargate provisions compute as needed
You must handle cluster optimization	Fargate handles cluster optimization
More granular control over infrastructure	Limited control, as infrastructure is automated

Fargate Launch Type

- The Fargate launch type allows you to run your containerized applications without the need to provision and manage the backend infrastructure. Just register your task definition and Fargate launches the container for you
- Fargate Launch Type is a serverless infrastructure managed by AWS

- Fargate only supports container images hosted on Elastic Container Registry (ECR) or Docker Hub

This diagram shows the general architecture:



EC2 Launch Type

- The EC2 launch type allows you to run your containerized applications on a cluster of Amazon EC2 instances that you manage
- Private repositories are only supported by the EC2 Launch Type

Images

Containers are created from a read-only template called an image which has the instructions for creating a Docker container

Images are built from a Dockerfile

Only Docker containers are currently supported

An image contains the instructions for creating a Docker container

Images are stored in a registry such as DockerHub or AWS ECR

The elastic container registry (ECR) is a managed AWS Docker registry service that is secure, scalable and reliable

ECR supports private Docker repositories with resource-based permissions using AWS IAM in order to access repositories and images

Developers can use the Docker CLI to push, pull and manage images

Tasks

A task definition is required to run Docker containers in Amazon ECS

A task definition is a text file in JSON format that describes one or more containers, up to a maximum of 10

Task definitions use Docker images to launch containers

You specify the number of tasks to run (i.e. the number of containers)

Some of the parameters you can specify in a task definition include:

- Which Docker images to use with the containers in your task
- How much CPU and memory to use with each container
- Whether containers are linked together in a task
- The Docker networking mode to use for the containers in your task
- What (if any) ports from the container are mapped to the host container instances
- Whether the task should continue if the container finished or fails
- The commands the container should run when it is started
- Environment variables that should be passed to the container when it starts
- Data volumes that should be used with the containers in the task
- IAM role the task should use for permissions

You can use Amazon ECS Run task to run one or more tasks once

Clusters

ECS Clusters are a logical grouping of container instances the you can place tasks on

A default cluster is created but you can then create multiple clusters to separate resources

ECS allows the definition of a specified number (desired count) of tasks to run in the cluster

Clusters can contain tasks using the Fargate and EC2 launch type

For clusters with the EC2 launch type clusters can contain different container instance types

Each container instance may only be part of one cluster at a time

"Services" provide auto-scaling functions for ECS

Clusters are region specific

You can create IAM policies for your clusters to allow or restrict users' access to specific clusters

Service Scheduler

You can schedule ECS using Service Scheduler and CustomScheduler

Ensures that the specified number of tasks are constantly running and reschedules tasks when a task fails

Can ensure tasks are registered against an ELB

Custom Scheduler

You can create your own schedulers to meet business needs

Leverage third party schedulers such as Blox

The Amazon ECS schedulers leverage the same cluster state information provided by the Amazon ECS API to make appropriate placement decisions

ECS Container Agent

The ECS container agent allows container instances to connect to the cluster

The container agent runs on each infrastructure resource on an ECS cluster

The ECS container agent is included in the Amazon ECS optimized AMI and can also be installed on any EC2 instance that supports the ECS specification (only supported on EC2 instances)

Linux and Windows based

For non-AWS Linux instances to be used on AWS you must manually install the ECS container agent

Security/SLA

EC2 instances use an IAM role to access ECS

IAM can be used to control access at the container level using IAM roles

The container agent makes calls to the ECS API on your behalf through the applied IAM roles and policies

You need to apply IAM roles to container instances before they are launched (EC2 launch type)

AWS recommend limiting the permissions that are assigned to the container instance's IAM roles

Assign extra permissions to tasks through separate IAM roles (IAM Roles for Tasks)

ECS tasks use an IAM role to access services and resources

Security groups attach at the instance or container level

You have root level access to the OS of the EC2 instances

The Compute SLA guarantees a Monthly Uptime Percentage of at least 99.99% for Amazon ECS

Limits

Soft limits (default):

- Clusters per region = 1000
- Instances per cluster = 1000
- Services per cluster = 500

Hard limits:

- One load balancer per service
- 1000 tasks per service (the "desired" count)
- Max 10 containers per task definition
- Max 10 tasks per instance (host)

AW Lambda

General Lambda Concepts

AWS Lambda lets you run code as functions without provisioning or managing servers

Lambda-based applications (also referred to as serverless applications) are composed of functions triggered by events

With serverless computing, your application still runs on servers, but all the server management is done by AWS

You cannot log in to the compute instances that run Lambda functions or customise the operating system or language runtime

Lambda functions:

- Consist of code and any associated dependencies
- Configuration information is associated with the function
- You specify the configuration information when you create the function
- API provided for updating configuration data

You specify the amount of memory you need allocated to your Lambda functions

AWS Lambda allocates CPU power proportional to the memory you specify using the same ratio as a general purpose EC2 instance type

Functions can access:

- AWS services or non-AWS services
- AWS services running in VPCs (e.g. RedShift, Elasticache, RDS instances)
- Non-AWS services running on EC2 instances in an AWS VPC

To enable your Lambda function to access resources inside your private VPC, you must provide additional VPC-specific configuration information that includes VPC subnet IDs and security group IDs

AWS Lambda uses this information to set up elastic network interfaces (ENIs) that enable your function

Compute resources:

- You can request additional memory in 64MB increments from 128MB to 3008MB
- Functions larger than 1536MB are allocated multiple CPU threads, and multi-threaded or multi-process code is needed to take advantage

There is a maximum execution timeout

- Max is 15 minutes (900 seconds), default is 3 seconds
- You pay for the time it runs
- Lambda terminates the function at the timeout

Code is invoked using API calls made using AWS SDKs

Lambda assumes an IAM role when it executes the function

The handler name refers to the method in your code where Lambda begins execution

The components of AWS Lambda are:

- A Lambda function which is comprised of your custom code and any dependent libraries
- Event sources such as SNS or a custom service that triggers your function and executes its logic
- Downstream resources such as DynamoDB or Amazon S3 buckets that your Lambda function calls once it is triggered
- Log streams are custom logging statements that allow you to analyze the execution flow and performance of your Lambda function

Lambda is an event-driven compute service where AWS Lambda runs code in response to events such as a changes to data in an S3 bucket or a DynamoDB table

An event source is an AWS service or developer-created application that produces events that trigger an AWS Lambda function to run

Event sources are mapped to Lambda functions

Event sources maintain the mapping configuration except for stream-based services (e.g. DynamoDB, Kinesis) for which the configuration is made on the Lambda side and Lambda performs the polling

Supported AWS event sources include:

- Amazon S3
- Amazon DynamoDB
- Amazon Kinesis Data Streams
- Amazon Simple Notification Service
- Amazon Simple Email Service
- Amazon Simple Queue Service
- Amazon Cognito
- AWS CloudFormation
- Amazon CloudWatch Logs
- Amazon CloudWatch Events
- AWS CodeCommit
- Scheduled Events (powered by Amazon CloudWatch Events)
- AWS Config
- Amazon Alexa
- Amazon Lex
- Amazon API Gateway
- AWS IoT Button
- Amazon CloudFront
- Amazon Kinesis Data Firehose
- Other Event Sources: Invoking a Lambda Function On Demand

Other event sources can invoke Lambda functions on-demand (application needs permissions to invoke the Lambda function)

Lambda can run code in response to HTTP requests using Amazon API gateway or API calls made using the AWS SDKs

AWS Lambda supports code written in Node.js (JavaScript), Python, Java (Java 8 compatible), C# (.NET Core), Ruby, Go and PowerShell

AWS Lambda stores code in Amazon S3 and encrypts it at rest

Continuous scaling - scales out not up

Lambda scales concurrently executing functions up to your default limit (1000)

Lambda functions are serverless and independent, 1 event = 1 function

Functions can trigger other functions so 1 event can trigger multiple functions

For non-stream-based event sources each published event is a unit of work, run in parallel up to your account limit (one Lambda function per event)2

For stream-based event sources the number of shards indicates the unit of concurrency (one function per shard)

Lambda works globally

To enable VPC support, you need to specify one or more subnets in a single VPC and a security group as part of your function configuration

Lambda functions provide access only to a single VPC. If multiple subnets are specified, they must all be in the same VPC

Lambda functions configured to access resources in a particular VPC will not have access to the Internet as a default configuration. If you need access to external endpoints, you will need to create a NAT in your VPC to forward this traffic and configure your security group to allow this outbound traffic

Versioning can be used to run different versions of your code

Each Lambda function has a unique Amazon Resource Name (ARN) which cannot be changed after publishing

Use cases fall within the following categories:

- Using Lambda functions with AWS services as event sources
- On-demand Lambda function invocation over HTTPS using Amazon API Gateway (custom REST API and endpoint)
- On-demand Lambda function invocation using custom applications (mobile, web apps, clients) and AWS SDKs, AWS Mobile SDKs, and the AWS Mobile SDK for Android
- Scheduled events can be configured to run code on a scheduled basis through the AWS Lambda Console

Building Lambda Apps

You can deploy and manage your serverless applications using the AWS Serverless Application Model (AWS SAM)

AWS SAM is a specification that prescribes the rules for expressing serverless applications on AWS

This specification aligns with the syntax used by AWS CloudFormation today and is supported natively within AWS CloudFormation as a set of resource types (referred to as "serverless resources")

You can automate your serverless application's release process using AWS CodePipeline and AWS CodeDeploy

You can enable your Lambda function for tracing with AWS X-Ray

Lambda Edge

Lambda@Edge allows you to run code across AWS locations globally without provisioning or managing servers, responding to end users at the lowest network latency

You just upload your Node.js code to AWS Lambda and configure your function to be triggered in response to an Amazon CloudFront request

The code is then ready to execute across AWS locations globally when a request for content is received, and scales with the volume of CloudFront requests globally

Limits

Memory - minimum 128MB, maximum 3008MB in 64MB increments

Ephemeral disk capacity (/tmp space) per invocation - 512 MB

Number of file descriptors - 1024

Number of processes and threads (combined) - 1024

Maximum execution duration per request - 900 seconds

Concurrent executions per account - 1000 (soft limit)

Operations and Monitoring

Lambda automatically monitors Lambda functions and reports metrics through CloudWatch

Lambda tracks the number of requests, the latency per request, and the number of requests resulting in an error

You can view the request rates and error rates using the AWS Lambda Console, the CloudWatch console, and other AWS resources

X-Ray is an AWS service that can be used to detect, analyse and optimise performance issues with Lambda applications

X-Ray collects metadata from the Lambda service and any upstream and downstream services that make up your application

Lambda is integrated with CloudTrail for capturing API calls and can deliver log files to your S3 bucket

Charges

Priced based on:

- Number of requests. First 1 million are free then $0.20 per 1 million
- Duration. Calculated from the time your code begins execution until it returns or terminates. Depends on the amount of memory allocated to a function

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

Considered a Platform as a Service (PaaS) solution

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

Supports the following languages and development stacks:

- Apache Tomcat for Java applications
- Apache HTTP Server for PHP applications
- Apache HTTP Server for Python applications
- Nginx or Apache HTTP Server for Node.js applications
- Passenger or Puma for Ruby applications
- Microsoft IIS 7.5, 8.0, and 8.5 for .NET applications
- Java SE
- Docker
- Go

Integrates with VPC

Integrates with IAM

Can provision most database instances

Allows full control of the underlying resources

Stores your application files and, optionally, server log files in Amazon S3

Application data can also be stored on S3

Multiple environments are supported to enable versioning

Changes from Git repositories are replicated

Linux and Windows 2008 R2 AMI support

Code is deployed using a WAR file or Git repository

Use the AWS toolkit for Visual Studio and the AWS toolkit for Eclipse to deploy Elastic Beanstalk

Fault tolerance within a single region

By default, applications are publicly accessible

Provides integration with CloudWatch

Can adjust application server settings

Can access logs without logging into application servers

Can use CloudFormation to deploy Elastic Beanstalk

Compute Practice Questions

Answers and explanations are provided below after the last question in this section.

Question 1:

Your organization is planning to go serverless in the cloud. Which of the following combinations of services provides a fully serverless architecture?

A. Lambda, API Gateway, DynamoDB, S3, CloudFront

B. Lambda, SQS, RDS, S3
C. API Gateway, EC2, DynamoDB, S3
D. EC2, EBS, Auto Scaling, ELB

Question 2:

An application stack includes an Elastic Load Balancer in a public subnet, a fleet of Amazon EC2 instances in an Auto Scaling Group, and an Amazon RDS MySQL cluster. Users connect to the application from the Internet. The application servers and database must be secure.

What is the most appropriate architecture for the application stack?

A. Create a private subnet for the Amazon EC2 instances and a public subnet for the Amazon RDS cluster
B. Create a private subnet for the Amazon EC2 instances and a private subnet for the Amazon RDS cluster
C. Create a public subnet for the Amazon EC2 instances and a private subnet for the Amazon RDS cluster
D. Create a public subnet for the Amazon EC2 instances and a public subnet for the Amazon RDS cluster

Question 3:

A call center application consists of a three-tier application using Auto Scaling groups to automatically scale resources as needed. Users report that every morning at 9:00am the system becomes very slow for about 15 minutes.

A Solutions Architect determines that a large percentage of the call center staff starts work at 9:00am, so Auto Scaling does not have enough time to scale to meet demand.

How can the Architect fix the problem?

A. Change the Auto Scaling group's scale out event to scale based on network utilization
B. Create an Auto Scaling scheduled action to scale out the necessary resources at 8:30am each morning
C. Use Reserved Instances to ensure the system has reserved the right amount of capacity for the scaling events
D. Permanently keep a steady state of instance that is needed at 9:00am to guarantee available resources, but use Spot Instances

Question 4:

You are using the Elastic Container Service (ECS) to run a number of containers using the EC2 launch type. To gain more control over scheduling containers you have decided to utilize Blox to integrate a third-party scheduler. The third-party scheduler will use the StartTask API to place tasks on specific container instances.

What type of ECS scheduler will you need to use to enable this configuration?

A. Service Scheduler
B. Cron Scheduler
C. ECS Scheduler
D. Custom Scheduler

Question 5:

A Solutions Architect is designing the compute layer of a serverless application. The compute layer will manage requests from external systems, orchestrate serverless workflows, and execute the business logic.

The Architect needs to select the most appropriate AWS services for these functions. Which services should be used for the compute layer? (choose 2)

A. Use Amazon ECS for executing the business logic
B. Use Amazon API Gateway with AWS Lambda for executing the business logic
C. Use AWS CloudFormation for orchestrating serverless workflows
D. Use AWS Step Functions for orchestrating serverless workflows
E. Use AWS Elastic Beanstalk for executing the business logic

Question 6:

A Solutions Architect is creating a solution for an application that must be deployed on Amazon EC2 hosts that are dedicated to the client. Instance placement must be automatic and billing should be per instance.

Which type of EC2 deployment model should be used?

A. Reserved Instance
B. Dedicated Instance
C. Dedicated Host
D. Cluster Placement Group

Question 7:

You just created a new subnet in your VPC and have launched an EC2 instance into it. You are trying to directly access the EC2 instance from the Internet and cannot connect. Which steps should you take to troubleshoot the issue? (choose 2)

A. Check that the instance has a public IP address
B. Check that there is a NAT Gateway configured for the subnet
C. Check that the route table associated with the subnet has an entry for an Internet Gateway
D. Check that you can ping the instance from another subnet
E. Check that Security Group has a rule for outbound traffic

Question 8:

You are a Solutions Architect at Digital Cloud Training. In your VPC you have a mixture of EC2 instances in production and non-production environments. You need to devise a way to segregate access permissions to different sets of users for instances in different environments.

How can this be achieved? (choose 2)

A. Add a specific tag to the instances you want to grant the users or groups access to
B. Add an environment variable to the instances using user data
C. Create an IAM policy with a conditional statement that matches the environment variables
D. Create an IAM policy that grants access to any instances with the specific tag and attach to the users and groups
E. Attach an Identity Provider (IdP) and delegate access to the instances to the relevant groups

Question 1 answer: A

Explanation:

Serverless is the native architecture of the cloud that enables you to shift more of your operational responsibilities to AWS, increasing your agility and innovation. Serverless allows you to build and run applications and services without thinking about servers.

Serverless services include Lambda, API Gateway, DynamoDB, S3, SQS, and CloudFront. EC2 and RDS are not serverless as they both rely on EC2 instances which must be provisioned and managed.

Question 2 answer: B

Explanation:

Typically, the nodes of an Internet-facing load balancer have public IP addresses and must therefore be in a public subnet. To keep your back-end instances secure you can place them in a private subnet. To do this you must associate a corresponding public and private subnet for each availability zone the ELB/instances are in).

For RDS, you create a DB subnet group which is a collection of subnets (typically private) that you create in a VPC and that you then designate for your DB instances.

Question 3 answer: B

Explanation:

Scaling based on a schedule allows you to set your own scaling schedule for predictable load changes. To configure your Auto Scaling group to scale based on a schedule, you create a scheduled action. This is ideal for situations where you know when and for how long you are going to need the additional capacity.

Changing the scale-out events to scale based on network utilization may not assist here. We're not certain the network utilization will increase sufficiently to trigger an Auto Scaling scale out action as the load may be more CPU/memory or number of connections. The main problem

however is that we need to ensure the EC2 instances are provisioned ahead of demand not in response to demand (which would incur a delay whilst the EC2 instances "warm up").

Using reserved instances ensures capacity is available within an AZ, however the issue here is not that the AZ does not have capacity for more instances, it is that the instances are not being launched by Auto Scaling ahead of the peak demand.

Keeping a steady state of Spot instances is not a good solution. Spot instances may be cheaper, but this is not guaranteed and keeping them online 24hrs a day is wasteful and could prove more expensive.

Question 4 answer: D

Explanation:

Amazon ECS provides a service scheduler (for long-running tasks and applications), the ability to run tasks manually (for batch jobs or single run tasks), with Amazon ECS placing tasks on your cluster for you. The service scheduler is ideally suited for long running stateless services and applications. Amazon ECS allows you to create your own schedulers that meet the needs of your business, or to leverage third party schedulers.

Custom schedulers use the StartTask API operation to place tasks on specific container instances within your cluster. Custom schedulers are only compatible with tasks using the EC2 launch type. If you are using the Fargate launch type for your tasks, the StartTask API does not workBlox is an open- source project that gives you more control over how your containerized applications run on Amazon ECS.

Blox enables you to build schedulers and integrate third-party schedulers with Amazon ECS while leveraging Amazon ECS to fully manage and scale your clusters.

A cron scheduler is used in UNIX/Linux but is not a type of ECS scheduler.

A service scheduler is not a type of third-party scheduler.

Question 5 answer: B,D

Explanation:

With Amazon API Gateway, you can run a fully managed REST API that integrates with Lambda to execute your business logic and includes traffic management, authorization and access control, monitoring, and API versioning.

AWS Step Functions orchestrates serverless workflows including coordination, state, and function chaining as well as combining long-running executions not supported within Lambda execution limits by breaking into multiple steps or by calling workers running on Amazon Elastic Compute Cloud (Amazon EC2) instances or on-premises.

The Amazon Elastic Container Service (ECS) is not a serverless application stack, containers run on EC2 instances.

AWS CloudFormation and Elastic Beanstalk are orchestrators that are used for describing and provisioning resources not actually performing workflow functions within the application.

Question 6 answer: *B*

Explanation:

Dedicated Instances are Amazon EC2 instances that run in a VPC on hardware that's dedicated to a single customer. Your Dedicated instances are physically isolated at the host hardware level from instances that belong to other AWS accounts. Dedicated instances allow automatic instance placement and billing is per instance.

An Amazon EC2 Dedicated Host is a physical server with EC2 instance capacity fully dedicated to your use. Dedicated Hosts can help you address compliance requirements and reduce costs by allowing you to use your existing server-bound software licenses. With dedicated hosts billing is on a per-host basis (not per instance).

Reserved instances are a method of reducing cost by committing to a fixed contract term of 1 or 3 years.

A Cluster Placement Group determines how instances are placed on underlying hardware to enable low-latency connectivity.

Question 7 answer: *A,C*

Explanation:

Public subnets are subnets that have:

- "Auto-assign public IPv4 address" set to "Yes"
- The subnet route table has an attached Internet Gateway

A NAT Gateway is used for providing outbound Internet access for EC2 instances in private subnets. Checking you can ping from another subnet does not relate to being able to access the instance remotely as it uses different protocols and a different network path.

Security groups are stateful and do not need a rule for outbound traffic. For this solution you would only need to create an inbound rule that allows the relevant protocol.

Question 8 answer: A,D

Explanation:

You can use the condition checking in IAM policies to look for a specific tag. IAM checks that the tag attached to the principal making the request matches the specified key name and value.

You cannot achieve this outcome using environment variables stored in user data and conditional statements in a policy. You must use an IAM policy that grants access to instances based on the tag.

You cannot use an IdP for this solution.

STORAGE

Amazon S3

General

Amazon S3 is object storage built to store and retrieve any amount of data from anywhere on the Internet

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

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

Amazon S3 provides a simple, standards-based REST web services interface that is designed to work with any Internet-development toolkit

Files can be from 0 bytes to 5TB

The largest object that can be uploaded in a single PUT is 5 gigabytes

For objects larger than 100 megabytes use the Multipart Upload capability

Updates to an object are atomic - when reading an updated object, you will either get the new object or the old one, you will never get partial or corrupt data

There is unlimited storage available

It is recommended to access S3 through SDKs and APIs (the console uses APIs)

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***

Requester pays function causes the requester to pay (removes anonymous access)

Can provide time-limited access to objects

Provides read after write consistency for PUTS of new objects

Provides eventual consistency for overwrite PUTS and DELETES (takes time to propagate)

You can only store files on S3, not possible for operating system's

HTTP 200 code indicates a successful write to S3

S3 data is made up of:

- ***Key (name)***
- ***Value (data)***
- ***Version ID***
- ***Metadata***
- ***Access Control Lists***

Amazon S3 automatically scales to high request rates

For example, your application can achieve at least 3,500 PUT/POST/DELETE and 5,500 GET requests per second per prefix in a bucket

There are no limits to the number of prefixes in a bucket. It is simple to increase your read or write performance exponentially

For read intensive requests you can also use CloudFront edge locations to offload from 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

Use Cases

Typical use cases include:

- **Backup and Storage** – Provide data backup and storage services for others
- **Application Hosting** – Provide services that deploy, install, and manage web applications
- **Media Hosting** – Build a redundant, scalable, and highly available infrastructure that hosts video, photo, or music uploads and downloads
- **Software Delivery** – Host your software applications that customers can download

S3 is a persistent, highly durable data store

Persistent data stores are non-volatile storage devices that retain data when powered off

This is in contrast to transient data stores and ephemeral data stores

The following table provides a description of persistent, transient and ephemeral data stores and which AWS service to use:

Storage Type	Description	Examples
Persistent Data Store	Data is durable and sticks around after reboots, restarts, or power cycles	S3, Glacier, EBS, EFS
Transient Data Store	Data is just temporarily stored and passed along to another process or persistent store	SQS, SNS
Ephemeral Data Store	Data is lost when the system is stopped	EC2 Instance Store, Memcached

Buckets

Files 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 to mimic folders

100 buckets per account by default

You can store unlimited objects in your buckets

You can create folders in your buckets (only available through the Console)

You cannot create nested buckets

Bucket ownership is not transferrable

Bucket names cannot be changed after they have been created

If a bucket is deleted its name becomes available again

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

URL is in this format: https://s3-***eu-west-1***.amazonaws.com/***<bucketname>***

Can backup a bucket to another bucket in another account

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

Sub-resources

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

Storage Classes

There are four S3 storage classes

- S3 Standard (durable, immediately available, frequently accessed)
- S3 Standard-IA (durable, immediately available, infrequently accessed)
- S3 One Zone-IA (lower cost for infrequently accessed data with less resilience)
- Amazon Glacier (archived data, where you can wait 3-5 hours for access)

	S3 Standard	S3 Standard-IA	S3 One Zone-IA	Amazon Glacier
Designed for durability	99.999999999%	99.999999999%	99.999999999%	99.999999999%
Designed for availability	99.99%	99.9%	99.5%	N/A
Availability SLA	99.9%	99%	99%	N/A
Availability Zones	≥3	≥3	1	≥3
Minimum capacity charge per object	N/A	128KB	128KB	N/A
Minimum storage duration charge	N/A	30 days	30 days	90 days
Retrieval fee	N/A	Per GB retrieved	Per GB retrieved	Per GB retrieved
First byte latency	milliseconds	milliseconds	milliseconds	Select minutes or hours
Storage type	Object	Object	Object	Object
Lifecycle transitions	Yes	Yes	Yes	Yes

For S3 Standard, S3 Standard-IA, and Amazon Glacier storage classes, your objects are automatically stored across multiple devices spanning a minimum of three Availability Zones

Objects stored in the S3 One Zone-IA storage class are stored redundantly within a single Availability Zone in the AWS Region you select

Access and Access Policies

There are four mechanisms for controlling access to Amazon S3 resources:

- IAM policies
- Bucket policies
- Access Control Lists (ACLs)
- Query string authentication (URL to an Amazon S3 object which is only valid for a limited time)

Access auditing can be configured by configuring an Amazon S3 bucket to create access log records for all requests made against it

For capturing IAM/user identity information in logs configure AWS CloudTrail Data Events

By default a bucket, its objects, and related sub-resources are all private

By default only a resource owner can access a bucket

The resource owner refers to the AWS account that creates the resource

With IAM the account owner rather than the IAM user is the owner

Within an IAM policy you can grant either programmatic access or AWS Management Console access to Amazon S3 resources

Amazon Resource Names (ARN) are used for specifying resources in a policy.

