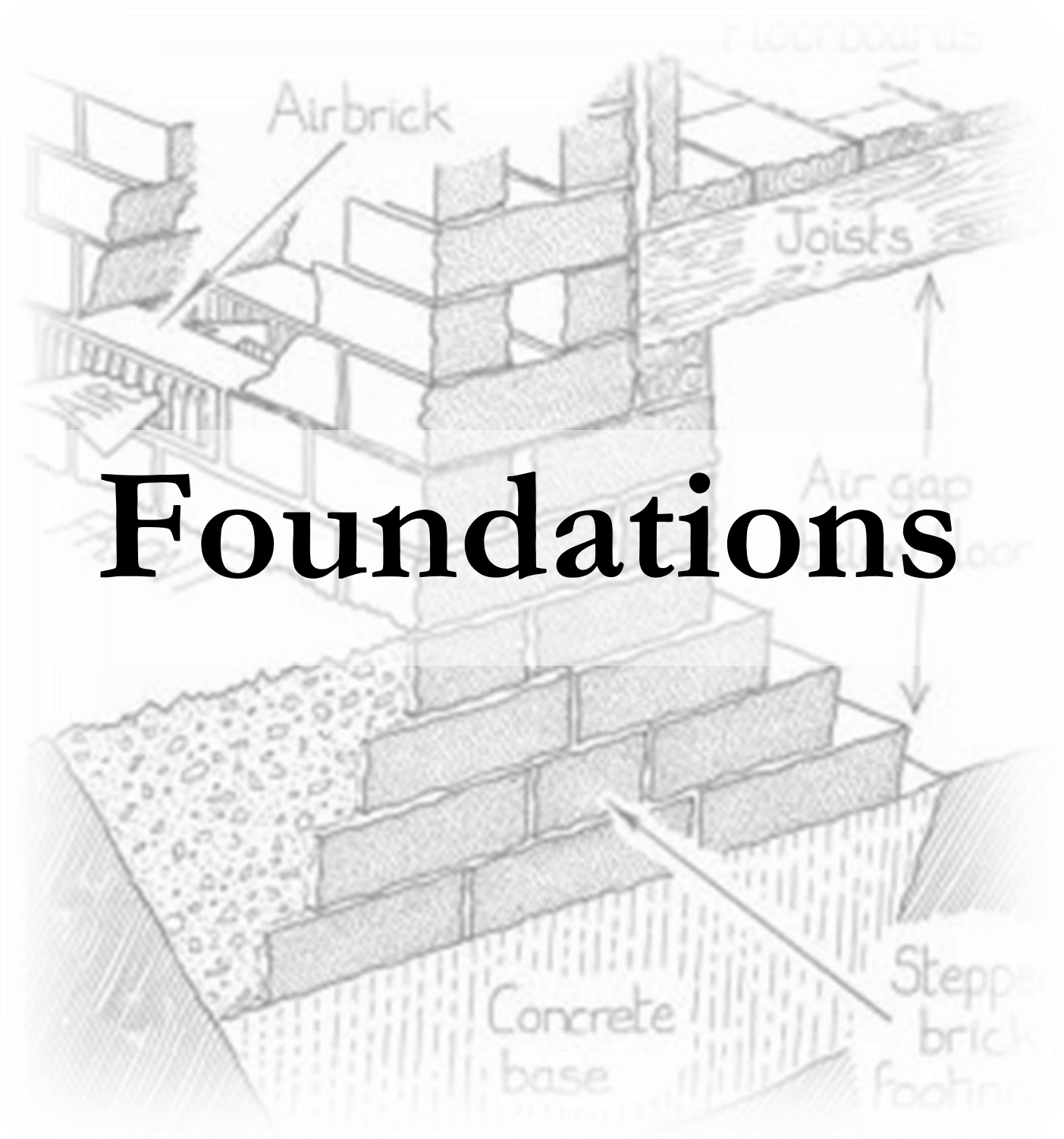


# Foundations



## \*Orchestra tunes up\*

|      |  |
|------|--|
| 1823 | Babbage given cash by British government |
| 1894 | Norbert Wiener born                      |

|             |                            |
|-------------|----------------------------|
| <b>1944</b> | <b>Colossus is working</b> |
| 1949        | EDSAC created              |

|             |  |
|-------------|--|
| <b>1964</b> | <b>IBM's SABRE comes online, and Bezos</b> |
| 1968        | Andy Jassy and Intel born                  |
| 1973        | Ethernet created                           |

|             |                                   |
|-------------|-----------------------------------|
| <b>1974</b> | <b>Xerox PARC Alto introduced</b> |
| 1975        | Microsoft founded                 |
| 1976        | Apple founded                     |
| 1983        | ARPANET switches to TCP/IP        |

|             |                                    |
|-------------|------------------------------------|
| <b>1984</b> | <b>Dell founded</b>                |
| 1989        | Berners-Lee starts the Web at CERN |
| 1990        | Windows 3.0 released               |







|             |  |
|-------------|--|
| <b>1994</b> | <b>“Amazon.com” registered</b>   |
|             | <a href="#">This</a> video starring Kate Bellingham is produced by the BBC |

|      |                        |
|------|------------------------|
| 2001 | Windows XP released    |
| 2004 | The Face book launched |

|             |                              |
|-------------|------------------------------|
| <b>2004</b> | <b>Amazon SQS announced</b>  |
| 2006        | S3 and EC2 announced         |
| 2008        | EBS and CloudFront announced |

|             |                             |
|-------------|-----------------------------|
| <b>2014</b> | <b>AWS Lambda announced</b> |
|-------------|-----------------------------|

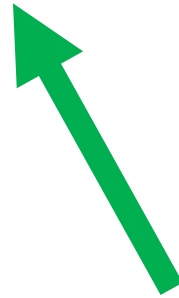
# Five Foundational Years

|      |   |   |
|------|---|---|
| 2004 | <br>SQS        |   |
| 2005 | In the Second Year Jeff Rested. (Also, Mechanical Turk was launched.)                           |   |
| 2006 | <br>S3        | <br>EC2         |
| 2007 | <br>SimpleDB |   |
| 2008 | <br>EBS      | <br>CloudFront |



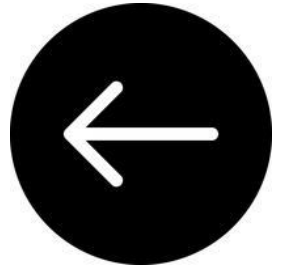


# Appendices





# Simple Queue Service (SQS)

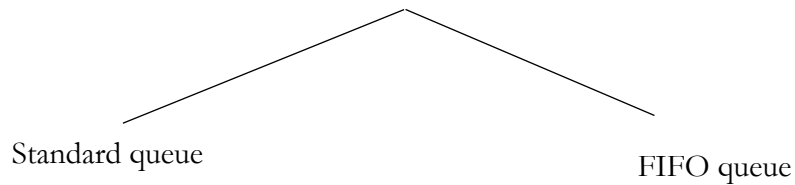


*[RETURN TO CONTENTS](#)*



sqs

## Types of queues



Amazon SQS has a broad distinction between two kinds of queue: Standard and FIFO. They are different in a few ways, not simply the property indicated by the names. However, this is a good starting point: FIFO stands for “first in, first out”. This means that the item entered into the queue first comes out first, like people cycling into a tunnel and coming out the same order they were in when they entered.

This is not what happens in busy elevators. In these, those who entered last (who are next to the doors) exit first. This would be more like “first in, last out” (FILO).

### Default retention period

The default retention period for an SQS queue is 14 days.

### My holiday in Quebec

I once went on a quaint holiday to Quebec (Canada), for the usually time, you know, two weeks. Before I left for the airport, I set up an SQS queue. There was just one item in it. When I got back from my holiday two weeks later, the item was still there!

- The **MAXIMUM** retention period for a queue is *14 days*.
- The **DEFAULT** retention period is just *4 days*.



Elevators are first-in-*last*-out!



## What is the Visibility Timeout (VT)?



This expression gets used, which is “visibility timeout”. I’m no fan of it.

When a reader picks up a message in a queue, it becomes invisible. The amount of time the message is invisible is the



“visibility timeout”. In my opinion, it should be called the ***IN***visibility timeout, because it is the amount of time the message is invisible, after all.

# Visibility timeout

If a job is processed within the “visibility timeout”, then the message will be deleted. If a job is *not* processed within the VT, the message will become visible again. Think of the VT as a kind of trial period for the message being properly deleted. VT jokingly stands for *Vanish Time*.

The maximum “visibility timeout” for an SQS message is 12 hours! So, it’s possible that a message disappears invisible at midnight, continues to be absent throughout the quiet, early hours of the morning—and finally, after it turns out it cannot be processed, it becomes visible again at lunchtime.

# What is the difference between “long polling” and “short polling”?

The truth is, there are a number of features that distinguish these approaches (in AWS, at least). But let's start with the most obvious point of difference. Long polling waits a LONG time between each “polling” of the queue. Short polling waits only a SHORT time between each “polling” of the queue. This explains the names. When I say “polling of the queue” I refer to the

# SP

Speak, Pronto!

Splashing Pennies

Subset Probing

Short Polling

# LP

Let's P rotect (our cash!)

Low P robability (of being  
wrong – we have *certainty*)

# What on earth are “message timers”?

## Amazon SQS message timers

[PDF](#) | [RSS](#)

Message timers let you specify an initial invisibility period for a message added to a queue. For example, if you send a message with a 45-second timer, the message isn't visible to consumers for its first 45 seconds in the queue. The default (minimum) delay for a message is 0 seconds. The maximum is 15 minutes. For information about sending messages with timers using the console, see [Sending messages to a queue \(console\)](#).

### Note

FIFO queues don't support timers on individual messages.

To set a delay period on *an entire queue*, rather than on individual messages, use [delay queues](#). A message timer setting for an individual message overrides any `DelaySeconds` value on an Amazon SQS delay queue.

# ApproximateNumberOfMessages

### 13. QUESTION

A new application will run across multiple Amazon ECS tasks. Front-end application logic will process data and then pass that data to a back-end ECS task to perform further processing and write the data to a datastore. The Architect would like to reduce interdependencies so failures do not impact other components.

Which solution should the Architect use?

- ☒ Create an Amazon SQS queue and configure the front-end to add messages to the queue and the back-end to poll the queue for messages
- ☐ Create an Amazon Kinesis Firehose delivery stream and configure the front-end to add data to the stream and the back-end to read data from the stream
- ☐ Create an Amazon Kinesis Firehose delivery stream that delivers data to an Amazon S3 bucket, configure the front-end to write data to the stream and the back-end to read data from Amazon S3
- ☐ Create an Amazon SQS queue that pushes messages to the back-end. Configure the front-end to add messages to the queue

Correct

Correct

#### Explanation:

This is a good use case for Amazon SQS. SQS is a service that is used for decoupling applications, thus reducing interdependencies, through a message bus. The front-end application can place messages on the queue and the back-end can then poll the queue for new messages. Please remember that Amazon SQS is pull-based (polling) not push-based (use SNS for push-based).

**CORRECT:** "Create an Amazon SQS queue and configure the front-end to add messages to the queue and the back-end to poll the queue for messages" is the correct answer.

**INCORRECT:** "Create an Amazon Kinesis Firehose delivery stream and configure the front-end to add data to the stream and the back-end to read data from the stream" is incorrect. Amazon Kinesis Firehose is used for streaming data. With Firehose the data is immediately loaded into a destination that can be Amazon S3, RedShift, Elasticsearch, or Splunk. This is not an ideal use case for Firehose as this is not streaming data and there is no need to load data into an additional AWS service.

**INCORRECT:** "Create an Amazon Kinesis Firehose delivery stream that delivers data to an Amazon S3 bucket, configure the front-end to write data to the stream and the back-end to read data from Amazon S3" is incorrect as per the previous explanation.

**INCORRECT:** "Create an Amazon SQS queue that pushes messages to the back-end. Configure the front-end to add messages to the queue" is incorrect as SQS is pull-based, not push-based. EC2 instances must poll the queue to find jobs to process.

## 3. QUESTION

A solutions architect is designing an application on AWS. The compute layer will run in parallel across EC2 instances. The compute layer should scale based on the number of jobs to be processed. The compute layer is stateless. The solutions architect must ensure that the application is loosely coupled and the job items are durably stored.

Which design should the solutions architect use?

- ☐ Create an Amazon SQS queue to hold the jobs that need to be processed. Create an Amazon EC2 Auto Scaling group for the compute application. Set the scaling policy for the Auto Scaling group to add and remove nodes based on network usage
- ☐ Create an Amazon SNS topic to send the jobs that need to be processed. Create an Amazon EC2 Auto Scaling group for the compute application. Set the scaling policy for the Auto Scaling group to add and remove nodes based on CPU usage
- ☐ Create an Amazon SNS topic to send the jobs that need to be processed. Create an Amazon EC2 Auto Scaling group for the compute application. Set the scaling policy for the Auto Scaling group to add and remove nodes based on the number of messages published to the SNS topic
- ☒ Create an Amazon SQS queue to hold the jobs that need to be processed. Create an Amazon EC2 Auto Scaling group for the compute application. Set the scaling policy for the Auto Scaling group to add and remove nodes based on the number of items in the SQS queue

Correct

## Explanation:

In this case we need to find a durable and loosely coupled solution for storing jobs. Amazon SQS is ideal for this use case and can be configured to use dynamic scaling based on the number of jobs waiting in the queue.

To configure this scaling you can use the *backlog per instance* metric with the target value being the *acceptable backlog per instance* to maintain. You can calculate these numbers as follows:

- **Backlog per instance:** To calculate your backlog per instance, start with the `ApproximateNumberOfMessages` queue attribute to determine the length of the SQS queue (number of messages available for retrieval from the queue). Divide that number by the fleet's running capacity, which for an Auto Scaling group is the number of instances in the `InService` state, to get the backlog per instance.
- **Acceptable backlog per instance:** To calculate your target value, first determine what your application can accept in terms of latency. Then, take the acceptable latency value and divide it by the average time that an EC2 instance takes to process a message.

This solution will scale EC2 instances using Auto Scaling based on the number of jobs waiting in the SQS queue.

**CORRECT:** "Create an Amazon SQS queue to hold the jobs that need to be processed. Create an Amazon EC2 Auto Scaling group for the compute application. Set the scaling policy for the Auto Scaling group to add and remove nodes based on the number of items in the SQS queue" is the correct answer.

**INCORRECT:** "Create an Amazon SQS queue to hold the jobs that need to be processed. Create an Amazon EC2 Auto Scaling group for the compute application. Set the scaling policy for the Auto Scaling group to add and remove nodes based on network usage" is incorrect as scaling on network usage does not relate to the number of jobs waiting to be processed.

**INCORRECT:** "Create an Amazon SNS topic to send the jobs that need to be processed. Create an Amazon EC2 Auto Scaling group for the compute application. Set the scaling policy for the Auto Scaling group to add and remove nodes based on CPU usage" is incorrect. Amazon SNS is a notification service so it delivers notifications to subscribers. It does store data durably but is less suitable than SQS for this use case. Scaling on CPU usage is not the best solution as it does not relate to the number of jobs waiting to be processed.

**INCORRECT:** "Create an Amazon SNS topic to send the jobs that need to be processed. Create an Amazon EC2 Auto Scaling group for the compute application. Set the scaling policy for the Auto Scaling group to add and remove nodes based on the number of messages published to the SNS topic" is incorrect. Amazon SNS is a notification service so it delivers notifications to subscribers. It does store data durably but is less suitable than SQS for this use case. Scaling on the number of notifications in SNS is not possible.

**References:**

QUESTION 8 OF 65

8. QUESTION

A retail company with many stores and warehouses is implementing IoT sensors to gather monitoring data from devices in each location. The data will be sent to AWS in real time. A solutions architect must provide a solution for ensuring events are received in order for each device and ensure that data is saved for future processing.

Which solution would be MOST efficient?

