# Amazon S3

* Amazon S3 is one of the main building blocks of AWS
* It’s advertised as ”infinitely scaling” storage
* Many websites use Amazon S3 as a backbone e.g. Nasdaq stores 7 years of data on S3 glacier, Sysco runs data analytics on data on S3
* Many AWS services use Amazon S3 as an integration as well

**Amazon S3 Use cases**

* Backup and storage
* Disaster Recovery
* Archive
* Hybrid Cloud storage
* Application hosting
* Media hosting
* Data lakes & big data analytics
* Software delivery
* Static website

**Amazon S3 - Buckets**

* Amazon S3 allows people to store objects (files) in “buckets” (directories)
* Buckets must have a globally unique name (across all regions all accounts)
* Buckets are defined at the region level
* S3 looks like a global service but buckets are created in a region
* Naming convention

1. No uppercase, No underscore
2. 3-63 characters long
3. Not an IP
4. Must start with lowercase letter or number
5. Must NOT start with the prefix xn--
6. Must NOT end with the suffix -s3alias

**Amazon S3 – Objects**

|  |  |
| --- | --- |
| * Objects (files) have a Key * The key is the FULL path:  1. s3://my-bucket/my\_file.txt 2. s3://my-bucket/my\_folder1/another\_folder/my\_file.txt  * The key is composed of prefix + object name   + s3://my-bucket/my\_folder1/another\_folder/my\_file.txt * There’s no concept of “directories” within buckets * (although the UI will trick you to think otherwise) * Just keys with very long names that contain slashes (“/”) | A green and black object  AI-generated content may be incorrect. |

* Object values are the content of the body:

1. Max. Object Size is 5TB (5000GB)
2. If uploading more than 5GB, must use “multi-part upload”

* Objects can also have metadata (list of text key / value pairs – system or user metadata)
* Ther are also tags (Unicode key / value pair – up to 10) – useful for security / lifecycle
* Can also have Version ID (if versioning is enabled)

## Amazon S3 – Security

* User-Based
  + IAM Policies – which API calls should be allowed for a specific user from IAM
* Resource-Based
  + Bucket Policies – bucket wide rules from the S3 console - allows cross account
  + Object Access Control List (ACL) – finer grain (can be disabled)
  + Bucket Access Control List (ACL) – less common (can be disabled)
* Note: an IAM principal can access an S3 object if
  + The user IAM permissions ALLOW it OR the resource policy ALLOWS it
  + AND there’s no explicit DENY
* Encryption: encrypted objects in Amazon S3 using encryption keys

**S3 Bucket Policies**

|  |  |
| --- | --- |
| * JSON based policies which contains   + Resources: buckets and objects where policy will apply. In the example it will apply to all objects in bucket, since \* is used   + Effect: Allow / Deny   + Actions: Set of API where policy will impact . In the example it will impact get Objects API   + Principal: The account or user to apply the policy to * Use case example for S3 bucket for policy:   + Grant public access to the bucket   + Force objects to be encrypted at upload   + Grant access to another account (Cross Account) | A screenshot of a computer code  AI-generated content may be incorrect. |

**Example: Public Access - Use Bucket Policy**

A person holding a blue object

AI-generated content may be incorrect.

First, we have to enable public access for bucket like from console. This but this is risky as it will make all objects public

A close-up of a text

AI-generated content may be incorrect.

* These settings were created to prevent company data leaks
* If you know your bucket should never be public, leave these on
* Can be set at the account level

Next, we must add bucket policy we can go to Bucket 🡪 Permission and add policy. We can use policy generators URL 🡪 <https://awspolicygen.s3.amazonaws.com/policygen.html>

A blue and white rectangular buttons

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

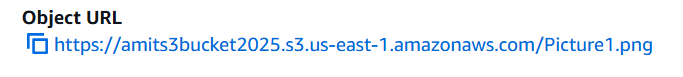
In Arn we have to provide bucket ARN and it has to be followed by star so that policy is applied to all objects in the bucket

arn:aws:s3:::amits3bucket2025/\*

Next we have to generate policy from policy generators and apply in Bucket policy and now we can access the file using public URL

A screenshot of a computer program

AI-generated content may be incorrect.



**Example: User Access to S3 – IAM permissions**

A blue line in a white background

Description automatically generated

**Example: EC2 instance access - Use IAM Roles**

A computer screen shot of a computer

Description automatically generated

**Advanced: Cross-Account Access – Use Bucket Policy**

A blue line in a white background

Description automatically generated

## Amazon S3 – Static Website Hosting

|  |  |
| --- | --- |
| * S3 can host static websites and have them accessible on the Internet * The website URL will be (depending on the region) * http://bucket-name.s3-website-aws-region.amazonaws.com * OR * http://bucket-name.s3-website.aws-region.amazonaws.com * If you get a 403 Forbidden error, make sure the bucket * policy allows public reads! | A screen shot of a computer  AI-generated content may be incorrect. |

Go to bucket settings at the end there will a section for Static website hosting we can enable it and provide a index.html for the code. Once we enable it in the same section we can find the website URL

## Amazon S3 -Versioning

|  |  |
| --- | --- |
| * You can version your files in Amazon S3 * It is enabled at the bucket level * Same key(file name with path) overwrite will change the “version”: 1, 2, 3…. * It is best practice to version your buckets  1. Protect against unintended deletes (ability to restore a version) 2. Easy roll back to previous version  * Notes:  1. Any file that is not versioned prior to enabling versioning will have version “null” 2. Suspending versioning does not delete the previous versions | A diagram of a bucket and a bucket of paper  AI-generated content may be incorrect. |

After we enable versions there is a toggle called show versions, we can turn on that to see the versions. For the files added before enabling versioning it will be null.

A screenshot of a computer

AI-generated content may be incorrect.

But any change we do after enabling versioning we can see the versions and if we delete the newer version, the older version will be restored

A group of black and white text

Description automatically generated

If we remove the version and just delete the file, it will not delete the file but add a delete marker. If we delete the version with delete marker it will restore the file.

A screenshot of a computer

AI-generated content may be incorrect.

## Replication

**Amazon S3 – Replication (CRR & SRR)**

|  |  |
| --- | --- |
| * To enable versioning must enable Versioning in source and destination buckets * Cross-Region Replication (CRR) * Same-Region Replication (SRR) * Buckets can be in different AWS accounts * Copying is asynchronous * Must give proper IAM permissions to S3 * Use cases: * CRR – compliance, lower latency access, replication across accounts * SRR – log aggregation, live replication between production and test accounts | A diagram of a bucket  AI-generated content may be incorrect. |

**Amazon S3 – Replication (Notes)**

* After you enable Replication, only new objects are replicated
* Optionally, you can replicate existing objects using S3 Batch Replication
  + Replicates existing objects and objects that failed replication
* For DELETE operations
  + Can replicate delete markers from source to target (optional setting) i.e. versions get copied
  + Deletions with a version ID are not replicated (to avoid malicious deletes) i.e. when we delete the actual file or its versions delete won’t happen in replica.
* There is no “chaining” of replication
  + If bucket 1 has replication into bucket 2, which has replication into bucket 3
  + Then objects created in bucket 1 are not replicated to bucket 3

Replication is under management section 🡪 replication rules

## S3 Storage Classes

* Amazon S3 Standard - General Purpose
* Amazon S3 Standard-Infrequent Access (IA)
* Amazon S3 One Zone-Infrequent Access
* Amazon S3 Glacier Instant Retrieval
* Amazon S3 Glacier Flexible Retrieval
* Amazon S3 Glacier Deep Archive
* Amazon S3 Intelligent Tiering
* Can move between classes manually or using S3 Lifecycle configurations

Durability:

1. High durability (99.999999999%, 11 9’s) of objects across multiple AZ
2. If you store 10,000,000 objects with Amazon S3, you can on average expect to incur a loss of a single object once every 10,000 years
3. Same for all storage classes

* Availability:

1. Measures how readily available a service is
2. Varies depending on storage class
3. Example: S3 standard has 99.99% availability = not available 53 minutes a year

**S3 Standard – General Purpose**

* 99.99% Availability
* Used for frequently accessed data
* Low latency and high throughput
* Sustain 2 concurrent facility failures
* Use Cases: Big Data analytics, mobile & gaming applications, content distribution…

**S3 Storage Classes – Infrequent Access**

* For data that is less frequently accessed, but requires rapid access when needed
* Lower cost than S3 Standard, cost on retrieval
* Amazon S3 Standard-Infrequent Access (S3 Standard-IA)

1. 99.9% Availability
2. Use cases: Disaster Recovery, backups

* Amazon S3 One Zone-Infrequent Access (S3 One Zone-IA)

1. High durability (99.999999999%) in a single AZ; data lost when AZ is destroyed
2. 99.5% Availability
3. Use Cases: Storing secondary backup copies of on-premises data, or data you can recreate

**Amazon S3 Glacier Storage Classes**

* Low-cost object storage meant for archiving / backup
* Pricing: price for storage + object retrieval cost
* Amazon S3 Glacier Instant Retrieval

1. Millisecond retrieval, great for data accessed once a quarter
2. Minimum storage duration of 90 days

* Amazon S3 Glacier Flexible Retrieval (formerly Amazon S3 Glacier):

1. Expedited (1 to 5 minutes), Standard (3 to 5 hours), Bulk (5 to 12 hours) – free
2. Minimum storage duration of 90 days

* Amazon S3 Glacier Deep Archive – for long term storage:

1. Standard (12 hours), Bulk (48 hours)
2. Minimum storage duration of 180 days

**S3 Intelligent-Tiering**

* Small monthly monitoring and auto-tiering fee
* Moves objects automatically between Access Tiers based on usage
* There are no retrieval charges in S3 Intelligent-Tiering
* Frequent Access tier (automatic): default tier
* Infrequent Access tier (automatic): objects not accessed for 30 days
* Archive Instant Access tier (automatic): objects not accessed for 90 days
* Archive Access tier (optional): configurable from 90 days to 700+ days
* Deep Archive Access tier (optional): config. from 180 days to 700+ days

We can go to files and change the storage tier of the file, going to properties 🡪 storage class of the file.

At bucket level we can do this using lifecycle rules. Bucket we want to change 🡪 management

**S3 Storage Classes Comparison**

[**https://aws.amazon.com/s3/storage-classes/**](https://aws.amazon.com/s3/storage-classes/)

A screenshot of a graph

Description automatically generated

**S3 Storage Classes – Price Comparison Example: us-east-1**

[**https://aws.amazon.com/s3/pricing/**](https://aws.amazon.com/s3/pricing/)

A screenshot of a blue and white table

Description automatically generated

**Questions**

**Top of Form**

**You're getting errors while trying to create a new S3 bucket named "dev". You're using a new AWS Account with no S3 buckets created before. And you double-checked and found that you have the correct IAM permissions to create S3 Buckets. What is a possible cause for this?**

* **Only AWS account root user can create S3 Buckets**
* **S3 bucket names must be globally unique and "dev" is already taken**

**Bottom of Form**

Ans B

**Top of Form**

**You have enabled versioning in your S3 bucket which already contains a lot of files. Which version will the existing files have?**

* **1**
* **0**
* **-1**
* **null**

**Bottom of Form**

Ans null

**Imp Top of Form**

**You have updated an S3 bucket policy to allow IAM users to read/write files in the S3 bucket, but one of the users complain that he can't perform a PutObject API call. What is a possible cause for this?**

* **The S3 bucket policy must be wrong**
* **The user is lacking permissions**
* **The IAM user must have an explicit DENY in the attached IAM Policy**
* **You need to contact AWS Support to lift this limit**

**Bottom of Form**

Even if the user doesn't have permissions, in the question the S3 bucket is explicitly allowing the user and that's enough from a permission's perspective so the answer is

Explicit DENY in an IAM Policy will take precedence over an S3 bucket policy.

**Top of Form**

**You want the content of an S3 bucket to be fully available in different AWS Regions. That will help your team perform data analysis at the lowest latency and cost possible. What S3 feature should you use?**

* **Amazon CloudFront Distributions**
* **S3 Versioning**
* **S3 Static Website Hosting**
* **S3 Replication**

**Bottom of Form**

S3 Replication allows you to replicate data from an S3 bucket to another in the same/different AWS Region.

**Top of Form**

**You have 3 S3 buckets. One source bucket A, and two destination buckets B and C in different AWS Regions. You want to replicate objects from bucket A to both bucket B and C. How would you achieve this?**

* **Configure replication from bucket A to bucket B, then from bucket A to bucket C**
* **Configure replication from bucket A to bucket B, then from bucket B to bucket C**
* **Configure replication from bucket A to bucket C, then from bucket C to bucket B**

**Bottom of Form**

Ans A

**Top of Form**

**Which of the following is NOT a Glacier Deep Archive retrieval mode?**

* **Expedited (1 - 5 minutes)**
* **Standard (12 hours)**
* **Bulk (48 hours)**

**Bottom of Form**

Ans A

**Top of Form**

**Which of the following is NOT a Glacier Flexible retrieval mode?**

* **Instant (10 seconds)**
* **Expedited (1 - 5 minutes)**
* **Standard (3 - 5 hours)**
* **Bulk (5 - 12 hours)**

**Bottom of Form**

Ans A

# Amazon S3 – Advanced

## Moving between Storage Classes

|  |  |
| --- | --- |
| * You can transition objects between storage classes * For infrequently accessed object, move them to Standard IA * For archive objects that you don’t need fast access to, move them to Glacier or Glacier Deep Archive * Moving objects can be automated using a Lifecycle Rules | A diagram of a standard  AI-generated content may be incorrect. |

**Amazon S3 – Lifecycle Rules**

Life cycle rules made of multiple items like

* Transition Actions – configure objects to transition to another storage class
  + Move objects to Standard IA class 60 days after creation
  + Move to Glacier for archiving after 6 months
* Expiration actions – configure objects to expire (delete) after some time
  + Access log files can be set to delete after 365 days
  + Can be used to delete old versions of files (if versioning is enabled)
  + Can be used to delete incomplete multi-Part uploads
* Rules can be created for a certain prefix (example: s3://mybucket/mp3/\*)
* Rules can be created for certain objects Tags (example: Department: Finance)

**Amazon S3 – Lifecycle Rules (Scenario 1)**

Your application on EC2 creates images thumbnails after profile photos are uploaded to Amazon S3. These thumbnails can be easily recreated and only need to be kept for 60 days. The source images should be able to be immediately retrieved for these 60 days, and afterwards, the user can wait up to 6 hours. How would you design this?

* S3 source images can be on Standard, with a lifecycle configuration to transition them to Glacier after 60 days
* S3 thumbnails can be on One-Zone IA, with a lifecycle configuration to expire them (delete them) after 60 days

**Amazon S3 – Lifecycle Rules (Scenario 2)**

A rule in your company states that you should be able to recover your deleted S3 objects immediately for 30 days, although this may happen rarely. After this time, and for up to 365 days, deleted objects should be recoverable within 48 hours.

* Enable S3 Versioning to have object versions, so that “deleted objects” are in fact hidden by a “delete marker” and can be recovered
* Transition the “noncurrent versions” of the object to Standard IA
* Transition afterwards the “noncurrent versions” to Glacier Deep Archive

**Amazon S3 Analytics – Storage Class Analysis**

|  |  |
| --- | --- |
| * Help you decide when to transition objects to the right storage class * Recommendations for Standard and Standard IA   + Does NOT work for One-Zone IA or Glacier * Report is updated daily * 24 to 48 hours to start seeing data analysis   Good first step to put together Lifecycle Rules (or improve them)! | A diagram of a data flow  AI-generated content may be incorrect. |

We can select the rules we want and then we have to add the rule

A white background with black text

AI-generated content may be incorrect.

Transition of current objects Transition of non current version

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

Description automatically generated

Expire current versions Permanently delete noncurrent versions

A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer error

Description automatically generated

Then we can see the summary below

A screenshot of a computer

AI-generated content may be incorrect.

**S3 – Requester Pays**

|  |  |
| --- | --- |
| * In general, bucket owners pay for all Amazon S3 storage and data transfer costs associated with their bucket * With Requester Pays buckets, the requester instead of the bucket owner pays the cost of the request and the data download from the bucket * Helpful when you want to share large datasets with other accounts * The requester must be authenticated in AWS (cannot be anonymous) | A diagram of a bucket  AI-generated content may be incorrect. |

S3 Event Notifications

|  |  |
| --- | --- |
| * S3:ObjectCreated, S3:ObjectRemoved,   S3:ObjectRestore, S3:Replication…   * Object name filtering possible (\*.jpg) * Use case: generate thumbnails of images uploaded to S3 * Can create as many “S3 events” as desired * S3 event notifications typically deliver events in seconds but can sometimes take a minute or longer | A diagram of a company  AI-generated content may be incorrect. |

**S3 Event Notifications – IAM Permissions**

For the Event Notifications to work, we need to have IAM permissions. So, the S3 service is sending data into an SNS topic. To make this possible, we need to attach what's called an SNS resource access policy. This is a IAM policy you attach to the SNS topic, and this will allow the S3 bucket to send messages directly into the SNS topic. Similarly, if we use SQS, we create an SQS resource access policy, which authorizes the S3 service to send data into our SQS queue.

So here we don't use IAM roles for Amazon S3. Instead, we define resource access policies on the SNS topic, on the SQS queue, or on the Lambda function.

A diagram of a computer

AI-generated content may be incorrect.

**S3 Event Notifications with Amazon EventBridge**

A diagram of a company

Description automatically generated

* Advanced filtering options with JSON rules (metadata, object size, name...)
* Multiple Destinations – ex Step Functions, Kinesis Streams / Firehose…
* EventBridge Capabilities – Archive, Replay Events, Reliable delivery

We can enable notification from bucker properties. We can create on the type of events like object creation, update , delete etc.

A screenshot of a computer program

AI-generated content may be incorrect.