The format for any resource on AWS is:

arn:partition:service:region:namespace:relative-id

For S3 resources:

- aws is a common partition name
- s3 is the service
- You don't specify Region and namespace
- For Amazon S3, it can be a bucket-name or a bucket-name/object-key. You can use wild card

The format for S3 resources is:

arn:aws:s3:::bucket_name

arn:aws:s3:::bucket_name/key_name

A bucket owner can grant cross-account permissions to another AWS account (or users in an account) to upload objects

- The AWS account that uploads the objects owns them
- The bucket owner does not have permissions on objects that other accounts own, however:
    - The bucket owner pays the charges
    - The bucket owner can deny access to any objects regardless of ownership
    - The bucket owner can archive any objects or restore archived objects regardless of ownership

Access to buckets and objects can be granted to:

- Individual users
- AWS accounts
- Everyone (public/anonymous)

- All authenticated users (AWS users)

Access policies define access to resources and can be associated with resources (buckets and objects) and users

You can use the AWS Policy Generator to create a bucket policy for your Amazon S3 bucket

The categories of policy are resource-based policies and user policies

Resource-based policies:

- Attached to buckets and objects
- ACL-based policies define permissions
- ACLs can be used to grant read/write permissions to other accounts
- Bucket policies can be used to grant other AWS accounts or IAM users permission to the bucket and objects

User policies:

- Can use IAM to manage access to S3 resources
- Using IAM you can create users, groups and roles and attach access policies to them granting them access to resources
- You cannot grant anonymous permissions in an IAM user policy as the policy is attached to a user
- User policies can grant permissions to a bucket and the objects in it

ACLs:

- S3 ACLs enable you to manage access to buckets and objects
- Each bucket and object has an ACL attached to it as a sub-resource
- Bucket and object permissions are independent of each other
- The ACL defines which AWS accounts (grantees) or pre-defined S3 groups are granted access and the type of access
- A grantee can be an AWS account or one of the predefined Amazon S3 groups
- When you create a bucket or an object, S3 creates a default ACL that grants the resource owner full control over the resource

Cross account access:

- You grant permission to another AWS account using the email address or the canonical user ID
- However, if you provide an email address in your grant request, Amazon S3 finds the canonical user ID for that account and adds it to the ACL
- Grantee accounts can then then delegate the access provided by other accounts to their individual users

Pre-defined Groups

Authenticated Users group:

- This group represents all AWS accounts
- Access permission to this group allows any AWS account access to the resource
- All requests must be signed (authenticated)
- Any authenticated user can access the resource

All Users group:

- Access permission to this group allows anyone in the world access to the resource
- The requests can be signed (authenticated) or unsigned (anonymous)
- Unsigned requests omit the authentication header in the request
- AWS recommends that you never grant the All Users group WRITE, WRITE_ACP, or FULL_CONTROL permissions

Log Delivery group:

- Providing WRITE permission to this group on a bucket enables S3 to write server access logs
- Not applicable to objects

The following table lists the set of permissions that Amazon S3 supports in an ACL

- The set of ACL permissions is the same for an object ACL and a bucket ACL
- Depending on the context (bucket ACL or object ACL), these ACL permissions grant permissions for specific buckets or object operations

The table below lists the permissions and describes what they mean in the context of objects and buckets:

Permission	When granted on a bucket	When granted on an object
READ	Allows grantee to list the objects in the bucket	Allows grantee to read the object data and its metadata
WRITE	Allows grantee to create, overwrite, and delete any object in the bucket	Not applicable
READ_ACP	Allows grantee to read the bucket ACL	Allows grantee to read the object ACL
WRITE_ACP	Allows grantee to write the ACL for the applicable bucket	Allows grantee to write the ACL for the applicable object
FULL_CONTROL	Allows grantee the READ, WRITE, READ_ACP, and WRITE_ACP permissions on the bucket	Allows grantee the READ, READ_ACP, and WRITE_ACP permissions on the object

Note the following:

- Permissions are assigned at the account level for authenticated users
- You cannot assign permissions to individual IAM users
- When Read is granted on a bucket it only provides the ability to list the objects in the bucket
- When Read is granted on an object the data can be read
- ACP means access control permissions and READ_ACP/WRITE_ACP control who can read/write the ACLs themselves
- WRITE is only applicable to the bucket level (except for ACP)

Bucket policies are limited to 20 KB in size

Object ACLs are limited to 100 granted permissions per ACL

The only recommended use case for the bucket ACL is to grant write permissions to the S3 Log Delivery group

There are limits to managing permissions using ACLs:

- You cannot grant permissions to individual users
- You cannot grant conditional permissions
- You cannot explicitly deny access

When granting other AWS accounts the permissions to upload objects, permissions to these objects can only be managed by the object owner using object ACLs

You can use bucket policies for:

- Granting users permissions to a bucket owned by your account
- Managing object permissions (where the object owner is the same account as the bucket owner)
- Managing cross-account permissions for all Amazon S3 permissions

You can use user policies for:

- Granting permissions for all Amazon S3 operations
- Managing permissions for users in your account
- Granting object permissions to users within the account

For an IAM user to access resources in another account the following must be provided:

- Permission from the parent account through a user policy
- Permission from the resource owner to the IAM user through a bucket policy, or the parent account through a bucket policy, bucket ACL or object ACL

If an AWS account owns a resource it can grant permissions to another account, that account can then delegate those permissions or a subset of them to uses in the account (permissions delegation)

An account that receives permissions from another account cannot delegate permissions cross-account to a third AWS account

Charges

No charge for data transferred between EC2 and S3 in the same region

Data transfer into S3 is free of charge

Data transferred to other regions is charged

Data Retrieval (applies to S3 Standard-IA and S3 One Zone-IA)

Charges are:

- Per GB/month storage fee
- Data transfer out of S3
- Upload requests (PUT and GET)
- Retrieval requests (S3-IA or Glacier)

Requester pays:

- The bucket owner will only pay for object storage fees
- The requester will pay for requests (uploads/downloads) and data transfers
- Can only be enabled at the bucket level

Multipart upload

Can be used to speed up uploads to S3

Multipart upload uploads objects in parts independently, in parallel and in any order

Performed using the S3 Multipart upload API

It is recommended for objects of 100MB or larger

- Can be used for objects from 5MB up to 5TB
- Must be used for objects larger than 5GB

If transmission of any part fails it can be retransmitted

Improves throughput

Can pause and resume object uploads

Can begin upload before you know the final object size

Copy

You can create a copy of objects up to 5GB in size in a single atomic operation

For files larger than 5GB you must use the multipart upload API

Can be performed using the AWS SDKs or REST API

The copy operation can be used to:

- Generate additional copies of objects
- Renaming objects
- Changing the copy's storage class or encryption at rest status
- Move objects across AWS locations/regions
- Change object metadata

Once uploaded to S3 some object metadata cannot be changed, copying the object can allow you to modify this information

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

Bucket names must be DNS compliance and cannot have periods between labels

Now HIPAA compliant

You can use multipart uploads with transfer acceleration

Must use one of the following endpoints:

- .s3-accelerate.amazonaws.com
- .s3-accelerate.dualstack.amazonaws.com (dual-stack option)

S3 Transfer Acceleration supports all bucket level features including multipart uploads

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

Requester pays does not work with website endpoints

Does not support HTTPS/SSL

Returns an HTML document

Supports object and bucket level redirects

Only supports GET and HEAD requests on objects

Supports publicly readable content only

To enable website hosting on a bucket, specify:

- An Index document (default web page)
- Error document (optional)

Key Difference	REST API Endpoint	Website Endpoint
Access Control	Supports both public and private content	Supports only publicly readable content
Error message handling	Returns an XML-formatted error response	Returns an HTML document
Redirection support	Not applicable	Supports both object-level and bucket-level redirects
Requests support	Supports all bucket and object operations	Supports only GET and HEAD requests on objects
Responses to GET and HEAD requests at the root of the bucket	Returns a list of the object keys in the bucket	Returns the Index document that is specified in the website configuration
SSL support	Supports SSL connections	Does not support SSL connections

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

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

Lifecycle Management

Use to optimize storage costs, adhere to data retention policies and to keep S3 volumes well-maintained

Bucket level configuration

The following actions can be performed:

- Transition to S3-IA (128Kb and 30 days after creation date)
- Archive to Glacier (30 days after IA if applicable)

Objects less than 128KB will not be transitioned to S3 Standard-IA

Objects must be stored in S3 Standard-IA for at least 30 days

An object must be in S3 Standard for at least 30 days before it can be transitioned to S3 Standard-IA

You cannot use a lifecycle policy to move an object from Glacier to S3 Standard or S3 Standard-IA (restore to S3 One Zone-IA and copy)

Cannot be used to change a storage class to S3 One Zone-IA

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

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

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

S3 CloudWatch Metrics

You can use the AWS Management Console to enable the generation of 1-minute CloudWatch request metrics for your S3 bucket or configure filters for the metrics using a prefix or object tag

Alternatively, you can call the S3 PUT Bucket Metrics API to enable and configure publication of S3 storage metrics

CloudWatch Request Metrics will be available in CloudWatch within 15 minutes after they are enabled

CloudWatch Storage Metrics are enabled by default for all buckets, and reported once per day

The S3 metrics that can be monitored include:

- S3 requests
- Bucket storage
- Bucket size
- All requests
- HTTP 4XX/5XX errors

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

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

You can set up CRR across AWS accounts to store your replicated data in a different account in the target region

Provides low latency access for data by copying objects to buckets that are closer to users

To activate CRR you need to configure the replication on the source bucket:

- Define the bucket in the other region to replicate to

- Specify to replicate all objects or a subset of objects with specific key name prefixes

The replicas will be exact replicas and share the same key names and metadata

You can specify a different storage class (by default the source storage class will be used)

AWS S3 will encrypt data in-transit with SSL

AWS S3 must have permission to replicate objects

Bucket owners must have permission to read the object and object ACL

Can be used across accounts but the source bucket owner must have permission to replicate objects into the destination bucket

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 sub-resources
- 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

Charges:

- Requests for upload
- Inter-region transfer
- S3 storage in both regions

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

AWS Glacier

Glacier is an archiving storage solution for infrequently accessed data

Archived objects are not available for real time access and you need to submit a retrieval request

Retrieval can take a few hours

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

There is no SLA

Glacier is designed to sustain the loss of two facilities

Glacier automatically encrypts data at rest using AES 256 symmetric keys and supports secure transfer of data over SSL

Glacier may not be available in all AWS regions

Glacier objects are visible through S3 only (not Glacier directly)

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

Glacier file archives of 1 byte - 4 GB can be performed in a single operation

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

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

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)

Glacier Charges:

There is no charge for data transfer between EC2 and Glacier in the same region

There is a charge if you delete data within 90 days

When you restore you pay for:

- The Glacier archive
- The requests
- The restored data on S3

Amazon EFS

General

EFS is a fully-managed service that makes it easy to set up and scale file storage in the Amazon Cloud

Implementation of an NFS file share and is accessed using the NFSv4.1 protocol

Elastic storage capacity, and pay for what you use (in contrast to EBS with which you pay for what you provision)

Multi-AZ metadata and data storage

Can configure mount-points in one, or many, AZs

Can be mounted from on-premises systems ONLY if using Direct Connect or a VPN connection

Alternatively, use the EFS File Sync agent

Good for big data and analytics, media processing workflows, content management, web serving, home directories etc.

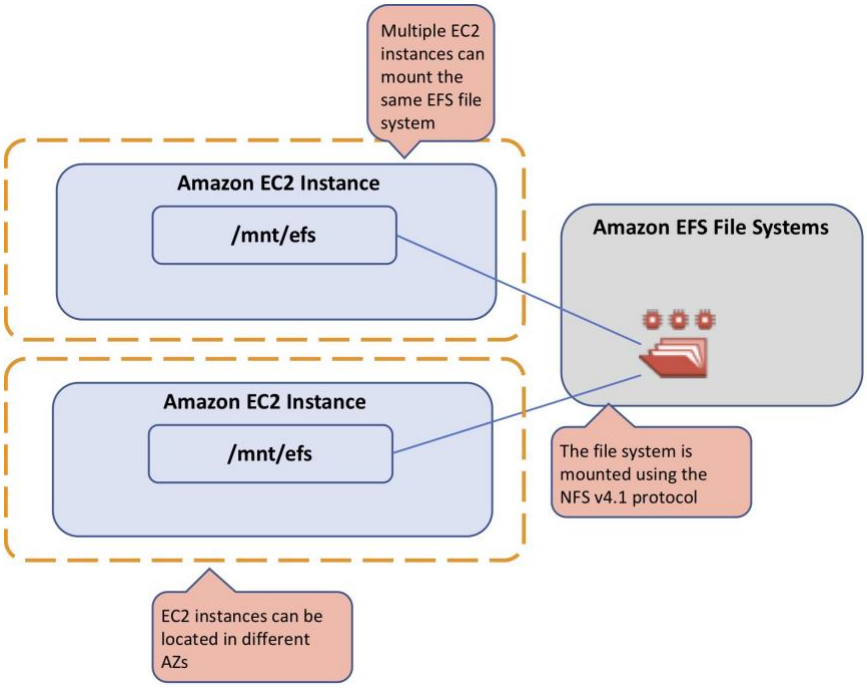
Pay for what you use (no pre-provisioning required)

Can scale up to petabytes

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

Can concurrently connect 1 to 1000s of EC2 instances, from multiple AZs

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



By default you can create up to 10 file systems per account

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

You mount an EFS file system on your on-premises Linux server using the standard Linux mount command for mounting a file system via the NFSv4.1 protocol

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

The 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)

Limited region support currently

Instances can be behind an ELB

Can also be mounted on an on-premises server (via Direct Connect)

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 and workflows, content management, web serving and home directories	Boot volumes, transactional and NoSQL databases, data warehousing and ETL

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

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 and block 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

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

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

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 industry standard Transport Layer Security (TLS) 1.2

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

Data can be encrypted in transit between your Amazon EFS file system and its clients by using the EFS mount helper

EFS File Sync

EFS File Sync provides a fast and simple way to securely sync existing file systems into Amazon EFS

EFS File Sync copies files and directories into Amazon EFS at speeds up to 5x faster than standard Linux copy tools, with simple setup and management in the AWS Console

EFS File Sync securely and efficiently copies files over the internet or an AWS Direct Connect connection

Copies file data and file system metadata such as ownership, timestamps, and access permissions

EFS File Sync provides the following benefits:

- Efficient high-performance parallel data transfer that tolerates unreliable and high-latency networks.
- Encryption of data transferred from your IT environment to AWS.
- Data transfer rate up to five times faster than standard Linux copy tools.
- Full and incremental syncs for repetitive transfers.

Note: EFS File Sync currently doesn't support syncing from an Amazon EFS source to an NFS destination

When deploying Amazon EFS File Sync on EC2, the instance size must be at least xlarge for your EFS File Sync to function.

Recommended to use one of the Memory optimized r4.xlarge instance types

Can choose to run EFS File Sync either on-premises as a virtual machine (VM), or in AWS as an EC2 instance

Supports VMware ESXi

Compatibility

EFS is integrated with a number of other AWS services, including CloudWatch, CloudFormation, CloudTrail, IAM, and Tagging services

CloudWatch allows you to monitor file system activity using metrics

CloudFormation allows you to create and manage file systems using templates

CloudTrail allows you to record all Amazon EFS API calls in log files

IAM allows you to control who can administer your file system

Tagging services allows you to label your file systems with metadata that you define

Pricing and Billing

You pay only for the amount of file system storage you use per month

When using the Provisioned Throughput mode, you pay for the throughput you provision per month

There is no minimum fee and there are no set-up charges

With EFS File Sync, you pay per-GB for data copied to EFS

AWS Storage Gateway

General

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 AWS S3 and Glacier

Often used in disaster recovery preparedness to sync data to AWS

Useful in cloud migrations

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:

New Name	Old Name	Interface	Use Case
File Gateway	None	NFS, SMB	Allow on-prem or EC2 instances to store objects in S3 via NFS or SMB mount points
Volume Gateway Stored Mode	Gateway-Stored Volumes	iSCSI	Asynchronous replication of on-prem data to S3
Volume Gateway Cached Mode	Gateway-Cached Volumes	iSCSI	Primary data stored in S3 with frequently accessed data cached locally on-prem
Tape Gateway	Gateway-Virtual Tape Library	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 NFS v3 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

Storage Practice Questions

Answers and explanations are provided below after the last question in this section.

Question 1:

A company has an on-premises data warehouse that they would like to move to AWS where they will analyze large quantities of data. What is the most cost-efficient EBS storage volume type that is recommended for this use case?

A. Throughput Optimized HDD (st1)
B. EBS Provisioned IOPS SSD (io1)
C. EBS General Purpose SSD (gp2)
D. Cold HDD (sc1)

Question 2:

Your company keeps unstructured data on a filesystem. You need to provide access to employees via EC2 instances in your VPC. Which storage solution should you choose?

A. Amazon S3
B. Amazon EBS
C. Amazon EFS
D. Amazon Snowball

Question 3:

A legacy application running on-premises requires a Solutions Architect to be able to open a firewall to allow access to several Amazon S3 buckets. The Architect has a VPN connection to AWS in place.

Which option represents the simplest method for meeting this requirement?

A. Create an IAM role that allows access from the corporate network to Amazon S3
B. Configure a proxy on Amazon EC2 and use an Amazon S3 VPC endpoint
C. Use Amazon API Gateway to do IP whitelisting
D. Configure IP whitelisting on the customer's gateway

Question 4:

A Solutions Architect is designing a mobile application that will capture receipt images to track expenses. The Architect wants to store the images on Amazon S3. However, uploading the images through the web server will create too much traffic.

What is the MOST efficient method to store images from a mobile application on Amazon S3?

A. Upload to a second bucket, and have a Lambda event copy the image to the primary bucket
B. Upload to a separate Auto Scaling Group of server behind an ELB Classic Load Balancer, and have the server instances write to the Amazon S3 bucket
C. Upload directly to S3 using a pre-signed URL
D. Expand the web server fleet with Spot instances to provide the resources to handle the images

Question 5:

A large quantity of data that is rarely accessed is being archived onto Amazon Glacier. Your CIO wants to understand the resilience of the service. Which of the statements below is correct about Amazon Glacier storage? (choose 2)

A. Provides 99.9% availability of archives
B. Data is resilient in the event of one entire region destruction

C. Data is resilient in the event of one entire Availability Zone destruction
D. Provides 99.999999999% durability of archives
E. Data is replicated globally

Question 6:

You have implemented the AWS Elastic File System (EFS) to store data that will be accessed by a large number of EC2 instances. The data is sensitive and you are working on a design for implementing security measures to protect the data. You need to ensure that network traffic is restricted correctly based on firewall rules and access from hosts is restricted by user or group.

How can this be achieved with EFS? (choose 2)

A. Use EFS Security Groups to control network traffic
B. Use AWS Web Application Firewall (WAF) to protect EFS
C. Use POSIX permissions to control access from hosts by user or group
D. Use IAM groups to control access by user or group
E. Use Network ACLs to control the traffic

Question 7:

You launched an EBS-backed EC2 instance into your VPC. A requirement has come up for some high-performance ephemeral storage and so you would like to add an instance-store backed volume. How can you add the new instance store volume?

A. You can specify the instance store volumes for your instance only when you launch an instance
B. You can use a block device mapping to specify additional instance store volumes when you launch your instance, or you can attach additional instance store volumes after your instance is running
C. You must shutdown the instance in order to be able to add the instance store volume
D. You must use an Elastic Network Adapter (ENA) to add instance store volumes. First, attach an ENA, and then attach the instance store volume

Question 8:

You are a Solutions Architect for an insurance company. An application you manage is used to store photos and video files that relate to insurance claims. The application writes data using the iSCSI protocol to a storage array. The array currently holds 10TB of data and is approaching capacity.

Your manager has instructed you that he will not approve further capital expenditure for on-premises infrastructure. Therefore, you are planning to migrate data into the cloud. How can you move data into the cloud whilst retaining low-latency access to frequently accessed data on-premise using the iSCSI protocol?

A. Use an AWS Storage Gateway File Gateway in cached volume mode
B. Use an AWS Storage Gateway Virtual Tape Library

C. Use an AWS Storage Gateway Volume Gateway in cached volume mode
D. Use an AWS Storage Gateway Volume Gateway in stored volume mode

Question 1 answer: *A*

Explanation:

Throughput Optimized HDD (st1) volumes are recommended for streaming workloads requiring consistent, fast throughput at a low price. Examples include Big Data warehouses and Log Processing. You cannot use these volumes as a boot volume.

EBS Provisioned IOPS SSD (io1) volumes are recommended for critical business applications that require sustained IOPS performance, or more than 16,000 IOPS or 250 MiB/s of throughput per volume.

EBS General Purpose SSD (gp2) volumes are recommended for most workloads including use as system boot volumes, virtual desktops, low-latency interactive apps, and development and test environments.

Cold HDD (sc1) volumes are recommended for throughput-oriented storage for large volumes of data that is infrequently accessed. This is the lowest cost HDD volume type. You cannot use these volumes as a boot volume.

Question 2 answer: *C*

Explanation:

EFS is the only storage system presented that provides a file system. EFS is accessed by mounting filesystems using the NFS v4.1 protocol from your EC2 instances. You can concurrently connect up to thousands of instances to a single EFS filesystem.

Amazon S3 is an object-based storage system that is accessed over a REST API.

Amazon EBS is a block-based storage system that provides volumes that are mounted to EC2 instances but cannot be shared between EC2 instances.

Amazon Snowball is a device used for migrating very large amounts of data into or out of AWS.

Question 3 answer: *A*

Explanation:

The solutions architect can create an IAM role that provides access to the required S3 buckets. With the on-premises firewall opened to allow outbound access to S3 (over HTTPS), a secure connection can be made, and the files can be uploaded. This is the simplest solution. You can use a condition in the IAM role that restricts access to a list of source IP addresses (your on-premise routed IPs).

Configuring a proxy on EC2 and using a VPC endpoint is not the simplest solution.

API Gateway is not suitable for performing IP whitelisting.

You cannot perform IP whitelisting on a VPN customer gateway.

Question 4 answer: *C*

Explanation:

Uploading using a pre-signed URL allows you to upload the object without having any AWS security credentials/permissions. Pre-signed URLs can be generated programmatically and anyone who receives a valid pre-signed URL can then programmatically upload an object. This solution bypasses the web server avoiding any performance bottlenecks.

Uploading to a second bucket (through the web server) does not solve the issue of the web server being the bottleneck.

Using Auto Scaling, ELB and fleets of EC2 instances (including Spot instances) is not the most efficient solution to the problem.

Question 5 answer: *C,D*

Explanation:

Glacier is designed for durability of 99.999999999% of objects across multiple Availability Zones. Data is resilient in the event of one entire Availability Zone destruction. Glacier supports SSL for data in transit and encryption of data at rest. Glacier is extremely low cost and is ideal for long-term archival.

Data is not resilient to the failure of an entire region.

Data is not replicated globally.

There is no availability SLA with Glacier.

Question 6 answer: *A,C*

Explanation:

You can control who can administer your file system using IAM. You can control access to files and directories with POSIX-compliant user and group-level permissions. POSIX permissions allows 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.

You cannot use AWS WAF to protect EFS data using users and groups. You do not use IAM to control access to files and directories by user and group, but you can use IAM to control who can administer the file system configuration.

You use EFS Security Groups to control network traffic to EFS, not Network ACLs.

Question 7 answer: *A*

Explanation:

You can specify the instance store volumes for your instance only when you launch an instance. You can't attach instance store volumes to an instance after you've launched it.

You can use a block device mapping to specify additional EBS volumes when you launch your instance, or you can attach additional EBS volumes after your instance is running.

An Elastic Network Adapter has nothing to do with adding instance store volumes.

Question 8 answer: *C*

Explanation:

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

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

File:

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

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

The question asks for an iSCSI (block) storage solution, so a file gateway is not the right solution

Volume:

- 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

The volume gateway is the correct solution choice as it provides iSCSI (block) storage which is compatible with the existing configuration

Tape:

- 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.

Amazon RDS

General RDS Concepts

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

RDS is an Online Transaction Processing (OLTP) type of database

Best for structured, relational data store requirements

Aims to be drop-in replacement for existing on-premise instances of the same databases

Automated backups and patching applied in customer-defined maintenance windows

Push-button scaling, replication and redundancy

Amazon RDS supports the following database engines

- Amazon Aurora
- MySQL
- MariaDB
- Oracle
- SQL Server
- PostgreSQL

RDS is a fully 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

Use Cases, Alternatives and Anti-Patterns

The table below provides guidance on when best to use RDS and several other AWS database/data store services:

Data Store	When to Use
Database on EC2	• Ultimate control over database • Preferred DB not available under RDS
Amazon RDS	• Need traditional relational database for OLTP • Your data is well-formed and structured • Existing apps requiring RDBMS
Amazon DynamoDB	• Name/value pair data or unpredictable data structure • In-memory performance with persistence • High I/O needs • Scale dynamically
Amazon RedShift	• Massive amounts of data • Primarily OLAP workloads
Amazon Neptune	• Relationships between objects a major portion of data value
Amazon ElastiCache	• Fast temporary storage for small amounts of data • Highly volatile data
Amazon S3	• BLOBs • Static websites

You can run databases on EC2. Consider the following points:

- You can run any database you like with full control and ultimate flexibility
- You must manage everything like backups, redundancy, patching and scaling
- Good option if you require a database not yet supported by RDS, such as IBM DB2 or SAP HANA
- Good option if it is not feasible to migrate to AWS-managed database

Anti-patterns are certain patterns in architecture or development that are considered bad, or sub-optimal practices - i.e. there may be a better service of method to produce the best result

The following table describes requirements that are not a good fit for RDS:

Requirement	More Suitable Service
Lots of large binary objects (BLOBs)	S3
Automated Scalability	DynamoDB
Name/Value Data Structure	DynamoDB
Data is not well structured or unpredictable	DynamoDB
Other database platforms like IBM DB2 or SAP HANA	EC2
Complete control over the database	EC2

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 RDS up (compute and storage)

You cannot decrease the allocated storage for an RDS instance

You can scale storage and change the storage type for all DB engines except MS SQL

For MS SQL the workaround is to create a new instance from a snapshot with the new configuration

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

Amazon Aurora supports a maximum DB size of 64 TiB

All other RDS DB types support a maximum DB size of 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	1,000–32,000 IOPS	200 GiB – 16 TiB
SQL Server, Web and Express editions	1,000–32,000 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

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 utilise **synchronous physical** replication

Multi-AZ deployments for the SQL Server engine use **synchronous logical** replication (SQL Server-native Mirroring technology)

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

The process for implementing maintenance activities is as follows:

- Perform operations on standby

- Promote standby to primary
- Perform operations on new standby (demoted primary)

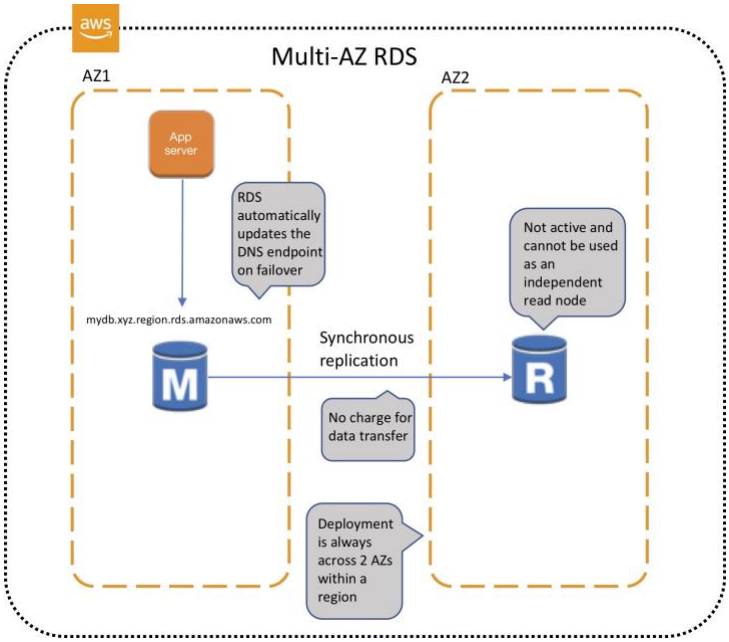
You can manually upgrade a DB instance to a supported DB engine version from the AWS Console

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 and Aurora (not SQL Server or Oracle)

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

Promotion of read replicas takes several minutes

Promoted read replicas retain:

- Backup retention window
- Backup window
- DB parameter group

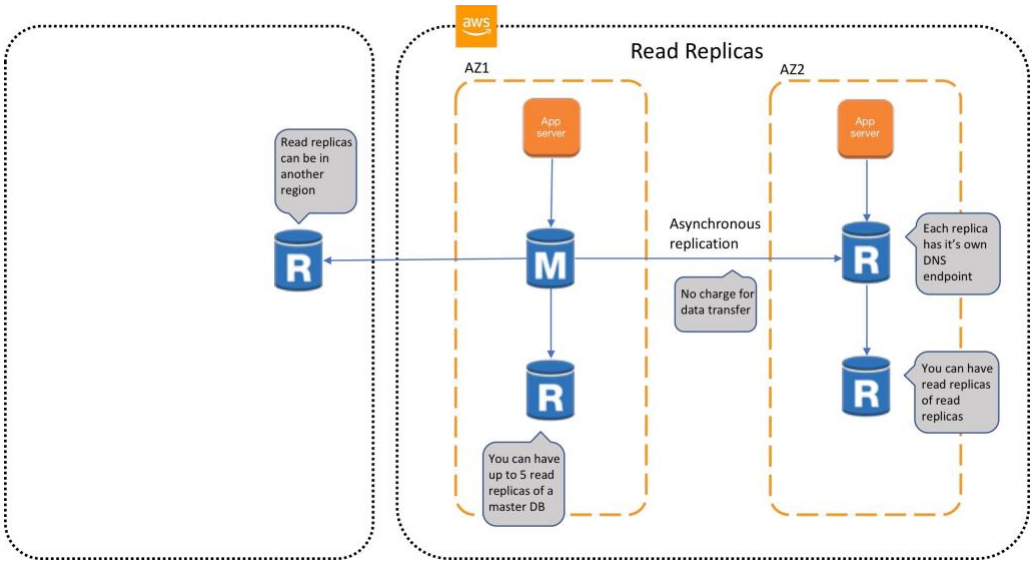
Existing read replicas continue to function as normal

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



Aurora

AWS proprietary database

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

Two types of replication: Aurora replica (up to 15), MySQL Read Replica (up to 5)

You can create read replicas for an Amazon Aurora database in up to five AWS regions. This capability is available for Amazon Aurora with MySQL compatibility

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

Self-healing

Automatic failover is available for Aurora replicas only

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

Backup

Automated Backups

Automated backups allow point in time recovery to any point within the retention period down to a second

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

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

You cannot restore from a DB snapshot to an existing DB - a new instance is created when you restore

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

Migration

AWS Database Migration Service helps you migrate databases to AWS quickly and securely

Use along with the Schema Conversion Tool (SCT) to migrate databases to AWS RDS or EC2-based databases

The source database remains fully operational during the migration, minimizing downtime to applications that rely on the database

The AWS Database Migration Service can migrate your data to and from most widely used commercial and open-source databases

Schema Conversion Tool can copy database schemas for homogenous migrations (same database) and convert schemas for heterogeneous migrations (different database)

DMS is used for smaller, simpler conversions and also supports MongoDB and DynamoDB

SCT is used for larger, more complex datasets like data warehouses

DMS has replication functions for on-premise to AWS or to Snowball or S3

Amazon DynamoDB

General DynamoDB Concepts

Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability

Multi-AZ NoSQL data store with Cross-Region Replication option

Push button scaling means that you can scale the DB at any time without incurring downtime

Defaults to eventual consistency reads but can request strongly consistent read via SDK parameter

Priced on throughput, rather than compute

Provision read and write capacity in anticipation of need

Autoscale capacity adjusts per configured min/max levels

On-Demand Capacity provides flexible capacity at a small premium cost

Can achieve ACID compliance with DynamoDB Transactions

SSD based and uses limited indexing on attributes for performance

DynamoDB is a Web service that uses HTTP over SSL (HTTPS) as a transport and JSON as a message serialisation format

Amazon DynamoDB stores three geographically distributed replicas of each table to enable high availability and data durability

Data is synchronously replicated across 3 facilities (AZs) in a region

Cross-region replication allows you to replicate across regions:

- Amazon DynamoDB global tables provides a fully managed solution for deploying a multi-region, multi-master database
- When you create a global table, you specify the AWS regions where you want the table to be available
- DynamoDB performs all of the necessary tasks to create identical tables in these regions, and propagate ongoing data changes to all of them

Provides low read and write latency

Scale storage and throughput up or down as needed without code changes or downtime

DynamoDB is schema-less

DynamoDB can be used for storing session state

Provides two read models

Eventually consistent reads (Default):

- The eventual consistency option maximises your read throughput (best read performance)
- An eventually consistent read might not reflect the results of a recently completed write
- Consistency across all copies reached within 1 second

Strongly consistent reads:

- A strongly consistent read returns a result that reflects all writes that received a successful response prior to the read (faster consistency)

Users/applications reading from DynamoDB tables can specify in their requests if they want strong consistency (default is eventually consistent)

Attributes consists of a name and a value or set of values

Attributes in DynamoDB are similar to fields or columns in other database systems

The primary key is the only required attribute for items in a table and it uniquely identifies each item

A primary key can either be one of the following types

Partition key:

- A simple primary key, composed of one attribute known as the partition key

Partition key and sort key:

- Referred to as a composite primary key
- Composed of two attributes: partition key and sort key

An item is a collection of attributes

The aggregate size of an item cannot exceed 400KB including keys and all attributes

Can store pointers to objects in S3, including items over 400KB

Tables are a collection of items and items are made up of attributes (columns)

Supports key-value and document data structures

Supports fast, in-place Atomic updates

Stores structured data in tables, indexed by a primary key

Supports GET/PUT operations using a user-defined primary key

DynamoDB provides flexible querying by letting you query on non-primary key attributes using Global Secondary Indexes and Local Secondary Indexes

You can create one or more secondary indexes on a table

A *secondary index* lets you query the data in the table using an alternate key, in addition to queries against the primary key

DynamoDB supports two kinds of secondary indexes:

- Global secondary index – An index with a partition key and sort key that can be different from those on the table
- Local secondary index – An index that has the same partition key as the table, but a different sort key

You can search using one of the following methods:

- Query operation - find items in a table or a secondary index using only the primary keys attributes
- Scan operation - reads every item in a table or a secondary index and by default will return all items

Use DynamoDB when relational features are not required and the DB is likely to need to scale

Not ideal for the following situations:

- Traditional RDS apps
- Joins and/or complex transactions
- BLOB data
- Large data with low I/O rate

DynamoDB Streams

DynamoDB Streams help you to keep a list of item level changes or provide a list of item level changes that have taken place in the last 24hrs

Amazon DynamoDB is integrated with AWS Lambda so that you can create triggers—pieces of code that automatically respond to events in DynamoDB Streams

If you enable DynamoDB Streams on a table, you can associate the stream ARN with a Lambda function that you write

Best practices

Keep item sizes small

If you are storing serial data in DynamoDB that will require actions based on date/time use separate tables for days, weeks, months

Store more frequently and less frequently accessed data in separate tables

If possible compress larger attribute values

Store objects larger than 400KB in S3 and use pointers (S3 Object ID) in DynamoDB

Integrations

ElastiCache can be used in front of DynamoDB for performance of reads on infrequently changed data

Triggers integrate with AWS Lambda to respond to triggers

Integration with RedShift:

- RedShift complements DynamoDB with advanced business intelligence
- When copying data from a DynamoDB table into RedShift you can perform complex data analysis queries including joins with other tables
- A copy operation from a DynamoDB table counts against the table's read capacity
- After data is copied, SQL queries do not affect the data in DynamoDB

DynamoDB is integrated with Apache Hive on EMR. Hive can allow you to:

- Read and write data in DynamoDB tables allowing you to query DynamoDB data using a SQL-like language (HiveQL)
- Copy data from a DynamoDB table to an S3 bucket and vice versa
- Copy data from a DynamoDB table into HDFS and vice versa
- Perform join operations on DynamoDB tables

Scalability

Push button scaling without downtime

You can scale down only 4 times per calendar day

AWS places some default limits on the throughput you can provision

These are the limits unless you request a higher amount:

US East (N. Virginia), US East (Ohio), US West (N. California), US West (Oregon), South America (São Paulo), EU (Frankfurt), EU (Ireland), Asia Pacific (Tokyo), Asia Pacific (Seoul), Asia Pacific (Singapore), Asia Pacific (Sydney), China (Beijing) Regions:

- Per table – 40,000 read capacity units and 40,000 write capacity units
- Per account – 80,000 read capacity units and 80,000 write capacity units

All Other Regions:

- Per table – 10,000 read capacity units and 10,000 write capacity units
- Per account – 20,000 read capacity units and 20,000 write capacity units

DynamoDB can throttle requests that exceed the provisioned throughput for a table

DynamoDB can also throttle read requests for an Index to prevent your application from consuming too many capacity units

When a request is throttled it fails with an HTTP 400 code (Bad Request) and a ProvisionedThroughputExceeded exception

Cross Region Replication with Global Tables

Amazon DynamoDB global tables provide a fully managed solution for deploying a multi-region, multi-master database

When you create a global table, you specify the AWS regions where you want the table to be available

DynamoDB performs all of the necessary tasks to create identical tables in these regions, and propagate ongoing data changes to all of them

DynamoDB global tables are ideal for massively scaled applications, with globally dispersed users

Global tables provide automatic multi-master replication to AWS regions world-wide, so you can deliver low-latency data access to your users no matter where they are located

A *global table* is a collection of one or more replica tables, all owned by a single AWS account.

A *replica table* (or *replica*, for short) is a single DynamoDB table that functions as a part of a global table. Each replica stores the same set of data items. Any given global table can only have one replica table per region

You can add replica tables to the global table, so that it can be available in additional AWS regions

With a global table, each replica table stores the same set of data items. DynamoDB does not support partial replication of only some of the items.

An application can read and write data to any replica table. If your application only uses eventually consistent reads, and only issues reads against one AWS region, then it will work without any modification.

However, if your application requires strongly consistent reads, then it must perform all of its strongly consistent reads and writes in the same region. DynamoDB does not support strongly consistent reads across AWS regions

It is important that each replica table and secondary index in your global table has identical write capacity settings to ensure proper replication of data

DynamoDB Auto Scaling

DynamoDB auto scaling uses the AWS Application Auto Scaling service to dynamically adjust provisioned throughput capacity on your behalf, in response to actual traffic patterns

This enables a table or a global secondary index to increase its provisioned read and write capacity to handle sudden increases in traffic, without throttling

When the workload decreases, Application Auto Scaling decreases the throughput so that you don't pay for unused provisioned capacity

How Application Auto Scaling works:

- You create a *scaling policy* for a table or a global secondary index
- The scaling policy specifies whether you want to scale read capacity or write capacity (or both), and the minimum and maximum provisioned capacity unit settings for the table or index.
- The scaling policy also contains a *target utilization*—the percentage of consumed provisioned throughput at a point in time
- Uses a *target tracking* algorithm to adjust the provisioned throughput of the table (or index) upward or downward in response to actual workloads, so that the actual capacity utilization remains at or near your target utilization

Currently, Auto Scaling does not scale down your provisioned capacity if your table's consumed capacity becomes zero

If you use the AWS Management Console to create a table or a global secondary index, DynamoDB auto scaling is enabled by default

Limits

256 tables per account per region

No limit on the size of a table

Read/write capacity unit limits vary per region

Capacity units

One read capacity unit represents one strongly consistent read per second, or two eventually consistent reads per second for items up to 4KB

For items larger than 4KB, DynamoDB consumes additional read capacity units

One write capacity unit represents one write per second for an item up to 1KB

Charges

There are two pricing models for DynamoDB:

- **On-demand capacity mode:** DynamoDB charges you for the data reads and writes your application performs on your tables. You do not need to specify how much read and write throughput you expect your application to perform because DynamoDB instantly accommodates your workloads as they ramp up or down

- **Provisioned capacity mode:** you specify the number of reads and writes per second that you expect your application to require. You can use auto scaling to automatically adjust your table's capacity based on the specified utilization rate to ensure application performance while reducing cost

Additional charges include:

- Data transfer out
- Backups per GB (continuous or on-demand)
- Global Tables
- DynamoDB Accelerator (DAX)
- DynamoDB Streams

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

Amazon ElastiCache

General ElastiCache Concepts

Fully managed implementations of two popular in-memory data stores – Redis and Memcached

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

Best for scenarios where the DB load is based on Online Analytics Processing (OLAP) transactions

Push-button scalability for memory, writes and reads

In-memory key/value store – not persistent in the traditional sense

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

Cached information may include the results of I/O-intensive database queries or the results of computationally-intensive calculations

Can be on-demand or reserved instances too (but not Spot instances)

Elasticache can be used for storing session state

A node is a fixed-sized chunk of secure, network-attached RAM and is the smallest building block

Each node runs an instance of the Memcached or Redis protocol-compliant service and has its own DNS name and port

Failed nodes are automatically replaced

Access to Elasticache nodes is controlled by VPC security groups and subnet groups (when deployed in a VPC)

Subnet groups are a collection of subnets designated for your Amazon ElastiCache Cluster

You cannot move an existing Amazon ElastiCache Cluster from outside VPC into a VPC

You need to configure subnet groups for Elasticache for the VPC that hosts the EC2 instances and the Elasticache cluster

When not using a VPC, Amazon ElastiCache allows you to control access to your clusters through Cache Security Groups (you need to link the corresponding EC2 Security Groups)

Elasticache nodes are deployed in clusters and can span more than one subnet of the same subnet group

A cluster is a collection of one or more nodes using the same caching engine

Applications connect to Elasticache clusters using endpoints

An endpoint is a node or cluster's unique address

Maintenance windows can be defined and allow software patching to occur

There are two types of ElastiCache engine:

- Memcached - simplest model, can run large nodes with multiple cores/threads, can be scaled in and out, can cache objects such as DBs
- Redis - complex model, supports encryption, master / slave replication, cross AZ (HA), automatic failover and backup/restore

Use Cases

The following table describes a few typical use cases for ElastiCache:

Use Case	Benefit
Web session store	In cases with load-balanced web servers, store web session information in Redis so if a server is lost, the session info is not lost and another web server can pick it up
Database caching	Use Memcached in front of AWS RDS to cache popular queries to offload work from RDS and return results faster to users
Leaderboards	Use Redis to provide a live leaderboard for millions of users of your mobile app
Streaming data dashboards	Provide a landing spot for streaming sensor data on the factory floor, providing live real-time dashboard displays

The table below describes the requirements that would determine whether to use the Memcached or Redis engine:

Memcached	Redis
Simple, no-frills	You need encryption
You need to scale-out and in as demand changes	You need HIPAA compliance
You need to run multiple CPU cores and threads	Support for clustering
You need to cache objects (e.g. database queries)	You need complex data types
	You need HA (replication
	Pub/Sub capability
	Geospacial Indexing
	Backup and restore

Memcached

Not persistent

Cannot be used as a data store

Supports large nodes with multiple cores or threads

Scales out and in, by adding and removing nodes

Ideal front-end for data stores (RDS, Dynamo DB etc.)

Use cases:

- Cache the contents of a DB
- Cache data from dynamically generated web pages
- Transient session data
- High frequency counters for admission control in high volume web apps

Max 100 nodes per region, 1-20 nodes per cluster (soft limits)

Can integrate with SNS for node failure/recovery notification

Supports auto-discovery for nodes added/removed from the cluster

Scales out/in (horizontally) by adding/removing nodes

Scales up/down (vertically) by changing the node family/type

Does not support multi-AZ failover or replication

Does not support snapshots

You can place nodes in different AZs

Redis

Data is persistent

Can be used as a datastore

Not multi-threaded

Scales by adding shards, not nodes

A Redis shard is a subset of the cluster's keyspace, that can include a primary node and zero or more read-replicas

Supports automatic and manual snapshots (S3)

Backups include cluster data and metadata

You can restore your data by creating a new Redis cluster and populating it from a backup

Supports master/slave replication

During backup you cannot perform CLI or API operations on the cluster

Automated backups are enabled by default (automatically deleted with Redis deletion)

You can only move snapshots between regions by exporting them from Elasticache before moving between regions (can then populate a new cluster with data)

Multi-AZ is possible using read replicas in another AZ in the same region

Clustering mode disabled:

- You can only have one shard

- One shard can have one read/write primary node and 0-5 read only replicas
- You can distribute the replicas over multiple AZs in the same region
- Replication from the primary node is asynchronous

Clustering mode enabled:

- Can have up to 15 shards
- Each shard can have one primary node and 0-5 read only replicas
- Taking snapshots can slow down nodes, best to take from the read replicas

Multi-AZ failover:

- Failures are detected by Elasticache
- Elasticache automatically promotes the replica that has the lowest replica lag
- DNS records remain the same but point to the IP of the new primary
- Other replicas start to sync with the new primary

You can have a fully automated, fault tolerant Elasticache-Redis implementation by enabling both cluster mode and multi-AZ failover

The following table compares the Memcached and Redis engines:

	Memcached	Redis (cluster mode disabled)	Redis (cluster mode enabled)
Engine versions	1.4.x	2.8.x and 3.2.x	3.2.x
Data types	Simple	Complex	Complex
Data partitioning	Yes	No	Yes
Cluster is modifiable	Yes	Yes	No
Online re-sharding	No	No	3.2.10
Encryption	No	3.2.6	3.2.6
HIPAA Compliance	No	3.2.6	3.2.6
Multi-threaded	Yes	No	No
Node type upgrade	No	Yes	No
Engine upgrading	Yes	Yes	No
High availability (replication)	No	Yes	Yes
Automatic failover	No	Optional	Required
Pub/Sub capabilities	No	Yes	Yes
Sorted sets	No	Yes	Yes
Backup and restore	No	Yes	Yes
Geospatial indexing	No	Yes	Yes

Charges

Pricing is per Node-hour consumed for each Node Type

Partial Node-hours consumed are billed as full hours

There is no charge for data transfer between Amazon EC2 and Amazon Elasticache within the same Availability Zone

High Availability for ElastiCache

Memcached:

- Because Memcached does not support replication, a node failure will result in data loss
- Use multiple nodes in each shard to minimize data loss on node failure
- Launch multiple nodes across available AZs to minimize data loss on AZ failure

Redis:

- Use multiple nodes in each shard and distribute the nodes across multiple AZs
- Enable Multi-AZ on the replication group to permit automatic failover if the primary nodes fails
- Schedule regular backups of your Redis cluster

Amazon RedShift

General RedShift Concepts

Amazon Redshift is a fast, fully managed data warehouse that makes it simple and cost-effective to analyze all your data using standard SQL and existing Business Intelligence (BI) tools

Clustered peta-byte scale data warehouse

RedShift is a SQL based data warehouse used for **analytics** applications

RedShift is an Online Analytics Processing (OLAP) type of DB

RedShift is used for running complex analytic queries against petabytes of structured data, using sophisticated query optimization, columnar storage on high-performance local disks, and massively parallel query execution

RedShift is ideal for **processing** large amounts of data for business intelligence

Extremely cost-effective as compared to some other on-premises data warehouse platforms

PostgreSQL compatible with JDBC and ODBC drivers available; compatible with most Business Intelligence tools out of the box

Features parallel processing and columnar data stores which are optimized for complex queries

Option to query directly from data files on S3 via RedShift Spectrum

RedShift is 10x faster than a traditional SQL DB

RedShift can store huge amounts of data but cannot ingest huge amounts of data in real time

RedShift uses columnar data storage:

- Data is stored sequentially in columns instead of rows
- Columnar based DB is ideal for data warehousing and analytics
- Requires fewer I/Os which greatly enhances performance

RedShift provides advanced compression:

- Data is stored sequentially in columns which allows for much better performance and less storage space
- RedShift automatically selects the compression scheme

RedShift provides good query performance and compression

RedShift provides Massively Parallel Processing (MPP) by distributing data and queries across all nodes

RedShift uses EC2 instances so you need to choose your instance type/size for scaling compute vertically, but you can also scale horizontally by adding more nodes to the cluster

You cannot have direct access to your AWS RedShift cluster nodes as a user, but you can through applications

HDD and SSD storage options

The size of a single node is 160GB and clusters can be created up to a petabyte or more

Multi-node consists of:

Leader node:

- Manages client connections and receives queries
- Simple SQL end-point
- Stores metadata
- Optimizes query plan
- Coordinates query execution

Compute nodes:

- Stores data and performs queries and computations
- Local columnar storage
- Parallel/distributed execution of all queries, loads, backups, restores, resizes
- Up to 128 compute nodes

Amazon RedShift Spectrum is a feature of Amazon Redshift that enables you to run queries against exabytes of unstructured data in Amazon S3, with no loading or ETL required

Availability and Durability

RedShift uses replication and continuous backups to enhance availability and improve durability and can automatically recover from component and node failures

Only available in one AZ but you can restore snapshots into another AZ

Alternatively, you can run data warehouse clusters in multiple AZ's by loading data into two Amazon Redshift data warehouse clusters in separate AZs from the same set of Amazon S3 input files

Redshift replicates your data within your data warehouse cluster and continuously backs up your data to Amazon S3

RedShift always keeps three copies of your data:

- The original
- A replica on compute nodes (within the cluster)
- A backup copy on S3

RedShift provides continuous/incremental backups:

- Multiple copies within a cluster
- Continuous and incremental backups to S3
- Continuous and incremental backups across regions
- Streaming restore

RedShift provides fault tolerance for the following failures:

- Disk failures
- Nodes failures
- Network failures
- AZ/region level disasters

For nodes failures the data warehouse cluster will be unavailable for queries and updates until a replacement node is provisioned and added to the DB

High availability for RedShift:

- Currently, RedShift does not support Multi-AZ deployments
- The best HA option is to use multi-node cluster which supports data replication and node recovery
- A single not RedShift cluster does not support data replication and you'll have to restore from a snapshot on S3 if a drive fails

RedShift can asynchronously replicate your snapshots to S3 in another region for DR

Single-node clusters do not support data replication (in a failure scenario you would need to restore from a snapshot)

Scaling requires a period of unavailability of a few minutes (typically during the maintenance window)

During scaling operations RedShift moves data in parallel from the compute nodes in your existing data warehouse cluster to the compute nodes in your new cluster

By default, Amazon Redshift retains backups for 1 day. You can configure this to be as long as 35 days

If you delete the cluster you can choose to have a final snapshot taken and retained

Manual backups are not automatically deleted when you delete a cluster

Security

You can load encrypted data from S3

Supports SSL Encryption in-transit between client applications and Redshift data warehouse cluster

VPC for network isolation

Encryption for data at rest (AES 256)

Audit logging and AWS CloudTrail integration

RedShift takes care of key management or you can manage your own through HSM or KMS

Charges

Charged for compute nodes hours, 1 unit per hour (only compute node, not leader node)

Backup storage - storage on S3

Data transfer - no charge for data transfer between RedShift and S3 within a region but for other scenarios you may pay charges

Database Practice Questions

Answers and explanations are provided below after the last question in this section.

Question 1:

Your company is starting to use AWS to host new web-based applications. A new two-tier application will be deployed that provides customers with access to data records. It is important that the application is highly responsive and retrieval times are optimized. You're looking for a persistent data store that can provide the required performance.

From the list below what AWS service would you recommend for this requirement?

A. ElastiCache with the Memcached engine
B. ElastiCache with the Redis engine

C. Kinesis Data Streams
D. RDS in a multi-AZ configuration

Question 2:

You are planning to launch a RedShift cluster for processing and analyzing a large amount of data. The RedShift cluster will be deployed into a VPC with multiple subnets. Which construct is used when provisioning the cluster to allow you to specify a set of subnets in the VPC that the cluster will be deployed into?

A. DB Subnet Group
B. Subnet Group
C. Availability Zone (AZ)
D. Cluster Subnet Group

Question 3:

A customer has asked you to recommend the best solution for a highly available database. The database is a relational OLTP type of database and the customer does not want to manage the operating system the database runs on. Failover between AZs must be automatic.

Which of the below options would you suggest to the customer?

A. Use DynamoDB
B. Use RDS in a Multi-AZ configuration
C. Install a relational database on EC2 instances in multiple AZs and create a cluster
D. Use RedShift in a Multi-AZ configuration

Question 4:

Your company runs a two-tier application on the AWS cloud that is composed of a web front-end and an RDS database. The web front-end uses multiple EC2 instances in multiple Availability Zones (AZ) in an Auto Scaling group behind an Elastic Load Balancer. Your manager is concerned about a single point of failure in the RDS database layer.

What would be the most effective approach to minimizing the risk of an AZ failure causing an outage to your database layer?

A. Take a snapshot of the database
B. Increase the DB instance size
C. Create a Read Replica of the RDS DB instance in another AZ
D. Enable Multi-AZ for the RDS DB instance

Question 5:

An application you manage exports data from a relational database into an S3 bucket. The data analytics team wants to import this data into a RedShift cluster in a VPC in the same account. Due to the data being sensitive the security team has instructed you to ensure that the data traverses the VPC without being routed via the public Internet.

Which combination of actions would meet this requirement? (choose 2)

A. Enable Amazon RedShift Enhanced VPC routing
B. Create a cluster Security Group to allow the Amazon RedShift cluster to access Amazon S3
C. Create a NAT gateway in a public subnet to allows the Amazon RedShift cluster to access Amazon S3
D. Set up a NAT gateway in a private subnet to allow the Amazon RedShift cluster to access Amazon S3
E. Create and configure an Amazon S3 VPC endpoint

Question 6:

A Solutions Architect requires a highly available database that can deliver an extremely low RPO. Which of the following configurations uses synchronous replication?

A. RDS Read Replica across AWS regions
B. DynamoDB Read Replica
C. RDS DB instance using a Multi-AZ configuration
D. EBS volume synchronization

Question 7:

A company is launching a new application and expects it to be very popular. The company requires a database layer that can scale along with the application. The schema will be frequently changes and the application cannot afford any downtime for database changes.

Which AWS service allows the company to achieve these requirements?

A. Amazon Aurora
B. Amazon RDS MySQL
C. Amazon DynamoDB
D. Amazon RedShift

Question: 8

You are a Solutions Architect at Digital Cloud Training. One of your clients runs an application that writes data to a DynamoDB table. The client has asked how they can implement a function that runs code in response to item level changes that take place in the DynamoDB table.

What would you suggest to the client?

A. Enable server access logging and create an event source mapping between AWS Lambda and the S3 bucket to which the logs are written
B. Enable DynamoDB Streams and create an event source mapping between AWS Lambda and the relevant stream
C. Create a local secondary index that records item level changes and write some custom code that responds to updates to the index
D. Use Kinesis Data Streams and configure DynamoDB as a producer

Question 1 answer: *B*

Explanation:

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

There are two different database engines with different characteristics as per below:

Memcached

- Not persistent
- Cannot be used as a data store
- Supports large nodes with multiple cores or threads
- Scales out and in, by adding and removing nodes

Redis

- Data is persistent
- Can be used as a datastore
- Not multi-threaded
- Scales by adding shards, not nodes

Kinesis Data Streams is used for processing streams of data, it is not a persistent data store

RDS is not the optimum solution due to the requirement to optimize retrieval times which is a better fit for an in-memory data store such as ElastiCache

Question 2 answer: *D*

Explanation:

You create a cluster subnet group if you are provisioning your cluster in your virtual private cloud (VPC). A cluster subnet group allows you to specify a set of subnets in your VPC.