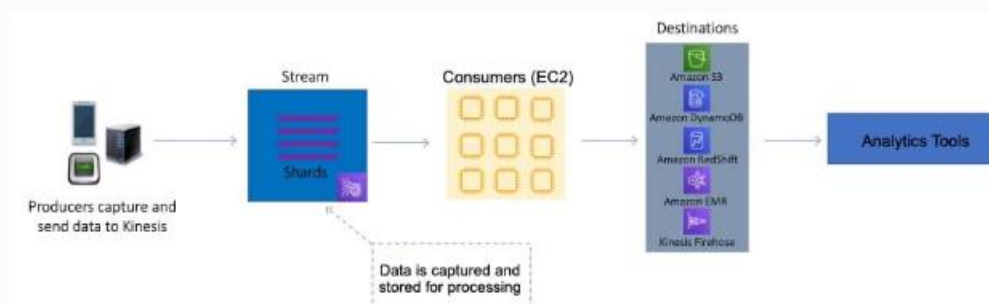
- ☐ Use an Amazon SQS standard queue for real-time events with one queue for each device. Trigger an AWS Lambda function from the SQS queue to save data to Amazon S3
- ☐ Use Amazon Kinesis Data Streams for real-time events with a partition key for each device. Use Amazon Kinesis Data Firehose to save data to Amazon S3
- ☐ Use an Amazon SQS FIFO queue for real-time events with one queue for each device. Trigger an AWS Lambda function for the SQS queue to save data to Amazon EFS
- ☐ Use Amazon Kinesis Data Streams for real-time events with a shard for each device. Use Amazon Kinesis Data Firehose to save data to Amazon EBS

Correct

Explanation:

Amazon Kinesis Data Streams collect and process data in real time. A *Kinesis data stream* is a set of **shards**. Each shard has a sequence of data records. Each data record has a **sequence number** that is assigned by Kinesis Data Streams. A *shard* is a uniquely identified sequence of data records in a stream.

A *partition key* is used to group data by shard within a stream. Kinesis Data Streams segregates the data records belonging to a stream into multiple shards. It uses the partition key that is associated with each data record to determine which shard a given data record belongs to.



For this scenario, the solutions architect can use a partition key for each device. This will ensure the records for that device are grouped by shard and the shard



Data is captured and stored for processing

For this scenario, the solutions architect can use a partition key for each device. This will ensure the records for that device are grouped by shard and the shard will ensure ordering. Amazon S3 is a valid destination for saving the data records.

**CORRECT:** "Use Amazon Kinesis Data Streams for real-time events with a partition key for each device. Use Amazon Kinesis Data Firehose to save data to Amazon S3" is the correct answer.

**INCORRECT:** "Use Amazon Kinesis Data Streams for real-time events with a shard for each device. Use Amazon Kinesis Data Firehose to save data to Amazon EBS" is incorrect as you cannot save data to EBS from Kinesis.

**INCORRECT:** "Use an Amazon SQS FIFO queue for real-time events with one queue for each device. Trigger an AWS Lambda function for the SQS queue to save data to Amazon EFS" is incorrect as SQS is not the most efficient service for streaming, real time data.

**INCORRECT:** "Use an Amazon SQS standard queue for real-time events with one queue for each device. Trigger an AWS Lambda function from the SQS queue to save data to Amazon S3" is incorrect as SQS is not the most efficient service for streaming, real time data.

References:

<https://docs.aws.amazon.com/streams/latest/dev/key-concepts.html>

#### 6. QUESTION

There are two applications in a company: a sender application that sends messages containing payloads, and a processing application that receives messages containing payloads. The company wants to implement an AWS service to handle messages between these two different applications. The sender application sends on average 1,000 messages each hour and the messages depending on the type sometimes take up to 2 days to be processed. If the messages fail to process, they must be retained so that they do not impact the processing of any remaining messages.

Which solution meets these requirements and is the MOST operationally efficient?

- ☐ Receive the messages from the sender application using an Amazon Kinesis data stream. Utilize the Kinesis Client Library (KCL) to integrate the processing application.
- ☐ Subscribe the processing application to an Amazon Simple Notification Service (Amazon SNS) topic to receive notifications. Write to the SNS topic using the sender application.
- ☐ Set up a Redis database on Amazon EC2. Configure the instance to be used by both applications. The messages should be stored, processed, and deleted, respectively.
- ☒ Provide an Amazon Simple Queue Service (Amazon SQS) queue for the sender and processor applications. Set up a dead-letter queue to collect failed messages.

Amazon Simple Queue Service (SQS) is a fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications. SQS eliminates the complexity and overhead associated with managing and operating message-oriented middleware and empowers developers to focus on differentiating work.

**CORRECT:** "Provide an Amazon Simple Queue Service (Amazon SQS) queue for the sender and processor applications. Set up a dead-letter queue to collect failed messages" is the correct answer (as explained above.)

**INCORRECT:** "Set up a Redis database on Amazon EC2. Configure the instance to be used by both applications. The messages should be stored, processed, and deleted, respectively" is incorrect, as the most operationally efficient way is to use the managed service Amazon SQS.

**INCORRECT:** "Receive the messages from the sender application using an Amazon Kinesis data stream. Utilize the Kinesis Client Library (KCL) to integrate the processing application" is incorrect, as the most operationally efficient way is to use the managed service Amazon SQS

**INCORRECT:** "Subscribe the processing application to an Amazon Simple Notification Service (Amazon SNS) topic to receive notifications. Write to the SNS topic using the sender application" is incorrect as Amazon SNS is not a queuing service, but a pub-sub one to many notification service and cannot be used as a queue.

#### 1. QUESTION

A company provides a REST-based interface to an application that allows a partner company to send data in near-real time. The application then processes the data that is received and stores it for later analysis. The application runs on Amazon EC2 instances.

The partner company has received many 503 Service Unavailable Errors when sending data to the application and the compute capacity reaches its limits and is unable to process requests when spikes in data volume occur.

Which design should a Solutions Architect implement to improve scalability?

- ☐ Use Amazon SQS to ingest the data. Configure the EC2 instances to process messages from the SQS queue.
- ☐ Use Amazon API Gateway in front of the existing application. Create a usage plan with a quota limit for the partner company.
- ☒ Use Amazon Kinesis Data Streams to ingest the data. Process the data using AWS Lambda functions.
- ☐ Use Amazon SNS to ingest the data and trigger AWS Lambda functions to process the data in near-real time.

Correct

**Explanation:**

Amazon Kinesis enables you to ingest, buffer, and process streaming data in real-time. Kinesis can handle any amount of streaming data and process data from hundreds of thousands of sources with very low latencies. This is an ideal solution for data ingestion.

To ensure the compute layer can scale to process increasing workloads, the EC2 instances should be replaced by AWS Lambda functions. Lambda can scale seamlessly by running multiple executions in parallel.

**CORRECT:** "Use Amazon Kinesis Data Streams to ingest the data. Process the data using AWS Lambda functions" is the correct answer.

**INCORRECT:** "Use Amazon API Gateway in front of the existing application. Create a usage plan with a quota limit for the partner company" is incorrect. A usage plan will limit the amount of data that is received and cause more errors to be received by the partner company.

**INCORRECT:** "Use Amazon SQS to ingest the data. Configure the EC2 instances to process messages from the SQS queue" is incorrect. Amazon Kinesis Data Streams should be used for near-real time or real-time use cases instead of Amazon SQS.

**INCORRECT:** "Use Amazon SNS to ingest the data and trigger AWS Lambda functions to process the data in near-real time" is incorrect. SNS is not a near-real time solution for data ingestion. SNS is used for sending notifications.

A company is working with a strategic partner that has an application that must be able to send messages to one of the company's Amazon SQS queues. The partner company has its own AWS account.

How can a Solutions Architect provide least privilege access to the partner?

- ☐ Update the permission policy on the SQS queue to grant all permissions to the partner's AWS account.
- ☐ Update the permission policy on the SQS queue to grant the `sqs:SendMessage` permission to the partner's AWS account.
- ☐ Create a cross-account role with access to all SQS queues and use the partner's AWS account in the trust document for the role.
- ☐ Create a user account that and grant the `sqs:SendMessage` permission for Amazon SQS. Share the credentials with the partner company.

Correct

**Explanation:**

Amazon SQS supports resource-based policies. The best way to grant the permissions using the principle of least privilege is to use a resource-based policy attached to the SQS queue that grants the partner company's AWS account the `sqs:SendMessage` privilege.

The following policy is an example of how this could be configured:

```
{
  "Version": "2012-10-17",
  "Id": "Queue1_Policy_UUID",
  "Statement": [{
    "Sid": "Queue1_SendMessage",
    "Effect": "Allow",
    "Principal": {
      "AWS": [
        "111122223333"
      ]
    },
    "Action": "sqs:SendMessage",
    "Resource": "arn:aws:sqs:us-east-2:444455556666:queue1"
  }]
}
```

**CORRECT:** "Update the permission policy on the SQS queue to grant the sqs:SendMessage permission to the partner's AWS account" is the correct answer.

**INCORRECT:** "Create a user account that and grant the sqs:SendMessage permission for Amazon SQS. Share the credentials with the partner company" is incorrect. This would provide the permissions for all SQS queues, not just the queue the partner company should be able to access.

**INCORRECT:** "Create a cross-account role with access to all SQS queues and use the partner's AWS account in the trust document for the role" is incorrect. This would provide access to all SQS queues and the partner company should only be able to access one SQS queue.

**INCORRECT:** "Update the permission policy on the SQS queue to grant all permissions to the partner's AWS account" is incorrect. This provides too many permissions; the partner company only needs to send messages to the queue.

#### 5. QUESTION

An eCommerce application consists of three tiers. The web tier includes EC2 instances behind an Application Load balancer, the middle tier uses EC2 instances and an Amazon SQS queue to process orders, and the database tier consists of an Auto Scaling DynamoDB table. During busy periods customers have complained about delays in the processing of orders. A Solutions Architect has been tasked with reducing processing times.

Which action will be MOST effective in accomplishing this requirement?

- ☐ Add an Amazon CloudFront distribution with a custom origin to cache the responses for the web tier.
- ☐ Use Amazon DynamoDB Accelerator (DAX) in front of the DynamoDB backend tier.
- ☒ Use Amazon EC2 Auto Scaling to scale out the middle tier instances based on the SQS queue depth.
- ☐ Replace the Amazon SQS queue with Amazon Kinesis Data Firehose.



**Explanation:**

The most likely cause of the processing delays is insufficient instances in the middle tier where the order processing takes place. The most effective solution to reduce processing times in this case is to scale based on the backlog per instance (number of messages in the SQS queue) as this reflects the amount of work that needs to be done.

**CORRECT:** "Use Amazon EC2 Auto Scaling to scale out the middle tier instances based on the SQS queue depth" is the correct answer.

**INCORRECT:** "Replace the Amazon SQS queue with Amazon Kinesis Data Firehose" is incorrect. The issue is not the efficiency of queuing messages but the processing of the messages. In this case scaling the EC2 instances to reflect the workload is a better solution.

**INCORRECT:** "Use Amazon DynamoDB Accelerator (DAX) in front of the DynamoDB backend tier" is incorrect. The DynamoDB table is configured with Auto Scaling so this is not likely to be the bottleneck in order processing.

**INCORRECT:** "Add an Amazon CloudFront distribution with a custom origin to cache the responses for the web tier" is incorrect. This will cache media files to speed up web response times but not order processing times as they take place in the middle tier.

**References:**

### 30. QUESTION

An application uses Amazon EC2 instances to retrieve messages from an Amazon SQS queue. The EC2 instances have an instance profile assigned that uses an IAM role to provide permissions to the queue. A security engineer has been asked to investigate why the instances are not able to retrieve messages. The solution should follow the principle of least privilege.

What actions should be taken to identify the cause of the? (Select TWO.)

- ☐ Check if an Amazon SQS policy explicitly denies access to the IAM role used by the instances.
- ☒ Check the configuration of the IAM role attached to the instance profile to ensure it has sufficient permissions.
- ☒ Check that the AmazonSQSFullAccess managed policy is attached to the IAM role used by the instances.
- ☐ Check if server-side encryption is enabled using an AWS KMS managed key.
- ☐ Check that a policy is attached to the IAM role used by the instances that grants the "sqs:AddPermission" permission.

Incorrect

Explanation:

There are two ways to give your users permissions to your Amazon SQS resources: using the Amazon SQS policy system and using the IAM policy system. You can use one or the other, or both.

The security engineer should check that the IAM role has the minimum permissions required to receive messages from the queue. The SQS policy system should also be checked to ensure that more restrictive permissions are not assigned there.

**CORRECT:** "Check the configuration of the IAM role attached to the instance profile to ensure it has sufficient permissions" is a correct answer (as explained above.)

**CORRECT:** "Check if an Amazon SQS policy explicitly denies access to the IAM role used by the instances" is also a correct answer (as explained above.)

**INCORRECT:** "Check that the AmazonSQSFullAccess managed policy is attached to the IAM role used by the instances" is incorrect.

This managed policy provides more permissions than are required by the instances and does not follow the principle of least privilege.

**INCORRECT:** "Check that a policy is attached to the IAM role used by the instances that grants the "sqs:AddPermission" permission" is incorrect.

This permission is not required to receive messages from an Amazon SQS queue, it allows sharing access to the queue.

**INCORRECT:** "Check if server-side encryption is enabled using an AWS KMS managed key" is incorrect.

Server-side encryption is not a requirement for receiving messages from the queue.

References:

<https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-using-identity-based-policies.html>

#### 17. QUESTION

An eCommerce company are running a promotional campaign and expect a large volume of user sign-ups on a web page that collects user information and preferences. The website runs on Amazon EC2 instances and uses an Amazon RDS for PostgreSQL DB instance. The volume of traffic is expected to be high and may be unpredictable with several spikes in activity. The traffic will result in a large number of database writes.

A solutions architect needs to build a solution that does not change the underlying data model and ensures that submissions are not dropped before they are committed to the database.

Which solution meets these requirements?

- ☐ Create an Amazon ElastiCache for Memcached cluster in front of the existing database instance to increase write performance.
- ☒ Create an Amazon SQS queue and decouple the application and database layers. Configure an AWS Lambda function to write items from the queue into the database.
- ☐ Migrate to Amazon DynamoDB and manage throughput capacity with automatic scaling.
- ☐ Use scheduled scaling to scale up the existing DB instance immediately before the event and then automatically scale down afterwards.

Correct

#### Explanation:

In order to avoid dropping records decoupling the application and database layers is the best solution for this specific scenario. This works as the application does not require synchronous responses (it's just writing the user information to the DB). The alternative is to increase write capacity on the database instance but as the traffic is unpredictable it's hard to know how much capacity to provision which could lead to underperformance or higher than necessary costs.

**CORRECT:** "Create an Amazon SQS queue and decouple the application and database layers. Configure an AWS Lambda function to write items from the queue into the database" is the correct answer.

**INCORRECT:** "Use scheduled scaling to scale up the existing DB instance immediately before the event and then automatically scale down afterwards" is incorrect. You cannot schedule RDS database instances to scale up or down.

**INCORRECT:** "Migrate to Amazon DynamoDB and manage throughput capacity with automatic scaling" is incorrect. DynamoDB is a NoSQL (non-relational) database whereas RDS is a relational database. This solution would change the underlying data model and is therefore not an option.

**INCORRECT:** "Create an Amazon ElastiCache for Memcached cluster in front of the existing database instance to increase write performance" is incorrect. ElastiCache is used for improving read performance, not write performance.



# TPN

1. ***Phenomenon1*** – the tendency of X to Y.
2. ***Phen2*** – the tendency of X to Y.
3. ***Phen3*** – the tendency of X to Y.
4. ***Phen4*** – the tendency of X to Y.
5. ***Phen5*** – the tendency of X to Y.
6. ***Phen6*** – the tendency of X to Y.
7. ***Phen7*** – the tendency of X to Y.
8. ***Phen8*** – the tendency of X to Y.
9. ***Phen9*** – the tendency of X to Y.
10. ***Phen10*** – the tendency of X to Y.

# Glossary

# Bibliography

- I. Official
- II. Unofficial
- III. Critical
- IV. General

## I. Official

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

### **[Wood 2020]**

Wood, Julian (2020). Choosing Events, Queues, Topics, and Streams in your Serverless Application. Available at:  
[https://www.youtube.com/watch?v=d9Jb1WKCLd8&ab\\_channel=AWSOnlineTechTalks](https://www.youtube.com/watch?v=d9Jb1WKCLd8&ab_channel=AWSOnlineTechTalks)

### **[AWS 2004]**

“Introducing the Amazon Simple Queue Service”. AWS.  
3<sup>rd</sup> Nov 2004. Available at:  
<https://aws.amazon.com/about-aws/whats-new/2004/11/03/introducing-the-amazon-simple-queue-service/>

### **[Barr 1]**

Barr, Jeff (2006). “Amazon Simple Queue Service Released”. *AWS News Blog*. 13<sup>th</sup> July 2006. Available at: [https://aws.amazon.com/blogs/aws/amazon\\_simple\\_q/](https://aws.amazon.com/blogs/aws/amazon_simple_q/)

## II. Unofficial

### [Levitt 2005]

Levitt, Jason (2005). Fun with Amazon’s simple queue service. Available at: <https://www.xml.com/pub/a/2005/01/05/sqs.html>

### [Barr 2]

Barr, Jeff (year). My first 12 years at Amazon.com. Jeff-barr.com. Available at: <http://jeff-barr.com/2014/08/19/my-first-12-years-at-amazon-dot-com/>  
[https://www.youtube.com/watch?v=s-E\\_V5Xyg6k&ab\\_channel=Parleys](https://www.youtube.com/watch?v=s-E_V5Xyg6k&ab_channel=Parleys)

## III. Critical

## IV. General

Elastic message queues. Ahmed El Rheddane and Noel De Palma

MSMQ is dead. David Boike.