**S3 – Baseline Performance**

* Amazon S3 automatically scales to high request rates, latency 100-200 milli seconds to get first byte out of S3
* Your application can achieve at least 3,500 PUT/COPY/POST/DELETE or 5,500 GET/HEAD requests per second per prefix in a bucket.
* There are no limits to the number of prefixes in a bucket.
* Example (object path => prefix): prefix example

1. bucket/folder1/sub1/file => /folder1/sub1/
2. bucket/folder1/sub2/file => /folder1/sub2/
3. bucket/1/file => /1/
4. bucket/2/file => /2/

* If you spread reads across all four prefixes evenly, you can achieve 22,000 requests per second for GET and HEAD

**S3 Performance optimization**

|  |  |
| --- | --- |
| Multi-Part upload:  • recommended for files > 100MB, must use for files > 5GB  • multiplarts help parallelize uploads (speed up transfers)  A diagram of a diagram of a file  AI-generated content may be incorrect. | S3 Transfer Acceleration   * Increase transfer speed by transferring files to an AWS edge location which will forward the data to the S3 bucket in the target region * Compatible with multi-part upload   A diagram of a network  AI-generated content may be incorrect. |

**S3 Performance – get operations**

S3 Byte-Range Fetches – getting files in byte ranges

|  |  |
| --- | --- |
| • Parallelize GETs by requesting specific byte ranges  • Better resilience in case of failures only small parts are retried in case of failures  Can be used to speed up downloads by parallel get  A screenshot of a computer screen  AI-generated content may be incorrect. | Can be used to retrieve only partial data (for example the header of a file)  A screenshot of a computer  AI-generated content may be incorrect. |

**S3 Batch Operations**

|  |  |
| --- | --- |
| • Allows to perform bulk operations on existing S3 objects with a single request, example:   1. Modify object metadata & properties of many S3 objects 2. Copy objects between S3 buckets 3. Encrypt un-encrypted objects 4. Modify ACLs, tags 5. Restore objects from S3 Glacier 6. Invoke Lambda function to perform custom action on each object   • A job consists of a   * list of objects, * the action to perform, * optional parameters   • S3 Batch Operations manages retries, tracks progress, sends completion notifications, generate reports …  • how to generate a list to pass to S3 Batch? Well, we can use S3 Inventory  to get your object list in an S3 bucket. And then you use Athena to query that list and filter the objects you want to pass to the S3 batch service.  So, using S3 Inventory and then Athena, we get a filtered list of objects that we want to include in our batch operation. We pass it to batch operation,  plus, whatever operation we want to do, as well as some parameters. And then the S3 Batch will do its thing and process the objects. | A diagram of a process  AI-generated content may be incorrect. |

**S3 – Storage Lens**

* Understand, analyze, and optimize storage across entire AWS Organization
* Discover anomalies, identify cost efficiencies, and apply data protection best practices across entire AWS Organization (30 days usage & activity metrics)
* Aggregate data for Organization, specific accounts, regions, buckets, or prefixes
* Default dashboard or create your own dashboards
* Can be configured to export metrics daily to an S3 bucket (CSV, Parquet)

**A diagram of a diagram of a diagram

AI-generated content may be incorrect.**

**Storage Lens – Default Dashboard**

* Visualize summarized insights and trends for both free and advanced metrics
* Default dashboard shows Multi-Region and Multi-Account data
* Preconfigured by Amazon S3
* Can’t be deleted, but can be disabled

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

<https://aws.amazon.com/blogs/aws/s3-storage-lens/> <https://aws.amazon.com/blogs/aws/s3-storage-lens/>

**Storage Lens – Metrics**

* Summary Metrics
  + General insights about your S3 storage
  + Storage Bytes, Object Count…
  + Use cases: identify the fastest growing (or not used) buckets and prefixes
* Cost-Optimization Metrics
  + Provide insights to manage and optimize your storage costs
  + NonCurrentVersionStorageBytes, IncompleteMultipartUploadStorageBytes…
  + Use cases: identify buckets with incomplete multipart uploaded older than 7 days, Identify which objects could be transitioned to lower-cost storage class

**Storage Lens – Metrics**

* Data-Protection Metrics
  + Provide insights for data protection features
  + VersioningEnabledBucketCount, MFADeleteEnabledBucketCount, SSEKMSEnabledBucketCount, CrossRegionReplicationRuleCount…
  + Use cases: identify buckets that aren’t following data-protection best practices
* Access-management Metrics
  + Provide insights for S3 Object Ownership
  + ObjectOwnershipBucketOwnerEnforcedBucketCount…
  + Use cases: identify which Object Ownership settings your buckets use
* Event Metrics
  + Provide insights for S3 Event Notifications
  + EventNotificationEnabledBucketCount (identify which buckets have S3 Event Notifications configured)
* Performance Metrics
  + Provide insights for S3 Transfer Acceleration
  + TransferAccelerationEnabledBucketCount (identify which buckets have S3 Transfer Acceleration enabled)
* Activity Metrics
  + Provide insights about how your storage is requested
  + AllRequests, GetRequests, PutRequests, ListRequests, BytesDownloaded…
* Detailed Status Code Metrics
  + Provide insights for HTTP status codes
  + 200OKStatusCount, 403ForbiddenErrorCount, 404NotFoundErrorCount

**Storage Lens– Free vs . Paid**

|  |  |
| --- | --- |
| * Free Metrics   + Automatically available for all customers   + Contains around 28 usage metrics   + Data is available for queries for 14 days * Advanced Metrics and Recommendations   + Additional paid metrics and features   + Advanced Metrics – Activity, Advanced Cost Optimization, Advanced Data Protection, Status Code   + CloudWatch Publishing – Access metrics in CloudWatch without additional charges   + Prefix Aggregation – Collect metrics at the prefix level   + Data is available for queries for 15 months | A screenshot of a web page  AI-generated content may be incorrect. |

## Questions

**Top of Form**

**How can you be notified when there's an object uploaded to your S3 bucket?**

* **S3 Select**
* **S3 Access Logs**
* **S3 Event Notifications**
* **S3 Analytics**

Ans CBottom of Form

**Top of Form**

**You have an S3 bucket that has S3 Versioning enabled. This S3 bucket has a lot of objects, and you would like to remove old object versions to reduce costs. What's the best approach to automate the deletion of these old object versions?**

* **S3 Lifecycle Rules - Transition Actions**
* **S3 Lifecycle Rules - Expiration Actions**
* **S3 Access Logs**

Ans B**Bottom of Form**

**Top of Form**

**How can you automate the transition of S3 objects between their different tiers?**

* **AWS Lambda**
* **CloudWatch Events**
* **S3 Lifecycle Rules**

Ans C**Bottom of Form**

**Top of Form**

**While you're uploading large files to an S3 bucket using Multi-part Upload, there are a lot of unfinished parts stored in the S3 bucket due to network issues. You are not using these unfinished parts and they cost you money. What is the best approach to remove these unfinished parts?**

* **Use AWS Lambda to loop on each old/unfinished part and delete them**
* **Request AWS Support to help you delete old/unfinished parts**
* **Use an S3 Lifecycle Policy to automate old/unfinished parts deletion**

Ans C**Bottom of Form**

**Top of Form**

**You are looking to get recommendations for S3 Lifecycle Rules. How can you analyze the optimal number of days to move objects between different storage tiers?**

* **S3 Inventory**
* **S3 Analytics**
* **S3 Lifecycle Rules Advisor**

Ans B**Bottom of Form**

**Top of Form**

**You are looking to build an index of your files in S3, using Amazon RDS PostgreSQL. To build this index, it is necessary to read the first 250 bytes of each object in S3, which contains some metadata about the content of the file itself. There are over 100,000 files in your S3 bucket, amounting to 50 TB of data. How can you build this index efficiently?**

* **Use the RDS Import feature to load the data from S3 to PostgreSQL, and run a SQL query to build the index**
* **Create an application that will traverse the S3 bucket, read all the files one by one, extract the first 250 bytes, and store that information in RDS**
* **Create an application that will traverse the S3 bucket, issue a Byte Range Fetch for the first 250 bytes, and store that information in RDS**
* **Create an application that will traverse the S3 bucket, use S3 Select to get the first 250 bytes, and store that information in RDS**

Ans C Ans AnsC**Bottom of Form**

**Top of Form**

**You have a large dataset stored on-premises that you want to upload to the S3 bucket. The dataset is divided into 10 GB files. You have good bandwidth but your Internet connection isn't stable. What is the best way to upload this dataset to S3 and ensure that the process is fast and avoid any problems with the Internet connection?**

* **Use Multi-part Upload Only**
* **Use S3 Select & Use S3 Transfer Acceleration**
* **Use S3 Multi-part Upload & S3 Transfer Acceleration**

Ans c

**Top of Form**

**A company is preparing for compliance and regulatory review on its infrastructure on AWS. Currently, they have their files stored on S3 buckets encrypted using S3 Default Encryption, which must be encrypted using KMS as required for compliance and regulatory review. Which S3 feature allows them to encrypt all files in their S3 buckets in the most efficient and cost-effective way?**

* **S3 Access Points**
* **S3 Cross-Region Replication**
* **S3 Batch Operations**
* **S3 Lifecycle Rules**

**Bottom of Form**

Ans C

**Top of Form**

**You have a 25 GB file that you're trying to upload to S3 but you're getting errors. What is a possible solution for this?**

* **The file size limit on S3 is 5 GB**
* **Update your bucket policy to allow the larger file**
* **Use Multi-Part upload when uploading files larger than 5GB**
* **Encrypt the file**

**Bottom of Form**

Ans Multi-Part Upload is recommended as soon as the file is over 100 MB. And mandatory for file greater than 5 GB

**Bottom of Form**

# Amazon S3 – Security

You can encrypt objects in S3 buckets using one of 4 methods

1. **Server-Side Encryption** (SSE)
   1. Server-Side Encryption with **Amazon S3-Managed Keys (SSE-S3)** – Enabled by Default
      1. Encrypts S3 objects using keys handled, managed, and owned by AWS
   2. Server-Side Encryption with **KMS Keys stored in AWS KMS (SSE-KMS)**
      1. Leverage AWS Key Management Service (AWS KMS) to manage encryption keys
   3. Server-Side Encryption with **Customer-Provided Keys (SSE-C)**
      1. When you want to manage your own encryption keys
2. **Client-Side Encryption**

**Amazon S3 Encryption**– SSE-S3

* Encryption using keys handled, managed, and owned by AWS we never have access to keys
* Object is encrypted server-side
* The Encryption type is AES-256
* Must set header "x-amz-server-side-encryption": "AES256"
* Enabled by default for new buckets & new objects

A diagram of a keyword

Description automatically generated

**Amazon S3 Encryption – SSE-KMS**

* Encryption using keys handled and managed by AWS KMS (Key Management Service)
* KMS advantages: user control + audit key usage using CloudTrail. so you can create keys yourself within KMS, and you can edit the key usage using CloudTrail. So anytime someone uses a key in KMS,this is going to be logged in a service that logs everything that happens in AWS called CloudTrail.
* Object is encrypted server side
* Must set header "x-amz-server-side-encryption": "aws:kms"

A screenshot of a computer

Description automatically generated

**SSE-KMS Limitation**

|  |  |
| --- | --- |
| • If you use SSE-KMS, you may be impacted by the KMS limits  • When you upload, it calls the GenerateDataKey KMS API  • When you download, it calls the Decrypt KMS API  • Count towards the KMS quota per second (5500, 10000, 30000 req/s based on region)  • You can request a quota increase using the Service Quotas Console | A diagram of a bucket and a key  AI-generated content may be incorrect. |

**Amazon S3 Encryption – SSE-C**

* Server-Side Encryption using keys fully managed by the customer outside of AWS
* Amazon S3 does NOT store the encryption key you provide
* HTTPS must be used
* Encryption key must provided in HTTP headers, for every HTTP request made
* This can only be done via SDK or CLI and not from consoleA screenshot of a computer

  AI-generated content may be incorrect.

**Amazon S3 Encryption – Client-Side Encryption**

* Use client libraries such as Amazon S3 Client-Side Encryption Library
* Clients must encrypt data themselves before sending to Amazon S3
* Clients must decrypt data themselves when retrieving from Amazon S3
* Customer fully manages the keys and encryption cycle

A diagram of a file transfer

Description automatically generated

**Amazon S3 – Encryption in transit (SSL/TLS)**

* Encryption in flight is also called SSL/TLS
* Amazon S3 exposes two endpoints:

1. HTTP Endpoint – non encrypted
2. HTTPS Endpoint – encryption in flight

* HTTPS is recommended
* HTTPS is mandatory for SSE-C
* Most clients would use the HTTPS endpoint by default

**Amazon S3 – Force Encryption in Transit aws:SecureTransport**

How would you go about forcing encryption in transits? For this, we could use a bucket policy. So you attach a bucket policy to your S3 bucket, and you attach this statement which is saying that you deny any GetObject operation if the condition is "aws:SecureTransport": "false". So SecureTransport is going to be true whenever using HTTPS and false whenever you're not using an encryption, an encrypted connection, and so, therefore, any user trying to use HTTP on your bucket is going to be blocked, but users using HTTPS may be allowed.

A screenshot of a computer

Description automatically generated

**Amazon S3 – Default Encryption vs. Bucket Policies**

* SSE-S3 encryption is automatically applied to new objects stored in S3 bucket
* Optionally, you can “force encryption” using a bucket policy and refuse any API call to PUT an S3 object without encryption headers (SSE-KMS or SSE-C)A computer screen shot of text

  Description automatically generated

Note: Bucket Policies are evaluated before “Default Encryption”

## CORS

* Cross-Origin Resource Sharing (CORS)
* Origin = scheme (protocol) + host (domain) + port
  + example: https://www.example.com (implied port is 443 for HTTPS, 80 for HTTP)
* Web Browser based mechanism to allow requests to other origins while visiting the main origin
* Same origin: http://example.com/app1 & http://example.com/app2
* Different origins: http://www.example.com & http://other.example.com
* The requests won’t be fulfilled unless the other origin allows for the requests, using CORS Headers (example: Access-Control-Allow-Origin)

The webserver on left(example.com) has a html page which asks for resources like image etc from web sever on left(other.com)

A computer network diagram with text

Description automatically generated with medium confidence

**Amazon S3 – CORS**

* If a client makes a cross-origin request on our S3 bucket, we need to enable the correct CORS headers
* You can allow for a specific origin or for \* (all origins)

A screenshot of a computer

Description automatically generated

[

    {

        "AllowedHeaders": [

            "Authorization"

        ],

        "AllowedMethods": [

            "GET"

        ],

        "AllowedOrigins": [

            "<url of first bucket with http://...without slash at the end>"

        ],

        "ExposeHeaders": [],

        "MaxAgeSeconds": 3000

    }

]

## Amazon S3 – MFA Delete

* MFA (Multi-Factor Authentication) – force users to generate a code on a device (usually a mobile phone or hardware) before doing important operations on S3
* MFA will be required to:

1. Permanently delete an object version
2. Suspend Versioning on the bucket

* MFA won’t be required to:

1. Enable Versioning
2. List deleted versions

* To use MFA Delete, Versioning must be enabled on the bucket
* Only the bucket owner (root account) can enable/disable MFA Delete from CLI

*# generate root access keys with a profile*

*# AWS CLI profiles allow you to manage multiple sets of configuration settings and credentials. By default, the AWS CLI uses the settings found in the profile named default. However, you can create and reference additional profiles to use different settings.*

aws configure --profile root-mfa-delete-demo

*# We have to enter the ARN of MFA device and the MFS code generated*

*# enable mfa delete*

aws s3api put-bucket-versioning --bucket mfa-demo-stephane --versioning-configuration Status=Enabled,MFADelete=Enabled --mfa "arn-of-mfa-device" "mfa-code" --profile root-mfa-delete-demo

*# disable mfa delete*

aws s3api put-bucket-versioning --bucket mfa-demo-stephane --versioning-configuration Status=Enabled,MFADelete=Disabled --mfa "arn-of-mfa-device" "mfa-code" --profile root-mfa-delete-demo

*# delete the root credentials in the IAM console!!!*

## S3 Access Logs

|  |  |
| --- | --- |
| * For audit purpose, you may want to log all access to S3 buckets * Any request made to S3, from any account, authorized or denied, will be logged into another S3 bucket * That data can be analyzed using data analysis tools… * The target logging bucket must be in the same AWS region * The log format is at:   <https://docs.aws.amazon.com/AmazonS3/latest/dev/LogFormat.html> | A green and black bucket with a handle  AI-generated content may be incorrect. |

**S3 Access Logs: Warning**

* Do not set your logging bucket to be the monitored bucket
* It will create a logging loop, and your bucket will grow exponentially

A screenshot of a computer

AI-generated content may be incorrect.

## Amazon S3 – Pre-Signed URLs

* Generate pre-signed URLs using the S3 Console, AWS CLI or SDK
* URL Expiration
  + S3 Console – 1 min up to 720 mins (12 hours)
  + AWS CLI – configure expiration with --expires-in parameter in seconds (default 3600 secs, max. 604800 secs ~ 168 hours)
* Users given a pre-signed URL inherit the permissions of the user that generated the URL for GET / PUT
* Examples:
  + Allow only logged-in users to download a premium video from your S3 bucket
  + Allow an ever-changing list of users to download files by generating URLs dynamically
  + Allow temporarily a user to upload a file to a precise location in your S3 BUCKET

## S3 Glacier Vault Lock

So the idea is that you want to lock your Glacier Vault to adapt a WORM model. That means Write Once Read Many. So the idea is that you take an object, you put it into your S3 Glacier Vault, and then you lock it so it cannot be ever modified or deleted. So for this, you need to create a Vault Lock Policy on top of your Glacier and then you lock the policy itself for future edits.

|  |  |
| --- | --- |
| * Adopt a WORM (Write Once Read Many) model * Create a Vault Lock Policy * Lock the policy for future edits (can no longer be changed or deleted) * Helpful for compliance and data retention | A diagram of a box and a circle  AI-generated content may be incorrect. |

There is the same option or something similar at least for the S3 buckets, but it's a bit more complicated.

**S3 Object Lock (versioning must be enabled)**

* Adopt a WORM (Write Once Read Many) model
* Block an object version deletion for a specified amount of time
* **Retention mode - Compliance:**
  + Object versions can't be overwritten or deleted by any user, including the root user
  + Objects retention modes can't be changed, and retention periods can't be shortened
* **Retention mode - Governance:**
  + Most users can't overwrite or delete an object version or alter its lock settings
  + Some users have special permissions to change the retention or delete the object
* **Retention Period:** protect the object for a fixed period, it can be extended
* **Legal Hold:**
  + protect the object indefinitely, independent from retention period
  + can be freely placed and removed using the s3:PutObjectLegalHold IAM permission

## S3 – Access Points

Lets say we have S3 bucker with finance data, and sales data, and we have different users or groups that want to access their data. We could create a very complicated S3 bucket policy and make it grow over time. The more users, the more data you have, the more unmanageable this may become.

Well, we can create what's called S3 access points. So we can, for example, create a finance access point that is going to be connected to the finance data. How is it connected to the finance data?

1. We're going to define an access point policy and this policy looks just like an S3 bucket policy and is going to grant read write access to the finance prefix.
2. Then we can define a sales access point. And, again, this will be connected to the sales data thanks to an access point policy, a different one attached to this access point, which is going to grant read and write access to the sales prefix
3. If I want to have an analytics access point, well, we can create it so that it points to finance and sales, but in read only access.

So you can see here we have pushed the security management from the S3 bucket policy into the access points and each access point will have its own security.

So by using access points, we define different ways to access our S3 bucket. And the result of that is that we have a very simple way to manage security.We have policies attached to each access point and also we have a very simple bucket policy on Amazon S3. Therefore, we can really scale access to our S3 buckets.

So to summarize access points, simplify security managements for S3 buckets, and each access point will have its own DNS name. That's how you connect to the access point.

A diagram of a diagram of a business

AI-generated content may be incorrect.