When provisioning a cluster, you provide the subnet group and Amazon Redshift creates the cluster on one of the subnets in the group.

A DB Subnet Group is used by RDS.

A Subnet Group is used by ElastiCache.

Availability Zones are part of the AWS global infrastructure, subnets reside within AZs but in RedShift you provision the cluster into Cluster Subnet Groups.

Question 3 answer: *B*

Explanation:

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. With RDS you can configure Multi-AZ which creates a replica in another AZ and synchronously replicates to it (DR only).

RedShift is used for analytics OLAP not OLTP.

If you install a DB on an EC2 instance you will need to manage to OS yourself and the customer wants it to be managed for them.

DynamoDB is a managed database of the NoSQL type. NoSQL DBs are not relational DBs.

Question 4 answer: *D*

Explanation:

Multi-AZ RDS creates a replica in another AZ and synchronously replicates to it. This provides a DR solution as if the AZ in which the primary DB resides fails, multi-AZ will automatically fail over to the replica instance with minimal downtime.

Read replicas are used for read heavy DBs and replication is asynchronous. Read replicas do not provide HA/DR as you cannot fail over to a read replica. They are used purely for offloading read requests from the primary DB.

Taking a snapshot of the database is useful for being able to recover from a failure so you can restore the database. However, this does not prevent an outage from happening as there will be significant downtime while you try and restore the snapshot to a new DB instance in another AZ.

Increasing the DB instance size will not provide any benefits to enabling high availability or fault tolerance, it will only serve to improve the performance of the DB.

Question 5 answer: *A,E*

Explanation:

Amazon RedShift Enhanced VPC routing forces all COPY and UNLOAD traffic between clusters and data repositories through a VPC.

Implementing an S3 VPC endpoint will allow S3 to be accessed from other AWS services without traversing the public network. Amazon S3 uses the Gateway Endpoint type of VPC endpoint with which a target for a specified route is entered into the VPC route table and used for traffic destined to a supported AWS service.

Cluster Security Groups are used with RedShift on EC2-Classic VPCs, regular security groups are used in EC2-VPC.

A NAT Gateway is used to allow instances in a private subnet to access the Internet and is of no use in this situation.

Question 6 answer: *C*

Explanation:

A Recovery Point Objective (RPO) relates to the amount of data loss that can be allowed, in this case a low RPO means that you need to minimize the amount of data lost so synchronous replication is required. Out of the options presented only Amazon RDS in a multi-AZ configuration uses synchronous replication.

RDS Read Replicas use asynchronous replication and are not used for DR.

DynamoDB Read Replicas do not exist.

EBS volume synchronization does not exist.

Question 7 answer: *C*

Explanation:

DynamoDB a NoSQL DB which means you can change the schema easily. It's also the only DB in the list that you can scale without any downtime.

Amazon Aurora, RDS MySQL and RedShift all require changing instance sizes in order to scale which causes an outage. They are also all relational databases (SQL) so changing the schema is difficult.

Question 8 answer: *B*

Explanation:

DynamoDB Streams help you to keep a list of item level changes or provide a list of item level changes that have taken place in the last 24hrs. Amazon DynamoDB is integrated with AWS Lambda so that you can create triggers—pieces of code that automatically respond to events in DynamoDB Streams.

If you enable DynamoDB Streams on a table, you can associate the stream ARN with a Lambda function that you write. Immediately after an item in the table is modified, a new record appears in the table's stream. AWS Lambda polls the stream and invokes your Lambda function synchronously when it detects new stream records.

An event source mapping identifies a poll-based event source for a Lambda function. It can be either an Amazon Kinesis or DynamoDB stream. Event sources maintain the mapping configuration except for stream-based services (e.g. DynamoDB, Kinesis) for which the configuration is made on the Lambda side and Lambda performs the polling.

You cannot configure DynamoDB as a Kinesis Data Streams producer.

You can write Lambda functions to process S3 bucket events, such as the object-created or object-deleted events. For example, when a user uploads a photo to a bucket, you might want Amazon S3 to invoke your Lambda function so that it reads the image and creates a thumbnail for the photo. However, the questions asks for a solution that runs code in response to changes in a DynamoDB table, not an S3 bucket.

A local secondary index maintains an alternate sort key for a given partition key value, it does not record item level changes.

MIGRATION

AWS Snowball

General

Petabyte scale data transport solution for transferring data into or out of AWS

Uses a secure storage device for physical transportation

AWS Snowball Client is software that is installed on a local computer and is used to identify, compress, encrypt, and transfer data

Uses 256-bit encryption (managed with the AWS KMS) and tamper-resistant enclosures with TPM

Snowball must be ordered from and returned to the same region

To speed up data transfer it is recommended to run simultaneous instances of the AWS Snowball Client in multiple terminals and transfer small files as batches

Snowball can import to S3 or export from S3

The Snowball Family

Several services are offered in the Snowball family

The table below describes these at a high-level:

Service	What it Is
AWS Import/Export	Ship an external hard drive to AWS. Someone at AWS plugs it in and copies your data to S3
AWS Snowball	Ruggedized NAS in a box that AWS ships to you. You can copy up to 80TB of data and ship it back to AWS. They copy the data over to S3
AWS Snowball Edge	Same as Snowball, but with onboard Lambda and clustering
AWS Snowmobile	A literal shipping container full of storage (up to 100PB) and a truck to transport it

Snowball (80TB) (50TB model available only in the USA)

Snowball Edge (100TB) comes with onboard storage and compute capabilities

Snowmobile - exabyte scale with up to 100PB per Snowmobile

AWS Import/export is when you send your own disks into AWS - this is being deprecated in favour of Snowball

Migration Practice Questions

Answers and explanations are provided below after the last question in this section.

Question 1:

The financial institution you are working for stores large amounts of historical transaction records. There are over 25TB of records and your manager has decided to move them into the AWS Cloud. You are planning to use Snowball as copying the data would take too long. Which of the statements below are true regarding Snowball? (choose 2)

A. Snowball can import to S3 but cannot export from S3
B. Uses a secure storage device for physical transportation
C. Can be used with multipart upload
D. Petabyte scale data transport solution for transferring data into or out of AWS
E. Snowball can be used for migration on-premise to on-premise

Question 1 answer: B,D

Explanation:

Snowball is a petabyte scale data transport solution for transferring data into or out of AWS. It uses a secure storage device for physical transportation.

The AWS Snowball Client is software that is installed on a local computer and is used to identify, compress, encrypt, and transfer data. It uses 256-bit encryption (managed with the AWS KMS) and tamper-resistant enclosures with TPM.

Snowball can import to S3 or export from S3.

Snowball cannot be used with multipart upload.

You cannot use Snowball for migration between on-premise data centers.

NETWORKING AND CONTENT DELIVERY

Amazon VPC

General

Amazon VPC lets you provision a logically isolated section of the Amazon Web Services (AWS) cloud where you can launch AWS resources in a virtual network that you define

Analogous to having your own DC inside AWS

Provides complete control over the virtual networking environment including selection of IP ranges, creation of subnets, and configuration of route tables and gateways

A VPC is logically isolated from other VPCs on AWS

Possible to connect the corporate data centre to a VPC using a hardware VPN (site-to-site)

VPCs are region wide

A default VPC is created in each region with a subnet in each AZ

By default you can create up to 5 VPCs per region

You can define dedicated tenancy for a VPC to ensure instances are launched on dedicated hardware (overrides the configuration specified at launch)

A default VPC is automatically created for each AWS account the first time Amazon EC2 resources are provisioned

The default VPC has all-public subnets

Public subnets are subnets that have:

- “Auto-assign public IPv4 address” set to “Yes”
- The subnet route table has an attached Internet Gateway

Instances in the default VPC always have both a public and private IP address

AZs names are mapped to different zones for different users (i.e. the AZ "ap-southeast-2a" may map to a different physical zone for a different user)

Components of a VPC:

- **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 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

Options for connecting to a VPC are:

- Hardware based VPN
- Direct Connect
- VPN CloudHub
- Software VPN

Routing

The VPC router performs routing between AZs within a region

The VPC router connects different AZs together and connects the VPC to the Internet Gateway

Each subnet has a route table the router uses to forward traffic within the VPC

Route tables also have entries to external destinations

Up to 200 route tables per VPC

Up to 50 route entries per route table

Each subnet can only be associated with one route table

Can assign one route table to multiple subnets

If no route table is specified a subnet will be assigned to the main route table at creation time

Cannot delete the main route table

You can manually set another route table to become the main route table

There is a default rule that allows all VPC subnets to communicate with one another - this cannot be deleted or modified

Routing between subnets is always possible because of this rule - any problems communicating is more likely to be security groups or NACLs

Subnets and Subnet Sizing

Types of subnet:

- If a subnet's traffic is routed to an internet gateway, the subnet is known as a **public subnet**

- If a subnet doesn't have a route to the internet gateway, the subnet is known as a **private subnet**
- If a subnet doesn't have a route to the internet gateway, but has its traffic routed to a virtual private gateway for a VPN connection, the subnet is known as a **VPN-only subnet**

The VPC is created with a master address range (CIDR block, can be anywhere from 16-28 bits), and subnet ranges are created within that range

New subnets are always associated with the default route table

Once the VPC is created you cannot change the CIDR block

You cannot create additional CIDR blocks that overlap with existing CIDR blocks

You cannot create additional CIDR blocks in a different RFC 1918 range

Subnets with overlapping IP address ranges cannot be created

The first 4 and last 1 IP addresses in a subnet are reserved

Subnets are created within availability zones (AZs)

Each subnet must reside entirely within one Availability Zone and cannot span zones

Availability Zones are distinct locations that are engineered to be isolated from failures in other Availability Zones

Availability Zones are connected with low latency, high throughput, and highly redundant networking

Can create private, public or VPN subnets

Subnets map 1:1 to AZs and cannot span AZs

You can only attach one Internet gateway to a custom VPC

IPv6 addresses are all public and the range is allocated by AWS

Internet Gateways

An Internet Gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in your VPC and the internet

An Internet Gateway serves two purposes:

- To provide a target in your VPC route tables for internet-routable traffic
- To perform network address translation (NAT) for instances that have been assigned public IPv4 addresses

Internet Gateways (IGW) must be created and then attached to a VPC, be added to a route table, and then associated with the relevant subnet(s)

No availability risk or bandwidth constraints

If your subnet is associated with a route to the Internet, then it is a public subnet

You cannot have multiple Internet Gateways in a VPC

IGW is horizontally scaled, redundant and HA

IGW performs NAT between private and public IPv4 addresses

IGW supports IPv4 and IPv6

IGWs must be detached before they can be deleted

Can only attach 1 IGW to a VPC at a time

Gateway terminology:

- Internet gateway (IGW) - AWS VPC side of the connection to the public Internet
- Virtual private gateway (VPG) - VPC endpoint on the AWS side
- Customer gateway (CGW) - representation of the customer end of the connection

To enable access to or from the Internet for instances in a VPC subnet, you must do the following:

- Attach an Internet Gateway to your VPC
- Ensure that your subnet's route table points to the Internet Gateway (see below)
- Ensure that instances in your subnet have a globally unique IP address (public IPv4 address, Elastic IP address, or IPv6 address).
- Ensure that your network access control and security group rules allow the relevant traffic to flow to and from your instance

Must update subnet route table to point to IGW, either:

- To all destinations, e.g. 0.0.0.0/0 for IPv4 or ::/0for IPv6
- To specific public IPv4 addresses, e.g. your company's public endpoints outside of AWS

Egress-only Internet Gateway:

- Provides outbound Internet access for IPv6 addressed instances
- Prevents inbound access to those IPv6 instances
- IPv6 addresses are globally unique and are therefore public by default
- Stateful – forwards traffic from instance to Internet and then sends back the response
- Must create a custom route for ::/0 to the Egress-Only Internet Gateway
- Use Egress-Only Internet Gateway instead of NAT for IPv6

Elastic Network Interfaces and IP Addresses

An Elastic Network Interface (ENI) is a logical networking component that represents a NIC

ENIs can be attached and detached from EC2 instances and the configuration of the ENI will be maintained

Every EC2 instance has a primary interface known as eth0 which cannot be detached

An Elastic IP address is a static IPv4 address that is associated with an instance or network interface

Amazon charges for Elastic IP addresses that are not associated with a running instance, or that are associated with stopped instances or an unattached network interface

Elastic IPs are retained in your account whereas auto-assigned public IPs are released

You can have up to 5 elastic IPs per account

VPC Wizard

VPC with a Single Public Subnet:

- Your instances run in a private, isolated section of the AWS cloud with direct access to the Internet
- Network access control lists and security groups can be used to provide strict control over inbound and outbound network traffic to your instances
- Creates a /16 network with a /24 subnet. Public subnet instances use Elastic IPs or Public IPs to access the Internet

VPC with Public and Private Subnets:

- In addition to containing a public subnet, this configuration adds a private subnet whose instances are not addressable from the Internet
- Instances in the private subnet can establish outbound connections to the Internet via the public subnet using Network Address Translation (NAT)
- Creates a /16 network with two /24 subnets
- Public subnet instances use Elastic IPs to access the Internet
- Private subnet instances access the Internet via Network Address Translation (NAT)

VPC with Public and Private Subnets and Hardware VPN Access:

- This configuration adds an IPsec Virtual Private Network (VPN) connection between your Amazon VPC and your data center - effectively extending your data center to the cloud while also providing direct access to the Internet for public subnet instances in your Amazon VPC
- Creates a /16 network with two /24 subnets
- One subnet is directly connected to the Internet while the other subnet is connected to your corporate network via an IPsec VPN tunnel

VPC with a Private Subnet Only and Hardware VPN Access:

- Your instances run in a private, isolated section of the AWS cloud with a private subnet whose instances are not addressable from the Internet
- You can connect this private subnet to your corporate data center via an IPsec Virtual Private Network (VPN) tunnel
- Creates a /16 network with a /24 subnet and provisions an IPsec VPN tunnel between your Amazon VPC and your corporate network

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 HA)

HA can be achieved by using Auto Scaling groups, multiple subnets in different AZ's and a script to automate failover

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 use as a bastion (jump) host

Can monitor traffic metrics

Not supported for IPv6 (use Egress-Only Internet Gateway)

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

Up to 5 Gbps bandwidth that can scale up to 45 Gbps

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

They are preferred by enterprises

No need to patch

Not associated with any security groups

Automatically assigned a public IP address

Remember to update route tables and point towards your gateway

More secure (e.g. you cannot access with SSH and there are no security groups to maintain)

No need to disable source/destination checks

Egress only NAT gateways operate on IPv6 whereas NAT gateways operate on IPv4

Port forwarding is not supported

Using the NAT Gateway as a Bastion host server is not supported

Traffic metrics are not supported

The table below highlights the key differences between both types of gateway:

	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

By default, custom security groups do not have inbound allow rules (all inbound traffic is denied by default)

By default, default security groups do have inbound allow rules (allowing traffic from within the group)

All outbound traffic is allowed by default in custom and default security groups

You cannot delete the security group that's created by default within a VPC

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

You can use the security group name 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

Up to 5 security groups can be added per EC2 instance interface

There is no limit on the number of EC2 instances within a security group

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

Network ACL's

Network ACL's function at the subnet level

The VPC router hosts the network ACL function

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

Recommended to leave spacing between network ACL numbers

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

Also recommended to have software firewalls installed on your instances

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 Connectivity

There are several methods of connecting to a VPC. These include:

- AWS Managed VPN
- AWS Direct Connect
- AWS Direct Connect plus a VPN
- AWS VPN CloudHub
- Software VPN
- Transit VPC
- VPC Peering
- AWS PrivateLink
- VPC Endpoints

Each of these will be further detailed below

AWS Managed VPN

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

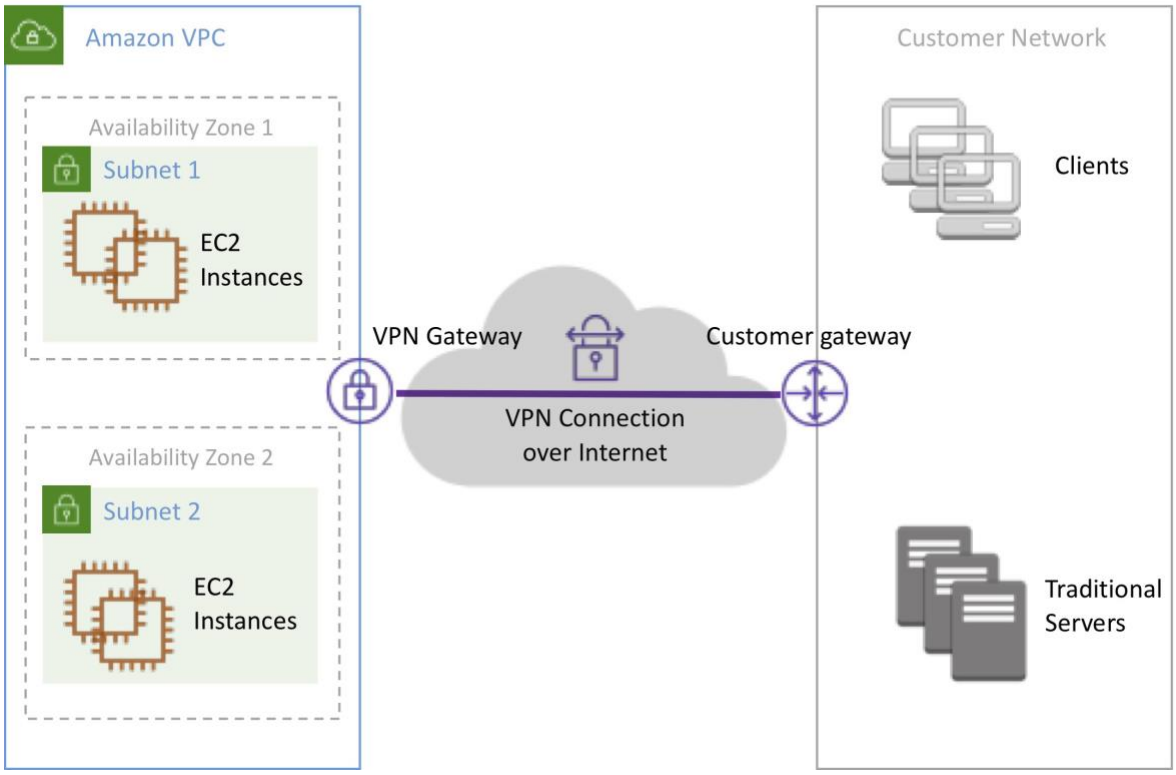
VPNs are quick, easy to deploy, and cost effective

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

A Customer Gateway is required on the customer side

The diagram below depicts an AWS Managed VPN configuration:

AWS Managed VPN



An Internet routable IP address is required on the customer gateway

Two tunnels per connection must be configured for redundancy

You cannot use a NAT gateway in AWS for clients coming in via a VPN

For route propagation you need to point your VPN-only subnet's route tables at the VGW

Must define the IP prefixes that can send/receive traffic through the VGW

VGW does not route traffic destined outside of the received BGP advertisements, static route entries, or its attached VPC CIDR

Cannot access Elastic IPs on your VPC via the VPN - Elastic IPs can only be connected to via the Internet

AWS Direct Connect

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

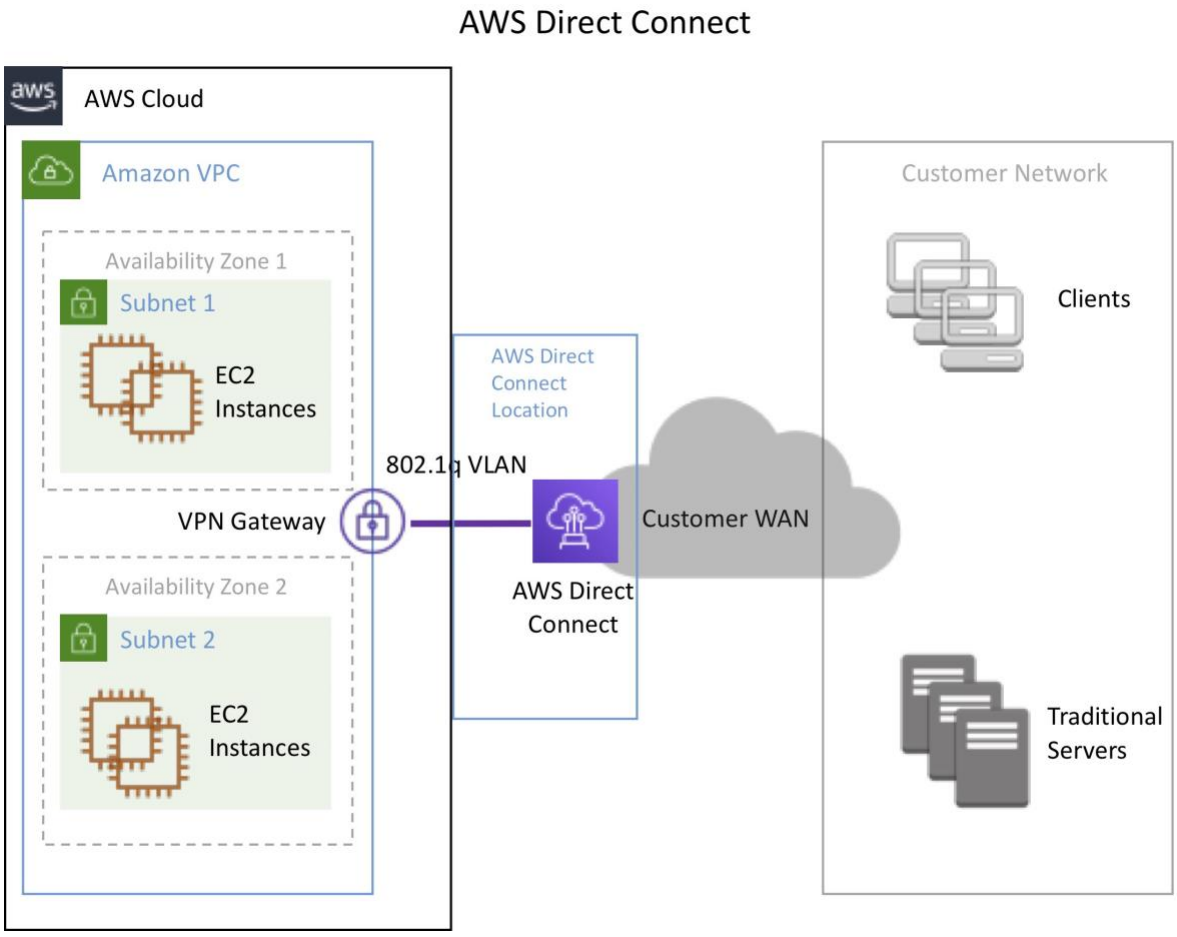
Using AWS Direct Connect, you can establish private connectivity between AWS and your data center, office, or collocated environment

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

The diagram below depicts an AWS Direct Connect configuration:



AWS Direct Connect Plus VPN

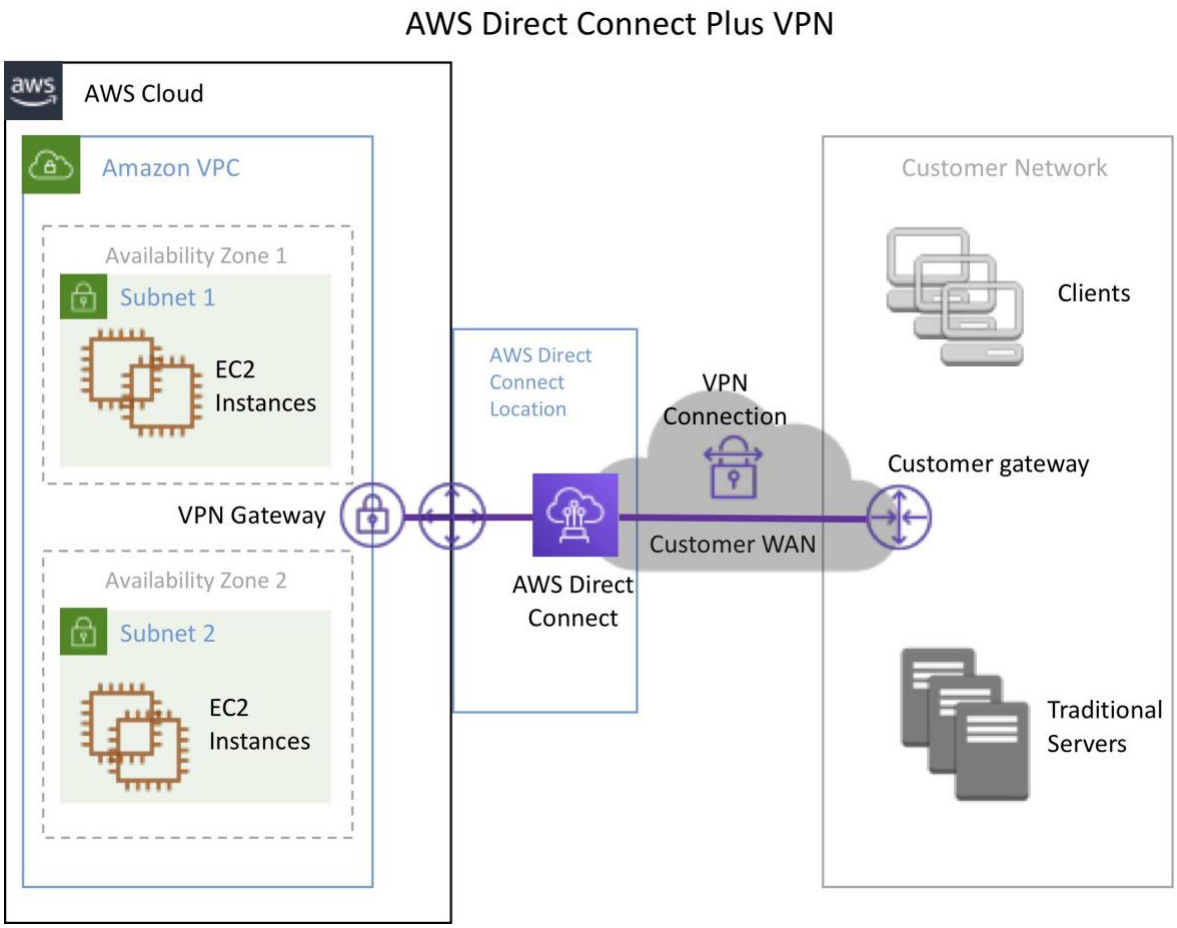
With AWS Direct Connect plus VPN, you can combine one or more AWS Direct Connect dedicated network connections with the Amazon VPC VPN

This combination provides an IPsec-encrypted private connection that also reduces network costs, increases bandwidth throughput, and provides a more consistent network experience than internet-based VPN connections

You can use AWS Direct Connect to establish a dedicated network connection between your network create a logical connection to public AWS resources, such as an Amazon virtual private gateway IPsec endpoint