## Chapter 18 – Distributed computing – models and methods. Leslie Lamport and Nancy Lynch.

### [Wikipedia 1]

Message queue. Wikipedia. Available at:  
[https://en.wikipedia.org/wiki/Message\\_queue](https://en.wikipedia.org/wiki/Message_queue)

### [Bajantri 1986]

Bajantri, M. and David B Skillicorn (1986). A fast multiprocessor message passing implementation. Information Processing Letters, 24 (6): 381-389. Available at:  
<<https://www.sciencedirect.com/science/article/abs/pii/0020019087901153>>

### [Christopher 1988]

Christopher, Thomas (1988). Message driven computing and its relationship to actors. OOPSLA/ECOOP '88: Proceedings of the 1988 ACM SIGPLAN workshop on Object-based concurrent programming. <https://doi.org/10.1145/67386.67405>.

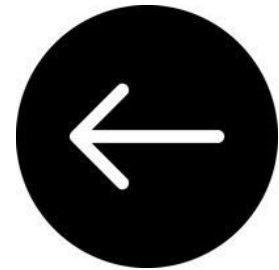
### [Slimmon 2016]

Slimmon, Dan (2016). The most important thing to understand about queues. Dan Slimmon [Blog]. Available at:  
<<https://blog.danslimmon.com/2016/08/26/the-most-important-thing-to-understand-about-queues/>>

<https://www.allaboutlean.com/fifo-benefits/>

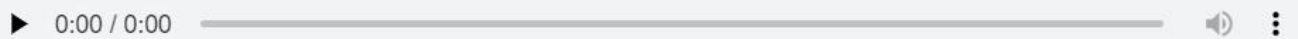
<https://www.infoq.com/news/2020/07/Core-MSMS/>

# Mechanical Turk



## Amazon's Mechanical Turk: The First Three Weeks

by [Jeff Barr](#) | on 21 NOV 2005 | [Permalink](#) | [Share](#)



Voiced by [Amazon Polly](#)

Things have been even busier than usual here at Amazon's headquarters in Seattle.

On the evening of Wednesday, November 2nd, we announced that Amazon's [Mechanical Turk](#) was ready for beta testing.

Momentum for most beta tests builds quietly at first. A few users and developers put a toe or two into the water, give it a try, and then start to spread the word via blog posts, newsletters, and private emails.

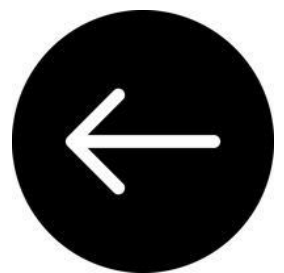
After a while some sort of critical mass is achieved, and information about the new product is seemingly to be found everywhere that you look. The servers are humming along, the bloggers are talking, and there's a real sense of action and excitement. For some products, getting to this point can take weeks or even months of concerted effort.

AWS's list of tasks that require root user credentials fails to include Mechanical Turk which absolutely does require it. Man, the company's attention to detail is being distracted by what I can only assume is AI. This grows concerning.

---

Email from Quinn on 22<sup>nd</sup> April 2024

# Simple Storage Service (S3)

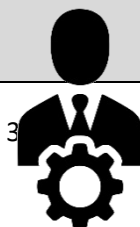


[\*RETURN TO CONTENTS\*](#)

This topic is so large that it must be dealt with in five modules.

|  |  |  |  |
|--|--|--|--|
|  |  |  |  |
|--|--|--|--|

|  |  |                      |   |
|--|--|----------------------|---|
|  | <b>MODULE ONE - Fundamentals</b>                   |                      |   |
|  |  |                      | The difference between object and block storage |
|  |  |                      | The storage classes on S3                       |
|  |  |                      |   |
|  | <b>MODULE TWO - Working with objects</b>           |                      |   |
|  |  | Working with objects |   |
|  |  |                      |   |
|  |  |                      | <b>Multipart upload</b>                         |
|  |  |                      | Copying objects                                 |
|  |  |                      | Downloading an object                           |
|  |  |                      | Checking object integrity                       |
|  |  |                      | Deleting objects                                |
|  |  |                      | Organizing and lifecycle rules                  |
|  |  |                      | <b>Using presigned URLs</b>                     |
|  |  |                      |   |
|  |  |                      |   |
|  | <b>MODULE THREE</b> – Working with buckets         |                      |   |
|  |  | Working with buckets |   |
|  |  |                      | Topic 1   |
|  |  |                      | <b>Transfer acceleration</b>                    |
|  |  |                      |   |
|  |  |                      |   |
|  |  |                      |   |
|  |  |                      |   |
|  |  |                      |   |
|  | <b>MODULE FOUR</b> – Security and managing storage |                      |   |

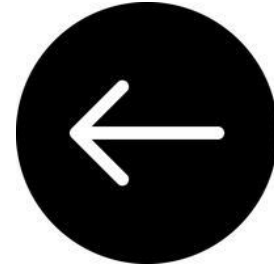




|  |  |    |  |
|--|--|----|--|
|  |  |    |  |
|  |  |    | Data encryption                          |
|  |  |    |  |
|  |  |    |  |
|  |  |    |  |
|  |  |    |  |
|  |  |    | Managing storage                         |
|  |  | 1  | Using versioning in S3 buckets           |
|  |  |    | Configuring MFA Delete                   |
|  |  | 2  | AWS Backup                               |
|  |  | 3  | Working with archived objects            |
|  |  | 4  | Using S3 Object Lock                     |
|  |  | 5  | Using S3 Storage classes                 |
|  |  | 6  | S3 Intelligent tiering                   |
|  |  | 7  | Managing your storage lifecycle          |
|  |  | 8  | Amazon S3 inventory                      |
|  |  | 9  | Replicating objects                      |
|  |  |    | <a href="#">Cross-region replication</a> |
|  |  | 10 | Using object tags                        |
|  |  | 11 | Billing and usage reporting              |
|  |  | 12 | <a href="#">Using S3 Select</a>          |
|  |  | 13 | Using Batch Operations                   |
|  |  |    |  |
|  |  |    |  |
|  |  |    | <b>MODULE 5 – Monitoring, analytics</b>  |
|  |  |    | Monitoring Amazon S3                     |
|  |  |    | Using analytics and insights             |
|  |  | 1  | Storage class analysis                   |
|  |  | 2  | S3 Storage lens                          |
|  |  | 3  | Tracing requests using X-ray             |
|  |  |    |  |
|  |  |    | Logging                                  |



# Module\_1 - Fundamentals



## The *appearance* of structure

### Amazon S3 Reduced Redundancy Storage

Reduced Redundancy Storage (RRS) is an Amazon S3 storage option that enables customers to store noncritical, reproducible data at lower levels of redundancy than Amazon S3's standard storage. It provides a highly available solution for distributing or sharing content that is durably stored elsewhere, or for storing thumbnails, transcoded media, or other processed data that can be easily reproduced. The RRS option stores objects on multiple devices across multiple facilities, providing 400 times the durability of a typical disk drive, but does not replicate objects as many times as standard Amazon S3 storage.

Reduced Redundancy Storage is:

- Backed with the [Amazon S3 Service Level Agreement](#) for availability.
- Designed to provide 99.99% durability and 99.99% availability of objects over a given year. This durability level corresponds to an average annual expected loss of 0.01% of objects.
- Designed to sustain the loss of data in a single facility.

#### Get Started with AWS Today

Try Amazon S3 for Free

AWS Free Tier includes 5GB storage, 20,000 Get Requests, and 2,000 Put Requests with Amazon S3.

[View AWS Free Tier Details »](#)

Figure 1 Snapshot of web page from 11th Sept 202

## Expressions:

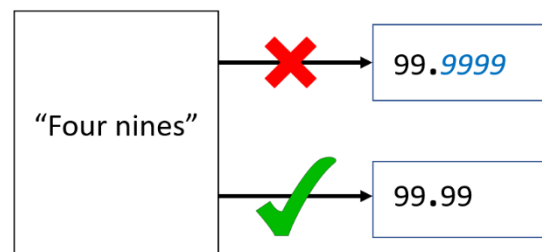
“Four nines”

“Five nines”

“Eleven nines”

The **E**xpressions

**E**ncompass **E**verything



QUESTION 12 OF 12

### 12. QUESTION

A team are planning to run analytics jobs on log files each day and require a storage solution. The size and number of logs is unknown and data will persist for 24 hours only.

What is the MOST cost-effective solution?

☐ Amazon S3 Glacier Deep Archive

☒ Amazon S3 Standard

Incorrect

Explanation:

S3 standard is the best choice in this scenario for a short term storage solution. In this case the size and number of logs is unknown and it would be difficult to fully assess the access patterns at this stage. Therefore, using S3 standard is best as it is cost-effective, provides immediate access, and there are no retrieval fees or minimum capacity charge per object.

**CORRECT:** "Amazon S3 Standard" is the correct answer.

**INCORRECT:** "Amazon S3 Intelligent-Tiering" is incorrect as there is an additional fee for using this service and for a short-term requirement it may not be beneficial.

**INCORRECT:** "Amazon S3 One Zone-Infrequent Access (S3 One Zone-IA)" is incorrect as this storage class has a minimum capacity charge per object (128 KB) and a per GB retrieval fee.

**INCORRECT:** "Amazon S3 Glacier Deep Archive" is incorrect as this storage class is used for archiving data. There are retrieval fees and it takes hours to retrieve data from an archive.

References:

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

Save time with our AWS cheat sheets:

<https://digitalcloud.training/amazon-s3-and-glacier/>

#### 7. QUESTION

A solutions architect needs to backup some application log files from an online ecommerce store to Amazon S3. It is unknown how often the logs will be accessed or which logs will be accessed the most. The solutions architect must keep costs as low as possible by using the appropriate S3 storage class.

Which S3 storage class should be implemented to meet these requirements?

☐ S3 Standard-Infrequent Access (S3 Standard-IA)

☐ S3 Glacier

Correct

**Explanation:**

The S3 Intelligent-Tiering storage class is designed to optimize costs by automatically moving data to the most cost-effective access tier, without performance impact or operational overhead.

It works by storing objects in two access tiers: one tier that is optimized for frequent access and another lower-cost tier that is optimized for infrequent access. This is an ideal use case for intelligent-tiering as the access patterns for the log files are not known.

**CORRECT:** "S3 Intelligent-Tiering" is the correct answer.

**INCORRECT:** "S3 Standard-Infrequent Access (S3 Standard-IA)" is incorrect as if the data is accessed often retrieval fees could become expensive.

**INCORRECT:** "S3 One Zone-Infrequent Access (S3 One Zone-IA)" is incorrect as if the data is accessed often retrieval fees could become expensive.

**INCORRECT:** "S3 Glacier" is incorrect as if the data is accessed often retrieval fees could become expensive. Glacier also requires more work in retrieving the data from the archive and quick access requirements can add further costs.

**References:**

It has been impossible to find an excellent explanation of the difference between block storage and file storage. It is clear the author does not know what they are talking about when they tell you something like:

Block storage stores data in blocks.

File storage stores data in files.

Object storage stores data in objects.

And this is the extent of the explanation. It is not for comedic effect. X storage stored data in X. *Oh, bravo.* Because this takes the biscuit, when I refer to it later, I'll just use TTB.



A classic example is found in a 2021 textbook on AWS. Piper and Clinton (2021) write:

With block-level storage, data on a raw physical storage device is divided into individual blocks whose use is managed by a filesystem.

p.60

I'm telling you—this has not impressed me with the unique nature of a block storage system. I'm not saying that what is written is *false*. I'm saying it doesn't delineate the concept well enough, from surrounding concepts (like file storage).

What we want to know is this: why has it been necessary to use distinct words (object, block, file) for units of data? If there were a lump of data on the table, what should cause me to label it a *block*, rather than an *object*? I'm interested in the necessary and sufficient conditions for some data being a block, or object, or file.

We can start by noting that all these things—block, object, file—consist of bytes. Now, let's try and move beyond this.

Let's start with this 2009 article explaining why the hard drive you bought has less space than advertised. It writes:

In addition to the format reducing the size by a given amount, the file system's "block size" will also change the efficiency of space use on the drive.

Straight away, this tells us that blocks are used by file systems. It seems, then, that a file system is entirely compatible with block storage. Why do multiple articles present them as opposed, then?

The [article](#) continues

Blocks are small chunks of the drive that are written to (like "cubby holes" for data), and in most formats (including HFS ) are evenly sized.

There we go, a definition of block, finally. It's as if they have a physical basis. I think an essential feature of a block is that it has a fixed size. Another essential feature of blocks is that they are of *equal* size.

Now onto a [page](#) I have come across from S.K. Chang at the University of Pittsburgh:

Blocking refers to the practice of placing several logical records within one physical record. The purposes of blocking are: (a) to reduce access time by reading/writing more data in one I/O access, and (b) to increase the amount of data stored on I/O devices.

This is in line with my belief that blocks have a physical basis. What's interesting about this is that we have a practice (blocking) not a *thing* (a block). To get the point across, the author has had to make a distinction between *physical records* and *logical records*.

Now I'm going to turn to a page provided by IBM. They write:

Block storage breaks up data into blocks and then stores those blocks as separate pieces, each with a unique identifier.

This is an example of TTB. Block storage breaks data into *blocks*. Really? The sentence could be improved by stating why the data can be considered *broken*. What, exactly, has been broken up? An object? A file? It should also state why we're able to say the data is stored as separate pieces. Do they have to be a certain distance away from one another to count as separate? The [article](#) also states:

Block storage, sometimes referred to as block-level storage, is a technology that is used to store data files on Storage Area Networks (SANs) or cloud-based storage environments.

So, again, block storage is used to store *files*. Block storage does not imply an absence of files. What's interesting here is the association between block storage and Storage-area networks (SAN). Note, these are kinds of networks. I stress this because later we'll come across NAS, which is the acronym reversed. But NAS is quite



different. The question is: *why* is block storage suited to Storage-area networks?

What's clear is that with block storage, entities (files, let's say) are distributed. They are scattered like ashes (Cinderblock?). And when you request them, the system has to scurry around, putting it back together:

When a user or application requests data from a block storage system, the underlying storage system reassembles the data blocks and presents the data to the user or application.

IBM have a distinct article on block storage, which is much longer, [here](#). You might think that I was not justified in my earlier criticism of the statement that:

Block storage breaks up data into blocks

The emphasis should be on “breaks”, you might argue. The invocation of “block”, to explain “block storage” can be forgiven, because what's really important is the idea that a single thing is fragmented. In response, I want to point you to their description of object storage:

[Object storage](#), which is also known as object-based storage, breaks data files up into pieces called objects.

So, “breaking” is not unique to block storage. They really do describe block storage as breaking into blocks and object storage as breaking into objects.

I've come across an article written by Laurent Denel, which makes it clear (finally!) that the file structure is *imposed on top of* the block device:

The file system, which creates a virtual tree structure, is an abstraction layer that “is superimpo”ed on the “block device” (management of block writing at the kernel level).

This is an obvious improvement, for the reasons mentioned above, but the file system has not killed block storage. And for good reason: the **addition of an abstraction layer leads to a decrease in IO performance**.

What this makes clear is, I think, the fact that block storage and file storage are not two, mutually exclusive categories. Data is usually blocked. But sometimes, we impose a file structure on top of this, and this makes it appropriate to refer to it as file storage.

Okay, a bit of history. Direct-attached storage came first. Then came the idea of having an AREA: a storage area network. Here is Denel again:

This led to the appearance of **Direct Attached Storage (DAS)**, which is the ability for a computer to access a disk connected to the machine as a device.

Then came the **Storage Area Network (SAN)**, a network attached hard disk system that allows a machine to access storage space via the Fibre Channel protocol in client/server mode.