* Access Points simplify security management for S3 Buckets
* Each Access Point has:
  + its own DNS name (Internet Origin or VPC Origin)
  + an access point policy (similar to bucket policy) – manage security at scale

**S3 – Access Points –VPC Origin**

You can choose it to be connected to the internet as an origin or a VPC for private traffic. And then you attach again an access point policy which is very similar to bucket policy. And this allows you to manage security at scale. Regarding the VPC origin of S3 access points, we can define them to be privately accessible. So that's, for example, an EC2 instance in a VPC access says without going through the internet our S3 bucket through the VPC access point through a VPC origin. So to do so, to get access to this VPC origin, we must create what's called a VPC endpoint to access the access point. So it's something in our VPC that will allow us to connect privately into the access point through our VPC origin. And then the VPC endpoint has a policy, and this policy must allow access to the target buckets and the access points. So the VPC endpoint policy will allow our EC2 instance to connect to both the VPC, the access points on Amazon S3 and the S3 buckets. So in this case, we have VPC endpoint for security. We also have security for the access point policy and security at the S3 bucket level.

|  |  |
| --- | --- |
| We can define the access point to be accessible only from within the VPC  • You must create a VPC Endpoint to access the Access Point (Gateway or Interface Endpoint)  • The VPC Endpoint Policy must allow access to the target bucket and Access Point | A diagram of a computer  AI-generated content may be incorrect. |

**S3 Object Lambda**

So there is another use case for EFS three access points and it's called S3 Object Lambda.

Say we have the cloud and we have an S3 bucket in it.

An E-commerce application maybe owns the data in this S3 bucket and so they're able to access directly the S3 buckets and put and get the original object out of it.

An analytics application may want to only have access to the redacted object. That means that some data has been deleted from the object. So instead of creating a new S3 bucket for this what we can do is that we create an S3 access point on top of a S3 bucket and is connected to a Lambda function. Lambda function allows you to run a bit of code in the cloud very easily. And so this Lambda function is going to redact the object as it is being retrieved. And on top of this Lambda function, we're going to create an S3 object Lambda access point. And, this is how the analytics application is going to access our S3 buckets.

So to summarize the analytics application accesses our S3 Object Lambda access points, which invokes our Lambda function. Our Lambda function is going to retrieve the data from the S3 bucket and runs some code to redact the data. And therefore the analytics application is obtaining a redacted object from the very same S3 bucket as the E-commerce application.

Now, a marketing application may want to have access to an enriched object, and they have a customer loyalty database to enhance the data. So instead of, again, creating a new S3 bucket and creating all the objects with all the enriched data. What we can do is, again, using a Lambda function. So another piece of code. And this one will enrich the data by looking it up from the customer loyalty database. And therefore we can also create an object Lambda access point on top of it. And therefore our marketing application can access this access point, this S3 object Lambda access point to get, again the enriched objects. As you can see, we only need one S3 bucket but we can create access points and object Lambda to modify the data as we wish.

A diagram of a cloud

AI-generated content may be incorrect.

* Use AWS Lambda Functions to change the object before it is retrieved by the caller application
* Only one S3 bucket is needed, on top of which we create S3 Access Point and S3 Object Lambda Access Points.
* Use Cases:
  + Redacting personally identifiable information for analytics or non- production environments.
  + Converting across data formats, such as converting XML to JSON.
  + Resizing and watermarking images on the fly using caller-specific details, such as the user who requested the object.

## Questions

**Top of Form**

**Your client wants to make sure that file encryption is happening in S3, but he wants to fully manage the encryption keys and never store them in AWS. You recommend him to use ............................**

* **SSE-S3**
* **SSE-KMS**
* **SSE-C**
* **Client-Side Encryption**

Bottom of Form

Ans With Client-Side Encryption, you have to do the encryption yourself and you have full control over the encryption keys. With SSE-C, the encryption happens in AWS and you have full control over the encryption keys. Ans C

**Top of Form**

**A company you're working for wants their data stored in S3 to be encrypted. They don't mind the encryption keys stored and managed by AWS, but they want to maintain control over the rotation policy of the encryption keys. You recommend them to use ....................**

* **SSE-S3**
* **SSE-KMS**
* **SSE-C**
* **Client-Side Encryption**

**Bottom of Form**

Ans With SSE-C, the encryption happens in AWS and you have full control over the encryption keys. You could define a rotation policy but this is against the question requirements (encryption keys must be stored and managed by AWS). With SSE-KMS, the encryption happens in AWS, and the encryption keys are managed by AWS but you have full control over rotation policy of the encryption key. Encryption keys stored in AWS. B

**Top of Form**

**Your company does not trust AWS for the encryption process and wants it to happen on the application. You recommend them to use ....................**

* **SSE-S3**
* **SSE-KMS**
* **SSE-C**
* **Client-Side Encryption**

**Bottom of Form**

Ans With Client-Side Encryption, you have to do the encryption yourself and you have full control over the encryption keys. You perform the encryption yourself and send the encrypted data to AWS. AWS does not know your encryption keys and cannot decrypt your data.

**Top of Form**

**You have a website that loads files from an S3 bucket. When you try the URL of the files directly in your Chrome browser it works, but when a website with a different domain tries to load these files it doesn't. What's the problem?**

* **The Bucket policy is wrong**
* **The IAM policy is wrong**
* **CORS is wrong**
* **Encryption is wrong**

**Bottom of Form**

Ans Cross-Origin Resource Sharing (CORS) defines a way for client web applications that are loaded in one domain to interact with resources in a different domain. To learn more about CORS, go here: <https://docs.aws.amazon.com/AmazonS3/latest/dev/cors.html>

**Top of Form**

**An e-commerce company has its customers and orders data stored in an S3 bucket. The company’s CEO wants to generate a report to show the list of customers and the revenue for each customer. Customer data stored in files on the S3 bucket has sensitive information that we don’t want to expose in the report. How do you recommend the report can be created without exposing sensitive information?**

* **Use S3 Object Lambda to change the objects before they are retrieved by the report generator application**
* **Create another S3 bucket. Create a lambda function to process each file, remove the sensitive information, and then move them to the new S3 bucket**
* **Use S3 Object Lock to lock the sensitive information from being fetched by the report generator application**

**Bottom of Form**

Ans A

**Top of Form**

**You suspect that some of your employees try to access files in an S3 bucket that they don't have access to. How can you verify this is indeed the case without them noticing?**

* **Enable S3 Access Logs and analyze them using Athena**
* **Restrict their IAM policies and look at CloudTail logs**
* **Use a bucket policy**

**Bottom of Form**

Ans S3 Access Logs log all the requests made to S3 buckets and Amazon Athena can then be used to run serverless analytics on top of the log files. For B This will be noticed by employees as they won't be able to access the files. For C A bucket policy would prevent access to the user but they would notice it.

**Top of Form**

**You are looking to provide temporary URLs to a growing list of federated users to allow them to perform a file upload on your S3 bucket to a specific location. What should you use?**

* **S3 CORS**
* **S3 Pre-Signed URL**
* **S3 Bucket Policies**

Bottom of Form

Ans S3 Pre-Signed URLs are temporary URLs that you generate to grant time-limited access to some actions in your S3 bucket.

**Top of Form**

**For compliance reasons, your company has a policy mandate that database backups must be retained for 4 years. It shouldn't be possible to erase them. What do you recommend?**

* **Glacier Vaults with Vault Lock Policies**
* **EFS network drives with restrictive Linux permissions**
* **S3 with Bucket Policies**

Ans A**Bottom of Form**

**Top of Form**

**You would like all your files in an S3 bucket to be encrypted by default. What is the optimal way of achieving this?**

* **Use a bucket policy that forces HTTPS connections**
* **Do nothing, Amazon S3 automatically encrypt new objects using Server-Side Encryption with S3-Managed Keys (SSE-S3)**
* **Enable Versioning**

Ans B**Bottom of Form**

**Top of Form**

**You have enabled versioning and want to be extra careful when it comes to deleting files on an S3 bucket. What should you enable to prevent accidental permanent deletions?**

* **Use a bucket policy**
* **Enable MFA Delete**
* **Encrypt the files**
* **Disable versioning**

**Bottom of Form**

Ans MFA Delete forces users to use MFA codes before deleting S3 objects. It's an extra level of security to prevent accidental deletions.

**Top of Form**

**A company has its data and files stored on some S3 buckets. Some of these files need to be kept for a predefined period of time and protected from being overwritten and deletion according to company compliance policy. Which S3 feature helps you in doing this?**

* **S3 Object Lock - Retention Governance Mode**
* **S3 Versioning**
* **S3 Object Lock - Retention Compliance Mode**
* **S3 Glacier Vault Lock**

Ans CBottom of Form

**Top of Form**

**Which of the following S3 Object Lock configuration allows you to prevent an object or its versions from being overwritten or deleted indefinitely and gives you the ability to remove it manually?**

* **Retention Governance Mode**
* **Retention Compliance Mode**
* **Legal Hold**

Ans CBottom of Form

Ans

# CloudFront & Global Accelerator

|  |  |
| --- | --- |
| • Content Delivery Network (CDN)  • Improves read performance, content is cached at the edge  • Improves users experience  • 216 Point of Presence globally (edge locations)  • DDoS protection (because worldwide), integration with Shield, AWS Web Application Firewall | A map of the world with people and a flag  AI-generated content may be incorrect. |

**CloudFront – Origins**

* S3 bucket
  + For distributing files and caching them at the edge
  + Enhanced security with CloudFront Origin Access Control (OAC)
  + OAC is replacing Origin Access Identity (OAI)
  + CloudFront can be used as an ingress (to upload files to S3)
* Custom Origin (HTTP)
  + Application Load Balancer
  + EC2 instance
  + S3 website (must first enable the bucket as a static S3 website)
  + Any HTTP backend you want

**CloudFront at a high level**

A screenshot of a computer

AI-generated content may be incorrect.

## CloudFront – S3 as an Origin

In cloud front we must select out bucket as origin

The we must select how we want Cloudfront to access S3

A screenshot of a computer

AI-generated content may be incorrect.

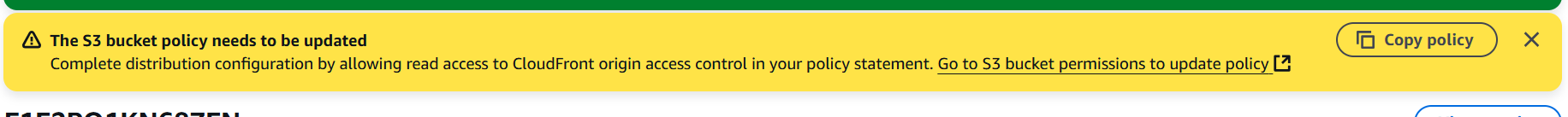
Can skip Web Application Firewall (WAF)

Select default root element

A screenshot of a computer

Description automatically generated

Once the Distribution is created it will tell us to update S3 bucket policy. Copy the policy and update in S3



We can access our Bucket content using distribution domain URL

A close up of a logo

AI-generated content may be incorrect.

After 1st request for any file next requests will be fast

A diagram of a cloud

AI-generated content may be incorrect.

**CloudFront vs S3 Cross Region Replication**

* CloudFront:
  + Global Edge network
  + Files are cached for a TTL (maybe a day)
  + Great for static content that must be available everywhere
* S3 Cross Region Replication:
  + Must be setup for each region you want replication to happen
  + Files are updated in near real-time
  + Read only
  + Great for dynamic content that needs to be available at low-latency in few regions

## CloudFront – ALB or EC2 as an origin

It's possible for CloudFront to access any custom HTTP backend. So this includes as well an EC2 instance or an application load balancer. The EC2 instance must be public otherwise the edge location will not be able to access the EC2 instance. Same foe ALB instance.

If we are using ALB then backing EC2 can be private

A screenshot of a computer

AI-generated content may be incorrect.

**CloudFront Geo Restriction**

* You can restrict who can access your distribution
  + Allowlist: Allow your users to access your content only if they're in one of the countries on a list of approved countries.
  + Blocklist: Prevent your users from accessing your content if they're in one of the countries on a list of banned countries.
* The “country” is determined using a 3rd party Geo-IP database
* Use case: Copyright Laws to control access to content

**CloudFront - Pricing**

* CloudFront Edge locations are all around the world
* The cost of data out per edge location varies

A screenshot of a table

AI-generated content may be incorrect.

**CloudFront – Price Classes**

* You can reduce the number of edge locations for cost reduction
* Three price classes:
  + Price Class All: all regions – best performance
  + Price Class 200: most regions, but excludes the most expensive regions
  + Price Class 100: only the least expensive regions

A screenshot of a survey

AI-generated content may be incorrect.

**CloudFront - Price Class**

A map of the world

AI-generated content may be incorrect.

**CloudFront – Cache Invalidations**

|  |  |
| --- | --- |
| • In case you update the back-end origin, CloudFront doesn’t know about it and will only get the refreshed content after the TTL has expired  • However, you can force an entire or partial cache refresh (thus bypassing the TTL) by performing a CloudFront Invalidation  • You can invalidate all files (\*) or a special path (/images/\*) | A diagram of a cloud front  AI-generated content may be incorrect. |

## Global accelerator

User case

|  |  |
| --- | --- |
| * You have deployed an application and have global users who want to access it directly. * They go over the public internet, which can add a lot of latency due to many hops * We wish to go as fast as possible through AWS network to minimize latency | A diagram of a computer network  AI-generated content may be incorrect. |

**Unicast IP vs Anycast IP**

|  |  |
| --- | --- |
| • Unicast IP: one server holds one IP address  • Anycast IP: all servers hold the same IP address and the client is routed to the nearest one | A computer network diagram with a computer and a client  AI-generated content may be incorrect. |

**AWS Global Accelerator**

It uses the concept of anycast IP

|  |  |
| --- | --- |
| • Leverage the AWS internal network to route to your application  • 2 Anycast IP are created for your application  • The Anycast IP send traffic directly to Edge Locations  • The Edge locations send the traffic to your application through private AWS network | A diagram of a network  AI-generated content may be incorrect. |

So the idea with this is that, we're going to use an Anycast IP and there's actually going to be two of those that are going to be created for your application and they're global. And the Anycast IP will send the traffic directly to the closest edge location of your users. And this is the power of Anycast IP. So the edge location will then send the traffic to you application, (mumbles) For example, through the private AWS network which is much more stable, has less latency and so on.

* Works with Elastic IP, EC2 instances, ALB, NLB, public or private
* Consistent Performance
  1. Intelligent routing to lowest latency and fast regional failover
  2. No issue with client cache (because the IP doesn’t change)
  3. Internal AWS network
* Health Checks
  1. Global Accelerator performs a health check of your applications
  2. Helps make your application global (failover less than 1 minute for unhealthy)
  3. Great for disaster recovery (thanks to the health checks)
* Security
  1. only 2 external IP need to be whitelisted
  2. DDoS protection thanks to AWS Shield

**AWS Global Accelerator vs CloudFront**

* They both use the AWS global network and its edge locations around the world
* Both services integrate with AWS Shield for DDoS protection.
* CloudFront
  1. Improves performance for both cacheable content (such as images and videos)
  2. Dynamic content (such as API acceleration and dynamic site delivery)
  3. Content is served at the edge
* Global Accelerator
* Improves performance for a wide range of applications over TCP or UDP
* Proxying packets at the edge to applications running in one or more AWS Regions.
* Good fit for non-HTTP use cases, such as gaming (UDP), IoT (MQTT), or Voice over IP
* Good for HTTP use cases that require static IP addresses
* Good for HTTP use cases that required deterministic, fast regional failover

## Questions

**Top of Form**

**You have a CloudFront Distribution that serves your website hosted on a fleet of EC2 instances behind an Application Load Balancer. All your clients are from the United States, but you found that some malicious requests are coming from other countries. What should you do to only allow users from the US and block other countries?**

* **Use CloudFront Geo Restriction**
* **Use Origin Access Control**
* **Set up a security group and attach it to your CloudFront Distribution**
* **Use a Route 53 Latency record and attach it to CloudFront**

**Bottom of Form**

Ans A

**Top of Form**

**You have a static website hosted on an S3 bucket. You have created a CloudFront Distribution that points to your S3 bucket to better serve your requests and improve performance. After a while, you noticed that users can still access your website directly from the S3 bucket. You want to enforce users to access the website only through CloudFront. How would you achieve that?**

* **Send an email to your clients and tell them to not use the S3 endpoint**
* **Configure your CloudFront Distribution and create an Origin Access Control (OAC), then update your S3 Bucket Policy to only accept requests from your CloudFront Distribution.**
* **Use S3 Access Points to redirect clients to CloudFront**

**Bottom of Form**

Ans B

**What does this S3 bucket policy do?**

**{**

**"Version": "2012-10-17",**

**"Id": "Mystery policy",**

**"Statement": [{**

**"Sid": "What could it be?",**

**"Effect": "Allow",**

**"Principal": {**

**"Service": "cloudfront.amazonaws.com"**

**},**

**"Action": "s3:GetObject",**

**"Resource": "arn:aws:s3:::examplebucket/\*",  
        "Condition": {  
            "StringEquals": {  
                "AWS:SourceArn": "arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE"  
            }  
        }**

**}]**

**}**

* **Forces GetObject request to be encrypted if coming from CloudFront**
* **Only allows the S3 bucket content to be accessed from your CloudFront Distribution**
* **Only allows GetObject type of request on the S3 bucket from anybody**

Ans B

**Top of Form**

**A WordPress website is hosted in a set of EC2 instances in an EC2 Auto Scaling Group and fronted by a CloudFront Distribution which is configured to cache the content for 3 days. You have released a new version of the website and want to release it immediately to production without waiting for 3 days for the cached content to be expired. What is the easiest and most efficient way to solve this?**

* **Open a support ticket with AWS Support to remove the CloudFront Cache**
* **CloudFront Cache Invalidation**
* **EC2 Cache Invalidation**

**Bottom of Form**

Ans B

**Top of Form**

**A company is deploying a media-sharing website to AWS. They are going to use CloudFront to deliver the content with low latency to their customers where they are located in both US and Europe only. After a while there a huge costs for CloudFront. Which CloudFront feature allows you to decrease costs by targeting only US and Europe?**

* **CloudFront Cache Invalidation**
* **CloudFront Price Classes**
* **CloudFront Cache Behavior**
* **Origin Access Control**

**Bottom of Form**

Ans B

**Top of Form**

**A company is migrating a web application to AWS Cloud and they are going to use a set of EC2 instances in an EC2 Auto Scaling Group. The web application is made of multiple components so they will need a host-based routing feature to route to specific web application components. This web application is used by many customers and therefore the web application must have a static IP address so it can be whitelisted by the customers’ firewalls. As the customers are distributed around the world, the web application must also provide low latency to all customers. Which AWS service can help you to assign a static IP address and provide low latency across the globe?**

* **AWS Global Accelerator + Application Load Balancer**
* **Amazon CloudFront**
* **Network Load Balancer**
* **Application Load Balancer**

**Bottom of Form**

Ans A

# AWS Storage Extras

AWS Snow Family

* Highly secure, portable devices to collect and process data at the edge, and migrate data into and out of AWS
* Helps migrate petabytes of data

Two kinds of Snowball edge devices.

A computer hard drive with a box

AI-generated content may be incorrect.