This solution combines the AWS managed benefits of the VPN solution with low latency, increased bandwidth, more consistent benefits of the AWS Direct Connect solution, and an end-to-end, secure IPsec connection

The diagram below depicts an AWS Direct Connect plus VPN configuration:



AWS VPN CloudHub

The AWS VPN CloudHub operates on a simple hub-and-spoke model that you can use with or without a VPC

Use this design if you have multiple branch offices and existing internet connections and would like to implement a convenient, potentially low-cost hub-and-spoke model for primary or backup connectivity between these remote offices

VPN CloudHub is used for hardware-based VPNs and allows you to configure your branch offices to go into a VPC and then connect that to the corporate DC (hub and spoke topology with AWS as the hub)

Can have up to 10 IPSec tunnels on a VGW by default

Uses eBGP

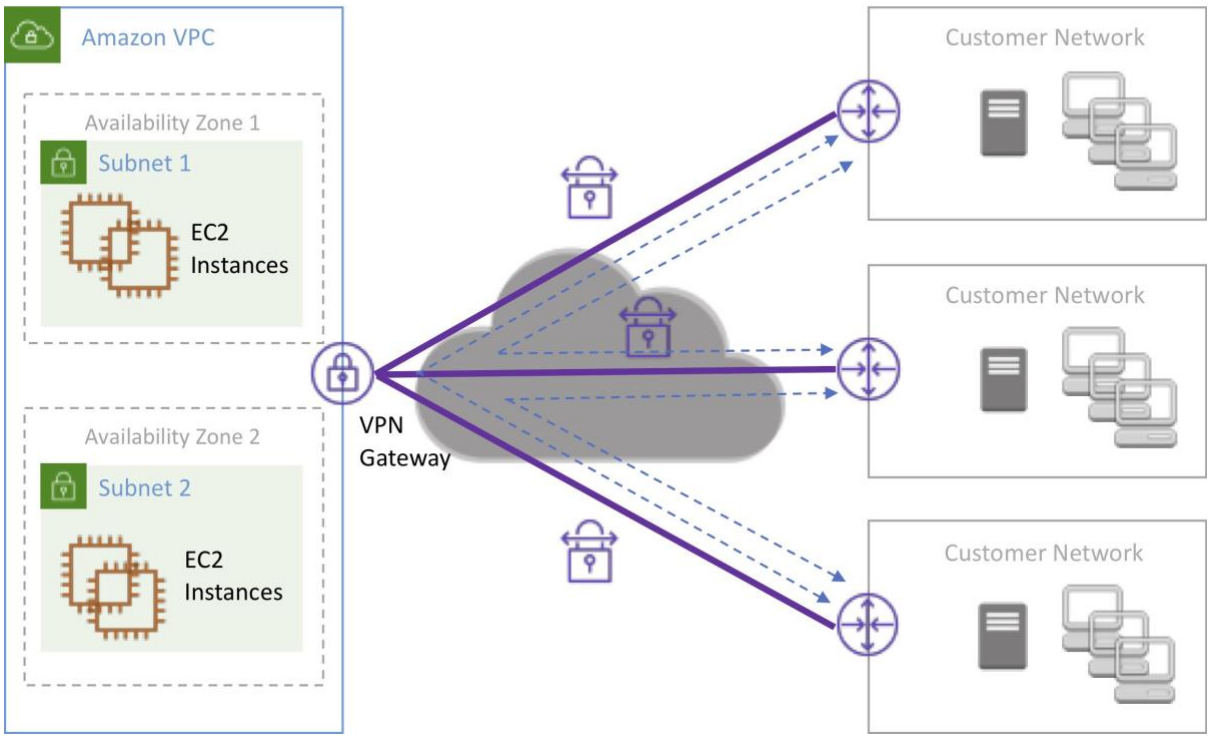
Branches can talk to each other (and provides redundancy)

Can have Direct Connect connections

Hourly rates plus data egress charges

The diagram below depicts an AWS VPN CloudHub configuration:

AWS VPN CloudHub



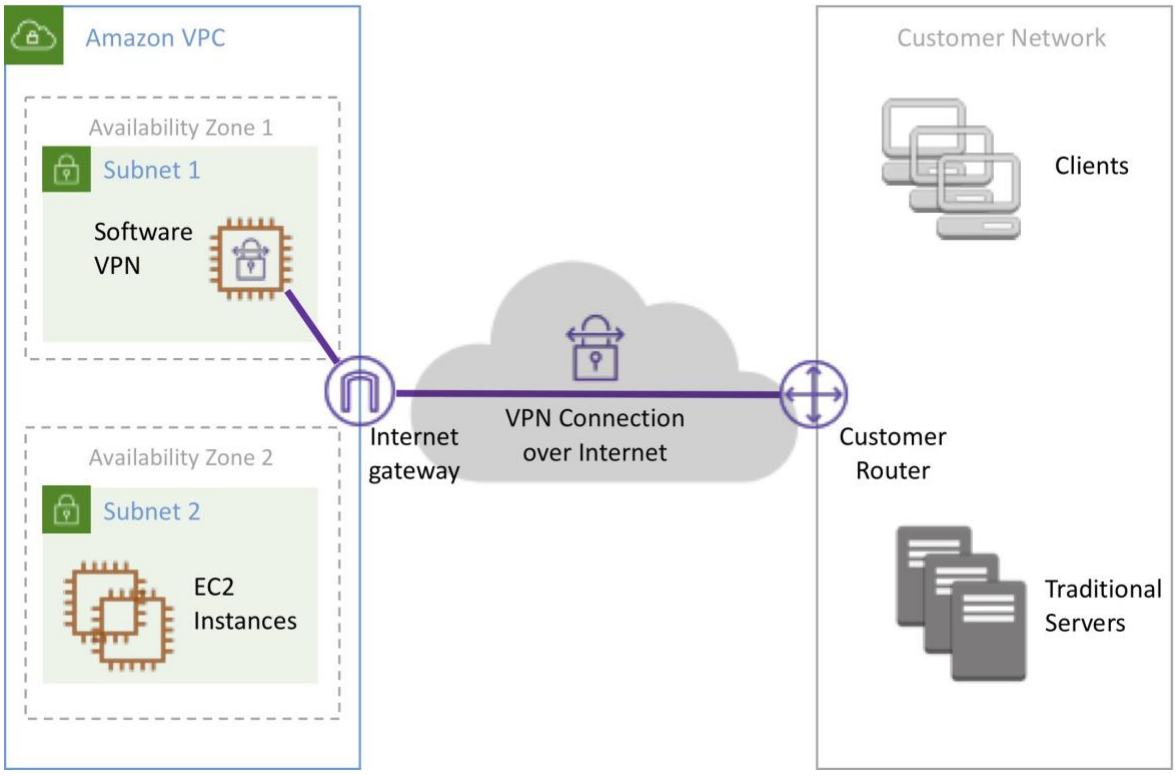
Software VPN

Amazon VPC offers you the flexibility to fully manage both sides of your Amazon VPC connectivity by creating a VPN connection between your remote network and a software VPN appliance running in your Amazon VPC network

This option is recommended if you must manage both ends of the VPN connection either for compliance purposes or for leveraging gateway devices that are not currently supported by Amazon VPC's VPN solution.

The diagram below depicts a Software VPN configuration:

Software VPN



Transit VPC

Building on the Software VPN design mentioned above, you can create a global transit network on AWS

A transit VPC is a common strategy for connecting multiple, geographically disperse VPCs and remote networks in order to create a global network transit center

A transit VPC simplifies network management and minimizes the number of connections required to connect multiple VPCs and remote networks

VPC Peering

What	AWS-provided network connectivity between two VPCs
When	Multiple VPCs need to communicate or access each other's resources
Pros	Uses AWS backbone without traversing the Internet
Cons	Transitive peering is not supported
How	VPC peering request made; accepter accepts request (either within or across accounts)

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

Limits are 50 VPC peers per VPC, up to 125 by request

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-"

AWS PrivateLink

What	AWS-provided network connectivity between VPCs and/or AWS services using interface endpoints
When	Keep Private Subnets truly private by using the AWS backbone to reach other AWS or Marketplace services rather than the public Internet
Pros	Redundant; uses the AWS backbone
Cons	
How	Create endpoint for required AWS or Marketplace service in all required subnets; access via the provided DNS hostname

VPC Endpoints

An Interface endpoint uses AWS PrivateLink and is an elastic network interface (ENI) with a private IP address that serves as an entry point for traffic destined to a supported service

Using PrivateLink you can connect your VPC to supported AWS services, services hosted by other AWS accounts (VPC endpoint services), and supported AWS Marketplace partner services

AWS PrivateLink access over Inter-Region VPC Peering:

- Applications in an AWS VPC can securely access AWS PrivateLink endpoints across AWS Regions using Inter-Region VPC Peering
- AWS PrivateLink allows you to privately access services hosted on AWS in a highly available and scalable manner, without using public IPs, and without requiring the traffic to traverse the Internet
- Customers can privately connect to a service even if the service endpoint resides in a different AWS Region
- Traffic using Inter-Region VPC Peering stays on the global AWS backbone and never traverses the public Internet.

A gateway endpoint is a gateway that is a target for a specified route in your route table, used for traffic destined to a supported AWS service

An interface VPC endpoint (interface endpoint) enables you to connect to services powered by AWS PrivateLink

The table below highlights some key information about both types of endpoint:

	Interface Endpoint	Gateway Endpoint
What	Elastic Network Interface with a Private IP	A gateway that is a target for a specific route
How	Uses DNS entries to redirect traffic	Uses prefix lists in the route table to redirect traffic
Which services	API Gateway, CloudFormation, CloudWatch etc.	Amazon S3, DynamoDB
Security	Security Groups	VPC Endpoint Policies

By default, IAM users do not have permission to work with endpoints

You can create an IAM user policy that grants users the permissions to create, modify, describe, and delete endpoints

Interface endpoints are available for:

- Amazon API Gateway
- Amazon CloudWatch Logs
- AWS CodeBuild
- Amazon EC2 API
- Elastic Load Balancing API
- AWS Key Management Service
- Amazon Kinesis Data Streams
- AWS Service Catalog
- Amazon SNS
- AWS Systems Manager
- Endpoint services hosted by other AWS accounts
- Supported AWS Marketplace partner services

Gateway endpoints are available for:

- DyanmoDB
- S3

VPC Flow Logs

Flow Logs capture information about the IP traffic going to and from network interfaces in a VPC

Flow log data is stored using Amazon CloudWatch 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

High Availability Approaches for Networking

By creating subnets in the available AZs, you create Multi-AZ presence for your VPC

Best practice is to create at least two VPN tunnels into your Virtual Private Gateway

Direct Connect is not HA by default, so you need to establish a secondary connection via another Direct Connect (ideally with another provider) or use a VPN

Route 53's health checks provide a basic level of redirecting DNS resolutions

Elastic IPs allow you flexibility to change out backing assets without impacting name resolution

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

Amazon CloudFront

General CloudFront Concepts

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

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

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

You can use a zone apex name on CloudFront

CloudFront supports wildcard CNAME

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

Regional Edge caches are used for custom origins, but not Amazon S3 origins

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

Origins

An origin is the origin of the files that the CDN will distribute

Origins can be either an 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 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 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

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

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:

- 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

For serving both the media player and media files you need two types of distributions:

- A web distribution for the media player
- An RTMP distribution for the media files

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

To delete a distribution it must first be disabled (can take up to 15 minutes)

Cache Behaviour

Allows you to configure a variety of CloudFront functionality for a given URL path pattern

For each cache behaviour 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 behaviour only allows a path pattern of /*

Additional cache behaviours need to be defined to change the path pattern following creation of the distribution

You can restrict access using the following methods:

- Restrict access to objects in CloudFront edge caches using signed cookies or signed URLs
- Restrict access to objects in your S3 bucket

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 behaviour

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

Domain Names

CloudFront typically creates a domain name such as a232323.cloudfront.net

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

Charges

There is an option for reserved capacity over 12 months or longer (starts at 10TB of data transfer in a single region)

You pay for:

- Data Transfer Out to Internet
- Data Transfer Out to Origin
- Number of HTTP/HTTPS Requests
- Invalidation Requests
- Dedicated IP Custom SSL
- Field level encryption requests

You do not pay for:

- Data transfer between AWS regions and CloudFront
- Regional edge cache
- AWS ACM SSL/TLS certificates
- Shared CloudFront certificates

Amazon Route 53

General Route 53 Concepts

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 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

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

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

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

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

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

Multi-value answer routing policy:

- Use for responding to DNS queries with up to eight healthy records selected at random

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
- Must total 100
- To stop sending traffic to a resource you can change the weight of the record to 0

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
- Route 53 Traffic Flow makes it easy for developers to create policies that route traffic based on the constraints they care most about, including latency, endpoint health, load, geo-proximity and geography
- You can use Amazon Route 53 Traffic Flow to assemble a wide range of routing scenarios, from adding a simple backup page in Amazon S3 for your website, to 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.

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)

You pay for queries

Latency-based routing queries are more expensive

Geo DNS and geo-proximity also have higher prices

Alias records are free of charge

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

Amazon API Gateway

General API Gateway Concepts

An Amazon API Gateway is a collection of resources and methods that are integrated with back-end HTTP endpoints, Lambda functions or other AWS services

API Gateway is a fully managed service that makes it easy for developers to publish, maintain, monitor, and secure APIs at any scale

API Gateway provides developers with a simple, flexible, fully managed, pay-as-you-go service that handles all aspects of creating and operating robust APIs for application back ends

API Gateway handles all of the tasks involved in accepting and processing up to hundreds of thousands of concurrent API calls

API calls include traffic management, authorisation and access control, monitoring, and API version management

Together with Lambda, API Gateway forms the app-facing part of the AWS serverless infrastructure

Back-end services include Amazon EC2, AWS Lambda or any web application (public or private endpoints)

Regionally based, private or edge optimized (deployed via CloudFront)

CloudFront is used as the public endpoint for API Gateway

Supports API keys and Usage Plans for user identification, throttling or quota management

Using CloudFront behind the scenes, custom domains, and SNI are supported

Can be published as products and monetized on AWS Marketplace

Collections can be deployed in stages

Permissions to invoke a method are granted using IAM roles and policies or API Gateway custom authorizers

An API can present a certificate to be authenticated by the back-end

All of the APIs created with Amazon API Gateway expose HTTPS endpoints only (does not support unencrypted endpoints)

By default API Gateway assigns an internal domain that automatically uses the API Gateway certificates

When configuring your APIs to run under a custom domain name you can provide your own certificate

APIs created with Amazon API Gateway expose HTTPS endpoints only

Supported data formats include JSON, XML, query string parameters, and request headers

Can enable Cross Origin Resource Sharing (CORS) for multiple domain use with Javascript/AJAX:

- Can be used to enable requests from domains other than the APIs domain
- Allows the sharing of resources between different domains
- The method (GET, PUT, POST etc) for which you will enable CORS must be available in the API Gateway API before you enable CORS
- If CORS is not enabled and an API resource received requests from another domain the request will be blocked
- Enable CORS on the APIs resources using the selected methods under the API Gateway

Data types used with API Gateway:

- Any payload sent over HTTP (always encrypted over HTTPS)
- Data formats include JSON, XML, query string parameters and request headers
- You can declare any content type for your APIs responses, and then use the transform templates to change the back-end response into your desired format

You can add caching to API calls by provisioning an Amazon API Gateway cache and specifying its size in gigabytes

Additional Features and Benefits

API Gateway provides several features that assist with creating and managing APIs:

- **Metering** - Define plans that meter and restrict third-party developer access to APIs
- **Security** - API Gateway provides multiple tools to authorize access to APIs and control service operation access
- **Resiliency** - Manage traffic with throttling so that backend operations can withstand traffic spikes
- **Operations Monitoring** - API Gateway provides a metrics dashboard to monitor calls to services
- **Lifecycle Management** - Operate multiple API versions and multiple stages for each version simultaneously so that existing applications can continue to call previous versions after new API versions are published

API Gateway provides robust, secure, and scalable access to backend APIs and hosts multiple versions and release stages for your APIs

You can create and distribute API Keys to developers

Option to use AWS Sig-v4 to authorize access to APIs

You can throttle and monitor requests to protect your backend

API Gateway allows you to maintain a cache to store API responses

SDK Generation for iOS, Android and JavaScript

Reduced latency and distributed denial of service protection through the use of CloudFront

Request/response data transformation and API mocking

Provides Swagger support

Resiliency through throttling rules based on the number of requests per second for each HTTP method (GET, PUT)

Throttling can be configured at multiple levels including Global and Service Call

A cache can be created and specified in gigabytes (not enabled by default)

Caches are provisioned for a specific stage of your APIs

Caching features include customizable keys and time-to-live (TTL) in seconds for your API data which enhances response times and reduces load on back-end services

API Gateway can scale to any level of traffic received by an API

Logging and Monitoring

The Amazon API Gateway logs (near real time) back-end performance metrics such as API calls, latency, and error rates to CloudWatch

You can monitor through the API Gateway dashboard (REST API) allowing you to visually monitor calls to the services

API Gateway also meters utilization by third-party developers and the data is available in the API Gateway console and through APIs

Amazon API Gateway is integrated with AWS CloudTrail to give a full auditable history of the changes to your REST APIs

All API calls made to the Amazon API Gateway APIS to create, modify, delete, or deploy REST APIs are logged to CloudTrail

Charges

With Amazon API Gateway, you only pay when your APIs are in use

There are no minimum fees or upfront commitments

You pay only for the API calls you receive and the amount of data transferred out

There are no data transfer out charges for Private APIs (however, AWS PrivateLink charges apply when using Private APIs in Amazon API Gateway)

Amazon API Gateway also provides optional data caching charged at an hourly rate that varies based on the cache size you select

The API Gateway free tier includes one million API calls per month for up to 12 months

AWS Direct Connect

AWS Direct Connect is a network service that provides an alternative to using the Internet to connect a customer's on-premise sites to AWS

Data is transmitted through a private network connection between AWS and a customer's datacenter or corporate network

Benefits:

- Reduce cost when using large volumes of traffic
- Increase reliability (predictable performance)
- Increase bandwidth (predictable bandwidth)
- Decrease latency

Each AWS Direct Connect connection can be configured with one or more virtual interfaces (VIFs)

Public VIFs allow access to public services such as S3, EC2, and DynamoDB

Private VIFs allow access to your VPC

Must use public IP addresses on public VIFs

Must use private IP addresses on private VIFs

Cannot do layer 2 over Direct Connect (L3 only)

From Direct Connect you can connect to all AZs **within the region**

You can establish IPSec connections over public VIFs to remote regions

Route propagation can be used to send customer side routes to the VPC

You can only have one 0.0.0.0/0 (all IP addresses) entry per route table

You can bind multiple ports for higher bandwidth

Virtual interfaces are configured to connect to either AWS public services (e.g. EC2/S3) or private services (e.g. VPC based resources)

Direct Connect is charged by port hours and data transfer

Available in 1Gbps and 10Gbps

Speeds of 50Mbps, 100Mbps, 200Mbps, 300Mbps, 400Mbps, and 500Mbps can be purchased through AWS Direct Connect Partners

Uses Ethernet trunking (802.1q)

Each connection consists of a single dedicated connection between ports on the customer router and an Amazon router

for HA you must have 2 DX connections - can be active/active or active/standby

Route tables need to be updated to point to a Direct Connect connection

VPN can be maintained as a backup with a higher BGP priority

Recommended to enable Bidirectional Forwarding Detection (BFD) for faster detection and failover

You cannot extend your on-premise VLANs into the AWS cloud using Direct Connect

Can aggregate up to 4 Direct Connect ports into a single connection using Link Aggregation Groups (LAG)

AWS Direct Connect supports both single (IPv4) and dual stack (IPv4/IPv6) configurations on public and private VIFs

Technical requirements for connecting virtual interfaces:

- A public or private ASN. If you are using a public ASN you must own it. If you are using a private ASN, it must be in the 64512 to 65535 range
- A new unused VLAN tag that you select
- **Private Connection (VPC)** - The VPC Virtual Private Gateway (VGW) ID
- **Public Connection** - Public IPs (/30) allocated by you for the BGP session

Direct Connect Gateway:

- Grouping of Virtual Private Gateways (VGWs) and Private Virtual Interfaces (VIFs) that belongs to the same AWS account
- Direct Connect Gateway enables you to interface with VPCs in any AWS Region (except AWS China Region)
- Can share private virtual interface to interface with more than one Virtual Private Clouds (VPCs) reducing the number of BGP sessions

Networking & Content Delivery Practice Questions

Answers and explanations are provided below after the last question in this section.

Question 1:

Your company shares some HR videos stored in an Amazon S3 bucket via CloudFront. You need to restrict access to the private content so users coming from specific IP addresses can access the videos and ensure direct access via the Amazon S3 bucket is not possible.

How can this be achieved?

A. Configure CloudFront to require users to access the files using a signed URL, create an origin access identity (OAI) and restrict access to the files in the Amazon S3 bucket to the OAI
B. Configure CloudFront to require users to access the files using signed cookies, create an origin access identity (OAI) and instruct users to login with the OAI
C. Configure CloudFront to require users to access the files using a signed URL, and configure the S3 bucket as a website endpoint

D. Configure CloudFront to require users to access the files using signed cookies, and move the files to an encrypted EBS volume

Question 2:

A customer is deploying services in a hybrid cloud model. The customer has mandated that data is transferred directly between cloud data centers, bypassing ISPs.

Which AWS service can be used to enable hybrid cloud connectivity?

A. IPSec VPN
B. Amazon Route 53
C. AWS Direct Connect
D. Amazon VPC

Question 3:

You have just created a new security group in your VPC. You have not yet created any rules. Which of the statements below are correct regarding the default state of the security group? (choose 2)

A. There is an outbound rule that allows all traffic to all IP addresses
B. There are no inbound rules and traffic will be implicitly denied
C. There are is an inbound rule that allows traffic from the Internet Gateway
D. There is an inbound rule allowing traffic from the Internet to port 22 for management
E. There is an outbound rule allowing traffic to the Internet Gateway

Question 4:

You have just created a new Network ACL in your VPC. You have not yet created any rules. Which of the statements below are correct regarding the default state of the Network ACL? (choose 2)

A. There is a default inbound rule denying all traffic
B. There is a default outbound rule allowing all traffic
C. There is a default inbound rule allowing traffic from the VPC CIDR block
D. There is a default outbound rule allowing traffic to the Internet Gateway
E. There is a default outbound rule denying all traffic

Question 5:

One of your clients is transitioning their web presence into the AWS cloud. As part of the migration the client will be running a web application both on-premises and in AWS for a period of time. During the period of co-existence the client would like 80% of the traffic to hit the AWS-based web servers and 20% to be directed to the on-premises web servers.

What method can you use to distribute traffic as requested?

A. Use a Network Load Balancer to distribute traffic based on Instance ID
B. Use an Application Load Balancer to distribute traffic based on IP address
C. Use Route 53 with a weighted routing policy and configure the respective weights

D. Use Route 53 with a simple routing policy

Question 6:

You need to setup a distribution method for some static files. The requests will be mainly GET requests and you are expecting a high volume of GETs often exceeding 2000 per second. The files are currently stored in an S3 bucket. According to AWS best practices, what can you do to optimize performance?

A. Integrate CloudFront with S3 to cache the content
B. Use cross-region replication to spread the load across regions
C. Use ElastiCache to cache the content
D. Use S3 Transfer Acceleration

Question 7:

A Solutions Architect has created a VPC and is in the process of formulating the subnet design. The VPC will be used to host a two-tier application that will include Internet facing web servers, and internal-only DB servers. Zonal redundancy is required.

How many subnets are required to support this requirement?

A. 1 subnet
B. 2 subnets
C. 4 subnets
D. 6 subnets

Question 8:

You have launched an EC2 instance into a VPC. You need to ensure that instances have both a private and public DNS hostname. Assuming you did not change any settings during creation of the VPC, how will DNS hostnames be assigned by default? (choose 2)

A. In a default VPC instances will be assigned a public and private DNS hostname
B. In a non-default VPC instances will be assigned a public and private DNS hostname
C. In a default VPC instances will be assigned a private but not a public DNS hostname
D. In all VPCs instances no DNS hostnames will be assigned
E. In a non-default VPC instances will be assigned a private but not a public DNS hostname

Question 9:

You are putting together an architecture for a new VPC on AWS. Your on-premise data center will be connected to the VPC by a hardware VPN and has public and VPN-only subnets. The security team has requested that traffic hitting public subnets on AWS that's destined to on-premise applications must be directed over the VPN to the corporate firewall.

How can this be achieved?

A. In the public subnet route table, add a route for your remote network and specify the customer gateway as the target
B. Configure a NAT Gateway and configure all traffic to be directed via the virtual private gateway
C. In the public subnet route table, add a route for your remote network and specify the virtual private gateway as the target
D. In the VPN-only subnet route table, add a route that directs all Internet traffic to the virtual private gateway

Question 1 answer: *A*

Explanation:

A signed URL includes additional information, for example, an expiration date and time, that gives you more control over access to your content. You can also specify the IP address or range of IP addresses of the users who can access your content.

If you use CloudFront signed URLs (or signed cookies) to limit access to files in your Amazon S3 bucket, you may also want to prevent users from directly accessing your S3 files by using Amazon S3 URLs. To achieve this you can create an origin access identity (OAI), which is a special CloudFront user, and associate the OAI with your distribution. You can then change the permissions either on your Amazon S3 bucket or on the files in your bucket so that only the origin access identity has read permission (or read and download permission).

Users cannot login with an OAI.

You cannot use CloudFront and an OAI when you're S3 bucket is configured as a website endpoint.

You cannot use CloudFront to pull data directly from an EBS volume.

Question 2 answer: C

Explanation:

With AWS Direct Connect, you can connect to all your AWS resources in an AWS Region, transfer your business-critical data directly from your datacenter, office, or colocation environment into and from AWS, bypassing your Internet service provider and removing network congestion.

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

An IPSec VPN can be used to connect to AWS however it does not bypass the ISPs or Internet.

Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources into a virtual network that you've defined.

Question 3 answer: *A,B*

Explanation:

Custom security groups do not have inbound allow rules (all inbound traffic is denied by default).

Default security groups do have inbound allow rules (allowing traffic from within the group). All outbound traffic is allowed by default in both custom and default security groups.

Security groups act like a stateful firewall at the instance level. Specifically, security groups operate at the network interface level of an EC2 instance. You can only assign permit rules in a security group; you cannot assign deny rules and 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. You can create ingress and egress rules.

Question 4 answer: *A,E*

Explanation:

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.

Network ACL's function at the subnet level and you can have permit and deny rules. 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.

Question 5 answer: *C*

Explanation:

Route 53 weighted routing policy is 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 which is a numerical value that favours one IP over another (values must total 100). To stop sending traffic to a resource you can change the weight of the record to 0.

Network Load Balancer can distribute traffic to AWS and on-premise resources using IP addresses (not Instance IDs).

Application Load Balancer can distribute traffic to AWS and on-premise resources using IP addresses but cannot be used to distribute traffic in a weighted manner.

Question 6 answer: ***A***

Explanation:

Amazon S3 automatically scales to high request rates. For example, your application can achieve at least 3,500 PUT/POST/DELETE and 5,500 GET requests per second per prefix in a bucket. There are no limits to the number of prefixes in a bucket.