Nope, we're not talking about one of those bizarre television channels you find by accidentally going 'down' instead of 'up' on the remote control—to a station airing round-the-clock programming for constipation sufferers.

Despite being an American creation, the name is in fact supposed to use the British spelling (fibre) so as to make it distinctive. Fibre Channel is all [standardised](#). An organisation called INCITS comes up with the standards. The committee concerned with Fibre Channel is called T11. We're [told](#) that T10, T11 and T13 all tend to 'work on block-based data'. (We don't talk about T12.)

Fibre Channel is usually described as an *INTERFACE*. Specifically, it is a "data transfer interface" ([Primmer 1996](#)). It operates over both copper wire and optical fibre. ANSI (the American National Standards Institute) gave the go-ahead for Fibre Channel in 1988, and the first standard was published in 1989. There are plenty of (quite bland) details that you can get your teeth into here: the five hierarchical functional levels, a frame, sequence and exchange. Consult [Merym Primmer](#) for the details.

SNIA tells us:

A [storage area network \(SAN\)](#) is a dedicated network used for storage connectivity between host servers and shared storage - typically shared [arrays](#) that deliver [block-level](#) data storage.

[This](#) document from IBM tells us:



For a good primer, read [this](#) chapter (Chapter 6).

What is the difference between SAN and NAS? [This](#) article from IBM is quite good for this, because it brings them into close comparison at the end. Here are two differences:

1. NAS can use several protocols to connect with servers, including NFS, SMB/CIFS, and HTTP; a SAN uses the SCSI protocol.
2. **Type of network:** NAS is connected to devices using a LAN or Ethernet network, while a SAN runs on high-speed Fibre channel.

A nice heuristic I've come up with is that SAN uses serial protocols (such as iSCSI and Fibre Channel). Once you've got an acronym with an S at the beginning (like NAS), get "serial" out of your head. (And—if you need more fibre in your life, try cereal).

In 2002, W Curtis Preston published a [book](#) entitled "Using SANs and NAS". On the front cover there is a hyrax and a pika. These animals look like mice.

## 2. QUESTION

A video production company is planning to move some of its workloads to the AWS Cloud. The company will require around 5 TB of storage for video processing with the maximum possible I/O performance. They also require over 400 TB of extremely durable storage for storing video files and 800 TB of storage for long-term archival.

Which combinations of services should a Solutions Architect use to meet these requirements?

- ☒ Amazon EBS for maximum performance, Amazon EFS for durable data storage, and Amazon S3 Glacier for archival storage.
- ☐ Amazon EC2 instance store for maximum performance, Amazon EFS for durable data storage, and Amazon S3 for archival storage.
- ☐ Amazon EBS for maximum performance, Amazon S3 for durable data storage, and Amazon S3 Glacier for archival storage.
- ☐ Amazon EC2 instance store for maximum performance, Amazon S3 for durable data storage, and Amazon S3 Glacier for archival storage.

Incorrect

Explanation:

The best I/O performance can be achieved by using instance store volumes for the video processing. This is safe to use for use cases where the data can be recreated from the source files so this is a good use case.

For storing data durably Amazon S3 is a good fit as it provides 99.999999999% of durability. For archival the video files can then be moved to Amazon S3 Glacier which is a low cost storage option that is ideal for long-term archival.

**CORRECT:** "Amazon EC2 instance store for maximum performance, Amazon S3 for durable data storage, and Amazon S3 Glacier for archival storage" is the correct answer.

**INCORRECT:** "Amazon EBS for maximum performance, Amazon S3 for durable data storage, and Amazon S3 Glacier for archival storage" is incorrect. EBS is not going to provide as much I/O performance as an instance store volume so is not the best choice for this use case.

**INCORRECT:** "Amazon EC2 instance store for maximum performance, Amazon EFS for durable data storage, and Amazon S3 for archival storage" is incorrect. EFS does not provide as much durability as Amazon S3 and will not be as cost-effective.

**INCORRECT:** "Amazon EBS for maximum performance, Amazon EFS for durable data storage, and Amazon S3 Glacier for archival storage" is incorrect. EBS and EFS are not the best choices here as described above.

References:

I thought that the question above, from Neal Davis, was very difficult. The reason I chose an option involving EBS is that I thought this carried performance advantages over the instance store, particularly in terms of I/O performance. I also made a decision to select an option involving EFS (Elastic File System) because the question explicitly mentioned "file".

I must admit I do not fully understand the justification provided for using the instance store. We were supposed to choose the instance store, S3 and S3 Glacier.

## S3 Is Not a Backup

I've gotten some flak recently for daring to suggest that S3's native features weren't a substitute for a thoughtful backup strategy.

I'm not alone in this opinion. Andreas Wittig calls out that S3 Versioning is insufficient on the grounds of three reasons:

1. Accidental deletion, since you can delete all versions at the same time,
2. Malicious deletion, which hits the same problem, and
3. At scale, recovery is going to *suck* for you.

I'd also extend that to MFA delete; it makes deleting things you want to get rid of SUPER obnoxious, while still not solving for everything. I suppose Object Lock might work except then you will never, ever be able to delete your data ever again. That's expensive and more than a little bit constraining.

Email from Corey Quinn on July 5<sup>th</sup> 2023

# TPN

1. ***Bucket name uniqueness***– the requirement that bucket names are globally unique, throughout the entire S3 system.
2. ***Specific URL*** – the fact that the general form of a URL will be  
s3.amazonaws.com/bucketname/filename
3. ***CLI addressing***– the fact that the general way to address a file, in S3, using the CLI, is:  
s3://bucketname/filename
4. ***Limit of an object***– the inability of a single object in S3 to be larger than 5TB.
5. ***Limit of an upload*** – the inability of a single upload to be larger than 5 gigabytes (GB).
6. ***Phen6*** – the tendency of X to Y.
7. ***Phen7*** – the tendency of X to Y.
8. ***Phen8*** – the tendency of X to Y.
9. ***Phen9*** – the tendency of X to Y.
10. ***Phen10*** – the tendency of X to Y.

## Review questions

1. How many S3 buckets can you create in one AWS account? 100
- 2.

# Glossary

## **Object storage**

Description of what term means here.

## **Block storage**

Description of what term means here.

## **TTB**

Stands for *taking the biscuit*. Refers to the phenomenon of people explaining block storage by saying that data is divided into blocks.

A classic example is found on page 60 of the 2021 Sybex (Wiley) textbook by Piper and Clinton.

## **Term3**

Description of what term means here.

# Bibliography

- I. Official
- II. Unofficial
- III. Critical
- IV. General

## I. Official

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

## II. Unofficial

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

## III. Critical

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.



## IV. General

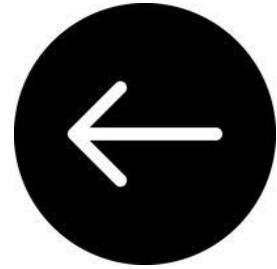
### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

# Module 2: working with objects



*MULTIPART  
UPLOAD*





## Using Presigned URLs

S3 tal57resignedthings called “presigned URLs”. At time of writing, the S3 documentation likes to use the unh57resigned, single word: *presigned*.

Straightaway, I want to note that the CloudFront documentation also talks about “signed57resigned believe that “presigned” and “signed” are interchangeable. This is because at present, the S3 documentation has this section where they nonchalantly switch between these two words:

### When does Amazon S3 check the expiration date and time of a presigned URL?

Amazon S3 checks the expiration date and time of a signed URL at the time of the HTTP request. For example, if a client begins to download a large file immediately before the expiration time, the download should complete even if the expiration time passes during the download. If the connection drops and the

I must note that I have tried to find some generic (non-AWS) information on signed URLs, and I have struggled. I am unable to locate a paper that first proposed the idea of a signed URL, nor a Request for Comments (RFC) that deals with these things.



However, we know signed URLs have been around since 2003 because there is a paper entitled:

Strongly authenticated URLs: Integrating Web Browsers and Applications with strong authentication

This paper appears to be a good introduction to signed URLs. The paper has four authors: Eddy Cheung, Andrew Goodchild, Hoylen Sue and Ben Fowler.

Also see [this](#) 2011 paper which mentions signed URLs. The point is, these things have been around for a while.

Piper and Clinton write:

If you cant to provide temporary access to an object that's otherwise private, you can generate a Presigned URL. The URL will be usable for a specified period of time, after which it will become invalid. You can build Presigned URL generation into your code to provide object access programmatically.

The following CLI command will return a URL that includes the required authentication string. The authentication will become invalid after 10 minutes (600 seconds). The default expiration value is 3,600 seconds (one hour).

## 2. QUESTION

A video production company is planning to move some of its workloads to the AWS Cloud. The company will require around 5 TB of storage for video processing with the maximum possible I/O performance. They also require over 400 TB of extremely durable storage for storing video files and 800 TB of storage for long-term archival.

Which combinations of services should a Solutions Architect use to meet these requirements?

- ☒ Amazon EBS for maximum performance, Amazon EFS for durable data storage, and Amazon S3 Glacier for archival storage.
- ☐ Amazon EC2 instance store for maximum performance, Amazon EFS for durable data storage, and Amazon S3 for archival storage.
- ☐ Amazon EBS for maximum performance, Amazon S3 for durable data storage, and Amazon S3 Glacier for archival storage.
- ☐ Amazon EC2 instance store for maximum performance, Amazon S3 for durable data storage, and Amazon S3 Glacier for archival storage.



**Incorrect**

**Explanation:**

The best I/O performance can be achieved by using instance store volumes for the video processing. This is safe to use for use cases where the data can be recreated from the source files so this is a good use case.

For storing data durably Amazon S3 is a good fit as it provides 99.999999999% of durability. For archival the video files can then be moved to Amazon S3 Glacier which is a low cost storage option that is ideal for long-term archival.

**CORRECT:** "Amazon EC2 instance store for maximum performance, Amazon S3 for durable data storage, and Amazon S3 Glacier for archival storage" is the correct answer.

**INCORRECT:** "Amazon EBS for maximum performance, Amazon S3 for durable data storage, and Amazon S3 Glacier for archival storage" is incorrect. EBS is not going to provide as much I/O performance as an instance store volume so is not the best choice for this use case.

**INCORRECT:** "Amazon EC2 instance store for maximum performance, Amazon EFS for durable data storage, and Amazon S3 for archival storage" is incorrect. EFS does not provide as much durability as Amazon S3 and will not be as cost-effective.

**INCORRECT:** "Amazon EBS for maximum performance, Amazon EFS for durable data storage, and Amazon S3 Glacier for archival storage" is incorrect. EBS and EFS are not the best choices here as described above.

**References:**

I thought that the question above, from Neal Davis, was very difficult. The reason I chose an option involving EBS is that I thought this carried performance advantages over the instance store, particularly in terms of I/O performance. I also made a decision to select an option involving EFS (Elastic File System) because the question explicitly mentioned "file".

I must admit I do not fully understand the justification provided for using the instance store. We were supposed to choose the instance store, S3 and S3 Glacier.





#### 17. QUESTION

A solutions architect is creating a document submission application for a school. The application will use an Amazon S3 bucket for storage. The solution must prevent accidental deletion of the documents and ensure that all versions of the documents are available. Users must be able to upload and modify the documents.

Which combination of actions should be taken to meet these requirements? (Select TWO.)

☒ Enable versioning on the bucket

☒ Enable MFA Delete on the bucket

☐ Attach an IAM policy to the bucket

☐ Set read-only permissions on the bucket

☐ Encrypt the bucket using AWS SSE-S3



#### Explanation:

None of the options present a good solution for specifying permissions required to write and modify objects so that requirement needs to be taken care of separately. The other requirements are to prevent accidental deletion and ensure that all versions of the document are available.

The two solutions for these requirements are versioning and MFA delete. Versioning will retain a copy of each version of the document and multi-factor authentication delete (MFA delete) will prevent any accidental deletion as you need to supply a second factor when attempting a delete.

**CORRECT:** "Enable versioning on the bucket" is a correct answer.

**CORRECT:** "Enable MFA Delete on the bucket" is also a correct answer.

**INCORRECT:** "Set read-only permissions on the bucket" is incorrect as this will also prevent any writing to the bucket which is not desired.

**INCORRECT:** "Attach an IAM policy to the bucket" is incorrect as users need to modify documents which will also allow delete. Therefore, a method must be implemented to just control deletes.

**INCORRECT:** "Encrypt the bucket using AWS SSE-S3" is incorrect as encryption doesn't stop you from deleting an object.

#### References:

# TPN

3.     ***Phenomenon1*** – the tendency of X to Y.
4.     ***Phen2*** – the tendency of X to Y.
5.     ***Phen3*** – the tendency of X to Y.
6.     ***Phen4*** – the tendency of X to Y.
7.     ***Phen5*** – the tendency of X to Y.
8.     ***Phen6*** – the tendency of X to Y.
9.     ***Phen7*** – the tendency of X to Y.
10.    ***Phen8*** – the tendency of X to Y.
11.    ***Phen9*** – the tendency of X to Y.
12.    ***Phen10*** – the tendency of X to Y.

# Glossary

## Term1

Description of what term means here.

## Term2

Description of what term means here.

## Term3

Description of what term means here.

# Bibliography

- V. Official
- VI. Unofficial
- VII. Critical
- VIII. General

## V. Official

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

## VI. Unofficial

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

<https://ieeexplore.ieee.org/abstract/document/7365920>

<https://www.ieee-security.org/TC/SP2011/PAPERS/2011/paper029.pdf>

Cheung, Eddy, Andrew Goodchild, Hoyle Sue, and Ben Fowler.  
"Strongly authenticated URLs: Integrating Web browsers  
and." *AUUGN* (2003): 47.

## VII. Critical

### [Surname1]

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

### [Surname1]

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

## VIII. General

### [Surname1]

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

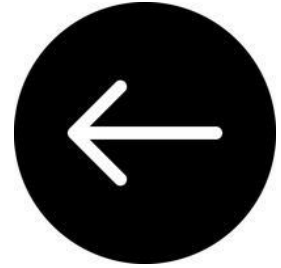
### [Surname1]

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.



# S3\_Module\_3

Working with buckets





# Transfer Acceleration



- Speed comparison tool
- [Bucketname.s3-accelerate.amazonaws.com](https://bucketname.s3-accelerate.amazonaws.com)



16. QUESTION

A company runs an application in an on-premises data center that collects environmental data from production machinery. The data consists of JSON files stored on network attached storage (NAS) and around 5 TB of data is collected each day. The company must upload this data to Amazon S3 where it can be processed by an analytics application. The data must be transferred securely.

Which solution offers the MOST reliable and time-efficient data transfer?

- ☐ Multiple AWS Snowcone devices.
- ☒ AWS DataSync over AWS Direct Connect.
- ☐ Amazon S3 Transfer Acceleration over the Internet.
- ☐ AWS Database Migration Service over the Internet.

The most reliable and time-efficient solution that keeps the data secure is to use AWS DataSync and synchronize the data from the NAS device directly to Amazon S3. This should take place over an AWS Direct Connect connection to ensure reliability, speed, and security.

AWS DataSync can copy data between Network File System (NFS) shares, Server Message Block (SMB) shares, self-managed object storage, AWS Snowcone, Amazon Simple Storage Service (Amazon S3) buckets, Amazon Elastic File System (Amazon EFS) file systems, and Amazon FSx for Windows File Server file systems.

**CORRECT:** "AWS DataSync over AWS Direct Connect" is the correct answer.

**INCORRECT:** "AWS Database Migration Service over the Internet" is incorrect. DMS is for migrating databases, not files.

**INCORRECT:** "Amazon S3 Transfer Acceleration over the Internet" is incorrect. The Internet does not offer the reliability, speed or performance that this company requires.

**INCORRECT:** "Multiple AWS Snowcone devices" is incorrect. This is not a time-efficient approach as it can take time to ship these devices in both directions.

References:

# TPN

13. ***Phenomenon1*** – the tendency of X to Y.
14. ***Phen2*** – the tendency of X to Y.
15. ***Phen3*** – the tendency of X to Y.
16. ***Phen4*** – the tendency of X to Y.
17. ***Phen5*** – the tendency of X to Y.
18. ***Phen6*** – the tendency of X to Y.
19. ***Phen7*** – the tendency of X to Y.
20. ***Phen8*** – the tendency of X to Y.
21. ***Phen9*** – the tendency of X to Y.
22. ***Phen10*** – the tendency of X to Y.

# Glossary

## Term1

Description of what term means here.

## Term2

Description of what term means here.