## Data Migrations with AWS Snow Family

|  |  |
| --- | --- |
| Challenges/ use case for snowball:   * Limited connectivity * Limited bandwidth * High network cost * Shared bandwidth (can’t maximize the line) * Connection stability   So, whenever you have these kind of challenges, or if it takes you, for example, over a week to transfer data over the network, the recommendation is to use a Snowball device. | Usual time taken for data transfer for specific bandwidth A table with text and numbers  AI-generated content may be incorrect. |

If you use Snowball, you're going to receive a physical Snowball device. So, it's a physical machine, and then you load it with the data you need, and then finally you ship it back to AWS. And then you will be having an export process within AWS to take your data from the Snowball onto, for example, an Amazon S3 bucket

AWS Snow Family: offline devices to perform data migrations If it takes more than a week to transfer over the network, use Snowball devices!

* Direct upload to S3:

A computer screen shot of a computer

AI-generated content may be incorrect.

* With Snow Family:

A green square with white text

AI-generated content may be incorrect.

**Snow Family – Usage Process**

1. Request Snowball devices from the AWS console for delivery

2. Install the snowball client / AWS OpsHub on your servers

3. Connect the snowball to your servers and copy files using the client

4. Ship back the device when you’re done (goes to the right AWS facility)

5. Data will be loaded into an S3 bucket

6. Snowball is completely wiped

## What is Edge Computing?

This is another use case

* Process data while it’s being created on an edge location
  + Example A truck on the road, a ship on the sea, a mining station underground. These locations may have limited internet and no access to computing power

A black and white logo

AI-generated content may be incorrect.

• We setup a Snowball Edge / Snowcone device to do edge computing

• Snowcone: 2 CPUs, 4 GB of memory, wired or wireless access

• Snowball Edge Compute Optimized (dedicated for that use case) & Storage Optimized

• Run EC2 Instances or Lambda functions at the edge

• Use cases: preprocess data, machine learning, transcoding media

**Solution Architecture: Snowball into Glacier**

• Snowball cannot import to Glacier directly

• You must use Amazon S3 first, in combination with an S3 lifecycle policy

A green and white logo with black text

AI-generated content may be incorrect.

## Amazon FSx – Overview

Allow to launch 3rd party high-performance file systems on AWS as Fully managed service

A green square with white text

AI-generated content may be incorrect.

**Amazon FSx for Windows (File Server)**

* FSx for Windows is a fully managed Windows file system share drive
* Supports SMB protocol & Windows NTFS
* Microsoft Active Directory integration, ACLs, user quotas
* Can be mounted on Linux EC2 instances
* Supports Microsoft's Distributed File System (DFS) Namespaces (group files across multiple FS) - if you have an existing Windows File Server somewhere, for example, on premises, then you can use the Microsoft Distributed File System, DFS feature, to group your file systems together and therefore join your FSx for Windows File Server to your on-premises Windows File Server
* Scale up to 10s of GB/s, millions of IOPS, 100s PB of data
* Storage Options:
  + SSD – latency sensitive workloads (databases, media processing, data analytics, …)
  + HDD – broad spectrum of workloads (home directory, CMS, …)
* Can be accessed from your on-premises infrastructure (VPN or Direct Connect)
* Can be configured to be multi-AZ (high availability)
* Data is backed up daily to S3

**Amazon FSx for Lustre**

* Lustre is a type of parallel distributed file system, for large-scale computing
* The name Lustre is derived from “Linux” and “cluster
* Used for Machine Learning, High Performance Computing (HPC)
  + Video Processing, Financial Modeling, Electronic Design Automation
* Massive scales up to 100 GB/s, millions of IOPS, sub-milliseconds latencies
* Storage Options:
  + SSD – low-latency, IOPS intensive workloads, small & random file operations
  + HDD – throughput-intensive workloads, large & sequential file operations
* Seamless integration with S3
  + Can “read S3” as a file system (through FSx)
  + Can write the output of the computations back to S3 (through FSx)
* Can be used from on-premises servers (VPN or Direct Connect)

**FSx Lustre - File System Deployment Options**

|  |  |
| --- | --- |
| * Scratch File System   + Temporary storage   + Data is not replicated (doesn’t persist if file server fails)   + High burst (6x faster, 200MBps per TiB)   + Usage: short-term processing, optimize costs * Persistent File System   + Long-term storage   + Data is replicated within same AZ   + Replace failed files within minutes   + Usage: long-term processing, sensitive data | A screenshot of a computer  AI-generated content may be incorrect. |

**Amazon FSx for NetApp ONTAP**

|  |  |
| --- | --- |
| * Managed NetApp ONTAP on AWS * File System compatible with NFS, SMB, iSCSI protocol * Move workloads running on ONTAP or NAS to AWS * Very broad compatibility works with:   + Linux   + Windows   + MacOS   + VMware Cloud on AWS   + Amazon Workspaces & AppStream 2.0   + Amazon EC2, ECS and EKS * Storage shrinks or grows automatically * Snapshots, replication, low-cost, compression and data de-duplication * Point-in-time instantaneous cloning (helpful for testing new workloads) | A diagram of a company  AI-generated content may be incorrect. |

**Amazon FSx for Open ZFS**

|  |  |
| --- | --- |
| * Managed Open ZFS file system on AWS * File System compatible with NFS (v3, v4, v4.1, v4.2) * Move workloads running on ZFS to AWS * Works with:   + Linux   + Windows   + MacOS   + VMware Cloud on AWS   + Amazon Workspaces & AppStream 2.0   + Amazon EC2, ECS and EKS * Up to 1,000,000 IOPS with < 0.5ms latency * Snapshots, compression and low-cost * Point-in-time instantaneous cloning (helpful for testing new workloads) | A diagram of a company  AI-generated content may be incorrect. |

## Hybrid Cloud for Storage

* AWS is pushing for ”hybrid cloud”
  + Part of your infrastructure is in the clouds.
  + Part of your infrastructure is on-premises
* This can be due to
  + Long cloud migrations
  + Security requirements
  + Compliance requirements
  + IT strategy

Example: say some of our data is present on S3, which is a proprietary storage technology (unlike EFS / NFS), so how do you expose the S3 data on-premises? The bridge between premise data and AWS is AWS Storage Gateway!

**AWS Storage Cloud Native Options**

A close-up of a logo

AI-generated content may be incorrect.

**AWS Storage Gateway**

It is the bridge between on-premises data and cloud data

* Use cases:
  + disaster recovery
  + backup & restore
  + tiered storage
  + on-premises cache & low-latency files access e.g.  have the majority of your data stored on AWS and use the storage gateway as an on-premises cache for low-latency file access.
* Types of Storage Gateway:
  + S3 File Gateway
  + FSx File Gateway
  + Volume Gateway
  + Tape Gateway

A green square with white outline and black arrows

AI-generated content may be incorrect.

**Amazon S3 File Gateway**

And we want to connect this S3 bucket to an on-premises application server, but we want to use a standard network file system. So, for this we're going to create an S3 File Gateway which is going to allow our application server to use the NFS or the SMB protocol. And by using these protocols, behind the scenes the S3 File Gateway is going to translate those requests into HTTPS requests for your Amazon S3 buckets. So, from an application server perspective, it looks like it's accessing a normal file share. But in fact, behind the scenes it is using an Amazon S3 bucket.

Then if you wanted to achieve some of these objects, you could create a lifecycle policy for your S3 bucket to transition objects after a while to S3 Glacier in order to have them archived.

A drawing of a bucket and text

AI-generated content may be incorrect.Glacier not supported

* All S3 buckets configured with gate way are accessible using the NFS and SMB protocol
* Most recently used data is cached in the file gateway
* Supports S3 Standard, S3 Standard IA, S3 One Zone A, S3 Intelligent Tiering
* Transition to S3 Glacier using a Lifecycle Policy
* Bucket access using IAM roles for each File Gateway
* If you use SMB Protocol(native to windows file system) it has integration with Active Directory (AD) for user authentication. So that means that your users can be authenticated with your S3 File Gateway before accessing it

**Amazon FSx File Gateway**

Discontinued by AWS

If you are using Amazon FSx for Windows File Server, you don't need anything special. This is already accessible for your on-premises system. So why would you go ahead with the trouble of creating an Amazon FSx File Gateway? Well, the idea is that if you do create the gateway, you're going to get a local cache for the frequently access data. So that means that some very important files will be cached locally on your corporate data center and you're going to have low latency access to them.

* Native access to Amazon FSx for Windows File Server
* Local cache for frequently accessed data
* Windows native compatibility (SMB, NTFS, Active Directory...)
* Useful for group file shares and home directoriesA diagram of a diagram

  AI-generated content may be incorrect.

**Volume Gateway**

* Block storage using iSCSI protocol backed by S3
* Backed by EBS snapshots which can help restore on-premises volumes!
* It’s of 2 type
  + Cached volumes: low latency access to most recent data
  + Stored volumes: entire dataset is on premise, scheduled backups to S3

A diagram of a network connection

AI-generated content may be incorrect.

**Tape Gateway**

* Some companies have backup processes using physical tapes (!)
* With Tape Gateway, companies use the same processes but, in the cloud
* Virtual Tape Library (VTL) backed by Amazon S3 and Glacier
* Back up data using existing tape-based processes (and iSCSI interface)
* Works with leading backup software vendors

The corporate data center has a backup server, which is tape-based. The Tape Gateway will do the interface into the cloud by storing the tapes into Amazon S3 or in Amazon Glacier.

A diagram of a tape gateway

AI-generated content may be incorrect.

**Storage Gateway – Hardware appliance**

The gateway must be installed on your corporate data center, it has to run within your corporate data center. But sometimes you do not have virtual servers to run this additional gateway. So, an option for you is to leverage hardware from AWS. So, it's called Storage Gateway Hardware Appliance. If you don't have virtualization on-premises, you can use a Storage Gateway Hardware Appliance.

You can order it literally on amazon.com. And then once you install this hardware appliance for this mini server into your infrastructure, then you can set it up as a File Gateway, a Volume Gateway or a Tape Gateway. And this is really something physical you have to install and will have enough CPU, memory, network and SSD cache resources to function correctly.

|  |  |
| --- | --- |
| * Using Storage Gateway means you need on-premises virtualization * Otherwise, you can use a Storage Gateway Hardware Appliance * You can buy it on amazon.com * Works with File Gateway, Volume Gateway, Tape Gateway * Has the required CPU, memory, network, SSD cache resources * Helpful for daily NFS backups in small data centers | A computer server with text on it  AI-generated content may be incorrect. |

So this is very helpful, for example, for daily NFS backups in small data centers where you don't have virtualization available.

**AWS Storage Gateway summary**

We have on-premises where we deploy a storage gateway VM or a hardware appliance, then we have the Storage Gateway service, and then we have the cloud of AWS. If we want to have a file gateway with a local cache, this is the use case where we have a user group file share and we want to access it over the NFS or the SMB protocol.

Option number one, we connect this into directly an S3 file gateway. Therefore, your data is going to be backed by Amazon S3. And that includes many storage tiers except Glacier and Glacier Deep Archive. But we can create a lifecycle policy to send this into any storage class Amazon S3, including S3 Glacier. If we were to instead use an FSx file gateway then we would send the data into Amazon FSx for Windows File Server, which is automatically backed up to Amazon S3, once in a while.

Now the other use case for a volume gateway is to have application servers mount volumes over the iSCSI protocol. What we do is that this volume gateway is going to be linked through Storage Gateway to Amazon S3. This is where the data of your volume is going to be. And then Amazon S3, this data can be transformed into AWS EBS volumes for really being restored on AWS.

Next, we have your backup applications connecting over the iSCSI VTL protocol to a tape gateway. And the Tape gateway is connected to Amazon S3 as a tape library. And then we can transition these tapes into the Glacier and Glacier Deep Archive tier to create an archive of your tapes.

A computer screen shot of a diagram

AI-generated content may be incorrect.

## AWS Transfer Family

You want to send data in and out of Amazon S3 or EFS, but you don't want to use the S3 APIs. You don't want to use the EFS network file system. You just want to use the FTP protocol. In this case, you need to use the transfer family service from AWS.

* Supported Protocols
  + AWS Transfer for FTP (File Transfer Protocol (FTP))
  + AWS Transfer for FTPS (File Transfer Protocol over SSL (FTPS))
  + AWS Transfer for SFTP (Secure File Transfer Protocol (SFTP))

using the FTP protocol, you can upload to S3 or EFS

* Managed infrastructure, Scalable, Reliable, Highly Available (multi-AZ)
* Pay per provisioned endpoint per hour + data transfers in GB
* Store and manage users’ credentials within the service
* Integrate with existing authentication systems (Microsoft Active Directory,
* LDAP, Okta, Amazon Cognito, custom)
* Usage: sharing files, public datasets, CRM, ERP, …

The transfer family has three flavors, and the users can access directly using the end points of the FTP or optional, you can use a DNS called route 53 to provide your own host name into the FTP service. And then the FTP service thus so the transfer for FTP service will have an IAM role that will be assumed to send or read through the files from Amazon S3 or Amazon EFS. And this is done transparently. You don't have to set many things up. And finally, if you wanted to secure the transfer family services then you could authenticate your users using an external authentication system such as Active Directory and LDAP.

A diagram of a computer

AI-generated content may be incorrect.

## AWS Data Sync

* Move large amounts of data to and from
  + On-premises / other cloud to AWS (NFS, SMB, HDFS, S3 API…) – needs agent
  + AWS to AWS (different storage services) – no agent needed
* Can synchronize to:
  + Amazon S3 (any storage classes – including Glacier)
  + Amazon EFS
  + Amazon FSx (Windows, Lustre, NetApp, Open ZFS...)
* Replication tasks are not continuous and wcan be scheduled hourly, daily, and weekly.
* File permissions and metadata are preserved (NFS POSIX, SMB…)
* One agent task can use 10 Gbps, can setup a bandwidth limit

**AWS Data Sync NFS / SMB to AWS (S3, EFS, FSx…)**

You have your on-premises and then your AWS region where DataSync is running. So here is your NFS or SMB server. And what you have to do is to install on-premises the AWS DataSync agent, and you will tell it to connect to your NFS or SMB server. And then the DataSync agent will establish a connection and connect in an encrypted fashion into the DataSync service. From there, you can tell it to go wherever you want. That could be any storage class for your Amazon S3 buckets, or it could be AWS, EFS, or it could be Amazon FSx. And the synchronization can happen one way from on-premises to AWS, but you can also synchronize from AWS back into on-premises. This is why it's called DataSync.

A screenshot of a computer

AI-generated content may be incorrect.

Now, sometimes at the exam we will tell you that we want to use DataSync, but we don't have the network capacity to do so. Therefore, what you must think about is to use the AWS Snowcone device specifically because the Snowcone device comes with the DataSync agent pre-install on it. So, you can run Snowcone on-premises, then it will pull your data, run the DataSync agents, then it will be shipped back into your AWS region and then synchronize your data to the storage resources of AWS.

**AWS Data Sync Transfer between AWS storage services**

But you can also use DataSync to just synchronize between different AWS storage services. For example, do you want to synchronize between Amazon S3, Amazon EFS, or Amazon FSx, back into Amazon S3, Amazon EFS, or Amazon FSx. And for this, again, we will use the AWS DataSync service, and it will copy the data of course, but also the metadata will be kept between the different AWS storage services, which is very important. And again, something that can come up in the exam. So, to remind you, DataSync can pretty much synchronize between anything, but it is not continuous. It is a scheduled task that can happen hourly, daily, weekly, and also it will preserve metadata and your file permissions. And finally, you need to run the DataSync agents if you are connecting to an NFS or SMB server.

A diagram of a cloud computing

AI-generated content may be incorrect.

## Storage Comparison

* **S3**: Object Storage
* **S3 Glacier**: Object Archival
* **EBS volumes**: Network storage for one EC2 instance at a time
* **Instance Storage**: Physical storage for your EC2 instance (high IOPS), super high performance
* **EFS**: Network File System for Linux instances, POSIX filesystem, can be mounted across multiple AZ
* **FSx for Windows**: Network File System for Windows servers
* **FSx for Lustre**: High Performance Computing Linux file system
* **FSx for NetApp ONTAP**: super High Operating System Compatibility for a network file system
* **FSx for OpenZFS**: If you want to get a managed ZFS file system
* **Storage Gateway**: S3 & FSx File Gateway, Volume Gateway (cache & stored), Tape Gateway
* **Transfer Family**: FTP, FTPS, SFTP interface on top of Amazon S3 or Amazon EFS
* **DataSync**: Schedule data sync from on-premises to AWS, or AWS to AWS
* **Snowcone / Snowball / Snowmobile**: to move large amount of data to the cloud, physically
* **Database**: for specific workloads, usually with indexing and querying

## Questions

**Top of Form**

**You need to move hundreds of Terabytes into Amazon S3, then process the data using a fleet of EC2 instances. You have a 1 Gbit/s broadband. You would like to move the data faster and possibly processing it while in transit. What do you recommend?**

* **Use your network**
* **Use Snowcone**
* **Use AWS Data Migration**
* **Use Snowball Edge**

**Bottom of Form**

Ans Snowball Edge is the right answer as it comes with computing capabilities and allows you to pre-process the data while it's being moved into Snowball.

**Top of Form**

**You want to expose virtually infinite storage for your tape backups. You want to keep the same software you're using and want an iSCSI compatible interface. What do you use?**

* **AWS Snowball**
* **AWS Storage Gateway - Tape Gateway**
* **AWS Storage Gateway - Volume Gateway**
* **AWS Storage Gateway - S3 File Gateway**

**Bottom of Form**

Ans B

**Top of Form**

**Your EC2 Windows Servers need to share some data by having a Network File System mounted on them which respects the Windows security mechanisms and has integration with Microsoft Active Directory. What do you recommend?**

* **Amazon FSx for Windows (File Server)**
* **Amazon EFS**
* **Amazon FSx for Lustre**
* **S3 File Gateway**

Ans ABottom of Form

B Linux

C This is for HPC and Linux.

**Top of Form**

**You have hundreds of Terabytes that you want to migrate to AWS S3 as soon as possible. You tried to use your network bandwidth and it will take around 3 weeks to complete the upload process. What is the recommended approach to using in this situation?**

* **AWS Storage Gateway - Volume Gateway**
* **S3 Multi-part Upload**
* **AWS Snowball Edge**
* **AWS Data Migration Service**

**Bottom of Form**

Anms C

**Top of Form**

**You have a large dataset stored in S3 that you want to access from on-premises servers using the NFS or SMB protocol. Also, you want to authenticate access to these files through on-premises Microsoft AD. What would you use?**

* **AWS Storage Gateway - Volume Gateway**
* **AWS Storage Gateway - S3 File Gateway**
* **AWS Storage Gateway - Tape Gateway**
* **AWS Data Migration Service**

**Bottom of Form**

Ans B

**Top of Form**

**You are planning to migrate your company's infrastructure from on-premises to AWS Cloud. You have an on-premises Microsoft Windows File Server that you want to migrate. What is the most suitable AWS service you can use?**

* **Amazon FSx for Windows (File Server)**
* **AWS Storage Gateway - S3 File Gateway**
* **AWS Managed Microsoft AD**

**Bottom of Form**

Ans A

**Top of Form**

**You would like to have a distributed POSIX compliant file system that will allow you to maximize the IOPS in order to perform some High-Performance Computing (HPC) and genomics computational research. This file system has to easily scale to millions of IOPS. What do you recommend?**

* **EFS with Max. IO enabled**
* **Amazon FSx for Lustre**
* **Amazon S3 mounted on the EC2 instances**
* **EC2 Instance Store**

**Bottom of Form**

A This is not meant for HPC.