If your workload is mainly sending GET requests, in addition to the preceding guidelines, you should consider using Amazon CloudFront for performance optimization. By integrating CloudFront with Amazon S3, you can distribute content to your users with low latency and a high data transfer rate.

Transfer Acceleration is used to accelerate object **uploads** to S3 over long distances (latency). Cross-region replication creates a replica copy in another region but should not be used for spreading read requests across regions.

There will be 2 S3 endpoints and CRR is not designed for 2-way sync so this would not work well. ElastiCache is used for caching database content not S3 content.

Question 7 answer: *C*

Explanation:

Zonal redundancy indicates that the architecture should be split across multiple Availability Zones. Subnets are mapped 1:1 to AZs.

A public subnet should be used for the Internet-facing web servers and a separate private subnet should be used for the internal-only DB servers. Therefore, you need 4 subnets - 2 (for redundancy) per public/private subnet.

Question 8 answer: *A,E*

Explanation:

When you launch an instance into a default VPC, we provide the instance with public and private DNS hostnames that correspond to the public IPv4 and private IPv4 addresses for the instance.

When you launch an instance into a nondefault VPC, we provide the instance with a private DNS hostname and we might provide a public DNS hostname, depending on the DNS attributes you specify for the VPC and if your instance has a public IPv4 address.

Question 9 answer: *C*

Explanation:

Route tables determine where network traffic is directed. In your route table, you must add a route for your remote network and specify the virtual private gateway as the target. This enables traffic from your VPC that's destined for your remote network to route via the virtual private gateway and over one of the VPN tunnels. You can enable route propagation for your route table to automatically propagate your network routes to the table for you.

You must select the virtual private gateway (AWS side of the VPN) not the customer gateway (customer side of the VPN) in the target in the route table. NAT Gateways are used to enable Internet access for EC2 instances in private subnets, they cannot be used to direct traffic to VPG.

You must create the route table rule in the route table attached to the public subnet, not the VPN-only subnet.

MANAGEMENT TOOLS

Amazon CloudWatch

Amazon CloudWatch is a monitoring service for AWS cloud resources and the applications you run on AWS

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

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

Amazon CloudWatch Logs lets you monitor and troubleshoot your systems and applications using your existing system, application and custom log files

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 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)

Dashboards allow you to create, customize, interact with, and save graphs of AWS resources and custom metrics

Alarms can be used to monitor any Amazon CloudWatch metric in your account

Events are a stream of system events describing changes in your AWS resources

Logs help you to aggregate, monitor and store logs

Basic monitoring = 5 mins (free for EC2 Instances, EBS volumes, ELBs and RDS DBs)

Detailed monitoring = 1 min (chargeable)

Metrics are provided automatically for a number of AWS products and services

There is no standard metric for memory usage on EC2 instances

A custom metric is any metric you provide to Amazon CloudWatch (e.g. time to load a web page or application performance)

Options for storing logs:

- CloudWatch Logs
- Centralized logging system (e.g. Splunk)
- Custom script and store on S3

Do not store logs on non-persistent disks:

Best practice is to store logs in CloudWatch Logs or S3

CloudWatch Logs subscription can be used across multiple AWS accounts (using cross account access)

Amazon CloudWatch uses Amazon SNS to send email

AWS CloudTrail

AWS CloudTrail is a web service that records activity made on your account and delivers log files to an 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

CloudTrail is about logging and saves a history of API calls for your AWS account

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

Not enabled by default

CloudTrail is per AWS account

Trails can be enabled per region or a trail can be applied to all regions

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

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

AWS OpsWorks

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

Automates how applications are configured, deployed and managed

Provide 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 CloudFormation

AWS CloudFormation is a service that gives developers and businesses an easy way to create a collection of related AWS resources and provision them in an orderly and predictable fashion

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

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

Think of CloudFormation as deploying infrastructure as code

Elastic Beanstalk is more focussed on deploying applications on EC2 (PaaS)

CloudFormation can deploy Elastic Beanstalk-hosted applications however the reverse is not possible

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

Concept of templates, stacks and change sets:

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
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:

- Architectural designs
- Create, update and delete templates
- Written in JSON or YAML
- CloudFormation determines the order of provisioning
- Don't need to worry about dependencies
- Modifies and updates templates in a controlled way (version control)
- Designer allows you to visualize using a drag and drop interface

Stacks:

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

Template elements:

- Mandatory:
  - File format and version
  - List of resources and associated configuration values

- Not mandatory:
  - Template parameters (limited to 60)
  - Output values (limited to 60)
  - List of data tables

Puppet and Chef integration is supported

Can use bootstrap scripts

Can define deletion policies

Provides WaitCondition function

Can create roles in IAM

VPCs can be created and customized

VPC peering in the same AWS account can be performed

Route 53 is supported

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

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
- 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

Best Practices

- AWS provides Python “helper scripts” which can help you install software and start services on your EC2 instances
- Use CloudFormation to make changes to your landscape rather than going directly into the resources
- Make use of Change Sets to identify potential trouble spots in your updates
- Use Stack Policies to explicitly protect sensitive portions of your stack
- Use a version control system such as CodeCommit or GitHub to track changes to templates

Charges:

- There is no additional charge for AWS CloudFormation
- You pay for AWS resources (such as Amazon EC2 instances, Elastic Load Balancing load balancers, etc.) created using AWS CloudFormation in the same manner as if you created them manually
- You only pay for what you use, as you use it; there are no minimum fees and no required upfront commitments

AWS Config

General

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 RDS?
- Is CloudTrail enabled on the AWS account?
- Are 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., 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 Systems Manager

AWS Systems Manager allows you to centralize operational data from multiple AWS services and automate tasks across your AWS 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 provides a central place to view and manage your AWS resources, so you can have complete visibility and control over your operations

Centralized console and toolset for a wide variety of system management tasks

Designed for managing a large fleet of systems – tens or hundreds

SSM Agent enables System Manager features and supports all OSs supported by OS as well as back to Windows Server 2003 and Raspbian

SSM Agent installed by default on recent AWS-provided base AMIs for Linux and Windows

Manages AWS-based and on-premises based systems via the agent

The AWS Systems Manager console integrates with AWS Resource Groups, and it offers grouping capabilities in addition to other native integrations

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 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:

- 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
- 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

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
- 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

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

Management Tools Practice Questions

Answers and explanations are provided below after the last question in this section.

Question 1:

You are a Solutions Architect at Digital Cloud Training. A client from a large multinational corporation is working on a deployment of a significant amount of resources into AWS. The client would like to be able to deploy resources across multiple AWS accounts and regions using a single toolset and template. You have been asked to suggest a toolset that can provide this functionality?

A. Use a CloudFormation template that creates a stack and specify the logical IDs of each account and region
B. Use a CloudFormation StackSet and specify the target accounts and regions in which the stacks will be created
C. Use a third-party product such as Terraform that has support for multiple AWS accounts and regions

D. This cannot be done, use separate CloudFormation templates per AWS account and region

Question 2:

A Solutions Architect needs to monitor application logs and receive a notification whenever a specific number of occurrences of certain HTTP status code errors occur. Which tool should the Architect use?

A. CloudWatch Events
B. CloudWatch Logs
C. CloudTrail Trails
D. CloudWatch Metrics

Question 3:

A Solutions Architect is designing the system monitoring and deployment layers of a serverless application. The system monitoring layer will manage system visibility through recording logs and metrics and the deployment layer will deploy the application stack and manage workload changes through a release management process.

The Architect needs to select the most appropriate AWS services for these functions. Which services and frameworks should be used for the system monitoring and deployment layers? (choose 2)

A. Use AWS X-Ray to package, test, and deploy the serverless application stack
B. Use Amazon CloudTrail for consolidating system and application logs and monitoring custom metrics
C. Use AWS Lambda to package, test, and deploy the serverless application stack
D. Use AWS SAM to package, test, and deploy the serverless application stack
E. Use Amazon CloudWatch for consolidating system and application logs and monitoring custom metrics

Question 4:

A systems integration consultancy regularly deploys and manages multi-tiered web services for customers on AWS. The SysOps team are facing challenges in tracking changes that are made to the web services and rolling back when problems occur.

Which of the approaches below would BEST assist the SysOps team?

A. Use AWS Systems Manager to manage all updates to the web services
B. Use CodeDeploy to manage version control for the web services
C. Use Trusted Advisor to record updates made to the web services
D. Use CloudFormation templates to deploy and manage the web services

Question 5:

An event in CloudTrail is the record of an activity in an AWS account. What are the two types of events that can be logged in CloudTrail? (choose 2)

A. System Events which are also known as instance level operations
B. Management Events which are also known as control plane operations
C. Platform Events which are also known as hardware level operations
D. Data Events which are also known as data plane operations
E. API events which are also known as CloudWatch events

Question 6:

Your company currently uses Puppet Enterprise for infrastructure and application management. You are looking to move some of your infrastructure onto AWS and would like to continue to use the same tools in the cloud. What AWS service provides a fully managed configuration management service that is compatible with Puppet Enterprise?

A. Elastic Beanstalk
B. CloudFormation
C. OpsWorks
D. CloudTrail

Question 7:

The operations team in your company are looking for a method to automatically respond to failed system status check alarms that are being received from an EC2 instance. The system in question is experiencing intermittent problems with its operating system software.

Which two steps will help you to automate the resolution of the operating system software issues? (choose 2)

A. Create a CloudWatch alarm that monitors the “StatusCheckFailed_System” metric
B. Create a CloudWatch alarm that monitors the “StatusCheckFailed_Instance” metric
C. Configure an EC2 action that recovers the instance
D. Configure an EC2 action that terminates the instance
E. Configure an EC2 action that reboots the instance

Question 1 answer: B

Explanation:

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.

A regular CloudFormation template cannot be used across regions and accounts. You would need to create copies of the template and then manage updates.

You do not need to use a third-party product such as Terraform as this functionality can be delivered through native AWS technology.

Question 2 answer: *B*

Explanation:

You can use **CloudWatch Logs** to monitor applications and systems using log data. For example, CloudWatch Logs can track the number of errors that occur in your application logs and send you a notification whenever the rate of errors exceeds a threshold you specify. This is the best tool for this requirement.

Amazon CloudWatch Events delivers a near real-time stream of system events that describe changes in Amazon Web Services (AWS) resources. Though you can generate custom application-level events and publish them to CloudWatch Events this is not the best tool for monitoring application logs.

CloudTrail is used for monitoring API activity on your account, not for monitoring application logs.

CloudWatch Metrics are the fundamental concept in CloudWatch. A metric represents a time-ordered set of data points that are published to CloudWatch. You cannot use a metric alone; it is used when setting up monitoring for any service in CloudWatch.

Question 3 answer: *D,E*

Explanation:

AWS Serverless Application Model (AWS SAM) is an extension of AWS CloudFormation that is used to package, test, and deploy serverless applications.

With Amazon CloudWatch, you can access system metrics on all the AWS services you use, consolidate system and application level logs, and create business key performance indicators (KPIs) as custom metrics for your specific needs.

AWS Lambda is used for executing your code as functions, it is not used for packaging, testing and deployment. AWS Lambda is used with AWS SAM.

AWS X-Ray lets you analyze and debug serverless applications by providing distributed tracing and service maps to easily identify performance bottlenecks by visualizing a request end-to-end.

Question 4 answer: *D*

Explanation:

When you provision your infrastructure with AWS CloudFormation, the AWS CloudFormation template describes exactly what resources are provisioned and their settings. Because these templates are text files, you simply track differences in your templates to track changes to your infrastructure, similar to the way developers control revisions to source code. For example, you can use a version control system with your templates so that you know exactly what changes

were made, who made them, and when. If at any point you need to reverse changes to your infrastructure, you can use a previous version of your template.

AWS Systems Manager gives you visibility and control of your infrastructure on AWS. Systems Manager provides a unified user interface so you can view operational data from multiple AWS services and allows you to automate operational tasks across your AWS resources. However, CloudFormation would be the preferred method of maintaining the state of the overall architecture.

AWS CodeDeploy is a deployment service that automates application deployments to Amazon EC2 instances, on-premises instances, or serverless Lambda function.

AWS Trusted Advisor is an online resource to help you reduce cost, increase performance, and improve security by optimizing your AWS environment, Trusted Advisor provides real time guidance to help you provision your resources following AWS best practices.

Question 5 answer: *B,D*

Explanation:

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

Question 6 answer: *C*

Explanation:

The only service that would allow you to continue to use the same tools is OpsWorks. AWS OpsWorks is a configuration management service that provides managed instances of Chef and Puppet. 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.

Question 7 answer: *B,E*

Explanation:

EC2 status checks are performed every minute and each returns a pass or a fail status. If all checks pass, the overall status of the instance is OK. If one or more checks fail, the overall status is impaired.

System status checks detect (StatusCheckFailed_System) problems with your instance that require AWS involvement to repair whereas **Instance status checks** (StatusCheckFailed_Instance) detect problems that require your involvement to repair.

The action to *recover* the instance is only supported on specific instance types and can be used only with StatusCheckFailed_System.

Configuring an action to terminate the instance would not help resolve system software issues as the instance would be terminated.

MEDIA SERVICES

Amazon Elastic Transcoder

Amazon Elastic Transcoder is a highly scalable, easy to use and cost-effective way for developers and businesses to convert (or "transcode") video and audio files from their source format into versions that will playback on devices like smartphones, tablets and PCs

Supports a wide range of input and output formats, resolutions, bitrates, and frame rates

Also offers features for automatic video bit rate optimization, generation of thumbnails, overlay of visual watermarks, caption support, DRM packaging, progressive downloads, encryption and more

Picks up files from an input S3 bucket and saves the output to an output S3 bucket

Uses a JSON API, and SDKs are provided for Python, Node.js, Java, .NET, PHP, and Ruby

Provides transcoding presets for popular formats

You are charged based on the duration of the content and the resolution or format of the media

Media Services Practice Questions

Answers and explanations are provided below after the last question in this section.

Question 1:

You are undertaking a project to make some audio and video files that your company uses for onboarding new staff members available via a mobile application. You are looking for a cost-effective way to convert the files from their current formats into formats that are compatible with smartphones and tablets. The files are currently stored in an S3 bucket.

What AWS service can help with converting the files?

A. MediaConvert
B. Data Pipeline
C. Elastic Transcoder
D. Rekognition

Question 2:

Which service provides a way to convert video and audio files from their source format into versions that will playback on devices like smartphones, tablets and PCs?

A. Amazon Elastic Transcoder
B. AWS Glue
C. Amazon Rekognition
D. Amazon Comprehend

Question 1 answer: C

Explanation:

Amazon Elastic Transcoder is a highly scalable, easy to use and cost-effective way for developers and businesses to convert (or "transcode") video and audio files from their source format into versions that will playback on devices like smartphones, tablets and PCs.

MediaConvert converts file-based content for broadcast and multi-screen delivery.

Data Pipeline helps you move, integrate, and process data across AWS compute and storage resources, as well as your on-premises resources.

Rekognition is a deep learning-based visual analysis service.

Question 2 answer: A

Explanation:

Amazon Elastic Transcoder is a highly scalable, easy to use and cost-effective way for developers and businesses to convert (or "transcode") video and audio files from their source format into versions that will playback on devices like smartphones, tablets and PCs.

AWS Glue is a fully managed extract, transform, and load (ETL) service that makes it easy for customers to prepare and load their data for analytics.

Amazon Rekognition makes it easy to add image and video analysis to your applications.

Amazon Comprehend is a natural language processing (NLP) service that uses machine learning to find insights and relationships in text.

ANALYTICS

Amazon EMR

Amazon EMR is a web service that enables businesses, researchers, data analysts, and developers to easily and cost-effectively process vast amounts of data

EMR utilizes a hosted Hadoop framework running on Amazon EC2 and Amazon S3

Managed Hadoop framework for processing huge amounts of data

Also support Apache Spark, HBase, Presto and Flink

Most commonly used for log analysis, financial analysis, or extract, translate and loading (ETL) activities

A Step is a programmatic task for performing some process on the data (e.g. count words)

A cluster is a collection of EC2 instances provisioned by EMR to run your Steps

EMR uses Apache Hadoop as its distributed data processing engine which is an open source, Java software framework that supports data-intensive distributed applications running on large clusters of commodity hardware

EMR is a good place to deploy Apache Spark, an open-source distributed processing used for big data workloads which utilizes in-memory caching and optimized query execution

You can also launch Presto clusters. Presto is an open-source distributed SQL query engine designed for fast analytic queries against large datasets

EMR launches all nodes for a given cluster in the same Amazon EC2 Availability Zone

You can access Amazon EMR by using the AWS Management Console, Command Line Tools, SDKS, or the EMR API

With EMR you have access to the underlying operating system (you can SSH in)

Amazon Kinesis

General

Amazon Kinesis makes it easy to collect, process, and analyze real-time, streaming data so you can get timely insights and react quickly to new information

Collection of services for processing streams of various data

Data is processed in “shards” – with each shard able to ingest 1000 records per second

There is a default limit of 500 shards, but you can request an increase to unlimited shards

A record consists of a partition key, sequence number, and data blob (up to 1 MB)

Transient data store – default retention of 24 hours, but can be configured for up to 7 days

There are four types of Kinesis service and these are detailed below

Kinesis Video Streams

Kinesis Video Streams makes it easy to securely stream video from connected devices to AWS for analytics, machine learning (ML), and other processing

Durably stores, encrypts, and indexes video data streams, and allows access to data through easy-to-use APIs

Producers provide data streams

Stores data for 24 hours by default, up to 7 days

Stores data in shards - 5 transaction per second for reads, up to a max read rate of 2MB per second and 1000 records per second for writes up to a max of 1MB per second

Consumers receive and process data

Can have multiple shards in a stream

Supports encryption at rest with server-side encryption (KMS) with a customer master key

Kinesis Data Streams

Kinesis Data Streams enables you to build custom applications that process or analyze streaming data for specialized needs

Kinesis Data Streams enables real-time processing of streaming big data

Kinesis Data Streams is useful for rapidly moving data off data producers and then continuously processing the data

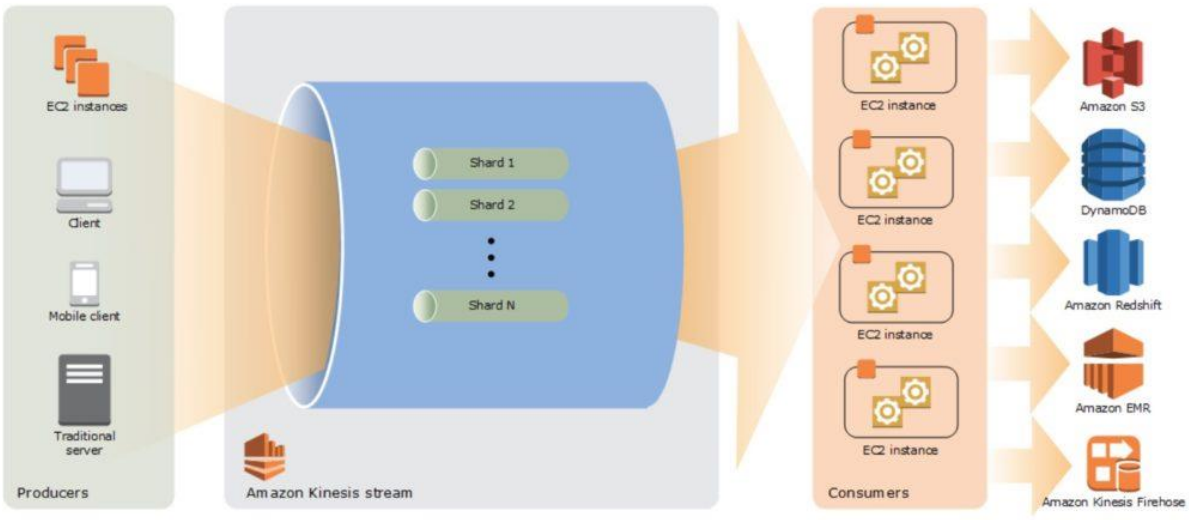
Kinesis Data Streams **stores data** for later processing by applications (key difference with Firehose which delivers data directly to AWS services)

Common use cases include:

- Accelerated log and data feed intake
- Real-time metrics and reporting
- Real-time data analytics
- Complex stream processing

The following diagram illustrates the high-level architecture of Kinesis Data Streams.

- Producers continually push data to Kinesis Data Streams
- Consumers process the data in real time
- Consumers can store their results using an AWS service such as Amazon DynamoDB, Amazon Redshift, or Amazon S3
- Kinesis Streams applications are consumers that run on EC2 instances
- Shards are uniquely identified groups or data records in a stream
- Records are the data units stored in a Kinesis Stream



A producer creates the data that makes up the stream

Producers can be used through the following:

- Kinesis Streams API
- Kinesis Producer Library (KPL)
- Kinesis Agent

A record is the unit of data stored in an Amazon Kinesis data stream

A record is composed of a sequence number, partition key, and data blob

By default, records of a stream are accessible for up to 24 hours from the time they are added to the stream (can be raised to 7 days by enabling extended data retention)

A data blob is the data of interest your data producer adds to a data stream

The maximum size of a data blob (the data payload before Base64-encoding) within one record is 1 megabyte (MB)

A shard is the base throughput unit of an Amazon Kinesis data stream

One shard provides a capacity of 1MB/sec data input and 2MB/sec data output

Each shard can support up to 1000 PUT records per second

A stream is composed of one or more shards

Consumers are the EC2 instances that analyze the data received from a stream

Consumers are known as Amazon Kinesis Streams Applications

When the data rate increases, add more shards to increase the size of the stream

Remove shards when the data rate decreases

Partition keys are used to group data by shard within a stream

Kinesis Streams uses KMS master keys for encryption

To read from or write to an encrypted stream the producer and consumer applications must have permission to access the master key

Kinesis Data Streams replicates synchronously across three AZs

Kinesis Data Firehose

Kinesis Data Firehose is the easiest way to load streaming data into data stores and analytics tools

Captures, transforms, and loads streaming data

Enables near real-time analytics with existing business intelligence tools and dashboards

Kinesis Data Streams can be used as the source(s) to Kinesis Data Firehose

You can configure Kinesis Data Firehose to transform your data before delivering it

With Kinesis Data Firehose you don't need to write an application or manage resources

Firehose can batch, compress, and encrypt data before loading it

Firehose synchronously replicates data across three AZs as it is transported to destinations

Each delivery stream stores data records for up to 24 hours

Can invoke a Lambda function to transform data before delivering it to destinations

A source is where your streaming data is continuously generated and captured

A delivery stream is the underlying entity of Amazon Kinesis Data Firehose

A record is the data of interest your data producer sends to a delivery stream

The maximum size of a record (before Base64-encoding) is 1000 KB

A destination is the data store where your data will be delivered

Firehose Destinations include:

- Amazon S3
- Amazon Redshift
- Amazon Elasticsearch Service
- Splunk

Producers provide data streams

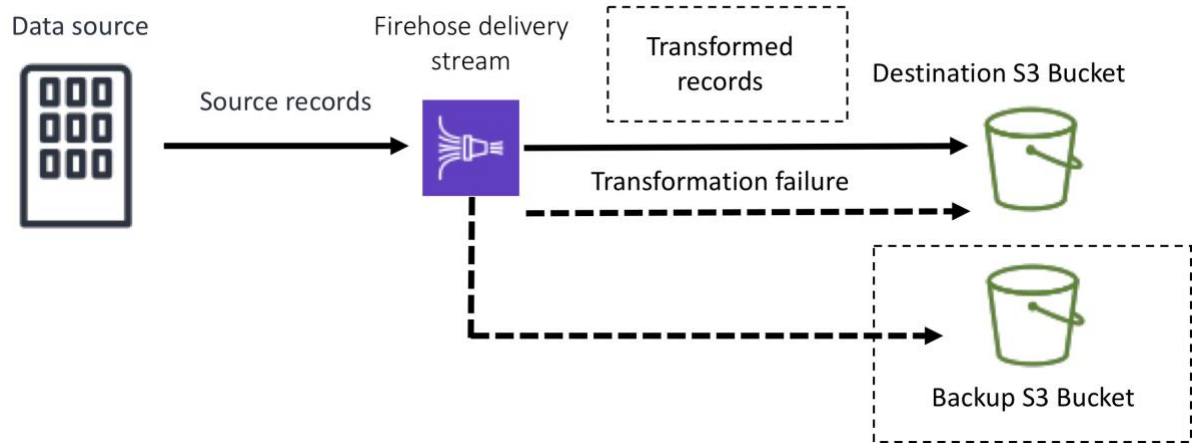
No shards, totally automated

Can encrypt data with an existing AWS Key Management Service (KMS) key

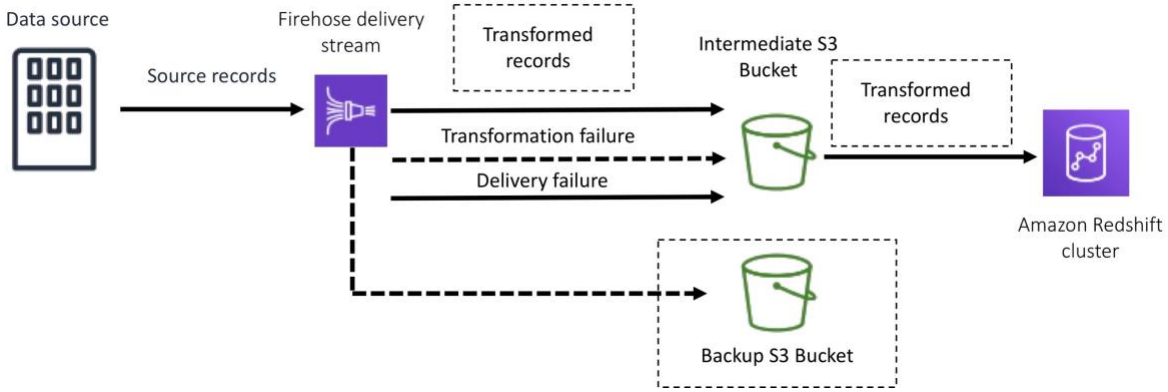
Server-side-encryption can be used if Kinesis Streams is used as the data source

Firehose can invoke an AWS Lambda function to transform incoming data before delivering it to a destination