## Term3

Description of what term means here.

# Bibliography

- IX. Official
- X. Unofficial
- XI. Critical
- XII. General

## IX. Official

### [Surname1]

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

### [Surname1]

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

## X. Unofficial

### [Surname1]

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

### [Surname1]

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

## XI. Critical

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

## XII. General

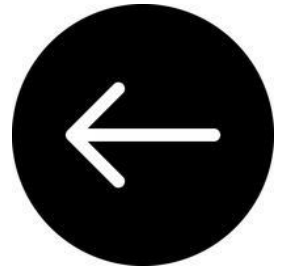
### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

# S3\_Module\_4



Security (encryption, managing access)  
and ways of managing storage

## Amazon S3 Inventory can include ACLs as object metadata in inventory reports

Posted On: Jul 14, 2023

With Amazon S3 Inventory, you can now easily review your access control lists (ACLs) on all of your objects to simplify review of access permissions. ACLs were the original way to manage object access when S3 launched in 2006. Now, when migrating to IAM-based [bucket policies](#) for access control, you can easily review all of the object ACLs in your buckets before enabling S3 Object Ownership.

S3 Inventory provides a complete list of objects in a bucket and their corresponding metadata. The new Object ACLs fields include details about the object owner and the grantee along with their permission granted. You can activate reporting on object ACLs by editing existing S3 Inventory configuration in the AWS Management Console or API.

By enabling [S3 Object Ownership](#), you can change how S3 performs access control for a bucket so that only IAM policies are used. S3 Object Ownership's 'Bucket owner enforced' setting disables ACLs for your bucket and the objects in it, and updates every object so that each object is owned by the bucket owner. We recommend that you carefully review your use of ACLs with inventory reports, migrate to IAM-based bucket policies, and then disable ACLs with S3 Object Ownership. For more information, see [Controlling ownership of objects and disabling ACLs for your bucket](#).

Amazon S3 Inventory support for Object ACL is generally available at no additional charge in all AWS Regions, excluding the AWS GovCloud (US) Regions and AWS China Regions. To learn more, please visit [Amazon S3 Inventory](#) and [Amazon S3 pricing](#).

## Configuring MFA Delete

#### 10. QUESTION

A company has uploaded some highly critical data to an Amazon S3 bucket. Management are concerned about data availability and require that steps are taken to protect the data from accidental deletion. The data should still be accessible, and a user should be able to delete the data intentionally.

Which combination of steps should a solutions architect take to accomplish this? (Select TWO.)

- ☐ Create a bucket policy on the S3 bucket.
- ☐ Create a lifecycle policy for the objects in the S3 bucket.
- ☐ Enable default encryption on the S3 bucket.
- ☒ Enable versioning on the S3 bucket.
- ☒ Enable MFA Delete on the S3 bucket.

#### Explanation:

Multi-factor authentication (MFA) delete adds an additional step before an object can be deleted from a versioning-enabled bucket.

With MFA delete the bucket owner must include the x-amz-mfa request header in requests to permanently delete an object version or change the versioning state of the bucket.

**CORRECT:** "Enable versioning on the S3 bucket" is a correct answer.

**CORRECT:** "Enable MFA Delete on the S3 bucket" is also a correct answer.

**INCORRECT:** "Create a bucket policy on the S3 bucket" is incorrect. A bucket policy is not required to enable MFA delete.

**INCORRECT:** "Enable default encryption on the S3 bucket" is incorrect. Encryption does protect against deletion.

**INCORRECT:** "Create a lifecycle policy for the objects in the S3 bucket" is incorrect. A lifecycle policy will move data to another storage class but does not protect against deletion.

#### References:

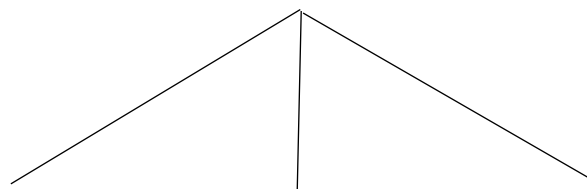
<https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingMFADelete.html>

Save time with our AWS cheat sheets:

<https://digitalcloud.training/amazon-s3-and-glacier/>

# x-amz-mfa

Controlling access



ACLs

Bucket  
policies

IAM  
policies

Piper and Clinton (page 67) write:

You can strategically open up access at the bucket and object levels using access control list (ACL) rules, finer-grained S3 bucket policies, or Identity and Access Management (IAM) policies.



Versioning



# Let's talk about **consistency** in general

## CONSISTENCY—CLIENT AND SERVER

There are two ways of looking at consistency. One is from the developer/client point of view: how they observe data updates. The second way is from the server side: how updates flow through the system and what guarantees systems can give with respect to updates.

### CLIENT-SIDE CONSISTENCY

The client side has these components:

Client-side consistency has to do with how and when observers (in this case the processes A, B, or C) see updates made to a data object in the storage systems. In the following examples illustrating the different types of consistency, process A has made an update to a data object:

- **Strong consistency.** After the update completes, any subsequent access (by A, B, or C) will return the updated value.
- **Weak consistency.** The system does not guarantee that subsequent accesses will return the updated value. A number of conditions need to be met before the value will be returned. The period between the update and the moment when it is guaranteed that any observer will always see the updated value is dubbed the *inconsistency window*.
- \* **Eventual consistency.** This is a specific form of weak consistency; the storage system guarantees that if no new updates are made to the object, eventually all accesses will return the last updated value. If no failures occur, the maximum size of the inconsistency window can be determined based on factors such as communication delays, the load on the system, and the number of replicas involved in the replication scheme. The most popular system that implements eventual consistency is DNS (Domain Name System). Updates to a name are distrib-

- **Causal consistency.** If process A has communicated to process B that it has updated a data item, a subsequent access by process B will return the updated value, and a write is guaranteed to supersede the earlier write. Access by process C that has no causal relationship to process A is subject to the normal eventual consistency rules.
- **Read-your-writes consistency.** This is an important model where process A, after it has updated a data item, always accesses the updated value and will never see an older value. This is a special case of the causal consistency model.
- **Session consistency.** This is a practical version of the previous model, where a process accesses the storage system in the context of a session. As long as the session exists, the system guarantees read-your-writes consistency. If the session terminates because of a certain failure
- **Monotonic write consistency.** In this case the system guarantees to serialize the writes by the same process. Systems that do not guarantee this level of consistency are notoriously hard to program.

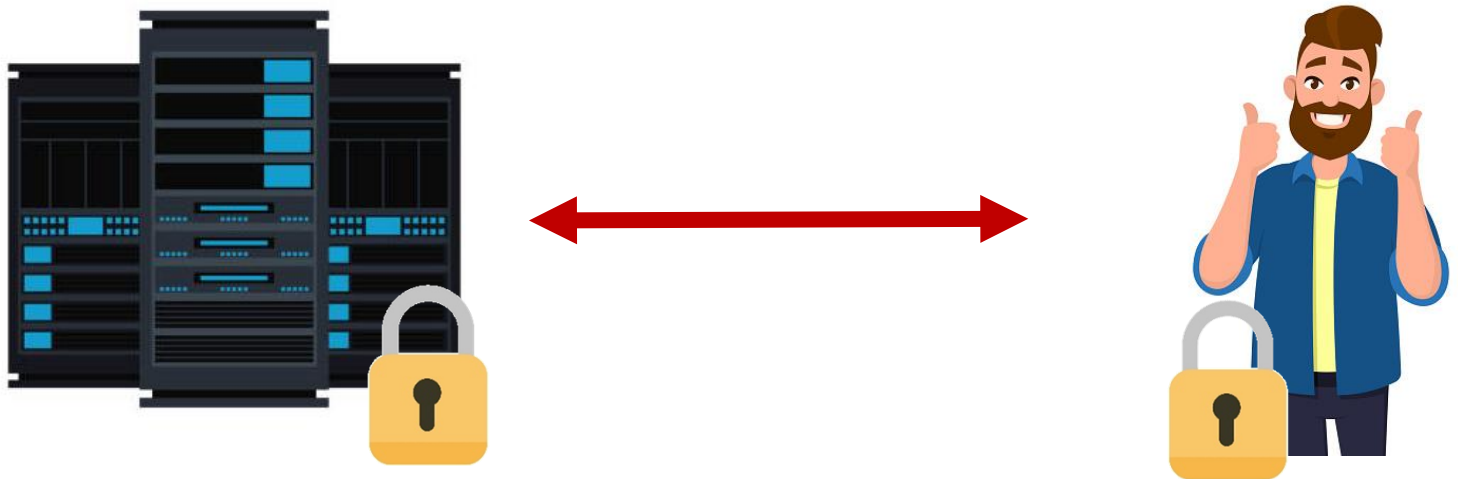
A number of these properties can be combined. For example, one can get monotonic reads combined with session-level consistency. From a practical point of view these two properties (monotonic reads and read-your-writes) are most desirable in an eventual consistency

Because there isn't the risk of corruption when creating new objects, S3 provides **read-after-write consistency** for creation (PUT) operations.

Piper and Clinton 2021, p66

# Data encryption

We can distinguish between SERVER-SIDE encryption and CLIENT-SIDE encryption.



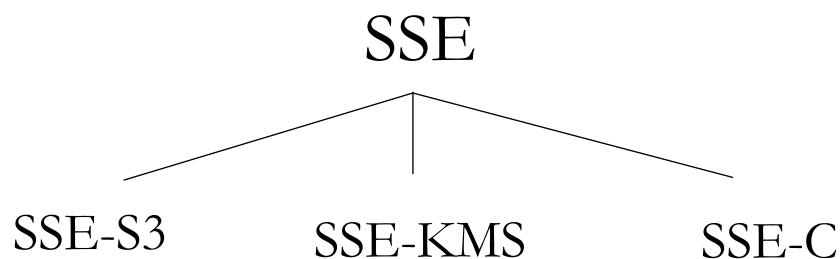
When we say “server-side”, we are referring to AWS and their tools. So, to achieve “server-side” encryption is to encrypt data using AWS tools, as opposed to non-AWS tools.

“Client-side encryption” refers to the practice of encrypting data before it’s transferred to S3. The client is encrypting their data, not (*just*) the server. In this case, the server is the AWS platform.

Note that KMS stands for Key Management Service.  
This is an AWS encryption service.

## How can we achieve **SERVER-SIDE** encryption?

There are three methods for achieving SSE. They are:



## How can we achieve **CLIENT-SIDE** encryption?

This can be done using a KMS-Managed Customer Master Key (CMK). A CMK produces a unique key for each object before it's uploaded.

You can also use a Client-Side Master Key, which you provide through the Amazon S3 encryption client.

Server-side encryption can greatly reduce the complexity of the process and is often preferred.

However, there might be regulations or a company policy that require client-side encryption.

## Managing your S3 data

# Object lock

**OBJECT**





# LOCK

## Cross-region Replication

# Cross-region replication

An aerial photograph of a scenic landscape featuring a large lake, rolling green hills, and distant mountains under a blue sky with scattered clouds. The text 'Cross-region replication' is overlaid in large white letters.

29. QUESTION

An organization want to share regular updates about their charitable work using static webpages. The pages are expected to generate a large amount of views from around the world. The files are stored in an Amazon S3 bucket. A solutions architect has been asked to design an efficient and effective solution.

Which action should the solutions architect take to accomplish this?

☒ Use Amazon CloudFront with the S3 bucket as its origin

☐ Generate presigned URLs for the files

☐ Use cross-Region replication to all Regions

☐ Use the geoproximity feature of Amazon Route 53

Correct

Explanation:

Amazon CloudFront can be used to cache the files in edge locations around the world and this will improve the performance of the webpages.

To serve a static website hosted on Amazon S3, you can deploy a CloudFront distribution using one of these configurations:

- Using a REST API endpoint as the origin with access restricted by an [origin access identity \(OAI\)](#)
- Using a website endpoint as the origin with anonymous (public) access allowed
- Using a website endpoint as the origin with access restricted by a Referer header

**CORRECT:** "Use Amazon CloudFront with the S3 bucket as its origin" is the correct answer.

**INCORRECT:** "Generate presigned URLs for the files" is incorrect as this is used to restrict access which is not a requirement.

**INCORRECT:** "Use cross-Region replication to all Regions" is incorrect as this does not provide a mechanism for directing users to the closest copy of the static webpages.

**INCORRECT:** "Use the geoproximity feature of Amazon Route 53" is incorrect as this does not include a solution for having multiple copies of the data in different geographic locations.

**CORRECT:** "Use Amazon CloudFront with the S3 bucket as its origin" is the correct answer.

**INCORRECT:** "Generate presigned URLs for the files" is incorrect as this is used to restrict access which is not a requirement.

**INCORRECT:** "Use cross-Region replication to all Regions" is incorrect as this does not provide a mechanism for directing users to the closest copy of the static webpages.

**INCORRECT:** "Use the geoproximity feature of Amazon Route 53" is incorrect as this does not include a solution for having multiple copies of the data in different geographic locations.







S3

Select

What is S3 Select?

## 2. QUESTION

A solutions architect is creating a document submission application for a school. The application will use an Amazon S3 bucket for storage. The solution must prevent accidental deletion of the documents and ensure that all versions of the documents are available. Users must be able to upload and modify the documents.

Which combination of actions should be taken to meet these requirements? (Select TWO.)

- ☐ Enable MFA Delete on the bucket
- ☐ Attach an IAM policy to the bucket
- ☐ Encrypt the bucket using AWS SSE-S3
- ☐ Set read-only permissions on the bucket
- ☐ Enable versioning on the bucket

# TPN

- 23. ***Phenomenon1*** – the tendency of X to Y.
- 24. ***Phen2*** – the tendency of X to Y.
- 25. ***Phen3*** – the tendency of X to Y.
- 26. ***Phen4*** – the tendency of X to Y.
- 27. ***Phen5*** – the tendency of X to Y.
- 28. ***Phen6*** – the tendency of X to Y.
- 29. ***Phen7*** – the tendency of X to Y.
- 30. ***Phen8*** – the tendency of X to Y.



31. ***Phen9*** – the tendency of X to Y.
32. ***Phen10*** – the tendency of X to Y.

# Glossary

## **SSE**

This stands for Server-Side Encryption (see *Server-Side Encryption*).

AWS and Piper and Clinton (Sybex 2021) use a hyphen. It is “server-side”, not “server side”.

## **SSE-C**

One of three ways to achieve SSE on S3. SSE using a Customer-provided Key.

## **SSE-S3**

One of three ways to achieve SSE on S3. SSE using a key provided by S3, which S3 manages.

## **SSE-KMS**

One of three ways to achieve SSE on S3. SSE using a key provided by AWS KMS. An envelope key is added along with a full audit trail for tracking key usage (Piper and Clinton 2021 p63.) See *envelope key*.

## **Server-side encryption**

Description of what term means here.

## **Envelope key**

Description of what term means here.

## **Versioning**

We say things like “is versioning *enabled* at the bucket level”. In S3, if versioning is enabled, then older overwritten copies of an object will be saved and remain accessible indefinitely.

It's called *versioning* simple because we're maintaining (holding on to) multiple *versions* or states, of a particular object.

## ACL

ACL – Each bucket and object has an ACL associated with it. An ACL is a list of grants identifying grantee and permission granted. You use ACLs to grant basic read/write permissions to other AWS accounts. ACLs use an Amazon S3-specific XML schema.

# Bibliography

- XIII. Official
- XIV. Unofficial
- XV. Critical
- XVI. General

## XIII. Official

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:

<URL here>.

## XIV. Unofficial

### [Surname1]

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.

Available at:

<URL here>.

### [Surname1]

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.

Available at:

<URL here>.

## XV. Critical

### [Surname1]

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.

Available at:

<URL here>.

### [Surname1]

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.

Available at:

<URL here>.

## XVI. General

### [Surname1]

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.

Available at:

<URL here>.

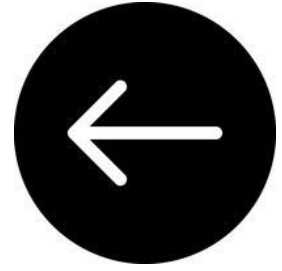
### [Surname1]

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.

Available at:

<URL here>.

# S3\_Module\_5



## 13. QUESTION

A company manages an application that runs on Amazon EC2 instances behind a Network Load Balancer (NLB). The NLB has access logs enabled which are being stored in an Amazon S3 bucket. A security engineer requires a solution to run ad hoc queries against the access logs to identify application access patterns.

How should the security engineer accomplish this task with the least amount of administrative overhead?