B

**Top of Form**

**Which deployment option in the FSx file system provides you with long-term storage that's replicated within AZ?**

* **Scratch File System**
* **Persistent File System**

**Bottom of Form**

Ans B Provides long-term storage where data is replicated within the same AZ. Failed files were replaced within minutes.

**Top of Form**

**Which of the following protocols is NOT supported by AWS Transfer Family?**

* **File Transfer Protocol (FTP)**
* **File Transfer Protocol over SSL (FTPS)**
* **Transport Layer Security (TLS)**
* **Secure File Transfer Protocol (SFTP)**

**Bottom of Form**

Ans C AWS Transfer Family is a managed service for file transfers into and out of S3 or EFS using the FTP protocol, thus TLS is not supported.

**Top of Form**

**A company uses a lot of files and data which is stored in an FSx for Windows File Server storage on AWS. Those files are currently used by the resources hosted on AWS. There’s a requirement for those files to be accessed on-premises with low latency. Which AWS service can help you achieve this?**

* **S3 File Gateway**
* **FSx for Windows File Server On-Premises**
* **FSx File Gateway**
* **Volume Gateway**

**Bottom of Form**

Ans C

**Top of Form**

**A Solutions Architect is working on planning the migration of a startup company from on-premises to AWS. Currently, their infrastructure consists of many servers and 30 TB of data hosted on a shared NFS storage. He has decided to use Amazon S3 to host the data. Which AWS service can efficiently migrate the data from on-premises to S3?**

* **AWS Storage Tape Gateway**
* **Amazon EBS**
* **AWS Transfer Family**
* **AWS DataSync**

**Bottom of Form**

Ans D

**Top of Form**

**Which AWS service is best suited to migrate a large amount of data from an S3 bucket to an EFS file system?**

* **AWS Snowball**
* **AWS DataSync**
* **AWS Transfer Family**
* **AWS Backup**

Bottom of Form

Ans B

**Top of Form**

**A Machine Learning company is working on a set of datasets that are hosted on S3 buckets. The company decided to release those datasets to the public to be useful for others in their research, but they don’t want to configure the S3 bucket to be public. And those datasets should be exposed over the FTP protocol. What can they do to do the requirement efficiently and with the least effort?**

* **Use AWS Transfer Family**
* **Create an EC2 instance with an FTP server installed then copy the data from S3 to the EC2 instance**
* **Use AWS Storage Gateway**
* **Copy the data from S3 to an EFS file system, then expose them over the FTP protocol**

**Bottom of Form**

Ans A

**Top of Form**

**Amazon FSx for NetApp ONTAP is compatible with the following protocols, EXCEPT ………………**

* **NFS**
* **SMB**
* **FTP**
* **iSCSI**

**Bottom of Form**

Ans C

**Top of Form**

**Which AWS service is best suited when migrating from an on-premises ZFS file system to AWS?**

* **Amazon FSx for OpenZFS**
* **Amazon FSx for NetApp ONTAP**
* **Amazon FSx for Windows File Server**
* **Amazon FSx for Luster**

Bottom of Form

Ans A

**Top of Form**

**A company is running Amazon S3 File Gateway to host their data on S3 buckets and is able to mount them on-premises using SMB. The data currently is hosted on S3 Standard storage class and there is a requirement to reduce the costs for S3. So, they have decided to migrate some of those data to S3 Glacier. What is the most efficient way they can use to move the data to S3 Glacier automatically?**

* **Create a Lambda function to migrate data to S3 Glacier and periodically trigger it every day using Amazon EventBridge**
* **Use S3 Batch Operations to loop through S3 files and move them to S3 Glacier every day**
* **Use S3 Lifecycle Policy**
* **Use AWS DataSync to replicate data to S3 Glacier every day**
* **Configure S3 File Gateway to send the data to S3 Glacier directly**

**Bottom of Form**

Ans C

**Top of Form**

**You have on-premises sensitive files and documents that you want to regularly synchronize to AWS to keep another copy. Which AWS service can help you with that?**

* **AWS Database Migration Service**
* **Amazon EFS**
* **AWS DataSync**

Bottom of Form

Ans C AWS DataSync is an online data transfer service that simplifies, automates, and accelerates moving data between on-premises storage systems and AWS Storage services, as well as between AWS Storage services.

**Top of Form**

**AWS DataSync supports the following locations, EXCEPT ....................**

* **Amazon S3**
* **Amazon EBS**
* **Amazon EFS**
* **Amazon FSx for Windows File Server**

**Bottom of Form**

Ans B

# AWS Integration & Messaging

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

A blue squares with white text

AI-generated content may be incorrect.

Section Introduction

* Synchronous between applications can be problematic if there are sudden spikes of traffic
* What if you need to suddenly encode 1000 videos but usually it’s 10?
* In that case, it’s better to decouple your applications,
  + using SQS: queue model
  + using SNS: pub/sub model
  + using Kinesis: real-time streaming model
* These services can scale independently from our application!

## SQS

A diagram of a diagram

AI-generated content may be incorrect.

Amazon SQS – Standard Queue

* Oldest offering (over 10 years old)
* Fully managed service, used to decouple applications
* Attributes:
  + Unlimited throughput (IOPS x request size), unlimited number of messages in queue
  + Default retention of messages: 4 days, maximum of 14 days
  + Low latency (<10 MS on publish and receive)
  + Limitation of 256KB per message sent
* Can have duplicate messages (at least once delivery, occasionally)
* Can have out of order messages (best effort ordering)

**SQS – Producing Messages**

* Application can send messages to SQS using the SDK using API Send Message API
* The message will be persisted in SQS until a consumer reads and deletes it
* Message retention: default 4 days, up to 14 days
* Example: send an order to be processed
  + Order id
  + Customer id
  + Any attributes you want
* SQS standard: unlimited throughput

A diagram of a button

AI-generated content may be incorrect.

**SQS – Consuming Messages**

* Consumers (running on EC2 instances, on premise servers, or AWS Lambda)…
* Consumers poll SQS for messages (receive up to 10 messages at a time)
* Consumers process the messages (example: insert the message into an RDS database)
* Delete the messages using the Delete Message API responsibility of consumer

A diagram of a message

AI-generated content may be incorrect.

**SQS – Multiple EC2 Instances Consumers**

|  |  |
| --- | --- |
| * Consumers can receive and process messages in parallel, in this example we have 3 on the right side * Each consumer will receive a different set of messages by calling the poll function * At least once delivery, which means at least one consumer will receive the message * Best-effort message ordering * Consumers delete messages after processing them otherwise other consumers will read them again. * We can scale consumers horizontally to improve throughput of processing | A diagram of a machine  AI-generated content may be incorrect. |

**SQS with Auto Scaling Group (ASG)**

This is a perfect use case for using SQS with your Auto Scaling groups, or ASG which will have EC2 as target. ASG can be scaling based on ApproximateNumberOfMessages. It is a CloudWatch Metric that's available in any SQS queue. And we could set up an alarm,

such as whenever the queue length goes over a certain level, then please set up a CloudWatch Alarm, and this alarm should increase the capacity of my Auto Scaling group by X amount

A diagram of a diagram

AI-generated content may be incorrect.

**Decouple application**

SQS can be used to decouple applications.

A diagram of a computer

Description automatically generated with medium confidence

**Amazon SQS – Security**

* Encryption:
  + In-flight encryption using HTTPS API
  + At-rest encryption using KMS keys
  + Client-side encryption if the client wants to perform encryption/decryption itself
* Access Controls: IAM policies to regulate access to the SQS API
* SQS Access Policies (like S3 bucket policies)
  + Useful for cross-account access to SQS queues
  + Useful for allowing other services (SNS, S3…) to write to an SQS queue

**SQS – Message Visibility Timeout**

* After a message is polled by a consumer, it becomes invisible to other consumers
* By default, the “message visibility timeout” is 30 seconds
* That means the message has 30 seconds to be processed and, in that period, message will not be visible to consumers
* After the message visibility timeout is over, the message is “visible” in SQS
* If the message is not delete, then the message will be read by another consumer

A diagram of a message

AI-generated content may be incorrect.

* If a message is not processed within the visibility timeout, it will be processed twice
* A consumer could call the ChangeMessageVisibility API to get more time
* If visibility timeout is high (hours), and consumer crashes, re-processing will take time
* If visibility timeout is too low (seconds), we may get duplicates

**Amazon SQS - Long Polling**

|  |  |
| --- | --- |
| * When a consumer requests messages from the queue, it can optionally “wait” for messages to arrive if there are none in the queue * This is called Long Polling * Long Polling decreases the number of API calls made to SQS while increasing the efficiency and reducing latency of your application * The wait time can be between 1 sec to 20 sec (20 sec preferable) * Long Polling is preferable to Short Polling * Long polling can be enabled at the queue level or at the API level using WaitTimeSeconds | A diagram of a diagram  AI-generated content may be incorrect. |

**Amazon SQS – FIFO Queue**

* FIFO = First in First Out (ordering of messages in the queue)

A close-up of a sign

AI-generated content may be incorrect.

* Limited throughput: 300 msg/s without batching, 3000 msg/s with batching
* Exactly-once send capability (by removing duplicates using De-duplication-Id) If the Id is seen twice in 5 minutes the duplicates will be removed.
* Messages are processed in order by the consumer
* Ordering by message group id i.e. all messages in the same group are ordered. This is a mandatory parameter.

**SQS with Auto Scaling Group (ASG)**

We can have an ASG to monitor Cloud watch matrix ApproximateNumberOfMessages. If there are too many messages, we can trigger Cloud watch alarm which will trigger more instance in ASG to consume messages.

A diagram of a system

AI-generated content may be incorrect.

Say if we are storing the message in DB after reading from producer application and if the load is too big, some transactions may be lost

A diagram of a software application

AI-generated content may be incorrect.

We can prevent this by adding SQS in between producer and consumer to prevent data loss

A diagram of a software flow

AI-generated content may be incorrect.

## Amazon SNS

• What if you want to send one message to many receivers? So you could have direct integration. But this is cumbersome because every time you have to add a new receiving service. You need to create and write that integration. Instead, what you may want to do is called a Pub/Sub or Publish-subscribe

A diagram of a service

AI-generated content may be incorrect.

**Amazon SNS**

* The “event producer” only sends message to one SNS topic
* To receive message we can have as many “event receivers” (subscriptions) as we want to listen to the SNS topic notifications
* Each subscriber to the topic will get all the messages (note: new feature to filter messages)
* Up to 12,500,000 subscriptions per topic
* 100,000 topics limit

A screenshot of a computer

AI-generated content may be incorrect.

**SNS integrates with a lot of AWS services**

Many AWS services can send data directly to SNS for notifications, some of them mentioned below.

A screenshot of a computer

AI-generated content may be incorrect.

**Amazon SNS – How to publish**

Topic Publish (using the SDK)

* Create a topic
* Create a subscription (or many)
* Publish to the topic

Direct Publish (for mobile apps SDK)

* Create a platform application
* Create a platform endpoint
* Publish to the platform endpoint
* Works with Google GCM, Apple APNS, Amazon ADM

**Amazon SNS – Security**

* Encryption:
  + In-flight encryption using HTTPS API
  + At-rest encryption using KMS keys
  + Client-side encryption if the client wants to perform encryption/decryption itself
* Access Controls: IAM policies to regulate access to the SNS API
* SNS Access Policies (similar to S3 bucket policies)
  + Useful for cross-account access to SNS topics
  + Useful for allowing other services ( S3…) to write to an SNS topic

**SNS + SQS: Fan Out**

The idea is that you want a message to be sent to multiple SQS queues, but if you send them individually to every SQS queue, there can be problems associated with it. For example, if your application crashes in between, if there are delivery failures, or if you add more SQS queues down the road. Therefore, we want to use the fan-out pattern.

A diagram of a diagram

AI-generated content may be incorrect.

The idea of this as well is that it's a fully decoupled model and there is no data loss. SQS will give you data persistence, delayed processing, as well as retries of work. And with this pattern, we can add more SQS queues as a subscriber of the SNS topic over time

* Push once in SNS, receive in all SQS queues that are subscribers
* Fully decoupled, no data loss
* SQS allows for: data persistence, delayed processing and retries of work
* Ability to add more SQS subscribers over time
* Make sure your SQS queue access policy allows for SNS to write
* Cross-Region Delivery: works with SQS Queues in other regions

**Fan out Example using S3 event to multiple queues**

There is a limitation of S3 event rules that for a combination of event type, for example, an object is being created and a prefix, for example, images/, you can only have one S3 event rule. But what if you want to send the same S3 event notification to multiple SQS queues? In that case, you would use the fan-out pattern. For example, we have S3 object created as an event appearing in your S3 bucket, and we will send this event into an SNS topic, and we will subscribe to many SQS queues to the SNS topic as a final pattern. But we could also subscribe to other types of application, email, Lambda functions, et cetera, et cetera. And then what we get from this is that the message of the event happening in Amazon S3 is going to go to many different destinations thanks to this fan-out pattern.

* For the same combination of event type (e.g. object create) and prefix (e.g. images/) you can only have one S3 Event rule
* If you want to send the same S3 event to many SQS queues, use fan-out

A diagram of a fan out

Description automatically generated

**Application: SNS to Amazon S3 through Kinesis Data Firehose**

Another architecture is that you can send data directly from SNS to Amazon S3 through Kinesis Data Firehose. Because SNS has a direct integration with KDF, then your buying service can send data into an SNS topic. Then Kinesis Data Firehose, KDF, will receive that information. And then from Kinesis Data Firehose, you can send it into your Amazon S3 bucket, or for that matter, any supported KDF specific destination, which allows you to be really extensible in the way you want to maybe persist your messages from your SNS topic.

SNS can send to Kinesis and therefore we can have the following solutions architecture:

A diagram of a diagram

AI-generated content may be incorrect.

**Amazon SNS – FIFO Topic**

We can apply the fan-out pattern to FIFO topics as well. So, Amazon SNS has a FIFO or FIFO capability, which is first in first out, which gives ordering of messages in the topic. So, the producer send the messages. One, two, three, four. And the subscribers can only be an SQS FIFO queue for now, which is receiving messages, one, two, three, four, in order. So, the idea is that with SNS FIFO, we get the same features as SQS FIFO, we get ordering by message group ID. We get deduplication using a deduplication ID or content-based deduplication. And both SQS standard and FIFO queues can be subscribers. In terms of throughput, you're limited. You get the same throughput as the SQS FIFO queue.

* FIFO = First In First Out (ordering of messages in the topic)

A red rectangles with white rectangles

Description automatically generated

* Similar features as SQS FIFO:
* Ordering by Message Group ID (all messages in the same group are ordered)
* Deduplication using a Deduplication ID or Content Based Deduplication
* Can have SQS Standard and FIFO queues as subscribers
* Limited throughput (same throughput as SQS FIFO)

So why do we need this? Well, in case you want to do a fan out using SQS FIFO. So, you need fan out, ordering and deduplication. So, the buying service will send the data into an SNS FIFO topic, and then it will fan out to two SQS FIFO queues, which can have the fraud service and the shipping service read from FIFO queues as well.

In case you need fan out + ordering + deduplication

A diagram of a diagram

Description automatically generated

**SNS – Message Filtering**

One last feature of SNS that can be handy regarding the fan-out pattern is that you can do message filtering in SNS. So what is message filtering? Well, it's a JSON policy that is used to filter messages sent to the SNS topics subscription. So, if a subscription doesn't have a filter policy, it will receive every message, and that's a default behavior. But let's take an example of what happens when we set up a message filtering policy. So, we have a buying service and it sends transactions into an SNS topic. For example, the transaction looked like there's an order number, there's a product, for example, a pencil, the quantity for, and the state which is placed. Now we want to create an SQS queue just for placed orders, not all the orders, but just for placed orders. So, for this, we will subscribe the SQS queue into the SNS topic, and we'll apply a filter policy in JSON, and we will specify in a policy that we want to have State equals Placed. And so only the messages matching the policy will go into the SQS queue, but then we could have an SQS queue for canceled orders. And so we can create our own filter policy for canceled orders and have those coming from the same SNS topic into the SQS queue. And so the placed orders and the canceled orders SQS queue will not have the same messages. We can also use the same filter policy, the canceled one, to create an email subscription for canceled orders. We could have a filter policy for declined orders, for example, and as another SQS queue, or we can create an SQS queue without a filter policy to have all messages from that SNS topic. So, using all these fan-out patterns and message filtering, FIFO queues and FIFO topics, we get lots of different possibilities and the exam will try to test you on all of those

* JSON policy used to filter messages sent to SNS topic’s subscriptions
* If a subscription doesn’t have a filter policy, it receives every message

A diagram of a diagram

Description automatically generated

## Kinesis Overview

It’s a service used to collect and store streaming data in real time. The keyword here is real time. It is data that is created and used on the spot. For example, whenever users click on a website, it's called a click stream, or when you have a device that's connected to the internet, for example, a connected bicycle, or when you have metrics and logs on a server, and you want to use them directly, so you want to send those into Amazon Kinesis Data Streams. And to do so we must use what's called producers. So, producers can be either applications and this is actually code you have to write to take data from your website or from your devices and send it into Kinesis Data Streams.

A computer screen shot of a computer

AI-generated content may be incorrect.

* Makes it easy to collect, process, and analyze streaming data in real-time
* Ingest real-time data such as: Application logs, Metrics, Website clickstreams, IoT telemetry data…

A purple rectangular object with white background

AI-generated content may be incorrect.

* Kinesis Data Streams: capture, process, and store data streams
* Kinesis Data Firehose: load data streams into AWS data stores
* Kinesis Data Analytics: analyze data streams with SQL or Apache Flink
* Kinesis Video Streams: capture, process, and store video streams

**Kinesis Data Streams**

A screenshot of a computer

Description automatically generated

**Kinesis Data Streams**

* Retention between 1 day to 365 days
* Ability to reprocess (replay) data
* Once data is inserted in Kinesis, it can’t be deleted (immutability)
* Data that shares the same partition goes to the same shard (ordering)
* Producers: AWS SDK, Kinesis Producer Library (KPL), Kinesis Agent
* Consumers:
  + Write your own: Kinesis Client Library (KCL), AWS SDK
  + Managed: AWS Lambda, Kinesis Data Firehose, Kinesis Data Analytics,

**Kinesis Data Streams – Capacity Modes**

* Provisioned mode:
  + You choose the number of shards provisioned, scale manually or using API more shards more throughput.
  + Each shard gets 1MB/s in (or 1000 records per second)
  + Each shard gets 2MB/s out (classic or enhanced fan-out consumer)
  + You pay per shard provisioned per hour
* On-demand mode:
  + No need to provision or manage the capacity
  + Default capacity provisioned (4 MB/s in or 4000 records per second)
  + Scales automatically based on observed throughput peak during the last 30 days
  + Pay per stream per hour & data in/out per GB

**Kinesis Data Streams Security**

|  |  |
| --- | --- |
| * Control access / authorization using IAM policies * Encryption in flight using HTTPS endpoints * Encryption at rest using KMS * You can implement encryption/decryption of data on client side (harder) * VPC Endpoints available for Kinesis to access within VPC * Monitor API calls using CloudTrail | A diagram of a computer  AI-generated content may be incorrect. |

## Kinesis Data Firehose

Amazon Data Firehose is a service to send data from sources to target destinations. So to get data into Amazon Data Firehose, you may have producers that could be your applications, your clients, or things you write, and you use the SDK to send data into Firehose or the Kinesis agents, as well as Firehose has the capability to pull directly from some services such as, for example, a Kinesis Data Streams or an Amazon CloudWatch logs and events, or for example, AWS IoT.