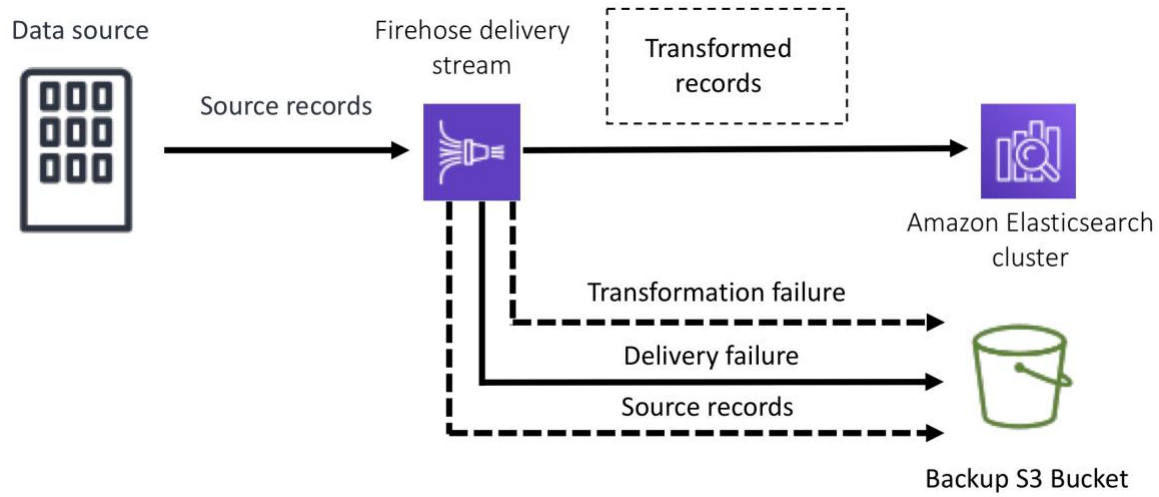
For Amazon S3 destinations, streaming data is delivered to your S3 bucket. If data transformation is enabled, you can optionally back up source data to another Amazon S3 bucket:



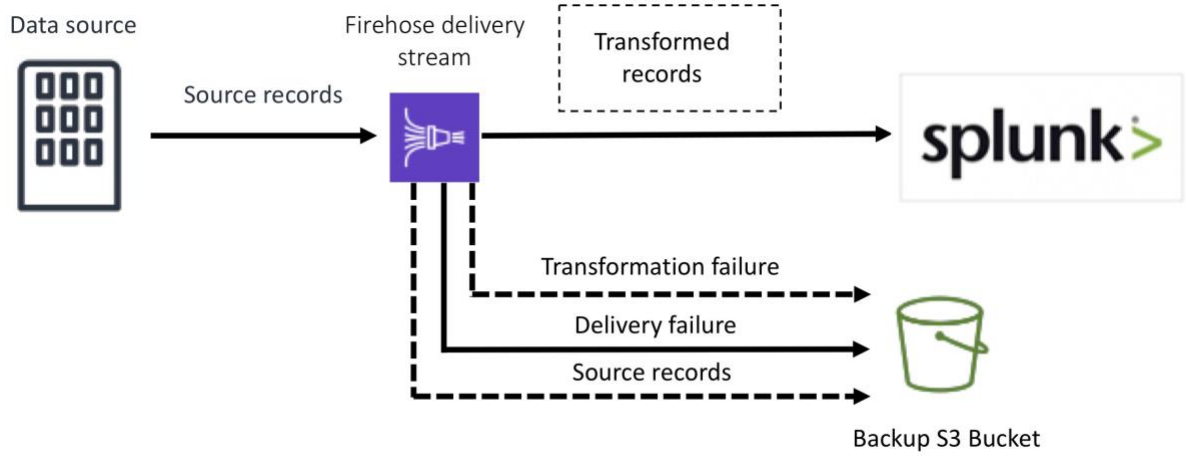
For Amazon Redshift destinations, streaming data is delivered to your S3 bucket first. Kinesis Data Firehose then issues an Amazon Redshift **COPY** command to load data from your S3 bucket to your Amazon Redshift cluster. If data transformation is enabled, you can optionally back up source data to another Amazon S3 bucket:



For Amazon Elaticsearch destinations, streaming data is delivered to your Amazon ES cluster, and it can optionally be backed up to your S3 bucket concurrently:



For Splunk destinations, streaming data is delivered to Splunk, and it can optionally be backed up to your S3 bucket concurrently:



Kinesis Data Analytics

Amazon Kinesis Data Analytics is the easiest way to process and analyze real-time, streaming data

Can use standard SQL queries to process Kinesis data streams

Provides real-time analysis

Use cases:

- Generate time-series analytics
- Feed real-time dashboards
- Create real-time alerts and notifications

Quickly author and run powerful SQL code against streaming sources

Can ingest data from Kinesis Streams and Kinesis Firehose

Output to S3, RedShift, Elasticsearch and Kinesis Data Streams

Sits over Kinesis Data Streams and Kinesis Data Firehose

A Kinesis Data Analytics application consists of three components:

- Input - the streaming source for your application
- Application code - a series of SQL statements that process input and produce output
- Output - one or more in-application streams to hold intermediate results

Kinesis Data Analytics supports two types of inputs: streaming data sources and reference data sources:

- A streaming data source is continuously generated data that is read into your application for processing
- A reference data source is static data that your application uses to enrich data coming in from streaming sources

Can configure destinations to persist the results

Supports Kinesis Streams and Kinesis Firehose (S3, RedShift, ElasticSearch) as destinations

IAM can be used to provide Kinesis Analytics with permissions to read records from sources and write to destinations

Analytics Practice Questions

Answers and explanations are provided below after the last question in this section.

Question 1:

A user is testing a new service that receives location updates from 5,000 rental cars every hour. Which service will collect data and automatically scale to accommodate production workload?

A. Amazon EC2
B. Amazon Kinesis Firehose
C. Amazon EBS
D. Amazon API Gateway

Question 2:

Which AWS service can be used to prepare and load data for analytics using an extract, transform and load (ETL) process?

A. AWS Lambda
B. Amazon Athena
C. AWS Glue
D. Amazon EMR

Question 3:

A Solutions Architect is designing a solution for a financial application that will receive trading data in large volumes. What is the best solution for ingesting and processing a very large number of data streams in near real time?

A. Amazon EMR

B. Amazon Kinesis Firehose
C. Amazon Redshift
D. Amazon Kinesis Data Streams

Question 4:

You have recently enabled Access Logs on your Application Load Balancer (ALB). One of your colleagues would like to process the log files using a hosted Hadoop service. What configuration changes and services can be leveraged to deliver this requirement?

A. Configure Access Logs to be delivered to DynamoDB and use EMR for processing the log files
B. Configure Access Logs to be delivered to S3 and use Kinesis for processing the log files
C. Configure Access Logs to be delivered to S3 and use EMR for processing the log files
D. Configure Access Logs to be delivered to EC2 and install Hadoop for processing the log files

Question 5:

A Solutions Architect is designing the messaging and streaming layers of a serverless application. The messaging layer will manage communications between components and the streaming layer will manage real-time analysis and processing of streaming data.

The Architect needs to select the most appropriate AWS services for these functions. Which services should be used for the messaging and streaming layers? (choose 2)

A. Use Amazon Kinesis for collecting, processing and analyzing real-time streaming data
B. Use Amazon EMR for collecting, processing and analyzing real-time streaming data
C. Use Amazon SNS for providing a fully managed messaging service
D. Use Amazon SWF for providing a fully managed messaging service
E. Use Amazon CloudTrail for collecting, processing and analyzing real-time streaming data

Question 1 answer: B

Explanation:

What we need here is a service that can streaming collect streaming data. The only option available is Kinesis Firehose which captures, transforms, and loads streaming data into "destinations" such as S3, RedShift, Elasticsearch and Splunk.

Amazon EC2 is not suitable for collecting streaming data.

EBS is a block-storage service in which you attach volumes to EC2 instances, this does not assist with collecting streaming data (see previous point).

Amazon API Gateway is used for hosting and managing APIs not for receiving streaming data.

Question 2 answer: C

Explanation:

AWS Glue is a fully managed extract, transform, and load (ETL) service that makes it easy for customers to prepare and load their data for analytics.

Amazon Elastic Map Reduce (EMR) provides a managed Hadoop framework that makes it easy, fast, and cost-effective to process vast amounts of data across dynamically scalable Amazon EC2 instances.

Amazon Athena is an interactive query service that makes it easy to analyze data in Amazon S3 using standard SQL.

AWS Lambda is a serverless application that runs code as functions in response to events.

Question 3 answer: *D*

Explanation:

Kinesis Data Streams enables you to build custom applications that process or analyze streaming data for specialized needs. It enables real-time processing of streaming big data and can be used for rapidly moving data off data producers and then continuously processing the data. Kinesis Data Streams stores data for later processing by applications (key difference with Firehose which delivers data directly to AWS services).

Kinesis Firehose can allow transformation of data and it then delivers data to supported services.

RedShift is a data warehouse solution used for analyzing data.

EMR is a hosted Hadoop framework that is used for analytics.

Question 4 answer: *C*

Explanation:

Access Logs can be enabled on ALB and configured to store data in an S3 bucket. Amazon EMR is a web service that enables businesses, researchers, data analysts, and developers to easily and cost-effectively process vast amounts of data. EMR utilizes a hosted Hadoop framework running on Amazon EC2 and Amazon S3.

Neither Kinesis nor EC2 provide a hosted Hadoop service.

You cannot configure access logs to be delivered to DynamoDB.

Question 5 answer: *A,C*

Explanation:

Amazon Kinesis makes it easy to collect, process, and analyze real-time streaming data. With Amazon Kinesis Analytics, you can run standard SQL or build entire streaming applications using SQL.

Amazon Simple Notification Service (Amazon SNS) provides a fully managed messaging service for pub/sub patterns using asynchronous event notifications and mobile push notifications for microservices, distributed systems, and serverless applications.

Amazon Elastic Map Reduce runs on EC2 instances so is not serverless.

Amazon Simple Workflow Service is used for executing tasks not sending messages.

Amazon CloudTrail is used for recording API activity on your account.

AWS IAM

General IAM Concepts

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 two 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 always setup multi-factor authentication on the root account

IAM is universal (global) and does not apply to regions

IAM is eventually consistent

IAM replicates data across multiple data centres around the world

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

Power user access allows all permissions except the management of groups and users in IAM

Temporary security credentials consist of the AWS access key ID, secret access key, and security token

IAM can assign temporary security credentials to provide users with temporary access to services/resources

To sign-in you must provide your account ID or account alias in addition to a user name and password

The sign-in URL includes the account ID or account alias, e.g:

https://***My_AWS_Account_ID***.signin.aws.amazon.com/console/

Alternatively you can sign-in at the following URL and enter your account ID or alias manually:

https://console.aws.amazon.com/

IAM integrates with many different AWS services

IAM supports PCI DSS compliance

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

IAM Infrastructure Elements

Principals:

- An entity that can take an action on an AWS resource
- Your administrative IAM user is your first principal
- You can allow users and services to assume a role

- IAM supports federated users
- IAM supports programmatic access to allow an application to access your AWS account
- IAM users, roles, federated users, and applications are all AWS principals

Requests:

- Principals send requests via the Console, CLI, SDKs, or APIs
- Requests are:
    - Actions (or operations) that the principal wants to perform
    - Resources upon which the actions are performed
    - Principal information including the environment from which the request was made
- Request context - AWS gathers the request information:
    - Principal (requester)
    - Aggregate permissions associated with the principal
    - Environment data, such as IP address, user agent, SSL status etc.
    - Resource data, or data that is related to the resource being requested

Authentication:

- A principal sending a request must be authenticated to send a request to AWS
- To authenticate from the console, you must sign in with your user name and password
- To authenticate from the API or CLI, you must provide your access key and secret key

Authorisation:

- IAM uses values from the request context to check for matching policies and determines whether to allow or deny the request
- IAM policies are stored in IAM as JSON documents and specify the permissions that are allowed or denied
- IAM policies can be:
    - User (identity) based policies
    - Resource-based policies
- IAM checks each policy that matches the context of your request
- If a single policy has a deny action IAM denies the request and stops evaluating (explicit deny)
- Evaluation logic:
    - By default all requests are denied (implicit deny)
    - An explicit allow overrides the implicit deny
    - An explicit deny overrides any explicit allows
- Only the root user has access to all resources in the account by default

Actions:

- Actions are defined by a service
- Actions are the things you can do to a resource such as viewing, creating, editing, deleting
- Any actions on resources that are not explicitly allowed are denied

- To allow a principal to perform an action you must include the necessary actions in a policy that applies to the principal or the affected resource

Resources:

- A resource is an entity that exists within a service
- E.g. EC2 instances, S3 buckets, IAM users, and DynamoDB tables
- Each AWS service defines a set of actions that can be performed on the resource
- After AWS approves the actions in your request, those actions can be performed on the related resources within your account

Authentication Methods

Console password:

- A password that the user can enter to sign in to 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

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

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

STS

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 a 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

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

Federation with Mobile Apps:

- Use Facebook/Amazon/Google or other OpenID providers to login

Cross Account Access:

- 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

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

IAM Best Practices

Lock away the AWS root user access keys

Create individual IAM users

Use AWS defined policies to assign permissions whenever possible

Use groups to assign permissions to IAM users

Grant least privilege

Use access levels to review IAM permissions

Configure a strong password policy for users

Enable MFA for privileged users

Use roles for applications that run on AWS EC2 instances

Delegate by using roles instead of sharing credentials

Rotate credentials regularly

Remove unnecessary credentials

Use policy conditions for extra security

Monitor activity in your AWS account

AWS Accounts

AWS Organizations

Root account with organizational units and AWS accounts behind the OU's

Policies can be assigned at different points in the hierarchy

Available in two feature sets:

- Consolidated billing
- All features

Consolidated billing separates paying accounts and linked accounts

Limit of 20 linked accounts for consolidated billing (default)

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

Resource Groups

Resource groups allow you to group resources and then tag them

Tag editor assists with finding resources and adding tags

Resource groups contain information such as:

- Region
- Name
- Health Checks

AWS Directory Service

General

AWS provide a number of directory types

The following three types currently feature on the exam and will be covered on this page:

- Active Directory Service for Microsoft Active Directory
- Simple AD
- AD Connector
-

As an alternative to the AWS Directory service you can build your own Microsoft AD DCs in the AWS cloud (on EC2)

- When you build your own you can join an existing on-premise Active Directory domain (replication mode)
- You must establish a VPN (on top of Direct Connect if you have it)
- Replication mode is less secure than establishing trust relationships

The table below summarises the directory services covered on this page as well as a couple of others, and provides some typical use cases:

Directory Service Option	Description	Use Case
AWS Cloud Directory	Cloud-native directory to share and control access to hierarchical data between applications	Cloud applications that need hierarchical data with complex relationships
Amazon Cognito	Sign-up and sign-in functionality that scales to millions of users and federated to public social media services	Develop consumer apps or SaaS
AWS Directory Service for Microsoft Active Directory	AWS-managed full Microsoft AD running on Windows Server 2012 R2	Enterprises that want hosted Microsoft AD or you need LDAP for Linux apps
AD Connector	Allows on-premises users to log into AWS services with their existing AD credentials. Also allows EC2 instances to join AD domain	Single sign-on for on-premises employees and for adding EC2 instances to the domain
Simple AD	Low scale, low cost, AD implementation based on Samba	Simple user directory, or you need LDAP compatibility

Active Directory Service for Microsoft Active Directory

Fully managed AWS services on AWS infrastructure

Best choice if you have more than 5000 users and/or need a trust relationship set up

Includes software pathing, replication, automated backups, replacing failed DCs and monitoring

Runs on a Windows Server

Can perform schema extensions

Works with SharePoint, Microsoft SQL Server and .Net apps

You can setup trust relationships to extend authentication from on-premises Active Directories into the AWS cloud

On-premise users and groups can access resources in either domain using SSO

Requires a VPN or Direct Connect connection

Can be used as a standalone AD in the AWS cloud

When used standalone users can access 3rd party applications such as Microsoft O365 through federation

You can also use Active Directory credentials to authenticate to the AWS management console without having to set up SAML authentication

AWS Microsoft AD supports AWS applications including Workspaces, WorkDocs, QuickSight, Chime, Amazon Connect, and RDS for Microsoft SQL Server

Includes security features such as:

- Fine-grained password policy management
- LDAP encryption through SSL/TLS
- HIPAA and PCI DSS approved
- Multi-factor authentication through integration with existing RADIUS-based MFA infrastructure

Monitoring provided through CloudTrail, notifications through SNS, daily automated snapshots

Scalable service that scales by adding Domain Controllers

Deployed in a HA configuration across two AZs in the same region

AWS Microsoft AD does not support replication mode where replication to an on-premise AD takes place

Two editions:

- Standard Edition is optimized to be a primary directory for small and midsize businesses with up to 5,000 employees. It provides you enough storage capacity to support up to 30,000 directory objects, such as users, groups, and computers
- Enterprise Edition is designed to support enterprise organizations with up to 500,000 directory objects
-

Directory Sharing:

- AWS Directory Service for Microsoft Active Directory allows you to use a directory in one account and share it with multiple accounts and VPCs
- There is an hourly sharing charge for each additional account to which you share a directory
- There is no sharing charge for additional VPCs to which you share a directory, or for the account in which you install the directory

Simple AD

An inexpensive Active Directory-compatible service with common directory features

Standalone, fully managed, directory on the AWS cloud

Simple AD is generally the least expensive option

Best choice for less than 5000 users and don't need advanced AD features

Powered by SAMBA 4 Active Directory compatible server

Can create users and control access to applications on AWS

Provides a subset of the features provided by AWS MS AD

Features include:

- Manage user accounts
- Manage groups
- Apply group policies
- Securely connect to EC2 instances
- Kerberos-based SSO
- Supports joining Linux or Windows based EC2 instances

AWS provides monitoring, daily snapshots, and recovery services

Manual snapshots possible

Simple AD is compatible with WorkSpaces, WorkDocs, Workmail and QuickSight

You can also sign on to the AWS management console with Simple AD user accounts to manage AWS resources

Available in two editions:

- Small - supports up to 500 users (approximately 2000 objects)
- Large - supports up to 5000 users (approximately 20,000 objects)

AWS creates two directory servers and DNS servers on two different subnets within an AZ

Simple AD does not support:

- DNS dynamic updates
- Schema extensions
- Multi-factor authentication
- Communication over LDAPS
- PowerShell AD cmdlets
- FSMO role transfer

Not compatible with RDS SQL server

Does not support trust relationships with other domains (use AWS MS AD)

AD Connector

AD Connector is a directory gateway for redirecting directory requests to your on-premise Active Directory

AD Connector eliminates the need for directory synchronization and the cost and complexity of hosting a federation infrastructure

Connects your existing on-premise AD to AWS

Best choice when you want to use an existing Active Directory with AWS services

AD Connector comes in two sizes:

- Small - designed for organizations up to 500 users
- Large - designed for organizations up to 5000 users

The VPC must be connected to your on-premise network via VPN or Direct Connect

When users log in to AWS applications AD connector forwards sign-in requests to your on-premise AD DCs

You can also join EC2 instances to your on-premise AD through AD Connector

You can also login to the AWS Management Console using your on-premise AD DCs for authentication

Not compatible with RDS SQL

You can use AD Connector for multi-factor authentication using RADIUS-based MFA infrastructure

AD Connector vs Simple AD

The table below describes some of the key differences to consider when choosing AD Connector or Simple AD:

AD Connector	Simple AD
Must have an existing AD	Standalone AD based on Samba
Existing AD users can access AWS assets via IAM roles	Supports user accounts, groups, group policies, and domains
Supports MFA via existing RADIUS-based MFA infrastructure	Kerberos-based SSO
	MFA not supported
	Trust relationships not supported

AWS KMS

AWS Key Management Store (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

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

You set usage policies on these keys that determine which users can use them to encrypt and decrypt data under which conditions

KMS is tightly integrated into many AWS services like Lambda, S3, EBS, EFS, DynamoDB, SQS etc.

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

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)

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.

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

Limits:

- You can create up to 1000 customer master keys per account per region
- As both enabled and disabled customer master keys count towards the limit, AWS recommend deleting disabled keys that you no longer use
- AWS managed master keys created on your behalf for use within supported AWS services do not count against this limit
- There is no limit to the number of data keys that can be derived using a master key and used in your application or by AWS services to encrypt data on your behalf

AWS CloudHSM

The AWS CloudHSM service helps you meet corporate, contractual and regulatory compliance requirements for data security by using dedicated Hardware Security Module (HSM) instances within the AWS cloud

AWS and AWS Marketplace partners offer a variety of solutions for protecting sensitive data within the AWS platform, but for some applications and data subject to contractual or regulatory mandates for managing cryptographic keys, additional protection may be necessary

CloudHSM complements existing data protection solutions and allows you to protect your encryption keys within HSMs that are designed and validated to government standards for secure key management

CloudHSM allows you to securely generate, store and manage cryptographic keys used for data encryption in a way that keys are accessible only by you

A Hardware Security Module (HSM) provides secure key storage and cryptographic operations within a tamper-resistant hardware device

HSMs are designed to securely store cryptographic key material and use the key material without exposing it outside the cryptographic boundary of the hardware

You can use the CloudHSM service to support a variety of use cases and applications, such as database encryption, Digital Rights Management (DRM), Public Key Infrastructure (PKI), authentication and authorization, document signing, and transaction processing

The table below describes the latest version of CloudHSM and how it differs from its predecessor:

	"Classic" CloudHSM	Current CloudHSM
Device	safeNET Luna SA	Proprietary
Pricing	Upfront cost required ($5000)	No upfront cost, pay per hour
High Availability	Have to buy a second device	Clustered
FIPS 140-2	Level 2	Level 3

When you use the AWS CloudHSM service you create a CloudHSM Cluster

Clusters can contain multiple HSM instances, spread across multiple Availability Zones in a region. HSM instances in a cluster are automatically synchronized and load-balanced

You receive dedicated, single-tenant access to each HSM instance in your cluster. Each HSM instance appears as a network resource in your Amazon Virtual Private Cloud (VPC)

Adding and removing HSMs from your Cluster is a single call to the AWS CloudHSM API (or on the command line using the AWS CLI)

After creating and initializing a CloudHSM Cluster, you can configure a client on your EC2 instance that allows your applications to use the cluster over a secure, authenticated network connection

Must be within a VPC and can be accessed via VPC Peering

Applications don't need to be in the same VPC but the server or instance on which your application and the HSM client are running must have network (IP) reachability to all HSMs in the cluster

Does not natively integrate with many AWS services like KMS, but instead requires custom application scripting

Offload SSL from web server, act as an issuing CA, enable TDE for Oracle databases

The table below compares CloudHSM against 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

Security, Identity & Compliance Practice Questions

Answers and explanations are provided below after the last question in this section.

Question 1:

A company needs to deploy virtual desktops for its customers in an AWS VPC, and would like to leverage their existing on-premise security principles. AWS Workspaces will be used as the virtual desktop solution.

Which set of AWS services and features will meet the company's requirements?

A. A VPN connection. AWS Directory Services
B. A VPN connection, VPC NACLs and Security Groups
C. A VPN connection, VPC NACLs and Security Groups
D. Amazon EC2, and AWS IAM

Question 2:

To improve security in your AWS account you have decided to enable multi-factor authentication (MFA). You can authenticate using an MFA device in which two ways? (choose 2)

A. Locally to EC2 instances
B. Through the AWS Management Console
C. Using biometrics
D. Using a key pair
E. Using the AWS API

Question 3:

Your company would like to restrict the ability of most users to change their own passwords whilst continuing to allow a select group of users within specific user groups.

What is the best way to achieve this? (choose 2)

A. Under the IAM Password Policy deselect the option to allow users to change their own passwords
B. Create an IAM Policy that grants users the ability to change their own password and attach it to the groups that contain the users
C. Create an IAM Role that grants users the ability to change their own password and attach it to the groups that contain the users
D. Create an IAM Policy that grants users the ability to change their own password and attach it to the individual user accounts
E. Disable the ability for all users to change their own passwords using the AWS Security Token Service

Question 4:

Your company has started using the AWS CloudHSM for secure key storage. A recent administrative error resulted in the loss of credentials to access the CloudHSM. You need access to data that was encrypted using keys stored on the hardware security module.

How can you recover the keys that are no longer accessible?

A. There is no way to recover your keys if you lose your credentials
B. Log a case with AWS support and they will use MFA to recover the credentials
C. Restore a snapshot of the CloudHSM
D. Reset the CloudHSM device and create a new set of credentials

Question 5:

The AWS Acceptable Use Policy describes permitted and prohibited behavior on AWS and includes descriptions of prohibited security violations and network abuse. According to the policy, what is AWS's position on penetration testing?

A. AWS do not allow any form of penetration testing
B. AWS allow penetration testing by customers on their own VPC resources
C. AWS allow penetration for some resources with prior authorization
D. AWS allow penetration testing for all resources

Question 6:

You have been asked to come up with a solution for providing single sign-on to existing staff in your company who manage on-premise web applications and now need access to the AWS management console to manage resources in the AWS cloud.

Which product combinations provide the best solution to achieve this requirement?

A. Use your on-premise LDAP directory with IAM
B. Use IAM and MFA
C. Use the AWS Secure Token Service (STS) and SAML
D. Use IAM and Amazon Cognito

Question 7:

You are a Developer working for Digital Cloud Training. You are planning to write some code that creates a URL that lets users who sign in to your organization's network securely access the AWS Management Console. The URL will include a sign-in token that you get from AWS that authenticates the user to AWS. You are using Microsoft Active Directory Federation Services as your identity provider (IdP) which is compatible with SAML 2.0.

Which of the steps below will you need to include when developing your custom identity broker? (choose 2)

A. Generate a pre-signed URL programmatically using the AWS SDK for Java or the AWS SDK for .NET

B. Call the AWS Security Token Service (AWS STS) AssumeRole or GetFederationToken API operations to obtain temporary security credentials for the user
C. Delegate access to the IdP through the "Configure Provider" wizard in the IAM console
D. Call the AWS federation endpoint and supply the temporary security credentials to request a sign-in token
E. Assume an IAM Role through the console or programmatically with the AWS CLI, Tools for Windows PowerShell or API

Question 8:

A health club is developing a mobile fitness app that allows customers to upload statistics and view their progress. Amazon Cognito is being used for authentication, authorization and user management and users will sign-in with Facebook IDs.

In order to securely store data in DynamoDB, the design should use temporary AWS credentials. What feature of Amazon Cognito is used to obtain temporary credentials to access AWS services?

A. User Pools
B. Identity Pools
C. SAML Identity Providers
D. Key Pairs

Question 1 answer: A

Explanation:

A security principle is an individual identity such as a user account within a directory. The AWS Directory service includes: Active Directory Service for Microsoft Active Directory, Simple AD, AD Connector. One of these services may be ideal depending on detailed requirements. The Active Directory Service for Microsoft AD and AD Connector both require a VPN or Direct Connect connection.

A VPN with NACLs and security groups will not deliver the required solution. AWS Directory Service with IAM or EC2 with IAM are also not sufficient for leveraging on-premise security principles. You must have a VPN.

Question 2 answer: A,E

Explanation:

You can authenticate using an MFA device in the following 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)

Question 3 answer: *A,B*

Explanation:

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 and you should attach this to the group that contains the users, not to the individual users themselves.

You cannot use an IAM role to perform this function.

The AWS STS is not used for controlling password policies.

Question 4 answer: *A*

Explanation:

Amazon does not have access to your keys or credentials and therefore has no way to recover your keys if you lose your credentials.

Question 5 answer: *C*

Explanation:

Permission is required for all penetration tests.

You must complete and submit the AWS Vulnerability / Penetration Testing Request Form to request authorization for penetration testing to or originating from any AWS resources.

There is a limited set of resources on which penetration testing can be performed.

Question 6 answer: C

Explanation:

Single sign-on using federation allows users to login to the AWS console without assigning IAM credentials.