- ☒ Create an Amazon Athena table that uses the S3 bucket containing the access logs. Run SQL queries using Athena.
- ☐ Use the S3 copy command to copy logs to a separate bucket. Enable S3 analytics to analyze access patterns.
- ☐ Write an AWS Lambda function to query the access logs. Use event notifications to trigger the Lambda functions when log entries are added.
- ☐ Import the access logs into Amazon CloudWatch Logs. Use CloudWatch Logs Insights to analyze the log data.

**Incorrect**

**Explanation:**

Amazon Athena is a serverless service you can use to run SQL queries against data in Amazon S3. You just need to point Athena to your data in Amazon S3, define the schema, and start querying using the built-in query editor. This is ideal for running ad-hoc queries on access logs stored in an S3 bucket.

**CORRECT:** "Create an Amazon Athena table that uses the S3 bucket containing the access logs. Run SQL queries using Athena" is the correct answer (as explained above.)

**INCORRECT:** "Use the S3 copy command to copy logs to a separate bucket. Enable S3 analytics to analyze access patterns" is incorrect.

There's no need to copy the data and S3 analytics is used to identify object access patterns for requests to S3 objects. It is used for storage class analytics. It does not help with identifying access patterns for your application by reading the file and looking at source IP addresses (for example).

**INCORRECT:** "Write an AWS Lambda function to query the access logs. Use event notifications to trigger the Lambda functions when log entries are added" is incorrect.

This will be more complex and is less useful for running ad hoc queries as it is something that will run every time a file is added.

**INCORRECT:** "Import the access logs into Amazon CloudWatch Logs. Use CloudWatch Logs Insights to analyze the log data" is incorrect.

You cannot natively import logs into CloudWatch Logs from Amazon S3. You may be able to achieve this with a custom Lambda function, but it will be more work.

**References:**

<https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-access-logs.html>

<https://aws.amazon.com/athena/features/>

# TPN

- 33. *Phenomenon1* – the tendency of X to Y.
- 34. *Phen2* – the tendency of X to Y.
- 35. *Phen3* – the tendency of X to Y.
- 36. *Phen4* – the tendency of X to Y.
- 37. *Phen5* – the tendency of X to Y.
- 38. *Phen6* – the tendency of X to Y.
- 39. *Phen7* – the tendency of X to Y.
- 40. *Phen8* – the tendency of X to Y.
- 41. *Phen9* – the tendency of X to Y.
- 42. *Phen10* – the tendency of X to Y.

# Glossary

## Term1

Description of what term means here.

## Term2

Description of what term means here.

## Term3

Description of what term means here.

# Bibliography

- XVII. Official
- XVIII. Unofficial
- XIX. Critical
- XX. General

## XVII. Official

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

## XVIII. Unofficial

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

## XIX. Critical



**[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

**[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

## XX. General

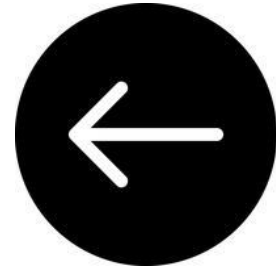
**[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.


**[Surname1]**

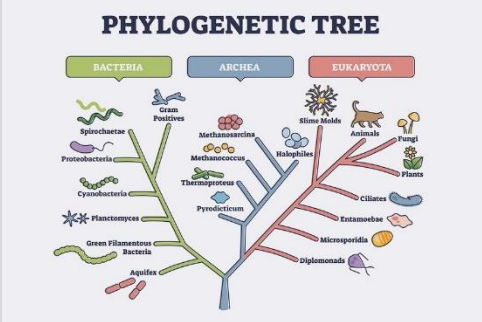


Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

# EC2



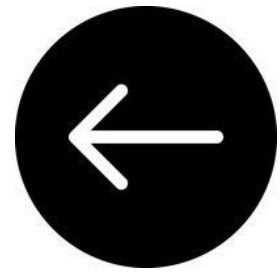
There is so much you need to know about this service that it is broken into five modules.

|   |  |  |
|---|--|--|
| 1 | Virtualization   |  |
|   | 1  |  |
|   | 2  |  |
|   | 3  |  |
|   | 4  |  |
|   | 5  |  |
|   | 6  |  |
| 2 | Amazon Machine Images  |  |
|   |  |  |
|   | 1  |  |
|   | 2  |  |
|   | 3  |  |
|   | 4  |  |
| 3 | Instance Types   |  |

|   |  |  |                |
|---|--|--|----------------|
|   |  |  |                |
|   |  |  |                |
|   | 1  |  |                |
|   | 2  |  |                |
|   | 3  |  |                |
|   | 4  | High Performance Computing (HPC)   |                |
|   |  |  |                |
|  |  |  |                |
|   | 1  |  |                |
|   | 2  |  |                |
|   | 3  |  |                |
|   | 4  |  |                |
|   | 5  |  |                |
|   | 6  | Instance lifecycle   |                |
|   |  |  |                |
|   |  |  |                |
|   |  |  | Stop and start |
|   |  |  | Hibernate      |
|   |  |  | Terminate      |
|   |  |  | Recover        |
| 5   | Configuring your instance<br> |  |                |
|   | 1  |  |                |
|   | 2  |  |                |
|   | 3  |  |                |
|   | 4  |  |                |

|  |   |   |
|--|---|---|
|  | 5 | Enhanced Networking: <ul style="list-style-type: none"> <li>• SR-IOV</li> </ul> |
|  |   |   |
|  | 6 | Placement groups  |
|  |   |   |

# Module 1 – Virtualization



[RETURN TO EC2](#)

Fifteen years ago today I wrote the blog post that launched the [Amazon EC2 Beta](#). As I recall, the launch was imminent for quite some time as we worked to finalize the feature set, the pricing model, and innumerable other details. The launch date was finally chosen and it happened to fall in the middle of a long-planned family vacation to Cabo San Lucas, Mexico.

Undaunted, I brought my laptop along on vacation, and had to cover it with a towel so that I could see the screen as I wrote. I am not 100% sure, but I believe that I actually clicked **Publish** while sitting on a lounge chair

near the pool! I spent the remainder of the week offline, totally unaware of just how much excitement had been created by this launch.



[Blog post](#) from Jeff Barr, recalling the launch of the EC2 service

# 1. What on earth is AMD?

In its section on Amazon Machine Images (AMIs), you'll be told something such as:

By default, Graviton instance types run on UEFI, and Intel and AMD instance types run on Legacy BIOS.

What on earth is "AMD"? First, it's not the eye disorder, *age-related macular degeneration* (AMD). It stands for "Advanced Micro Devices" and is the name of a company based in Santa Clara, California. This [city](#) is in the heart of Silicon Valley. The Spanish named this city after Saint Clare (patron saint of [television screens](#) and eye diseases). Taking a look at [this list](#) of the top ten employers in this city tells you a lot. The fourth employer is an [amusement park](#) which sadly [announced](#) it is closing recently. The top employers are Applied Material, Intel, and then AMD, which employs 3000. They manufacture computer processors. Small ones. Which are advanced.

AMD is usually said to be founded by "Jerry Sanders and seven others", the year of the moonlanding ([Britannica](#)). These were executives who came from a large company called Fairchild Semiconductor Corporation.

# 2. What is Intel?

At time of writing, Wikipedia tells you that Intel is "the world's largest semiconductor chip manufacturer by revenue". The word

“Intel” comes from crunching together “integrated” and “electronics”. I should at this point explain what a “semiconductor chip” is.

You know about conductors: materials which an electric current flows through easily (examples are metals such as copper and gold). Then, there are insulators, which an electric current cannot flow through (examples are rubber, and wood, which is packed full of carbon). The category SEMICONDUCTOR is reserved for those materials whose conductivity we can *manipulate*. We can heat them up to increase their conductivity; we can add impurities to change the conductivity. Chemists have a funny name for elements on the periodic table that *sort of* behave like metals: “metalloid”. There’s silicon, which sits below carbon, and gallium, which sits below aluminium. Often, these *metalloids* are the elements found in semiconductor devices.

Semiconductor devices are electronic components. That is, they are removable parts of an electronic circuit. Put crudely, consider circuit diagrams in GCSE Physics, with a battery and lamp with wires attached. On the same level as battery and lamp, is *semiconductor device*. They can be added to circuits or removed from them, as a unit. Examples of *semiconductor devices* include things called *diodes* and *transistors*.

The electronic circuit had started to be refined ten years before the moon landing. In 1959 Jack Kilby [proposed](#): “a body of semiconductor material ... wherein all the components of the electronic circuit are” completely integrated.” This was the beginning of the concept of an *integrated circuit* (IC). We can define an integrated circuit as

a circuit in which all or some of the circuit elements are inseparably associated and electrically interconnected so that it is considered to be indivisible for the purposes of construction and commerce.

[JEDEC]

Incredibly, the whole circuit is built in a single, monolithic piece of silicon. There will not be discrete components (such as transistors), instead the circuit will *be* the component. The IC roughly become an “array of transistors” ([Horowitz 1989: 61](#)). Because we have one or several circuits on a single piece (or chip) of material, many people will refer to an integrated circuit as a *chip*.

### 3. What is a chip?

SO, back to Intel. Recall that it arose from “Integrated Electronics”. Intel was basically founded the same year as AMD (1968 is the official line). It was founded by Gordon Moore and Robert Noyce. This is the namesake of Moore’s Law who originally studied chemistry at Berkeley. They thought for a bit about calling the company “Moore-Noyce”, but they rejected this because it sounds too similar to “more noise”.

Intel also has headquarters in Santa Clara. It really is 6 minutes by car, from the AMD headquarters to Intel. Watch Conan O’Brien take a visit to Intel [here](#).

Meltdown and Spectre

## 4. What on earth is “x86”?

The “x” is a variable. It signifies that lots of different things could go before the 86. The expression “x86” denotes a family of instruction set architectures. We’ll get onto instruction set architectures shortly.

Why 86? I don’t know. If it’s in your interests to be able to recall this number, join me for some interesting facts. According to [Snopes](#), “eighty six” is American slang used for items of food or drink that are no longer available. It’s also a kind of codeword for customers who need to be ejected from a restaurant. It can also take a darker meaning, used to refer to *killing* people [[Green](#) 2005]. [Here](#) is Gordon Moore being interviewed aged 86, in the year his law celebrated its 50<sup>th</sup> birthday.

Anyhow, it all started with the eighty variety: the Intel 8086. “The 8086 gave rise to the [x86](#) architecture, which eventually became Intel’s most successful line of processors.” [[Wikipedia](#)] We are told that “Several successors to Intel’s 8086 processor have been



released with the names 80186, 80286, 80386 and 80486.”

So, they started putting three numbers before the 86. Examples include: 801, 802, 803, 804. We’re counting up.

## 5. What is HVM?

## 6. What is PV?

What is  
Graviton?

## 7. What is UEFI?

## 8. What is SoC?

SoC is short for System on a Chip. An SoC combines all of the various components of a computer or other system.

An SoC is an integrated circuit. However, SoCs do more than plain old integrated circuits. SoC include things you might need such as memory, input/output functions and secondary storage—all in one place.

We can better understand SoCs by comparing them to CPUs:

“An SoC is only a little larger than a CPU but packs a lot more functionality into that space. A CPU can’t function without dozens of other chips, but you can build a complete computer with a single SoC, and that computer can be much smaller, and much cheaper.”

[[Suse.com](#)]

To begin with SoCs, I refer you to this 2016 video by [[TechQuickie](#)].

## 9. What is UEFI?

To readers who enjoy refereeing under the rules of Union of European Football Associations, this is *not* for you. However, your intention of providing control, and of following rules which cannot be reprogrammed—this is our concern. We’re talking about [firmware](#).

UEFI stands for Unified Extensible Firmware Interface. This is a specification that defines the software interface between an OS and platform firmware. UEFI is an alternative to BIOS, or Basic Input/output System. Let's break this down.

It didn't always have the "unified". There is a thing just called "Extensible Firmware Interface" (EFI). The EFI was developed by Intel, with the initial specification released in 1999. Various proposals were considered, with the aim to "clean up the boot interface". And "with the advent of 64-bit computing... and the industry's need to have a commonly owned specification, the UEFI 2.0 specification appeared in 2005". The "various proposals" included "Open Firmware and Advanced Risk Computing (ARC)". In a great book, entitled [\*Beyond BIOS\*](#), we're told that "ultimately, though, EFI prevailed and its architecture-neutral interface was adopted".

In the early 2000s, the UEFI Forum is established as a Washington non-profit Corporation. It manages the evolution of a unified EFI Specification. Promoter members of the early specification include AMD, AMI, Apple Dell, IBM, Intel and Microsoft. (AMI is [American Megatrends International](#), founded in 1985.)

Itanium (settled on after an engineer, who preferred efficient pronouns, was locked in a room with the latte-loving marketing guru, until a compromise was reached) was a family of Intel processors.

## 10. What is going on with "ARM"?

Occasionally, you will see the term ARM, Arm, or arm. I want to make things clear here—because this can refer to a few different, but related things.

A company called “[Acorn Computers Ltd](#)” was established in Cambridge, England in 1978.

Now, a company called “**Advanced RISC Machines Ltd**” was established in England in 1990. Twelve employees from the original “Acorn Computers Ltd” were involved. Apple provided about three million dollars in investment. Apple said something like “~~remove the word “acorn”, there’s only one plant-based brand in town~~”. Thus, the word “advanced” was put in place.

This company—*Advanced* RISC Machines Ltd—started to just called itself “ARM Holdings” by 1998. By 2017, the conversion to cool was complete, with *arm*.

Now let’s remove ourselves from the commercial world. If you ask a computer scientist what ARM stands for, they will say “Advanced RISC Machine”. Even though this *is* a product, I want to now focus on the idea of an Advanced RISC Machine (ARM).

ARM is one of the most used, and most licensed processor cores in the world.

There have been different versions. From 1993 to 2001, [ARM7](#) was released. Later came [the ARM10 family](#). [ARM11](#) cores were released from 2002 to 2005. Nowadays, ARM Cortex-A and Cortex-R cores are preferred.

There are technical differences between them. ARM7 had a three-stage pipeline depth; ARM11 an eight-stage one. ARM7 has a Von Neumann architecture; ARM11 had a Harvard architecture. ARM7 typically achieved 80 mega Hertz; ARM11 was 335 [[Chuang](#)].

Some helpful PowerPoint slides, which I have relied on, are [here](#). To go further, get a book with a blue cover called *ARM System Developer’s Guide* (2004) by Sloss, Symes and Wright. Also see Steve Furber’s [ARM System on Chip Architecture](#) (2000).

# 11. What is Graviton?

The Graviton processor is a processor developed by AWS. Let's just step back for a moment, and consider AWS's work in the area of hardware. Ali Saidi [tells us](#) that:

“We’ve been innovating in silicon across three major areas in AWS.

The first is the AWS Nitro system, where we took components of the hypervisor (that’s the piece of software that takes a big machine and cuts it into virtual machines) and start moving it onto special-purpose chips, that accelerated I/O, raised the bar in security and let you use all of the resources on that host processor for your processing.

The other two areas we’ve been investing in are

Graviton (Graviton2 and Graviton2, now): powerful and efficient host-compute, and also

Our machine-learning chips: Inferentia (for inference) and Trainium (for training).

[\[Saidi 2021\]](#)

James Hamilton says “Graviton is one of our four semiconductor product lines”. Graviton is a central processing unit (CPU). Hamilton [says](#) “we’re now a leading semiconductor design house”. They launched the first Graviton CPU in 2018.

When Graviton processors were announced in November 2018, Joel Hruska investigated what it involved, saying: “It’s based on the Cortex-A72, with a maximum clock speed of 2.3GHz”.

Frumusani [wrote](#) on the Graviton2 in 2020.

## 12. What is HVM?

I have to try to get this correct. It's not [HMV](#), the British retailer that sells CDs and DVDs, and seems to go into administration every 2 years. HVM is just VM (virtual machine) with another word in front. That word is *Hardware*.

AWS offers two virtualisation types, Hardware Virtual Machine (HVM) and Paravirtual (PV). [This](#) 2022 article notes how, traditionally, AWS offered HVM for instances using the Windows OS and PV for those using Linux.

The Wikipedia article entitled “paravirtualization” tells us that it's a

virtualization technique that presents a software interface to the virtual machines which is similar, yet not identical, to the underlying hardware–software interface.

And it claims that the first use of the word “paravirtualization” was in a 2002 [paper](#) by Whitaker, Shaw and Gribble (*Denali: A Scalable Isolation Kernel*).