And so all these services are either going to be pushing into Amazon Data Firehose or Firehose are going to be pulling from, for example, a Kinesis data stream and it is going to receive records. Now these records are going to be received, and then they can optionally be transformed using a Lambda function, if you wanted to do some data conversion format, for example. And they're going to accumulate into a buffer and the buffer is going to be flushed occasionally to do batch writing into a lot of destinations.

So, the first destinations are going to be AWS specific destinations, such as sending data into Amazon S3 or into Amazon Redshift to perform analytics, or into Amazon OpenSearch as well. You also have the option to send from Data Firehose directly into third-party partner destinations such as Datadog, Splunk, New Relic, and MongoDB. And finally, if your destination is not supported, you can simply use the HTTP endpoint integration to send it anywhere you want. Now, Firehose will write this data, but you also have the option to write all or just the failed data that has been sent into an S3 bucket for a backup.

A diagram of a firehose

AI-generated content may be incorrect.

So Amazon Data Firehose used to be called Kinesis Data Firehose for the longest time, but now it's called Amazon Data Firehose because it does way more than just Kinesis

**Kinesis Data Firehose**

* Fully Managed Service, no administration, automatic scaling, serverless
  + AWS: Redshift / Amazon S3 / OpenSearch
  + 3rd party partner: Splunk / MongoDB / DataDog / NewRelic / …
  + Custom: send to any HTTP endpoint
* Automatic scaling, serverless, pay for use
* Pay for data going through Firehose
* Near Real Time
  + Buffer interval: 0 seconds (no buffering) to 900 seconds
  + Buffer size: minimum 1MB
* Supports many data formats, like CSV, JSON, Parquet, Avro, Raw Text binary
* Conversion to Parquet/ORC , compression with gzip/snappy
* Supports custom data transformations using AWS Lambda
* Can send failed or all data to a backup S3 bucket

**Kinesis Data Streams vs Firehose**



|  |  |
| --- | --- |
| • Streaming service for ingest at scale  • Write custom code (producer /  consumer)  • Real-time (~200 ms)  • Manage scaling (shard splitting /  merging)  • Data storage for 1 to 365 days  • Supports replay capability | • Load streaming data into S3 / Redshift / OpenSearch / 3rd party / custom HTTP  • Fully managed  • Near real-time  • Automatic scaling  • No data storage  • Doesn’t support replay capability |

**Ordering data into Kinesis**

|  |  |
| --- | --- |
| * Imagine you have 100 trucks (truck\_1, truck\_2, … truck\_100) on the road sending their GPS positions regularly into AWS. * You want to consume the data in order for each truck, so that you can track their movement accurately. * How should you send that data into Kinesis? * Answer: send using a “Partition Key” value of the “truck\_id” * The same key will always go to the same shard | A diagram of a computer program  AI-generated content may be incorrect. |

**Ordering data into SQS**

* For SQS standard, there is no ordering.
* For SQS FIFO, if you don’t use a Group ID, messages are consumed in the order they are sent, with only one consumer



* You want to scale the number of consumers, but you want messages to be “grouped” when they are related to each other
* Then you use a Group ID (similar to Partition Key in Kinesis)

A diagram of a company

Description automatically generated with medium confidence

**Kinesis vs SQS ordering**

* Let’s assume 100 trucks, 5 kinesis shards, 1 SQS FIFO
* Kinesis Data Streams:
  + On average you’ll have 20 trucks per shard
  + Trucks will have their data ordered within each shard
  + The maximum amount of consumers in parallel we can have is 5
  + Can receive up to 5 MB/s of data
* SQS FIFO
  + You only have one SQS FIFO queue
  + You will have 100 Group ID
  + You can have up to 100 Consumers (due to the 100 Group ID)
  + You have up to 300 messages per second (or 3000 if using batching)

## SQS vs SNS vs Kinesis

|  |  |  |
| --- | --- | --- |
| SQS:  • Consumer “pull data”  • Data is deleted after being  consumed  • Can have as many workers  (consumers) as we want  • No need to provision  throughput  • Ordering guarantees only on  FIFO queues  • Individual message delay  capability | SNS:  • Push data to many  subscribers  • Up to 12,500,000 subscribers  • Data is not persisted (lost if  not delivered)  • Pub/Sub  • Up to 100,000 topics  • No need to provision  throughput  • Integrates with SQS for fan- out architecture pattern  • FIFO capability for SQS FIFO | Kinesis:  • Standard: pull data  • 2 MB per shard  • Enhanced-fan out: push data  • 2 MB per shard per consumer  • Possibility to replay data  • Meant for real-time big data,  analytics and ETL  • Ordering at the shard level  • Data expires after X days  • Provisioned mode or ondemand capacity mode |

## Amazon MQ

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

A white background with black and white clouds

AI-generated content may be incorrect.

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

**Amazon MQ – High Availability**

A diagram of a company

AI-generated content may be incorrect.

## Questions

**Top of Form**

**You have an e-commerce website and you are preparing for Black Friday which is the biggest sale of the year. You expect that your traffic will increase by 100x. Your website already using an SQS Standard Queue, and you're running a fleet of EC2 instances in an Auto Scaling Group to consume SQS messages. What should you do to prepare your SQS Queue?**

* **Contact AWS Support to pre-warm your SQS Standard Queue**
* **Enable Auto Scaling in your SQS queue**
* **Increase the capacity of the SQS queue**
* **Do nothing, SQS scales automatically**

**Bottom of Form**

Ans D

**Top of Form**

**You have an SQS Queue where each consumer polls 10 messages at a time and finishes processing them in 1 minute. After a while, you noticed that the same SQS messages are received by different consumers resulting in your messages being processed more than once. What should you do to resolve this issue?**

* **Enable Long Polling**
* **Add DelaySeconds parameter to the messages when being produced**
* **Increase the Visibility Timeout**
* **Decrease the Visibility Timeout**

**Bottom of Form**

Ans A When SQS Long Polling is enabled, Amazon SQS reducing the number of empty responses when there are no messages available to return and eliminating false empty responses (when SQS messages are available but aren't included in a response).

B SQS Delay Queues is a period of time during which Amazon SQS keeps new SQS messages invisible to consumers. In SQS Delay Queues, a message is hidden when it is first added to the queue. (default: 0 mins, max.: 15 mins)

C correct SQS Visibility Timeout is a period of time during which Amazon SQS prevents other consumers from receiving and processing the message again. In Visibility Timeout, a message is hidden only after it is consumed from the queue. Increasing the Visibility Timeout gives more time to the consumer to process the message and prevent duplicate reading of the message. (default: 30 sec., min.: 0 sec., max.: 12 hours)

**Top of Form**

**Which SQS Queue type allows your messages to be processed exactly once and in order?**

* **SQS Standard Queue**
* **SQS Dead Letter Queue**
* **SQS Delay Queue**
* **SQS FIFO Queue**

**Bottom of Form**

Ans A SQS Standard Queue doesn't have ordering capability and doesn't guarantee to process your messages exactly once.

B SQS Dead Letter Queue is where other SQS queues (source queues) can send messages that can't be processed (consumed) successfully. It's useful for debugging as it allows you to isolate problematic messages so you can debug why their processing doesn't succeed.

C SQS Delay Queues is a period of time during which Amazon SQS keeps new SQS messages invisible to consumers. In SQS Delay Queues, a message is hidden when it is first added to the queue. (default: 0 mins, max.: 15 mins)

D Correct SQS FIFO (First-In-First-Out) Queues have all the capabilities of the SQS Standard Queue, plus the following two features. First, The order in which messages are sent and received are strictly preserved and a message is delivered once and remains available until a consumer process and deletes it. Second, duplicated messages are not introduced into the queue.

**Top of Form**

**You have 3 different applications that you'd like to send them the same message. All 3 applications are using SQS. What is the best approach would you choose?**

* **Use SQS Replication Feature**
* **Use SNS + SQS Fan Out Pattern**
* **Send messages individually to 3 SQS queues**

**Bottom of Form**

Ans A There's no such feature.

B Correct This is a common pattern where only one message is sent to the SNS topic and then "fan-out" to multiple SQS queues. This approach has the following features: it's fully decoupled, no data loss, and you have the ability to add more SQS queues (more applications) over time.

C This is not a common pattern and doesn't scale well.

**Top of Form**

**You have a Kinesis data stream with 6 shards provisioned. This data stream usually receiving 5 MB/s of data and sending out 8 MB/s. Occasionally, your traffic spikes up to 2x and you get a ProvisionedThroughputExceeded exception. What should you do to resolve the issue?**

* **Add more Shards**
* **Enable Kinesis Replication**
* **Use SQS as a buffer to Kinesis**

**Bottom of Form**

Ans A The capacity limits of a Kinesis data stream are defined by the number of shards within the data stream. The limits can be exceeded by either data throughput or the number of reading data calls. Each shard allows for 1 MB/s incoming data and 2 MB/s outgoing data. You should increase the number of shards within your data stream to provide enough capacity.

B Unfortunately, this feature doesn't exist!

C This is not a good design.

**Top of Form**

**You have a website where you want to analyze clickstream data such as the sequence of clicks a user makes, the amount of time a user spends, and where the navigation begins and how it ends. You decided to use Amazon Kinesis, so you have configured the website to send these clickstream data all the way to a Kinesis data stream. While you checking the data sent to your Kinesis data stream, you found that the users' data is not ordered and the data for one individual user is spread across many shards. How would you fix this problem?**

* **There are too many shards, you should only use 1 shard**
* **You shouldn't use multiple consumers, only one and it should re-order data**
* **For each record sent to Kinesis add a partition key that represents the identity of the user**

**Bottom of Form**

Ans A Using 1 shard would guarantee total ordering but wouldn't allow us to scale.

B This wouldn't allow us to scale.

C correct Kinesis Data Stream uses the partition key associated with each data record to determine which shard a given data record belongs to. When you use the identity of each user as the partition key, this ensures the data for each user is ordered hence sent to the same shard.

**Top of Form**

**You are running an application that produces a large amount of real-time data that you want to load into S3 and Redshift. Also, these data need to be transformed before being delivered to their destination. What is the best architecture would you choose?**

* **SQS + AWS Lambda**
* **SNS + HTTP Endpoint**
* **Kinesis Data Streams + Kinesis Data Firehose**

**Bottom of Form**

Ans A This doesn't have a direct integration with S3 or Redshift.

B This doesn't have a direct integration with S3 or Redshift.

C correct This is a perfect combo of technology for loading data near real-time data into S3 and Redshift. Kinesis Data Firehose supports custom data transformations using AWS Lambda.

**Top of Form**

**Which of the following is NOT a supported subscriber for AWS SNS?**

* **Amazon Kinesis Data Streams**
* **Amazon SQS**
* **HTTP(S) Endpoint**
* **AWS Lambda**

**Bottom of Form**

Ans A correct Note: Kinesis Data Firehose is now supported, but not Kinesis Data Streams.

**Top of Form**

**Which AWS service helps you when you want to send email notifications to your users?**

* **Amazon SQS with AWS Lambda**
* **Amazon SNS**
* **Amazon Kinesis**

**Bottom of Form**

Ans A You would be re-implementing a feature from SNS.

B Correct

C Amazon Kinesis on its own can not send emails. You would need to chain it with a consumer, and in the end, you would re-create something like SNS.

**Top of Form**

**You're running many micro-services applications on-premises and they communicate using a message broker that supports MQTT protocol. You're planning to migrate these applications to AWS without re-engineering the applications and modifying the code. Which AWS service allows you to get a managed message broker that supports the MQTT protocol?**

* **Amazon SQS**
* **Amazon SNS**
* **Amazon Kinesis**
* **Amazon MQ**

**Bottom of Form**

Ans D Amazon MQ supports industry-standard APIs such as JMS and NMS, and protocols for messaging, including AMQP, STOMP, MQTT, and WebSocket.

**Top of Form**

**An e-commerce company is preparing for a big marketing promotion that will bring millions of transactions. Their website is hosted on EC2 instances in an Auto Scaling Group and they are using Amazon Aurora as their database. The Aurora database has a bottleneck and a lot of transactions have been failed in the last promotion they have made as they had a lot of transaction and the Aurora database wasn’t prepared to handle these too many transactions. What do you recommend to handle those transactions and prevent any failed transactions?**

* **Use SQS as a buffer to write to Aurora**
* **Host the website in AWS Fargate instead of EC2 instances**
* **Migrate Aurora to RDS for SQL Server**

**Bottom of Form**

Ans A

**Top of Form**

**A company is using Amazon Kinesis Data Streams to ingest clickstream data and then do some analytical processes on it. There is a campaign in the next few days and the traffic is unpredictable which might grow up to 100x. What Kinesis Data Stream capacity mode do you recommend?**

* **Provisioned Mode**
* **On-demand Mode**

**Bottom of Form**

Ans B

# Containers on AWS

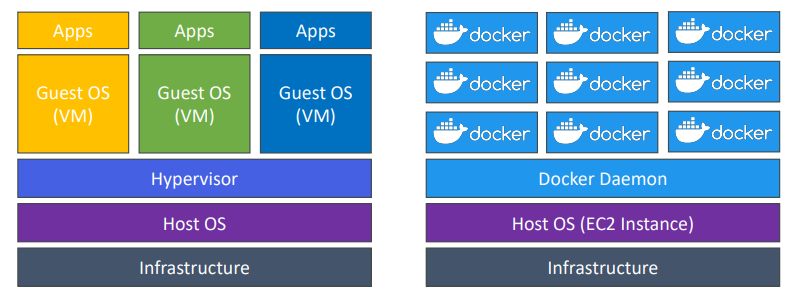
|  |  |
| --- | --- |
| What is Docker?   * Docker is a software development platform to deploy apps * Apps are packaged in containers that can be run on any OS * Apps run the same, regardless of where they’re run * Any machine * No compatibility issues * Predictable behavior * Less work * Easier to maintain and deploy * Works with any language, any OS, any technology * Use cases: microservices architecture, lift-and-shift apps from on- premises to the AWS cloud, … |  |

**Where are Docker images stored?**

* Docker images are stored in Docker Repositories
* Docker Hub (https://hub.docker.com)
* Public repository
* Find base images for many technologies or OS (e.g., Ubuntu, MySQL, …)
* Amazon ECR (Amazon Elastic Container Registry)
* Private repository
* Public repository (Amazon ECR Public Gallery <https://gallery.ecr.aws>)

**Docker vs. Virtual Machines**

* Docker is ”sort of” a virtualization technology, but not exactly
* Resources are shared with the host => many containers on one server



**Getting Started with Docker**

A diagram of a docker

AI-generated content may be incorrect.

**Docker Containers Management on AWS**

|  |  |
| --- | --- |
| * Amazon Elastic Container Service (Amazon ECS)   + Amazon’s own container platform * Amazon Elastic Kubernetes Service (Amazon EKS)   + Amazon’s managed Kubernetes (open source) * AWS Fargate   + Amazon’s own Serverless container platform   + Works with ECS and with EKS * Amazon ECR:   + Store container images |  |

## ECS launch types

**Amazon ECS - EC2 Launch Type**

|  |  |
| --- | --- |
| * When you launch Docker Containers on AWS, you are launching what's called an ECS Task on ECS Cluster. * And an ECS Cluster is made of things. And with the EC2 Launch Type, well these things are EC2 instances. * Each EC2 Instance must run the ECS Agent to register in the ECS Cluster. This Agent is going to register each, EC2 Instance into the Amazon ECS service and the specified ECS Cluster. * AWS takes care of starting / stopping containers there after on EC2 |  |

**Amazon ECS – Fargate Launch Type**

|  |  |
| --- | --- |
| * Launch Docker containers on AWS * You do not provision the infrastructure (no EC2 instances to manage) * It’s all Serverless! * You just create task definitions i.e. docker containers * AWS just runs ECS Tasks for you based on the CPU / RAM you need * To scale, just increase the number of tasks. Simple - no more EC2 instances |  |

**Amazon ECS – IAM Roles for ECS**

|  |  |
| --- | --- |
| * **ECS instance profile**  **(**used by EC2 launch types only). Lets say we have an EC2 instance running ECS agent. ECS agent will use EC2 Instance Profile for :   + Makes API calls to ECS service   + Send container logs to CloudWatch Logs   + Pull Docker image from ECR   + Reference sensitive data in Secrets Manager or SSM Parameter Store * **ECS Task Role**: (used by both EC2 and Fargate launch tyoes) ECS tasks are going to get ECS tasks role   + Allows each task to have a specific role   + Use different roles for the different ECS Services you run   + Task Role is defined in the task definition |  |

**Amazon ECS – Load Balancer Integrations**

|  |  |
| --- | --- |
| * Application Load Balancer supported and works for most use cases * Network Load Balancer recommended only for high throughput / high performance use cases, or to pair it with AWS Private Link * Classic Load Balancer supported but not recommended (no advanced features – no Fargate) |  |

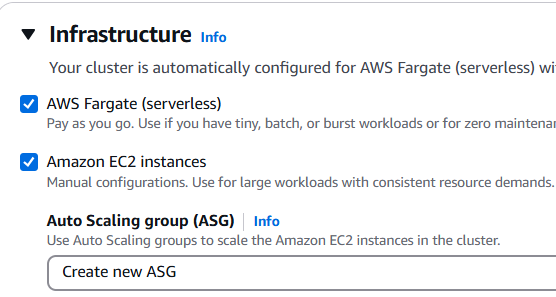
**Amazon ECS – Data Volumes (EFS)**

|  |  |
| --- | --- |
| * Mount EFS file systems onto ECS tasks * Works for both EC2 and Fargate launch types * Tasks running in any AZ will share the same data in the EFS file system * Fargate + EFS = Serverless * Use cases: persistent multi-AZ shared storage for your containers * Note:   + Amazon S3 cannot be mounted as a file system |  |