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 (such as federated users from an on-premise directory).

Federation (typically Active Directory) uses SAML 2.0 for authentication and grants temporary access based on the users AD credentials. The user does not need to be a user in IAM.

You cannot use your on-premise LDAP directory with IAM, you must use federation.

Enabling multi-factor authentication (MFA) for IAM is not a federation solution.

Amazon Cognito is used for authenticating users to web and mobile apps not for providing single sign-on between on-premises directories and the AWS management console.

Question 7 answer: *B,D*

Explanation:

The aim of this solution is to create a single sign-on solution that enables users signed in to the organization's Active Directory service to be able to connect to AWS resources. When developing a custom identity broker you use the AWS STS 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). The steps performed by the custom identity broker to sign users into the AWS management console are:

1. Verify that the user is authenticated by your local identity system
2. Call the AWS Security Token Service (AWS STS) AssumeRole or GetFederationToken API operations to obtain temporary security credentials for the user
3. Call the AWS federation endpoint and supply the temporary security credentials to request a sign-in token
4. Construct a URL for the console that includes the token
5. Give the URL to the user or invoke the URL on the user's behalf

You cannot generate a pre-signed URL for this purpose using SDKs, delegate access through the IAM console or directly assume IAM roles.

Question 8 answer: *B*

Explanation:

With an identity pool, users can obtain temporary AWS credentials to access AWS services, such as Amazon S3 and DynamoDB.

A user pool is a user directory in Amazon Cognito. With a user pool, users can sign in to web or mobile apps through Amazon Cognito, or federate through a third-party identity provider (IdP).

SAML Identity Providers are supported IDPs for identity pools but cannot be used for gaining temporary credentials for AWS services.

Key pairs are used in Amazon EC2 for access to instances.

APPLICATION INTEGRATION

Amazon SNS

Amazon Simple Notification Service (Amazon SNS) is a web service that makes it easy to set up, operate, and send notifications from the cloud

Amazon SNS is used for building and integrating loosely-coupled, distributed applications

Provides instantaneous, push-based delivery (no polling)

Uses simple APIs and easy integration with applications

Flexible message delivery is provided over multiple transport protocols

Offered under an inexpensive, pay-as-you-go model with no up-front costs

The web-based AWS Management Console offers the simplicity of a point-and-click interface

Data type is JSON

SNS supports a wide variety of needs including event notification, monitoring applications, workflow systems, time-sensitive information updates, mobile applications, and any other application that generates or consumes notifications

SNS Subscribers:

- HTTP
- HTTPS
- Email
- Email-JSON
- SQS
- Application
- Lambda

SNS supports notifications over multiple transport protocols:

- HTTP/HTTPS - subscribers specify a URL as part of the subscription registration
- Email/Email-JSON - messages are sent to registered addresses as email (text-based or JSON-object)
- SQS - users can specify an SQS standard queue as the endpoint
- SMS - messages are sent to registered phone numbers as SMS text messages

Topic names are limited to 256 characters

SNS supports CloudTrail auditing for authenticated calls

SNS provides durable storage of all messages that it receives (across multiple AZs)

Users pay $0.50 per 1 million Amazon SNS Requests, $0.06 per 100,000 notification deliveries over HTTP, and $2.00 per 100,000 notification deliveries over email

Amazon SQS

General SQS Concepts

Amazon Simple Queue Service (Amazon SQS) is a web service that gives you access to message queues that store messages waiting to be processed

SQS offers a reliable, highly-scalable, hosted queue for storing messages in transit between computers

SQS is used for distributed/decoupled applications

SQS can be used with RedShift, DynamoDB, EC2, ECS, RDS, S3 and Lambda

SQS uses a message-oriented API

SQS uses pull based (polling) not push based

Messages are 256KB in size

Messages can be kept in the queue from 1 minute to 14 days (default is 4 days)

The visibility timeout is the amount of time a message is invisible in the queue after a reader picks up the message

If a job is processed within the visibility timeout the message will be deleted

If a job is not processed within the visibility timeout the message will become visible again (could be delivered twice)

The maximum visibility timeout for an Amazon SQS message is 12 hours

An Amazon SQS message can contain up to 10 metadata attributes

Polling

SQS uses short polling and long polling

Short polling:

- Does not wait for messages to appear in the queue
- It queries only a subset of the available servers for messages (based on weighted random execution)
- Short polling is the default
- ReceiveMessageWaitTime is set to 0
- More requests are used, which implies higher cost

Long polling:

- Uses fewer requests and reduces cost
- Eliminates false empty responses by querying all servers
- SQS waits until a message is available in the queue before sending a response
- Requests contain at least one of the available messages up to the maximum number of messages specified in the ReceiveMessage action

- Shouldn’t be used if your application expects an immediate response to receive message calls
- ReceiveMessageWaitTime is set to a non-zero value (up to 20 seconds)
- Same charge per million requests as short polling

Queues

Queue names must be unique within a region

Queues can be either standard or first-in-first-out (FIFO)

Standard queues provide a loose-FIFO capability that attempts to preserve the order of messages

Because standard queues are designed to be massively scalable using a highly distributed architecture, receiving messages in the exact order they are sent is not guaranteed

Standard queues provide at-least-once delivery, which means that each message is delivered at least once

FIFO (first-in-first-out) queues preserve the exact order in which messages are sent and received

FIFO queues are available in limited regions currently

If you use a FIFO queue, you don't have to place sequencing information in your message

FIFO queues provide exactly-once processing, which means that each message is delivered once and remains available until a consumer processes it and deletes it

Limits

In-flight messages are messages that have been picked up by a consumer but not yet deleted from the queue

Standard queues have a limit of 120,000 in-flight messages per queue

FIFO queues have a limit of 20,000 in-flight messages per queue

Queue names can be up to 80 characters

Messages are retained for 4 days by default up to 14 days

FIFO queues support up to 3000 messages per second when batching or 300 per second otherwise

The maximum messages size is 256KB

Scalability and Durability

You can have multiple queues with different priorities

Scaling is performed by creating more queues

SQS stores all message queues and messages within a single, highly-available AWS region with multiple redundant AZs

Security

You can use IAM policies to control who can read/write messages

Authentication can be used secure messages within queues (who can send and receive)

SQS supports HTTPS and supports TLS versions 1.0, 1.1, 1.2

SQS is PCI DSS level 1 compliant and HIPAA eligible

Server-side encryption (SSE) lets you transmit sensitive data in encrypted queues (AWS KMS):

- SSE encrypts messages as soon as SQS receives them
- The messages are stored in encrypted form and SQS decrypts messages only when they are sent to an authorized consumer
- Uses AES 256 bit encryption
- Not available in all regions
- Standard and FIFO queues
- Body of message is encrypted
- The following is not encrypted:
    - Queue metadata
    - Message metadata
    - Per-queue metrics

Monitoring

CloudWatch is integrated with SQS and you can view and monitor queue metrics

CloudWatch metrics are automatically collected every 5 minutes

CloudWatch considers a queue to be active for up to 6 hours if it contains any messages or if any API action accesses it

No charge for CloudWatch (no detailed monitoring)

CloudTrail captures API calls from SQS and logs to a specified S3 bucket

Charges

The cost of SQS is calculated per request, plus data transfer charges for data transferred out of SQS

The following table describes related services and typical use cases for them:

Service	What it does	Suggested Use Case
Step Functions	Out-of-the-box coordination of AWS service components	Order processing workflow
Simple Workflow Service	Need to support external processes or specialized execution logic	Loan application process with manual review steps
Simple Queue Service	Messaging queue; store and forward patterns	Image resize process
AWS Batch	Scheduled or reoccurring tasks that do not require heavy logic	Rotate logs daily on firewall appliance

Amazon SWF

Amazon Simple Workflow Service (SWF) is a web service that makes it easy to coordinate work across distributed application components

Create distributed asynchronous systems as workflows

Supports both sequential and parallel processing

Tracks the state of your workflow which you interact and update via API

Best suited for human-enabled workflows like an order fulfilment system or for procedural requests

AWS recommends that for new applications customers consider Step Functions instead of SWF

SWF enables applications for a range of use cases, including media processing, web application back-ends, business process workflows, and analytics pipelines, to be designed as a coordination of tasks

Registration is a one-time step that you perform for each different type of workflow and activity

SWF has a completion time of up to 1 year for workflow executions

SWF uses a task-oriented API

SWF ensures a task is assigned once and never duplicated

SWF keeps track of all the tasks and events in an application

A domain is a logical container for application resources such as workflows, activities, and executions

Workers are programs that interact with Amazon SWF to get tasks, process received tasks, and return the results

The decider is a program that controls the coordination of tasks, i.e. their ordering, concurrency, and scheduling according to the application logic

SWF applications include the following logical components:

- Domains
- Workflows
- Activities
- Task Lists
- Workers
- Workflow Execution

The following table describes related services and typical use cases for them:

Amazon MQ

Amazon MQ is a managed message broker service for ActiveMQ that makes it easy to set up and operate message brokers in the cloud, so you can migrate your messaging and applications without rewriting code

Amazon MQ supports industry-standard APIs and protocols so you can migrate messaging and applications without rewriting code

Amazon MQ provides cost-efficient and flexible messaging capacity - you pay for broker instance and storage usage as you go

Amazon MQ manages the administration and maintenance of ActiveMQ brokers and automatically provisions infrastructure for high availability

With Amazon MQ, you can use the AWS Management Console, AWS CloudFormation, the Command Line Interface (CLI), or simple API calls to launch a production-ready message broker in minutes

It's a managed implementation of Apache ActiveMQ

Fully managed and highly available within a region

Amazon MQ stores your messages redundantly across multiple Availability Zones (AZs)

Active/standby brokers are designed for high availability. In the event of a failure of the broker, or even a full AZ outage, Amazon MQ automatically fails over to the standby broker so you can continue sending and receiving messages

ActiveMQ API and support for JMS, NMS, MQTT, and WebSockets

Designed as a drop-in replacement for on-premise message brokers

Use SQS if you're creating a new application from scratch

Use MQ if you want an easy low-hassle path to migrate from existing message brokers to AWS

Amazon MQ provides encryption of your messages at rest and in transit

It's easy to ensure that your messages are securely stored in an encrypted format. Connections to the broker use SSL, and access can be restricted to a private endpoint within your Amazon VPC, which allows you to isolate your broker in your own virtual network

You can configure security groups to control network access to your broker

Amazon MQ is integrated with Amazon CloudWatch and AWS CloudTrail. With CloudWatch you can monitor metrics on your brokers, queues, and topics

AWS Step Functions

AWS Step Functions makes it easy to coordinate the components of distributed applications as a series of steps in a visual workflow

You can quickly build and run state machines to execute the steps of your application in a reliable and scalable fashion

How it works:

1. Define the steps of your workflow in the JSON-based Amazon States Language. The visual console automatically graphs each step in the order of execution.
2. Start an execution to visualize and verify the steps of your application are operating as intended. The console highlights the real-time status of each step and provides a detailed history of every execution.
3. AWS Step Functions operates and scales the steps of your application and underlying compute for you to help ensure your application executes reliably under increasing demand.

Managed workflow and orchestration platform

Scalable and highly available

Define your app as a state machine

Create tasks, sequential steps, parallel steps, branching paths or timers

Amazon State Language declarative JSON

Apps can interact and update the stream via Step Function API

Visual interface describes flow and real-time status

Detailed logs of each step execution

Benefits and Features:

- **Built-in error handling -** AWS Step Functions tracks the state of each step, so you can automatically retry failed or timed-out tasks, catch specific errors, and recover gracefully, whether the task takes seconds or months to complete
- **Automatic Scaling -** AWS Step Functions automatically scales the operations and underlying compute to run the steps of your application for you in response to changing workloads. Step Functions scales automatically to help ensure the performance of your application workflow remains consistently high as the frequency of requests increases
- **Pay per use -** With AWS Step Functions, you pay only for the transition from one step of your application workflow to the next, called a state transition. Billing is metered by state transition, regardless of how long each state persists (up to one year)
- **Execution event history -** AWS Step Functions creates a detailed event log for every execution, so when things do go wrong, you can quickly identify not only where, but

why. All of the execution history is available visually and programmatically to quickly troubleshoot and remediate failures

- **High availability** - AWS Step Functions has built-in fault tolerance. Step Functions maintains service capacity across multiple Availability Zones in each region to help protect application workflows against individual machine or data center facility failures. There are no maintenance windows or scheduled downtimes
- **Administrative security** - AWS Step Functions is integrated with AWS Identity and Access Management (IAM). IAM policies can be used to control access to the Step Functions APIs

Application Integration Practice Questions

Answers and explanations are provided below after the last question in this section.

Question 1:

There is expected to be a large increase in write intensive traffic to a website you manage that registers users onto an online learning program. You are concerned about writes to the database being dropped and need to come up with a solution to ensure this does not happen. Which of the solution options below would be the best approach to take?

A. Update the application to write data to an SQS queue and provision additional EC2 instances to process the data and write it to the database
B. Use RDS in a multi-AZ configuration to distribute writes across AZs
C. Update the application to write data to an S3 bucket and provision additional EC2 instances to process the data and write it to the database
D. Use CloudFront to cache the writes and configure the database as a custom origin

Question 2:

You are using a series of Spot instances that process messages from an SQS queue and store results in a DynamoDB table. Shortly after picking up a message from the queue AWS terminated the Spot instance. The Spot instance had not finished processing the message. What will happen to the message?

A. The message will be lost as it would have been deleted from the queue when processed
B. The message will remain in the queue and be immediately picked up by another instance
C. The message will become available for processing again after the visibility timeout expires
D. The results may be duplicated in DynamoDB as the message will likely be processed multiple times

Question 3:

You are developing a multi-tier application that includes loosely-coupled, distributed application components and need to determine a method of sending notifications instantaneously. Using SNS which transport protocols are supported? (choose 2)

A. FTP
B. Email-JSON
C. HTTPS
D. Amazon SWF
E. AWS Lambda

Question 4:

A Solutions Architect is creating the business process workflows associated with an order fulfilment system. What AWS service can assist with coordinating tasks across distributed application components?

A. Amazon STS
B. Amazon SQS
C. Amazon SWF
D. Amazon SNS

Question 5:

You are a developer at Digital Cloud Training. An application stack you are building needs a message bus to decouple the application components from each other. The application will generate up to 300 messages per second without using batching. You need to ensure that a message is only delivered once, and duplicates are not introduced into the queue. It is not necessary to maintain the order of the messages.

Which SQS queue type will you use?

A. Standard queues
B. Long polling queues
C. Auto Scaling queues
D. FIFO queues

Question 6:

A client is in the design phase of developing an application that will process orders for their online ticketing system. The application will use a number of front-end EC2 instances that pick-up orders and place them in a queue for processing by another set of back-end EC2 instances. The client will have multiple options for customers to choose the level of service they want to pay for.

The client has asked how he can design the application to process the orders in a prioritized way based on the level of service the customer has chosen?

A. Create multiple SQS queues, configure the front-end application to place orders onto a specific queue based on the level of service requested and configure the back-end instances to sequentially poll the queues in order of priority
B. Create a combination of FIFO queues and Standard queues and configure the applications to place messages into the relevant queue based on priority
C. Create a single SQS queue, configure the front-end application to place orders on the queue in order of priority and configure the back-end instances to poll the queue and pick up messages in the order they are presented
D. Create multiple SQS queues, configure exactly-once processing and set the maximum visibility timeout to 12 hours

Question 1 answer: *A*

Explanation:

This is a great use case for Amazon Simple Queue Service (Amazon SQS). SQS is a web service that gives you access to message queues that store messages waiting to be processed and offers a reliable, highly-scalable, hosted queue for storing messages in transit between computers. SQS is used for distributed/decoupled applications. In this circumstance SQS will reduce the risk of writes being dropped and it the best option presented.

RDS in a multi-AZ configuration will not help as writes are only made to the primary database.

Though writing data to an S3 bucket could potentially work, it is not the best option as SQS is recommended for decoupling application components.

The CloudFront option is bogus as you cannot configure a database as a custom origin in CloudFront.

Question 2 answer: *C*

Explanation:

The visibility timeout is the amount of time a message is invisible in the queue after a reader picks up the message. If a job is processed within the visibility timeout the message will be deleted. If a job is not processed within the visibility timeout the message will become visible again (could be delivered twice). The maximum visibility timeout for an Amazon SQS message is 12 hours.

The message will not be lost and will not be immediately picked up by another instance. As mentioned above it will be available for processing in the queue again after the timeout expires.

As the instance had not finished processing the message it should only be fully processed once. Depending on your application process however it is possible some data was written to DynamoDB.

Question 3 answer: *B,C*

Explanation:

Note that the questions asks you which transport protocols are supported, NOT which subscribers - therefore Lambda is not supported

SNS supports notifications over multiple transport protocols:

- HTTP/HTTPS – subscribers specify a URL as part of the subscription registration
- Email/Email-JSON – messages are sent to registered addresses as email (text-based or JSON-object)
- SQS – users can specify an SQS standard queue as the endpoint
- SMS – messages are sent to registered phone numbers as SMS text messages

Question 4 answer: C

Explanation:

Amazon Simple Workflow Service (SWF) is a web service that makes it easy to coordinate work across distributed application components. SWF enables applications for a range of use cases, including media processing, web application back-ends, business process workflows, and analytics pipelines, to be designed as a coordination of tasks.

Amazon Security Token Service (STS) is used for requesting temporary credentials.

Amazon Simple Queue Service (SQS) is a message queue used for decoupling application components.

Amazon Simple Notification Service (SNS) is a web service that makes it easy to set up, operate, and send notifications from the cloud.

SNS supports notifications over multiple transports including HTTP/HTTPS, Email/Email-JSON, SQS and SMS.

Question 5 answer: D

Explanation:

The key fact you need to consider here is that duplicate messages cannot be introduced into the queue. For this reason alone you must use a FIFO queue. The statement about it not being necessary to maintain the order of the messages is meant to confuse you, as that might lead you to think you can use a standard queue, but standard queues don't guarantee that duplicates are not introduced into the queue.

FIFO (first-in-first-out) queues preserve the exact order in which messages are sent and received – note that this is not required in the question but exactly once processing is. FIFO queues provide exactly-once processing, which means that each message is delivered once and remains available until a consumer processes it and deletes it.

Standard queues provide a loose-FIFO capability that attempts to preserve the order of messages. Standard queues provide at-least-once delivery, which means that each message is delivered at least once.

Long polling is configuration you can apply to a queue, it is not a queue type.

There is no such thing as an Auto Scaling queue.

Question 6 answer: *A*

Explanation:

The best option is to create multiple queues and configure the application to place orders onto a specific queue based on the level of service. You then configure the back-end instances to poll these queues in order or priority so they pick up the higher priority jobs first.

Creating a combination of FIFO and standard queues is incorrect as creating a mixture of queue types is not the best way to separate the messages, and there is nothing in this option that explains how the messages would be picked up in the right order.

Creating a single queue and configuring the applications to place orders on the queue in order of priority would not work as standard queues offer best-effort ordering so there's no guarantee that the messages would be picked up in the correct order.

Creating multiple SQS queues and configuring exactly-once processing (only possible with FIFO) would not ensure that the order of the messages is prioritized.

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 Solutions Architect 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.

Assess your exam readiness with our pool of 500 practice questions

The Digital Cloud Training practice questions are the closest to the actual exam question format you can find and the only exam-difficulty questions on the market. If you can pass our exams, you're well set to smash the real thing! To sign up, click the link below:

AWS Certified Solutions Architect Associate Practice Exams

Get Hands-On experience with AWS!

AWS certification exams such as the Solutions Architect Associate test your hands-on knowledge and experience with the AWS platform. It's therefore super important to have some real experience before you sit the exam.

Our AWS Certified Solutions Architect Associate Hands-On Labs course provides a practical approach to learning. Through over 20 hours of videos you will learn how to architect and build solutions on Amazon Web Services. By the end of the course you will have a strong experience-based skillset. This is the best way to develop strong hands-on skills and will really help you when it comes time to answer exam questions.

Reach out with any questions you may have

Join our private Facebook group to ask questions and share knowledge and exam tips with the AWS community: https://www.facebook.com/groups/awscertificationqa

For technical support send an email to support@digitalcloud.training

DigitalCloud
TRAINING

Best wishes for your AWS certification journey!

OTHER BOOKS & COURSES BY NEAL DAVIS

AWS Certified Solutions Architect Associate (online) Practice Tests

AVAILABLE ON DIGITALCLOUD.TRAINING

Are you looking for top-quality practice questions so you can ace your AWS Certified Solutions Architect Associate exam? Well, you're in the right place!

Digital Cloud Training provides you a unique learning experience based on top quality AWS Training using practice questions and detailed AWS Certification resources so you pass your AWS Solution Architect exam first time.

The training package includes practice exams in simulation mode, training mode and knowledge reviews:

- **Simulation mode:** the number of questions, time limit, and pass mark are the same as the real AWS exam. You must complete the exam before you are able to check your score and review answers and explanations.
- **Training mode:** You are shown the answer and explanation for every question after clicking "check". Upon completion of the exam the score report shows your overall score and performance in each knowledge area.
- **Knowledge reviews:** Collections of practice questions for a specific knowledge area. When you complete a practice exam you can use the score report to identify your strengths and weaknesses and then use the knowledge reviews to focus your efforts where they're needed most.

There are 6 practice exams in simulation and training mode with 65 questions each, and each exam includes questions from the five domains of the AWS exam blueprint.

All questions are also available in the knowledge reviews where they are split into more than 15 categories for focussed training.

Click here to fast-track your AWS Certified Solutions Architect Associate Exam Success

AWS Certified Solutions Architect Associate (offline) Practice Tests

AVAILABLE ON AMAZON ONLY

The **AWS Solutions Architect Associate certification** is extremely valuable in the Cloud Computing industry today and preparing to answer the difficult scenario-based questions requires a significant commitment in time and effort.

The **latest SAA-C01 exam** is composed entirely of scenario-based questions that test your knowledge and experience working with Amazon Web Services. Our practice tests are patterned to reflect the difficulty of the AWS exam and are the closest to the real AWS exam experience available anywhere.

There are **6 practice exams with 65 questions each** covering the five domains of the AWS 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:

- To simulate the exam experience and assess your exam readiness, use the "**PRACTICE QUESTIONS ONLY**" sets.
- 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.

With more than 20 years of experience in the IT industry, **Neal Davis** is a true expert in virtualization and cloud computing. Neal's practice tests have been used by over 20,000 students and are highly regarded for their quality and similarity to the real AWS exam.

These Practice Questions will prepare you for your AWS exam in the following ways:

- **Master the new exam pattern**: All 390 practice questions are based on the SAA-C01 exam blueprint and use the question format of the real AWS exam
- **6 sets of exam-difficulty practice questions**: Presented with and without answers so you can study or simulate an exam
- **Ideal exam prep tool that will shortcut your study time**: Assess your exam readiness to maximize your chance of passing the AWS exam first time

The exam covers a broad set of technologies and it's vital to ensure you are armed with the knowledge to answer whatever questions come up in your certification exam, so we recommend reviewing these practice questions until you're confident in all areas and **ready to ace your AWS exam**.

AWS Certified Solutions Architect Associate Hands-on Labs Video Course

The only AWS Certified Solutions Architect Associate (SAA-C01) course delivered through practical AWS Hands-On Labs.

With our practical AWS Labs approach, you will learn how to architect and build solutions on Amazon Web Services and will have a strong experience-based skillset by the end of the course. You will be looking over my shoulder and building applications using guided AWS Practice Labs.

We start with the basics of setting up an account and use a process of repetition and incremental learning to build practical skills and theoretical knowledge. We take you from opening your first AWS Free Tier account through to creating complex multi-tier architectures, always sticking to the SAA-C01 exam blueprint to ensure you're learning practical skills and also preparing for your exam. Our method ensures that you retain the knowledge as repeated practice is the best way to learn and build your cloud skills.

How is this course different?

- **You won't just learn, you'll do**. Our AWS Hands-On Labs teach you how to design and build multi-tier web architectures with services such as EC2 Auto Scaling, Elastic Load Balancing, Route 53, ECS, Lambda, API Gateway and Elastic File System.
- We back the +20 hours of AWS Hands-On Labs with **high-quality logical diagrams** so you can visualize what you're building and check your progress
- We use a **process of logical progression**, building upon knowledge and skills in each section so you reuse and remember.

To enrol on the Udemy platform for less than $10, click here.

AWS Certified Cloud Practitioner Training Notes

Save valuable time by getting straight to the facts you need to know to be successful and ensure you pass your AWS Certified Cloud Practitioner exam first time!

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.

The Cloud Practitioner certification is a great first step into the world of Cloud Computing and requires a foundational knowledge of the AWS Cloud, its architectural principles, value proposition, billing and pricing, key services and more.

AWS Solutions Architect and successful instructor, Neal Davis, has consolidated the information you need to be successful from numerous training sources and AWS FAQ pages to save you time.

In addition to the book, you are provided with access to a 65-question practice exam on an interactive exam simulator to evaluate your progress and ensure you're prepared for the style and difficulty of the real AWS exam.

This book can help you prepare for your AWS exam in the following ways:

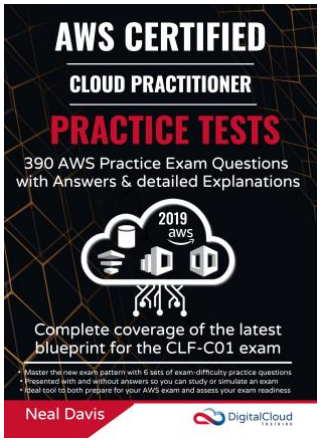
• Deep dive into the CLF-C01 exam objectives with over 150 pages of detailed facts, tables, and diagrams – everything you need to know!

• Familiarize yourself with the exam question format with the practice questions included in each section.

• Use our online exam simulator to evaluate progress and ensure you're ready for the real thing.

AWS Certified Cloud Practitioner (offline) Practice Tests

AVAILABLE ON AMAZON ONLY



The **AWS Cloud Practitioner** exam is a foundational level exam that nonetheless includes tricky questions that test your knowledge and experience of the AWS Cloud. Our practice tests are patterned to reflect the difficulty of the AWS exam and are the closest to the real AWS exam experience available anywhere.

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.

These Practice Questions will prepare you for your AWS exam in the following ways:

- **Master the latest exam pattern**: All 390 practice questions are based on the latest version of the CLF-C01 exam blueprint and use the question format of the real AWS exam
- **6 sets of exam-difficulty practice questions**: Presented with and without answers so you can study or simulate an exam
- **Ideal exam prep tool that will shortcut your study time**: Assess your exam readiness to maximize your chance of passing the AWS exam first time.

ABOUT THE AUTHOR

Neal Davis is the founder of Digital Cloud Training, an AWS Cloud Solutions Architect and a 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. With over 20,000 students currently enrolled in Digital Cloud Training courses, Neal's focus is on creating additional course content and growing his student base.

Connect with Neal on social media:

digitalcloud.training

facebook.com/digitalcloudtraining/

Instagram @nealkdavis

linkedin.com/company/digitalcloudtraining

Twitter @DigitalCloudT