The prefix para- is often used for phenomena that are distinct from something which is, nevertheless, resembled. Paramilitary organisations are distinct from militaries but resemble them. Paradoxes are distinct from the ordinary) but arise from things we normally accept (*doxy* = accepted, normal). Paranormal ghosts supposedly are distinct from normal humans but resemble them. Let's explore *how* exactly paravirtualization is distinct from virtualisation.

The best known example of paravirtualization is the open-source product Xen. The other famous company for this is VMWare. Paravirtualisation ‘installs a guest OS... directly on the hypervisor’ (Golden 2007). Sometimes, it's often best couched in terms of “knowledge” possessed by the OS. [\[VMWare\]](#). Does the OS know it's being virtualised? With paravirtualization, it *does*. In fact, a crucial element of paravirtualization is that the guest OS (the operating system “on top”) is communicating with the hypervisor.

Paravirtualization involved *modifying the OS kernel*. We go in and tinker with it, so that the guest OS can communicate with the hypervisor. If the guest OS is communicating with the hypervisor, then clearly the OS *knows* it is being virtualized. Paravirtualization involves “modifying the OS kernel to replace non-virtualizable instructions with hypercalls that communicate directly with the virtualization layer hypervisor” ([VMWare](#)). Hypercall? The term is analogous to *system call*. It's a call, by the OS, for the hypervisor to perform some process.

Bernard Golden wrote a book called *Virtualization for Dummies* (2007). He writes:

Rather than modify the guest operating system at runtime via binary translation, paravirtualization requires that guest operating systems be modified prior to execution to include code that can interact with the paravirtualization hypervisor.

After modification, when a guest operating system accesses the processor or memory, the modified code interacts with the hypervisor, which then coordinates access to those resources.

The following helpful diagram is created by Brendan Gregg:

## AWS EC2 Virtualization Types

|                 |    | Importance<br>Most → Least |             |             |                   |                    |                    |                   |  |
|-----------------|----|----------------------------|-------------|-------------|-------------------|--------------------|--------------------|-------------------|--|
|                 |    | With                       | CPU, Memory | Network I/O | Local Storage I/O | Remote Storage I/O | Interrupts, Timers | Motherboard, Boot |  |
| Old<br>↓<br>New | VM | Fully Emulated             | VS          | VS          | VS                | VS                 | VS                 | VS                |  |
|                 | VM | Xen PV 3.0                 | P           | P           | P                 | P                  | VS                 | VS                |  |
|                 | VM | Xen HVM 3.0                | VH          | P           | P                 | P                  | VS                 | VS                |  |
|                 | VM | Xen HVM 4.0.1              | VH          | P           | P                 | P                  | P                  | VS                |  |
|                 | VM | Xen AWS 2013               | VH          | VH          | P                 | P                  | P                  | VS                |  |
|                 | VM | Xen AWS 2017               | VH          | VH          | VH                | P                  | P                  | VS                |  |
|                 | VM | AWS Nitro 2017             | VH          | VH          | VH                | VH                 | VH                 | VS                |  |
|                 | HW | AWS Bare Metal 2017        | H           | H           | H                 | H                  | H                  | H                 |  |
|                 |    | Bare Metal                 | H           | H           | H                 | H                  | H                  | H                 |  |

VM: Virtual Machine. HW: Hardware.

VS: Virt. in software. VH: Virt. in hardware. P: Paravirt. Not all combinations shown.

SR-IOV(net): ixgbe/ena driver. SR-IOV(storage): nvme driver.

<http://www.brendangregg.com/blog/2017-11-29/aws-ec2-virtualization-2017.html>



Wednesday, July 19, 2023

I derive great pleasure from telling folks which AWS services I absolutely do not recommend. But this might be the first time that AWS agrees with me ...

*(Want to read this article online or share it with your friends on Twitter or Slack? As always, you can find it right here: [It's Extremely Likely You Should Not Use GovCloud](#))*

---

## **It's Extremely Likely You Should Not Use GovCloud**

Until the recent Public Sector AWS Summit in Washington, D.C., I'd gone my



# Bibliography

1. [Official](#)
2. Unofficial
3. Critical
4. General

## Official

<https://brendangregg.com/blog/2017-11-29/aws-ec2-virtualization-2017.html>

Xen and the art of virtualisation. Available at:  
<https://www.cl.cam.ac.uk/research/srg/netos/papers/2003-xensosp.pdf>

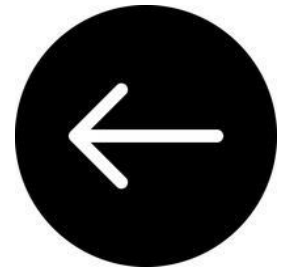
One VM to rule them all. Available at:  
[http://lafo.ssw.uni-linz.ac.at/papers/2013\\_OneVMToRuleThemAll.pdf](http://lafo.ssw.uni-linz.ac.at/papers/2013_OneVMToRuleThemAll.pdf)

Live migration of virtual machines. Available at:  
[https://github.com/papers-we-love/papers-we-love/blob/master/virtual\\_machines/live-migration-of-virtual-machines.pdf](https://github.com/papers-we-love/papers-we-love/blob/master/virtual_machines/live-migration-of-virtual-machines.pdf)

A Kivity, Y Kamay, D Laou, U Lublin, A Liguori. KVM: the Linux virtual machine monitor.

<https://archive.ph/20210824035659/https://aws.amazon.com/blogs/aws/happy-15th-birthday-amazon-ec2/>

# Module 2 - AMIs



[RETURN TO EC2](#)

QUESTION 6 OF 65

## 6. QUESTION

A company's application is running on Amazon EC2 instances in a single Region. In the event of a disaster, a solutions architect needs to ensure that the resources can also be deployed to a second Region.

Which combination of actions should the solutions architect take to accomplish this? (Select TWO.)

- ☐ Launch a new EC2 instance in the second Region and copy a volume from Amazon S3 to the new instance
- ☐ Copy an Amazon Machine Image (AMI) of an EC2 instance and specify the second Region for the destination
- ☒ Copy an Amazon Elastic Block Store (Amazon EBS) volume from Amazon S3 and launch an EC2 instance in the second Region using that EBS volume
- ☐ Detach a volume on an EC2 instance and copy it to an Amazon S3 bucket in the second Region
- ☒ Launch a new EC2 instance from an Amazon Machine Image (AMI) in the second Region

Incorrect

Explanation:

You can copy an Amazon Machine Image (AMI) within or across AWS Regions using the AWS Management Console, the AWS Command Line Interface or SDKs, or the Amazon EC2 API, all of which support the CopyImage action.

Using the copied AMI the solutions architect would then be able to launch an instance from the same EBS volume in the second Region.

**Note:** the AMIs are stored on Amazon S3, however you cannot view them in the S3 management console or work with them programmatically using the S3 API.

**CORRECT:** "Copy an Amazon Machine Image (AMI) of an EC2 instance and specify the second Region for the destination" is a correct answer.

**CORRECT:** "Launch a new EC2 instance from an Amazon Machine Image (AMI) in the second Region" is also a correct answer.

**INCORRECT:** "Detach a volume on an EC2 instance and copy it to an Amazon S3 bucket in the second Region" is incorrect. You cannot copy EBS volumes directly from EBS to Amazon S3.

**INCORRECT:** "Launch a new EC2 instance in the second Region and copy a volume from Amazon S3 to the new instance" is incorrect. You cannot create an EBS volume directly from Amazon S3.

**INCORRECT:** "Copy an Amazon Elastic Block Store (Amazon EBS) volume from Amazon S3 and launch an EC2 instance in the second Region using that EBS volume" is incorrect. You cannot create an EBS volume directly from Amazon S3.

References:

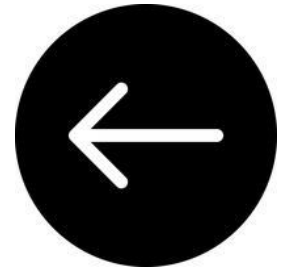
<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/CopyingAMIs.html>

# Bibliography

1. Official
2. Unofficial
3. Critical
4. General



# Module 3 – Instance Types



This module includes a discussion of high-performance computing.

[RETURN TO EC2](#)



# General Purpose instance types

|     |    |     |     |
|-----|----|-----|-----|
| A1  | T3 | T3a | T2  |
| M6g | M5 | M5a | M5n |
| M4  |    |     |     |
|     |    |     |     |
|     |    |     |     |



# Compute optimized



| C5 | C5n | C4 |  |
|----|-----|----|--|
|    |     |    |  |
|    |     |    |  |
|    |     |    |  |
|    |     |    |  |

# Memory optimized



| R5 | R5a | R5n | X1e |
|----|-----|-----|-----|
| X1 | Z1d |     |     |
|    |     |     |     |
|    |     |     |     |
|    |     |     |     |

# Memory optimized

## High Performance Computing (HPC)

### An Introduction to High Performance Computing

Sérgio Almeida \*

CENTRA, Departamento de Física, Instituto Superior Técnico,  
Universidade Técnica de Lisboa - UTL

September, 2013

#### **Abstract**

High Performance Computing (HPC) has become an essential tool in every researchers arsenal. Most research problems nowadays can be simulated, clarified or experimentally tested by using computational simulations. Researchers struggle with computational problems while they should be focusing on their research problems. Since most researchers have little-to-no knowledge in low-level computer science, they tend to look at computer programs as extensions of their minds and bodies instead of completely autonomous systems. Since computers don't work the same way as humans, the result is usually *Low Performance Computing* where HPC would be expected.

# Why is it called the “Elastic Fabric Adapter”?

Todd Lammle (in his 2009 textbook for the Network+ exam) tells us that Cisco have something called a “switch fabric”:

## **VLAN Identification Methods**

VLAN identification is what switches use to keep track of all those frames as they’re traversing a switch [fabric](#). It’s how switches identify which frames belong to which VLANs, and there’s more than one trunking method.

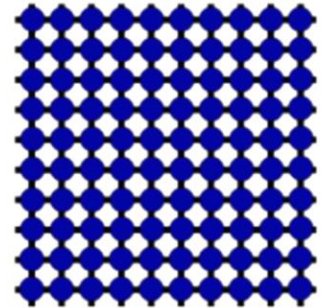
# Fabric computing

From Wikipedia, the free encyclopedia

**Fabric computing** or **unified computing** involves constructing a **computing fabric** consisting of interconnected nodes that look like a *weave* or a *fabric* when seen collectively from a distance.<sup>[1]</sup>

Usually the phrase refers to a consolidated [high-performance computing](#) system consisting of [loosely coupled storage](#), [networking](#) and [parallel processing](#) functions linked by [high bandwidth](#) interconnects (such as [10 Gigabit Ethernet](#) and [InfiniBand](#))<sup>[2]</sup> but the term has also been used to describe platforms such as the [Azure Services Platform](#) and [grid computing](#) in general (where the common theme is interconnected nodes that appear as a single logical unit).<sup>[3]</sup>

The fundamental components of fabrics are "nodes" (processor(s), memory, and/or peripherals) and "links" (functional connections between nodes).<sup>[2]</sup> While the term "fabric" has also been used in association with [storage area networks](#) and with [switched fabric networking](#), the introduction of [compute](#) resources provides a complete "unified" computing system.<sup>[citation needed]</sup> Other terms used to describe such fabrics include "unified fabric",<sup>[4]</sup> "data center fabric" and "unified data center fabric".<sup>[5]</sup>



## 5. QUESTION

A legacy tightly-coupled High Performance Computing (HPC) application will be migrated to AWS. Which network adapter type should be used?

- ☐ Elastic IP Address
- ☐ Elastic Fabric Adapter (EFA)
- ☐ Elastic Network Interface (ENI)
- ☐ Elastic Network Adapter (ENA)

Skip question

Check

Correct

Explanation:

An Elastic Fabric Adapter is an AWS Elastic Network Adapter (ENA) with added capabilities. The EFA lets you apply the scale, flexibility, and elasticity of the AWS Cloud to tightly-coupled HPC apps. It is ideal for tightly coupled app as it uses the Message Passing Interface (MPI).

**CORRECT:** "Elastic Fabric Adapter (EFA)" is the correct answer.

**INCORRECT:** "Elastic Network Interface (ENI)" is incorrect. The ENI is a basic type of adapter and is not the best choice for this use case.

**INCORRECT:** "Elastic Network Adapter (ENA)" is incorrect. The ENA, which provides Enhanced Networking, does provide high bandwidth and low inter-instance latency but it does not support the features for a tightly-coupled app that the EFA does.

**INCORRECT:** "Elastic IP Address" is incorrect. An Elastic IP address is just a static public IP address, it is not a type of network adapter.

References:

# Bibliography

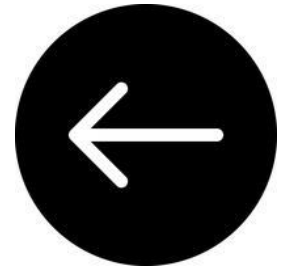
1. Official
2. Unofficial
3. Critical
4. General







# Module 4 – Instance Purchasing Options



[RETURN TO EC2](#)

# Coffee!



*For when you've run out  
of **capacity reservations!***

# Hibernating an instance



What, exactly, is unique about hibernating an instance?

AWS tell us:

When you hibernate an instance, Amazon EC2 signals the operating system to perform hibernation (suspend-to-disk). Hibernation saves the contents from the instance memory (RAM) to your Amazon Elastic Block Store (Amazon EBS) root volume. Amazon EC2 persists the instance's EBS root volume and any attached EBS data volumes. When you start your instance:

- The EBS root volume is restored to its previous state
- The RAM contents are reloaded



- The processes that were previously running on the instance are resumed
- Previously attached data volumes are reattached and the instance retains its instance ID

### 3. QUESTION

A company plans to make an Amazon EC2 Linux instance unavailable outside of business hours to save costs. The instance is backed by an Amazon EBS volume. There is a requirement that the contents of the instance's memory must be preserved when it is made unavailable.

How can a solutions architect meet these requirements?

- ☒ Stop the instance outside business hours. Start the instance again when required.
- ☐ Hibernate the instance outside business hours. Start the instance again when required.
- ☐ Use Auto Scaling to scale down the instance outside of business hours. Scale up the instance when required.
- ☐ Terminate the instance outside business hours. Recover the instance again when required.

Incorrect

Explanation:

When you hibernate an instance, Amazon EC2 signals the operating system to perform hibernation (suspend-to-disk). Hibernation saves the contents from the instance memory (RAM) to your Amazon Elastic Block Store (Amazon EBS) root volume. Amazon EC2 persists the instance's EBS root volume and any attached EBS data volumes. When you start your instance:

- The EBS root volume is restored to its previous state
- The RAM contents are reloaded
- The processes that were previously running on the instance are resumed
- Previously attached data volumes are reattached and the instance retains its instance ID

**CORRECT:** "Hibernate the instance outside business hours. Start the instance again when required" is the correct answer.

**INCORRECT:** "Stop the instance outside business hours. Start the instance again when required" is incorrect. When an instance is stopped the operating system is shut down and the contents of memory will be lost.

**INCORRECT:** "Use Auto Scaling to scale down the instance outside of business hours. Scale out the instance when required" is incorrect. Auto Scaling scales does not scale up and down, it scales in by terminating instances and out by launching instances. When scaling out new instances are launched and no state will be available from terminated instances.

**INCORRECT:** "Terminate the instance outside business hours. Recover the instance again when required" is incorrect. You cannot recover terminated instances, you can recover instances that have become impaired in some circumstances.

References:

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Hibernate.html>

Save time with our AWS cheat sheets:

<https://digitalcloud.training/amazon-ec2/>

#### 8. QUESTION

A company runs a large batch processing job at the end of every quarter. The processing job runs for 5 days and uses 15 Amazon EC2 instances. The processing must run uninterrupted for 5 hours per day. The company is investigating ways to reduce the cost of the batch processing job.

Which pricing model should the company choose?

☐ Dedicated Instances

☒ On-Demand Instances

☐ Spot Instances

☐ Reserved Instances

Correct

Explanation:

Each EC2 instance runs for 5 hours a day for 5 days per quarter or 20 days per year. This time duration is insufficient to warrant reserved instances as these require a commitment of a minimum of 1 year and the discounts would not outweigh the costs of having the reservations unused for a large percentage of time. In this case, there are no options presented that can reduce the cost and therefore on-demand instances should be used.

**CORRECT:** "On-Demand Instances" is the correct answer.

**INCORRECT:** "Reserved Instances" is incorrect. Reserved instances are good for continuously running workloads that run for a period of 1 or 3 years.