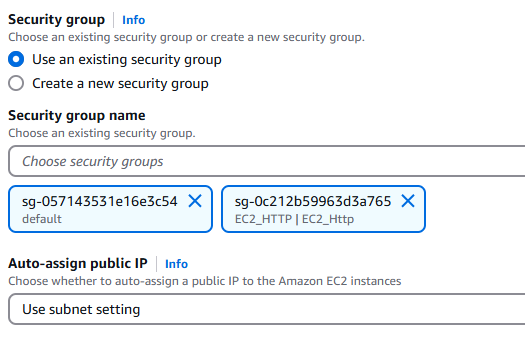
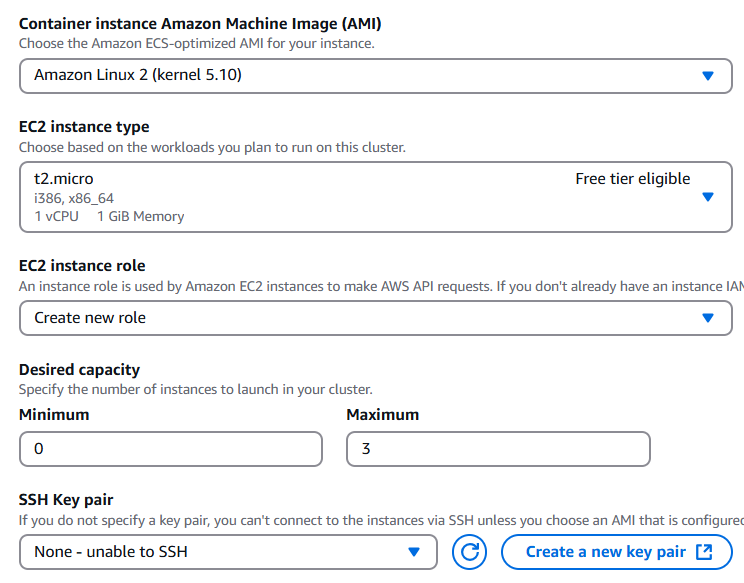
## ECS HandOn

Steps

1. Create a cluster can select Fargate and EC2 and set ASG
2. Create Task definition mention the image port etc.
3. Go to cluster and create a service and select the task which has image details

While creating ECS cluster we can select both Fargate and EC2 instances  


Can select other needed details like instance type, Security group etc



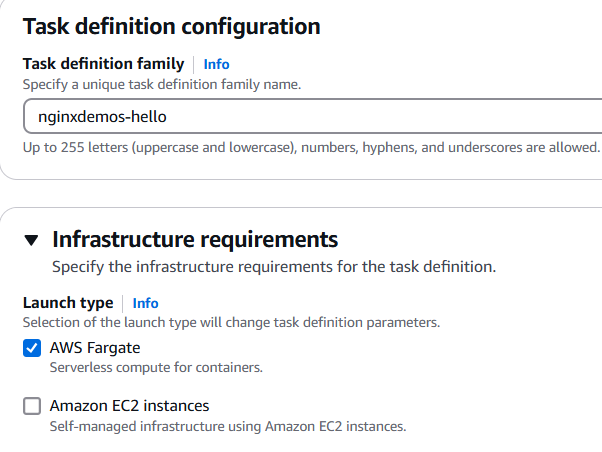
This will trigger a Auto scaling group automatically with min and max capacity and spread across AZ in that region. This ASG is regarding the no of instances in our cluster and not related to services. The instances in these clusters i.e. EC2 will register itself with CLuster

Inside the cluster we can see 3 capacity providers Fargate, Fargate Spot and ASG provider. When we change the ASG min capacity to 1 an EC2 instance is going to be created. And when it's created, it's going to register itself into the Demo Cluster, and then we can see it under container instances.

A screenshot of a computer

AI-generated content may be incorrect.

Next we need to create a task definition. We will try creating with Fargate



Select the capacity of service

A screenshot of a computer

AI-generated content may be incorrect.

So task role is an IAM role that we can assign to our task if we want to make API request to AWS services. Task execution role is we don’t provide it will be created by default. It  is used by the container agent to make AWS API requests on your behalf.

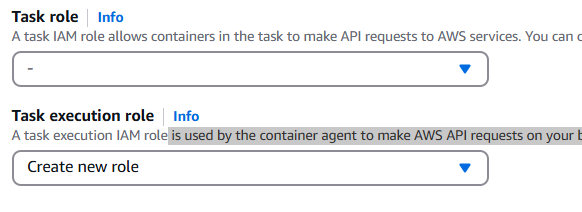


Image options

A screenshot of a computer

AI-generated content may be incorrect.

Port mapping

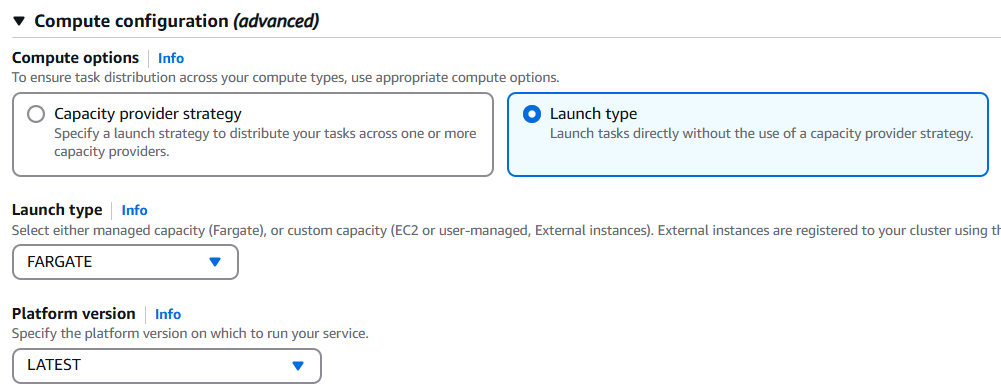
A screenshot of a computer

AI-generated content may be incorrect.

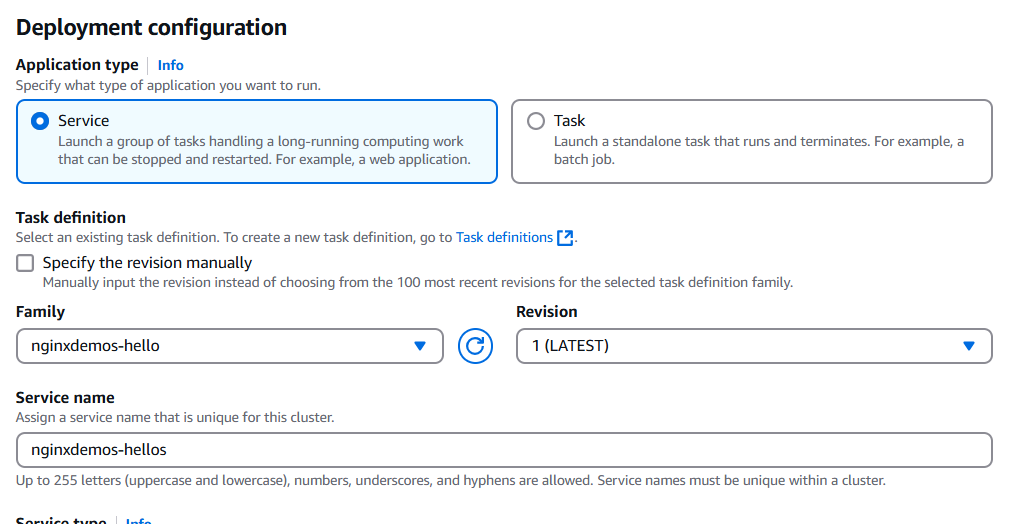
So next, let's launch this task definition as a service. So let's go into Clusters and then DemoCluster. And under services I'm going to create a service

A screenshot of a computer

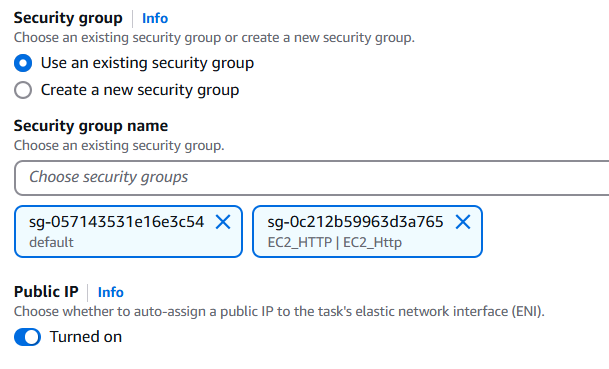
AI-generated content may be incorrect.



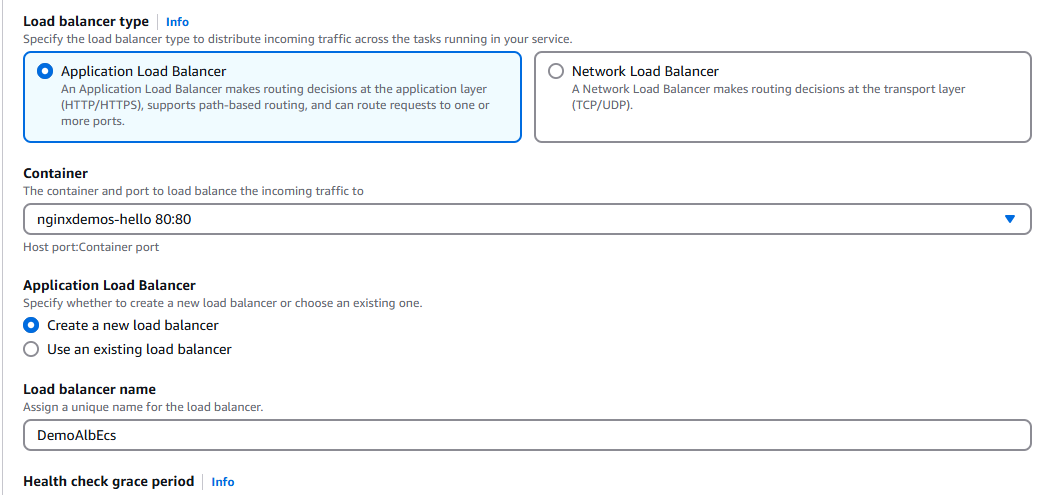
Next, we select task definition for the service



Select security group in network section for allowing traffic



Select load balancers



Once we create a service it will be linked to ASG, and an ALB will also be created

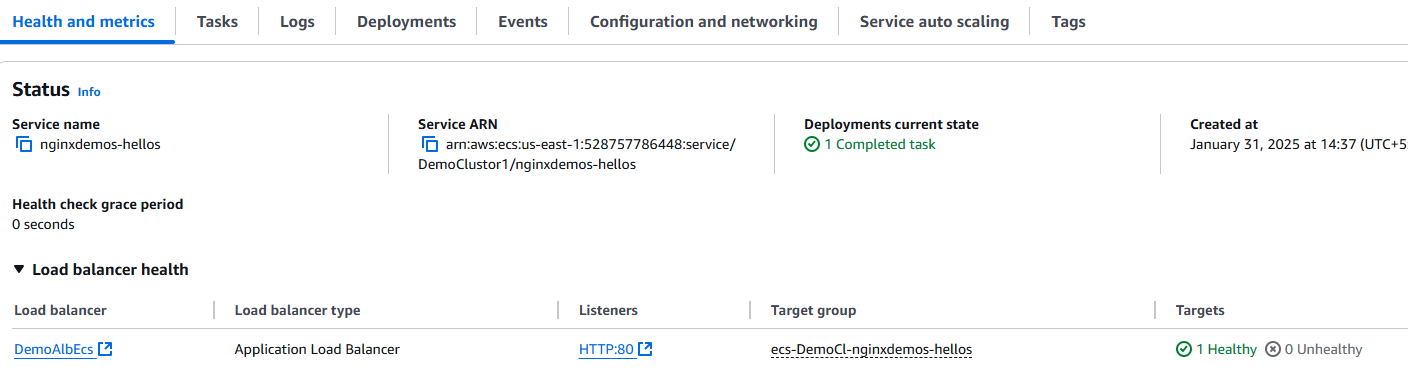
When we check in our cluster, we see the service created , and it is linked to ALB and a Target group. This target group is different to the ASG created earlier

Cluster

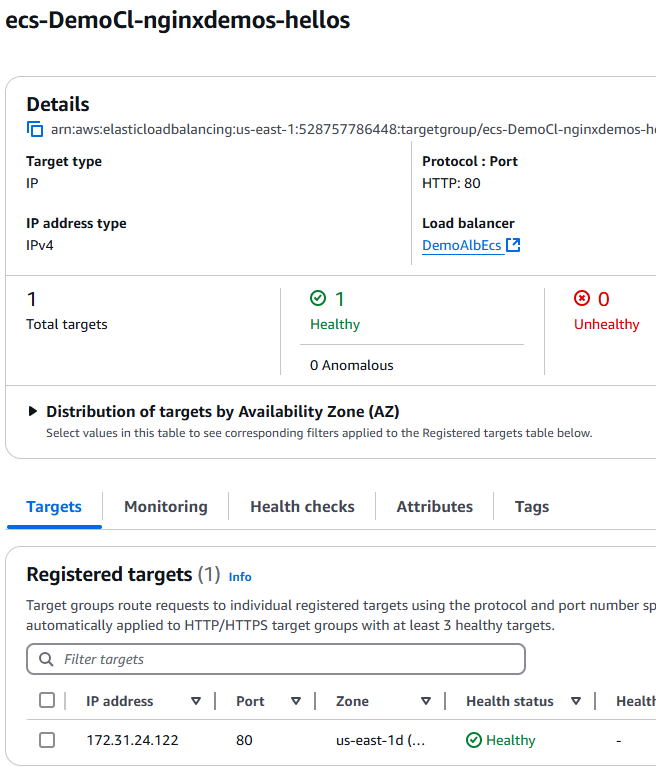
A screenshot of a computer

AI-generated content may be incorrect.

Service



If we open the target group, we can see the IP address of our Service



In the ALB we can see the public URL of the ALB with which we can access the service

A screenshot of a computer

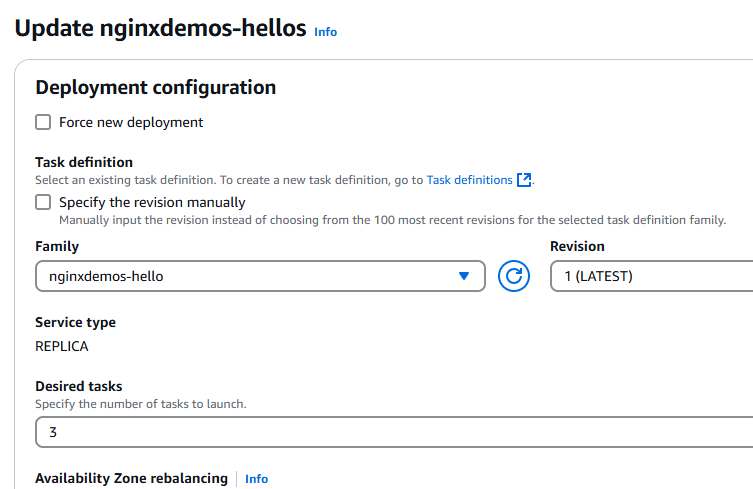
AI-generated content may be incorrect.

When we open the public URL of the ALB we can access nginx page of our service. It will have the IP of the container. This matcher with our service IP

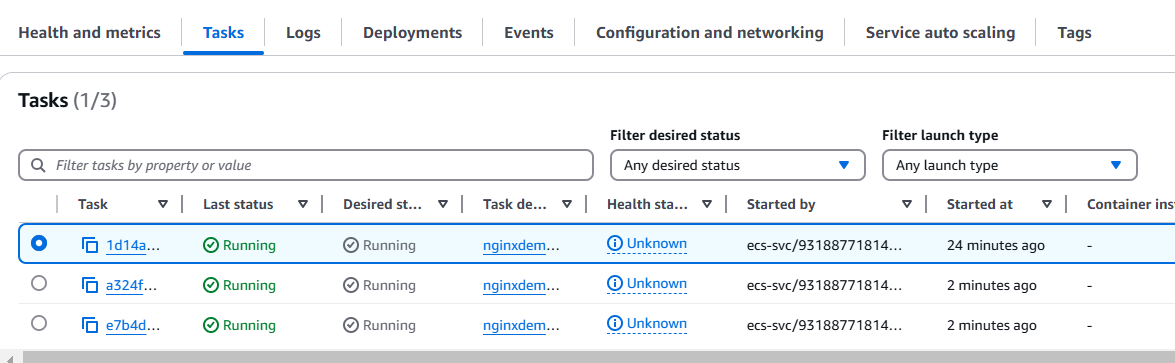
A screenshot of a computer

Description automatically generated

We can update the service to increase the number of containers



In tasks we can see more containers created in tasks. We can also monitor the logs, Events etc.



Now if we refresh the ALB URL, we can see the Ip changing .

We can update the service to zero, save on any costs. Also, in ASG we can make sure that the minimum no of instances in ASG is zero.

## ECS Service Auto Scaling

This section is for scaling the number of service not EC2

* Automatically increase/decrease the desired number of ECS tasks
* Amazon ECS Auto Scaling uses AWS Application Auto Scaling
  + ECS Service Average CPU Utilization
  + ECS Service Average Memory Utilization - Scale on RAM
  + ALB Request Count Per Target – metric coming from the ALB
* Target Tracking – scale based on target value for a specific CloudWatch metric
* Step Scaling – scale based on a specified CloudWatch Alarm
* Scheduled Scaling – scale based on a specified date/time (predictable changes)
* ECS Service Auto Scaling (task level) ≠ EC2 Auto Scaling (EC2 instance level)
* Fargate Auto Scaling is much easier to set up (because Serverless)

**EC2 Launch Type – Auto Scaling EC2 Instances**

This section is about the scaling of the number of EC2 instances and not the service instance(not applicable for fargate)

* Accommodate ECS Service Scaling by adding underlying EC2 Instances
  + Auto Scaling Group Scaling
  + Scale your ASG based on CPU Utilization
* Add EC2 instances over time
* ECS Cluster Capacity Provider
  + Used to automatically provision and scale the infrastructure for your ECS Tasks
  + Capacity Provider paired with an Auto Scaling Group
  + Add EC2 Instances when you’re missing capacity (CPU, RAM…)

## ECS Solution Architect examples

**ECS Scaling – Service CPU Usage Example**

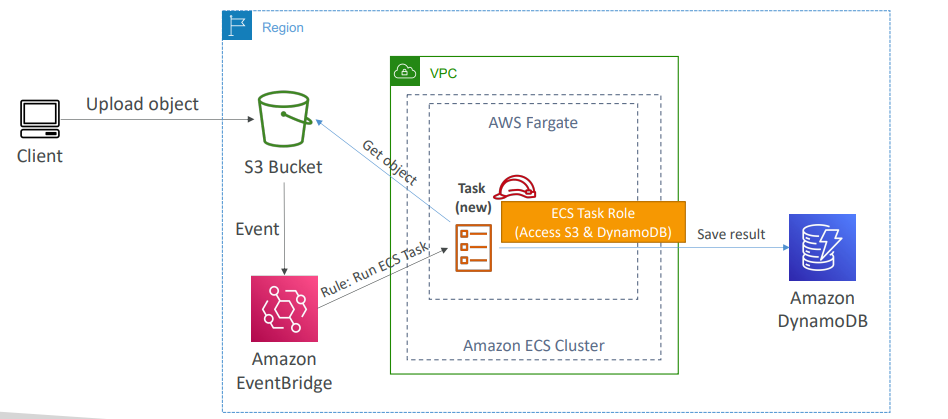
So we have a Service A with two tasks. And we have CPU Usage. And it's going to be auto scaled by the AWS Application Auto Scaling. But let's assume we have more users, and therefore your CPU usage goes really up, then your CloudWatch Metric, which monitors the CPU Usage at the ECS service level again, is going to trigger a CloudWatch Alarm, which will trigger a scaling activity in your Auto Scaling for your ECS service. The desired capacity will increase for your ECS Service, and a new task will be created. And optionally, if this service is running on the EC2 launch type, then the ECS Capacity Providers can help you scale your ECS cluster backed by EC2 instances.

A screenshot of a computer

AI-generated content may be incorrect.

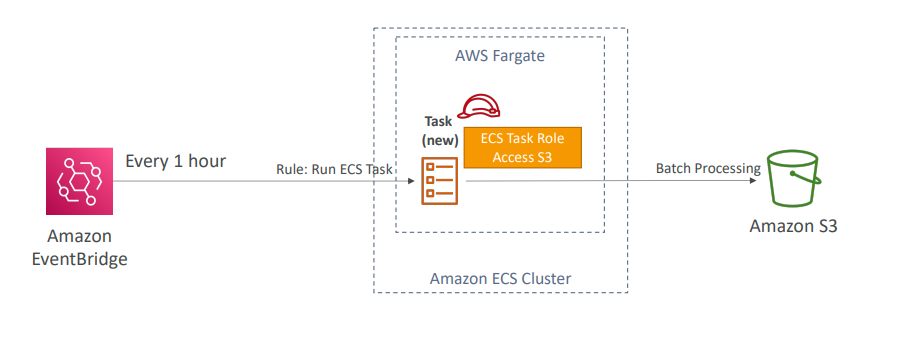
**ECS tasks invoked by Event Bridge**

So for example, say we have an Amazon ECS cluster, it's backed by Fargate, and we have S3 buckets. Our users are going to upload objects into our S3 buckets, And these S3 buckets can be, for example, integrated with Amazon Event Bridge to send all the events to it. And Amazon Event Bridge can have a rule to run ECS tasks on the go. Now, when ECS tasks are going to be created, they will have an ECS task role associated with them, and from the task itself what it can do is that it can get the objects, process it, and then send the results into Amazon DynamoDB. And that is thanks to the fact that we have an ECS task role associated with it. And so effectively here, what we've done is that we've created serverless architecture to process images, or to process objects, from your S3 buckets using a Docker container. And that is using Amazon Event Bridge ECS in the Fargate mode, as well as an ECS task role to talk to Amazon S3 and Amazon DynamoDB.



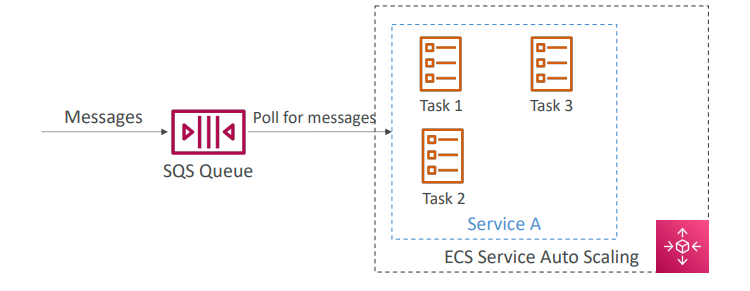
**ECS tasks invoked by Event Bridge Schedule**

Another architecture using, again, Event Bridge, is to use an Event Bridge schedule. So we have an Amazon ECS cluster backed by Fargate and Amazon Event Bridge, and we schedule a rule to be triggered every 1 hour. Now, this rule is going to run ECS tasks for us in Fargate, and so that means that every 1 hour, a new task will be created in our Fargate cluster, and the task can do whatever we want. For example, we can create an ECS task role with access to Amazon S3, and therefore our task, our Docker container, our program can, for example, do every 1 hour some batch processing against some files in Amazon S3. And again, all of that architecture is fully serverless.



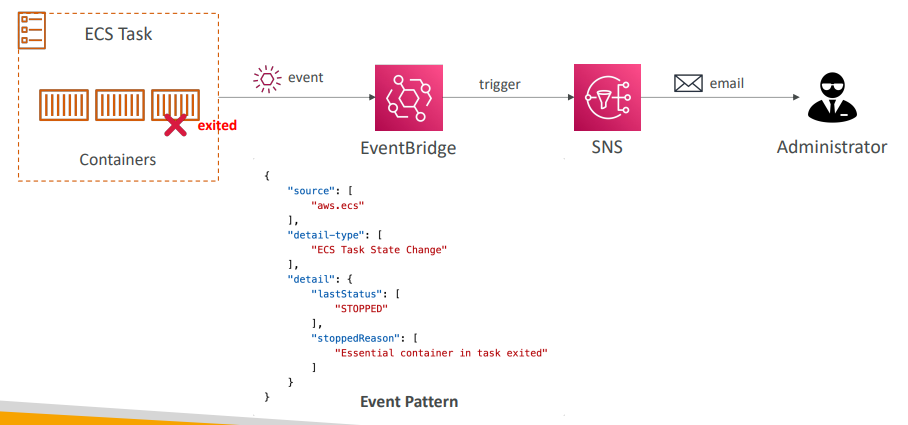
**E** **CS – SQS Queue Example**

A last example is using ECS and an SQS queue, so we could have a service on ECS with two ECS tasks, and messages are being sent into an SQS queue, and the service itself is pulling for messages from the SQS queue, and processing them. We can enable ECS Service Auto Scaling on top of this service. That means that, for example, the more messages we have in our SQS queue, the more tasks we're going to have into our ECS service, thanks to auto-scaling



**ECS – Intercept Stopped Tasks using EventBridge**

Another integration is when you want to use Event Bridge to actually intercept events from within your ECS cluster. So, for example, say you wanted to react to tasks being exited. In that case, any task exiting or starting in your ECS cluster can be triggered as an event in Event Bridge, and it will look like this. For example, the ECS task state change for "stopped" and the stopped reason. Then from there, for example, we could alert an SNS topic and send emails to your administrators. So, bottom line, Event Bridge does allow you to understand the lifecycle of your containers in your ECS cluster.



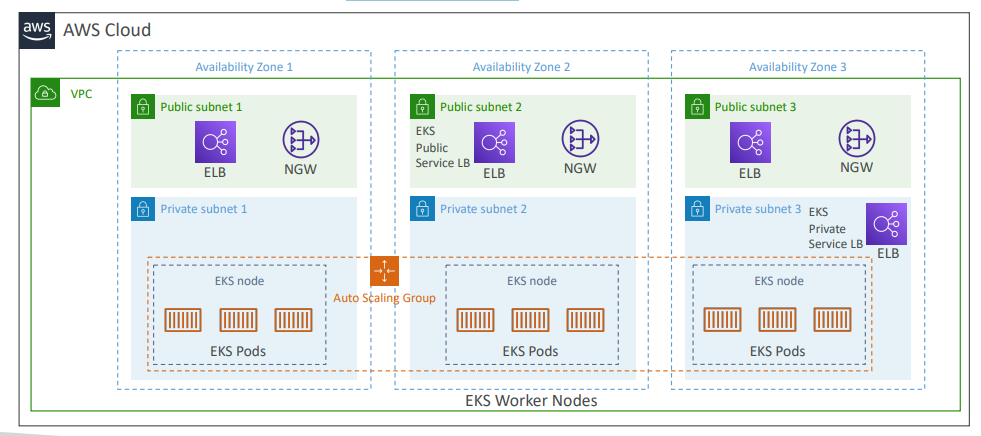
## Amazon ECR

|  |  |
| --- | --- |
| * ECR = Elastic Container Registry * Store and manage Docker images on AWS * Private and Public repository (Amazon ECR Public Gallery https://gallery.ecr.aws) * Fully integrated with ECS, images backed by Amazon S3 * So your ECR repository may contain different Docker images and then your ECS cluster. And for example, an EC2 instance on your ECS cluster may want to pull these images. So to do so while we're going to sign an IAM role to our EC2 instance and this IAM role will allow our instance to pull Docker images. Access is controlled through IAM (permission errors => policy)   Supports image vulnerability scanning, versioning, image tags, image lifecycle, … | A diagram of a computer  AI-generated content may be incorrect. |

## Amazon EKS

* Amazon EKS = Amazon Elastic Kubernetes Service
* It is a way to launch managed Kubernetes clusters on AWS
* Kubernetes is an open-source system for automatic deployment, scaling and management of containerized (usually Docker) application
* It’s an alternative to ECS, similar goal but different API
* EKS supports EC2 if you want to deploy worker nodes or Fargate to deploy serverless containers
* Use case: if your company is already using Kubernetes on-premises or in another cloud, and wants to migrate to AWS using Kubernetes
* Kubernetes is cloud-agnostic (can be used in any cloud – Azure, GCP…)
* For multiple regions, deploy one EKS cluster per region
* Collect logs and metrics using CloudWatch Container Insights

**Amazon EKS – Diagram**



**Amazon EKS – Node Types**

* Managed Node Groups
  + Creates and manages Nodes (EC2 instances) for you
  + Nodes are part of an ASG managed by EKS
  + Supports On-Demand or Spot Instances
* Self-Managed Nodes
  + Nodes created by you and registered to the EKS cluster and managed by an ASG
  + You can use prebuilt AMI - Amazon EKS Optimized AMI
  + Supports On-Demand or Spot Instances
* AWS Fargate
  + No maintenance required; no nodes managed

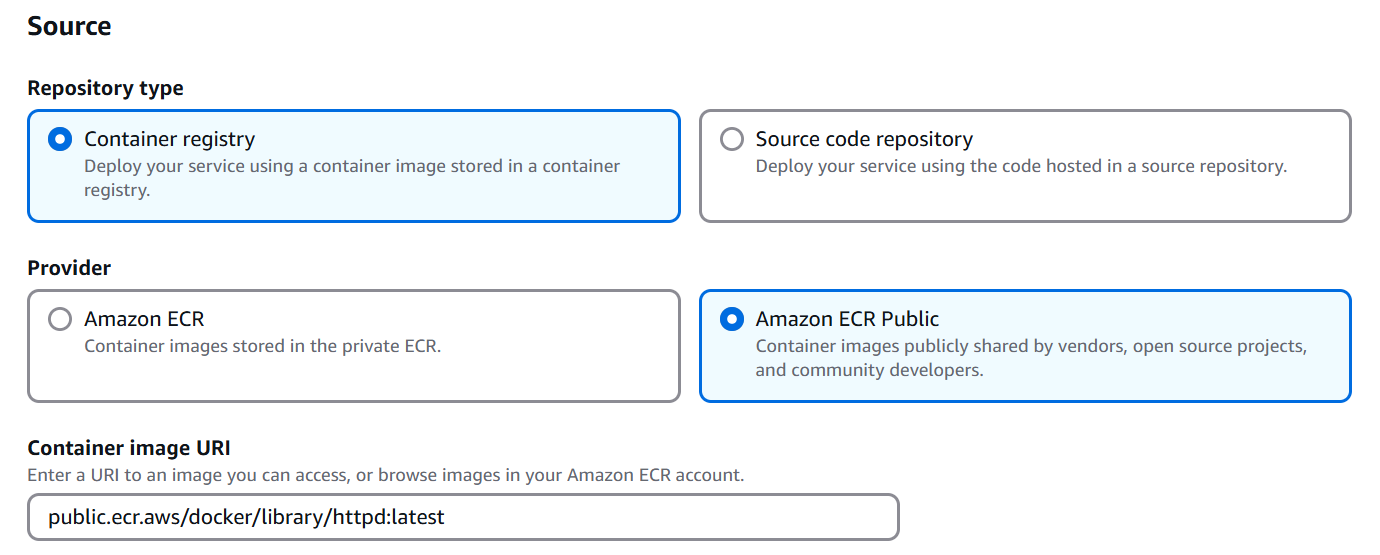
**Amazon EKS – Data Volumes**

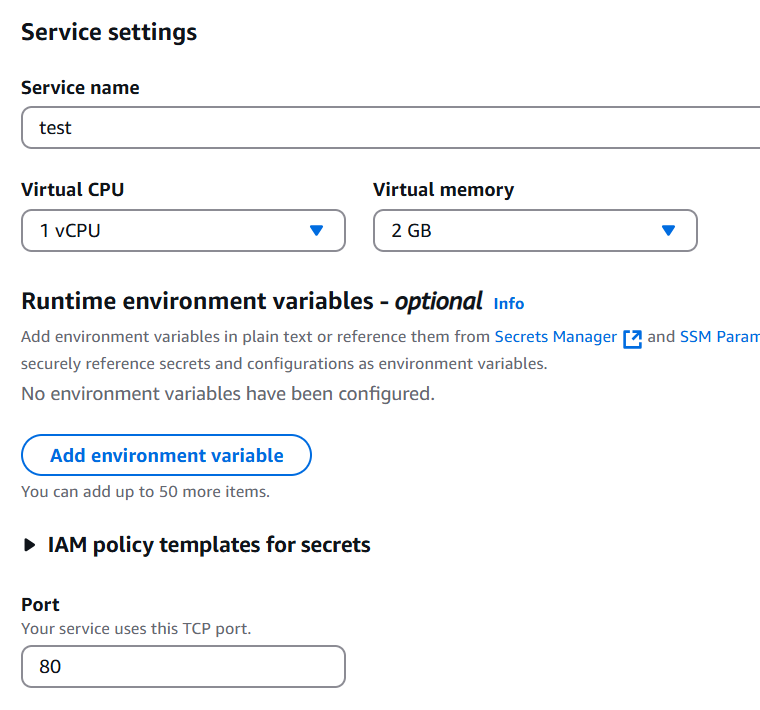
|  |  |
| --- | --- |
| * Need to specify StorageClass manifest on your EKS cluster * Leverages a Container Storage Interface (CSI) compliant driver * Support for… * Amazon EBS * Amazon EFS (works with Fargate) * Amazon FSx for Lustre * Amazon FSx for NetApp ONTAP |  |

## AWS App Runner

|  |  |
| --- | --- |
| * Fully managed service that makes it easy to deploy web applications and APIs at scale * No infrastructure experience required * Start with your source code or container image * You need to configure the settings like, how many VCPS you want, how much memory you want for your containers, if you want autoscaling and a few health checks. And then automatically, it will start building and deploying the web app using the App Runner survey. * Automatic scaling, highly available, load balancer, encryption * VPC access support * Connect to database, cache, and message queue services * Use cases: web apps, APIs, microservices, rapid production deployments | A diagram of a software development process  Description automatically generated |

First we have to select from where it will start building container registry or Git hub





Can configure Auto scalling, health check etc. etc.

I will provide a domain URL to access the service

## AWS App2Container (A2C)

* CLI tool for migrating and modernizing Java and .NET web apps into Docker Containers
* Lift-and-shift apps running in on-premises bare metal, virtual machines and you migrate it to AWS
* Accelerate modernization, no code changes, migrate legacy app. And so, the idea is that you accelerate modernization, but you don't change any code, and you migrate legacy apps this way

to the cloud without making any code changes.

* It will generates CloudFormation templates (compute, network…)
* Will also register generated Docker containers to ECR
* Deploy to ECS, EKS, or App Runner
* Supports pre-built CI/CD pipelines

A screenshot of a computer

AI-generated content may be incorrect.

## Questions

**Top of Form**

**You have multiple Docker-based applications hosted on-premises that you want to migrate to AWS. You don't want to provision or manage any infrastructure; you just want to run your containers on AWS. Which AWS service should you choose?**

* **Elastic Container Service (ECS) in EC2 Launch Mode**
* **Elastic Container Registry (ECR)**
* **AWS Fargate on ECS**

**Bottom of Form**

Ans C AWS Fargate allows you to run your containers on AWS without managing any servers.

**Top of Form**

**Amazon Elastic Container Service (ECS) has two Launch Types: .................. and ..................**

* **Amazon EC2 Launch Type and Fargate Launch Type**
* **Amazon EC2 Launch Type and EKS Launch Type**
* **Fargate Launch Type and EKS Launch Type**

Bottom of Form

Ans A

**Top of Form**

**You have an application hosted on an ECS Cluster (EC2 Launch Type) where you want your ECS tasks to upload files to an S3 bucket. Which IAM Role for your ECS Tasks should you modify?**

* **EC2 Instance Profile**
* **ECS Task Role**

**Bottom of Form**

Ans Correct answer ->ECS Task Role is the IAM Role used by the ECS task itself. Use when your container wants to call other AWS services like S3, SQS, etc.

EC2 Instance Profile is the IAM Role used by the ECS Agent on the EC2 instance to execute ECS-specific actions such as pulling Docker images from ECR and storing the container logs into CloudWatch Logs.

**Top of Form**

**You're planning to migrate a WordPress website running on Docker containers from on-premises to AWS. You have decided to run the application in an ECS Cluster, but you want your docker containers to access the same WordPress website content such as website files, images, videos, etc. What do you recommend to achieve this?**

* **Mount an EFS volume**
* **Mount an EBS volume**
* **Use an EC2 Instance Store**

**Bottom of Form**

Ans EFS volume can be shared between different EC2 instances and different ECS Tasks. It can be used as a persistent multi-AZ shared storage for your containers.

EBS Vol This can be used when all your ECS tasks run on the same EC2 instance. It's not suitable when your tasks are spread across multiple EC2 instances and can't be used with AWS Fargate.

Ec2 Instance store EC2 Instance Store can't be shared between different EC2 instances.

**Top of Form**

**You are deploying an application on an ECS Cluster made of EC2 instances. Currently, the cluster is hosting one application that is issuing API calls to DynamoDB successfully. Upon adding a second application, which issues API calls to S3, you are getting authorization issues. What should you do to resolve the problem and ensure proper security?**

* **Edit the EC2 instance role to add permissions to S3**
* **Create an IAM task role for the new application**
* **Enable the Fargate mode**
* **Edit the S3 bucket policy to allow the ECS task**

**Bottom of Form**

Ans

A-> This would allow the first application to access S3 as well, which is a security risk.

B-> Correct ans

**Top of Form**

**You are migrating your on-premises Docker-based applications to Amazon ECS. You were using Docker Hub Container Image Library as your container image repository. Which is an alternative AWS service which is fully integrated with Amazon ECS?**

* **AWS Fargate**
* **Elastic Container Registry (ECR)**
* **Elastic Kubernetes Service (EKS)**
* **Amazon EC2**

**Bottom of Form**

Ans Amazon ECR is a fully managed container registry that makes it easy to store, manage, share, and deploy your container images. ECR is fully integrated with Amazon ECS, allowing easy retrieval of container images from ECR while managing and running containers using ECS.

**Top of Form**

**Amazon EKS supports the following node types, EXCEPT ………………..**

* **Managed Node Groups**
* **Self-Managed Nodes**
* **AWS Fargate**
* **AWS Lambda**Bottom of Form

Ans D

**Top of Form**

**A developer has a running website and APIs on his local machine using containers and he wants to deploy both of them on AWS. The developer is new to AWS and doesn’t know much about different AWS services. Which of the following AWS services allows the developer to build and deploy the website and the APIs in the easiest way according to AWS best practices?**

* **AWS App Runner**
* **EC2 Instances + Application Load Balancer**
* **Amazon ECS**
* **AWS Fargate**

**Bottom of Form**

Ans A