**INCORRECT:** "Spot Instances" is incorrect. Spot instances may be interrupted and this is not acceptable. Note that Spot Block is deprecated and unavailable to new customers.

**INCORRECT:** "Dedicated Instances" is incorrect. These do not provide any cost advantages and will actually be more expensive.

References:

<https://aws.amazon.com/ec2/pricing/>

Save time with our AWS cheat sheets:

<https://digitalcloud.training/amazon-ec2/>

Next

8. QUESTION

Amazon EC2 instances in a development environment run between 9am and 5pm Monday-Friday. Production instances run 24/7. Which pricing models should be used to optimize cost and ensure capacity is available? (Select TWO.)

☒ On-demand capacity reservations for the development environment

☒ Use Reserved instances for the production environment

☐ Use On-Demand instances for the production environment

☐ Use Reserved instances for the development environment

☐ Use Spot instances for the development environment

**Explanation:**

Capacity reservations have no commitment and can be created and canceled as needed. This is ideal for the development environment as it will ensure the capacity is available. There is no price advantage but none of the other options provide a price advantage whilst also ensuring capacity is available

Reserved instances are a good choice for workloads that run continuously. This is a good option for the production environment.

**CORRECT:** "On-demand capacity reservations for the development environment" is a correct answer.

**CORRECT:** "Use Reserved instances for the production environment" is also a correct answer.

**INCORRECT:** "Use Spot instances for the development environment" is incorrect. Spot Instances are a cost-effective choice if you can be flexible about when your applications run and if your applications can be interrupted. Spot instances are not suitable for the development environment as important work may be interrupted.

**INCORRECT:** "Use Reserved instances for the development environment" is incorrect as they require a long-term commitment which is not ideal for a development environment.

**INCORRECT:** "Use On-Demand instances for the production environment" is incorrect. There is no long-term commitment required when you purchase On-Demand Instances. However, you do not get any discount and therefore this is the most expensive option.

**References:**

<https://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/instance-purchasing-options.html>

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-capacity-reservations.html>

**15. QUESTION**

A solutions architect is creating a system that will run analytics on financial data for several hours a night, 5 days a week. The analysis is expected to run for the same duration and cannot be interrupted once it is started. The system will be required for a minimum of 1 year.

What should the solutions architect configure to ensure the EC2 instances are available when they are needed?

- ☐ Savings Plans
- ☐ On-Demand Instances
- ☒ On-Demand Capacity Reservations
- ☐ Regional Reserved Instances



Correct

Explanation:

On-Demand Capacity Reservations enable you to reserve compute capacity for your Amazon EC2 instances in a specific Availability Zone for any duration. This gives you the ability to create and manage Capacity Reservations independently from the billing discounts offered by Savings Plans or Regional Reserved Instances.

By creating Capacity Reservations, you ensure that you always have access to EC2 capacity when you need it, for as long as you need it. You can create Capacity Reservations at any time, without entering a one-year or three-year term commitment, and the capacity is available immediately.

The table below shows the difference between capacity reservations and other options:

|                  | Capacity Reservations  | Zonal Reserved Instances   | Regional Reserved Instances                                 | Savings Plans |
|------------------|--|--|---|---------------|
| Term             | No commitment required. Can be created and canceled as needed. | Requires a fixed one-year or three-year commitment                     |   |               |
| Capacity benefit | Capacity reserved in a specific Availability Zone.             |  | No capacity reserved.                                       |               |
| Billing discount | No billing discount. †   | Provides a billing discount.   |   |               |
| Instance Limits  | Your On-Demand Instance limits per Region apply.               | Default is 20 per Availability Zone. You can request a limit increase. | Default is 20 per Region. You can request a limit increase. | No limit.     |

CORRECT: "On-Demand Capacity Reservations" is the correct answer.

INCORRECT: "Regional Reserved Instances" is incorrect. This type of reservation does not reserve capacity.

INCORRECT: "On-Demand Instances" is incorrect. This does not provide any kind of capacity reservation.

INCORRECT: "Savings Plans" is incorrect. This pricing option does not provide a capacity reservation.

# TPN

- 43. ***Phenomenon1*** – the tendency of X to Y.
- 44. ***Phen2*** – the tendency of X to Y.
- 45. ***Phen3*** – the tendency of X to Y.
- 46. ***Phen4*** – the tendency of X to Y.
- 47. ***Phen5*** – the tendency of X to Y.
- 48. ***Phen6*** – the tendency of X to Y.
- 49. ***Phen7*** – the tendency of X to Y.
- 50. ***Phen8*** – the tendency of X to Y.
- 51. ***Phen9*** – the tendency of X to Y.
- 52. ***Phen10*** – the tendency of X to Y.

# Glossary

## Term1

Description of what term means here.

## Term2

Description of what term means here.

## Term3

Description of what term means here.

# Bibliography

- XXI. Official
- XXII. Unofficial
- XXIII. Critical
- XXIV. General

## XXI. Official

### [Surname1]

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

### [Surname1]

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

## XXII. Unofficial

### [Surname1]

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

### [Surname1]

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

## XXIII. Critical

### [Surname1]

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.

Available at:

<URL here>.

### [Surname1]

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.

Available at:

<URL here>.

## XXIV. General

### [Surname1]

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.

Available at:

<URL here>.

### [Surname1]

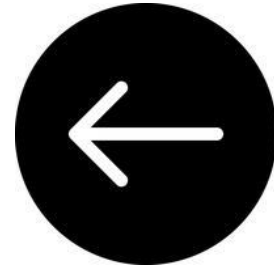
Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.

Available at:

<URL here>.



# Module 5 – Configuring your instance



[RETURN TO EC2](#)

## 1. What is Amazon Linux 2?

Amazon’s Linux-themed OS, “[Amazon Linux 2](#)” was announced on [26<sup>th</sup> June 2018](#), the very same day Japan [unveiled](#) a Hello-Kitten themed bullet train. It’s important to note that AWS had provided the Amazon Linux *AMI* since 2010.

But “Amazon Linux 2” is not supported any more. AWS [announced](#) “[Amazon Linux 2022](#)” in the winter of 2021.

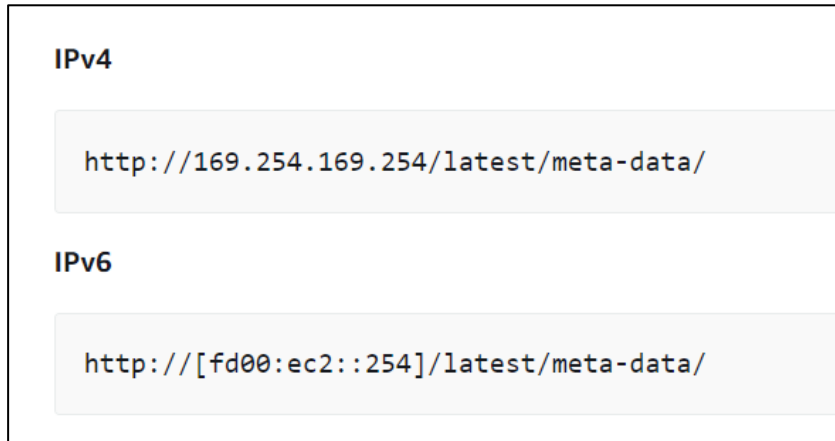
## 2. What are the instance metadata categories?

*Instance metadata* simple denotes data about your instance. For example, if I tell you the ID of the kernel used to launch an instance, I’ve given you some instance metadata. If I tell you the instance’s MAC address, I’ve given you some instance metadata. I might give you the public IPv4 address associated with the instance, the AWS Region in which the instance was launched, or the ID of the instance. These are all examples of instance metadata.

Or are they? Strictly, the examples I gave were examples of *types* of instance metadata. An example of actual instance metadata would be something like i-3983nj3902dejk (an example of an instance ID). So, the examples I gave were of types, or *categories*.

AWS lists 68 instance metadata categories, according to my most recent count. They are listed in the Documentation.

However, the categories can also be listed on the console. “To view all categories of instance metadata from within a running instance, use the following IPv4 or IPv6 URIs:



You’ll notice that within that URI (Universal Resource Indicator) there is an IP address. Specifically, it is 169.254.169.254. Please note that this IP address will only work from within the instance! We say that it is “link-local”. Wikipedia [tells us](#) that link-local addresses are

valid only for communications within the [subnetwork](#) that the host is connected to

Link-local addresses are not guaranteed to be unique beyond the sub-network.

Now, let me consider this psychological question: How can I remember that the address is 169.254.169.154? I have to use some feature of the concept of “metadata” as the feeder. The solution must not presuppose that I already know the answer and if it is bizarre, this is a virtue.

Metadata is knowledge *about* things. It is not data itself, it is somewhat external, and often called “data about data”. One area in which humanity wants to expand its knowledge is extra-terrestrial exploration. We’d circle the moon one hunnnnnndred times just to find out something new about that beautiful sphere. We landed on the moon in 1969. So, ~~moon~~data, metadata, gets me to 169. Why 254? In binary, this is **11111110** or:

1111 111 

### 3. What does IMDS stand for?

IMDS stands for Instance Metadata *Service*. We know what instance metadata is at this point. Examples of instance metadata categories include the public IPv4 address of the instance or the AWS Region in which the instance was launched.

The reason it's called a service is that it *is* a service. You use the terminal to input commands, and you're provided with answers. It's therefore referred to in the AWS documentation as the *instance metadata service* (IMDS).

### 4. What is IMDSv2?

In an AWS [blog post](#), Colm MacCarthaigh announced that a new version of the Instance Metadata Service would be rolled out. The post, written on 19<sup>th</sup> Nov 2019, explains why the IMDS was helpful:

The IMDS solved a big security headache for cloud users by providing access to temporary, frequently rotated credentials, removing the need to hardcode or distribute sensitive credentials t148resigned148icallyually or programatically.

Mark Ryland has even given a whole [presentation](#) on the Instance Metadata Service. Look, this isn't just a cool party trick, for you (a human) to type things into the console. The IMDS gives knowledge to *software*:

Attached locally to every EC2 instance, the IMDS runs on a special “link local” IP address of 169.254.169.254 that means only software running on the instance can access it.

The idea is that *applications* can access the IMDS:

For applications with access to IMDS, it makes available metadata about the instance, its network, and its storage.

The IMDS is about self-knowledge (as Maarek notes); it's about introspection (as Ryland [puts it](#)).





Mark Ryland (2019) explaining how the instance metadata service works

## 5. Retrieve dynamic data

In the documentation, AWS tells us that to retrieve dynamic data, we do the following:

To retrieve dynamic data from within a running instance, use the following URI.

```
http://169.254.169.254/latest/dynamic/
```

But what is “dynamic data”? All the AWS documentation says is this:

EC2 instances can also include *dynamic data*, such as an instance identity document that is generated when the instance is launched.

This is clearly data that changes in some way.

## 6. What is an *instance identity document*?

EC2 has this idea of an *instance identity document*. We’re told that:

You can retrieve the instance identity document from a running instance at any time.

The instance identity document includes the following information:

|    | Data                    | Description  |
|----|-------------------------|--|
| 1  | devpayProductCodes      | Deprecated.  |
| 2  | marketplaceProductCodes | The AWS Marketplace product code of the AMI used to launch the instance. |
| 3  | availabilityZone        | The Availability Zone in which the instance is running.                  |
| 4  | privateIp               | The private IPv4 address of the instance.                                |
| 5  | version                 | The version of the instance identity document format.                    |
| 6  | instanceId              | The ID of the instance.  |
| 7  | billingProducts         | The billing products of the instance.                                    |
| 8  | instanceType            | The instance type of the instance.                                       |
| 9  | accountId               | The ID of the AWS account that launched the instance.                    |
| 10 | imageId                 | The ID of the AMI used to launch the instance.                           |

|    |              |  |
|----|--------------|--|
| 11 | pendingTime  | The date and time that the instance was launched.                                |
| 12 | architecture | The architecture of the AMI used to launch the instance (i386   x86_64   arm64). |
| 13 | kernelId     | The ID of the kernel associated with the instance, if applicable.                |
| 14 | ramdiskId    | The ID of the RAM disk associated with the instance, if applicable.              |
| 15 | region       | The Region in which the instance is running.                                     |

Some of these items appear similar to instance metadata categories. For example, there was an instance metadata category called:

Place/region

And one of the items in the Instance Identity Document is “the Region in which the instance is running”.

## 7. Why are role credentials so important?

## 8. Manage software on your instance

It’s not true that a new EC2 instance is like a blank slate. Amazon Linux instances launch with repositories already installed. These instances come with software packages and utilities that are required for basic server operations.

There’s a command you can use to view the installed packages on your instance:

```
[ec2-user ~]$ yum list installed
```

As you can see, Amazon Linux instances manage their software using the yum package manager.

YUM was originally created to manage a physics department in North Caroline. Standing for Yellowdog Updater, Modified, it was started by Michael Stenner and Seth Vidal, who sadly died in a [bicycle crash](#) in 2003.

## I/O

### 9. scheduler

### 10. Set the time

## 11. Dynamic DNS

Dynamic (DNS) is useful for servers that change their IP address frequently. Recall that your paradigmatic DNS record is an A-record, which maps a domain name to an IP address. The resource record can be thought of as having the domain name on the lefthand side and the IP address on the right-hand side. A large number of computers needed names; to meet scalability issues, DHCP was invented:

The [Dynamic Host Configuration Protocol](#) (DHCP) allowed enterprises and [Internet service providers](#) (ISPs) to assign addresses to computers automatically as they powered up.

[Wikipedia]

DHCP was first defined in “RFC 1531 in October 1993” [\[Wikipedia\]](#). So, there are things called DHCP servers which dish out names. But if a computer has been dishd out a name, it needs to let the nameservers know about it:

The first implementations of *dynamic DNS* fulfilled this purpose: Host computers gained the feature to notify their

respective DNS server of the address they had received from a DHCP server or through self-configuration.

There are things called Dynamic DNS providers.

And the idea is this:

You can use a dynamic DNS provider with Amazon EC2 and configure the instance to update the IP address associated with a public DNS name each time the instance starts.

## 12. Core, I'm tired

## 13. IMDSv2

What on earth

## 14. What is Elastic Inference?

You know what inference is. If I tell you that 5 people went into Denny's and two people left, you can infer that 3 people remain. We're now getting machines to make inferences like this, within machine learning (ML). Elastic Inference has something to do with machine learning.

AWS says it's a *resource*, namely:

“a resource you can attach to your Amazon EC2 CPU instances to accelerate your deep learning (DL) inference workloads

So, we're talking about deep learning, the subdomain of ML, which tends to use artificial neural networks, the “deep” referring to the many, many layers of the network. AWS [announced](#) Elastic Inference on 28<sup>TH</sup> Nov 2018, which can only be described as a sort of Wacky Wednesday, on which AWS announced the most services they ever have in one day.

Now, GPU acceleration is the process of using a Graphical Processing Unit (GPU) in addition to a *Central* Processing Unit (CPU) in order to speed up data-intensive applications. This addition of GPU seems to be what you get with EI. Amazon write:

Amazon Elastic Inference allows you to attach low-cost GPU-powered acceleration to Amazon EC2 and SageMaker instances or Amazon ECS tasks, to reduce the cost of running deep learning inference by up to 75%.

Elastic Inferences is currently presented as providing a Third Way between two undesirable options:

(1) Firstly, standalone GPU instances are typically designed for model training - not for inference.

CPU instances are not specialized for matrix operations, and thus are often too slow for d

(2) deep learning inference.

Part of the reason (1) is unacceptable stems from the unique character of inferences jobs. They ‘usually process a single input in real time, and thus consume a small amount of GPU compute’. What they don’t do it process many samples in parallel.

AWS provide an example, involving a P3 instance type. They say that this provides “a range of up to 1000 TFLOPS”. The point is that this range is large. TFLOPS, by the way, stands for

Trillion Floating Point Operations per Second.

In contrast, “Elastic Inference can provide as little as a single-precision TFLOPS... or as much as 32 mixed-precision TFLOPS”. It calls this “a much more appropriate range”.

[Here](#) is Andy Jassy announcing Elastic Inference in 2018.

What is “user data”? In the context of EC2, “user data” refers to data that *you provide*. So, it’s somewhat intuitive. However, it is nevertheless, a ‘thing’. The console takes you through a number of steps to configure your EC2 instance, and just because you’re providing data (your preferences) at each point, doesn’t mean you’re providing User Data. No, the console has a particular

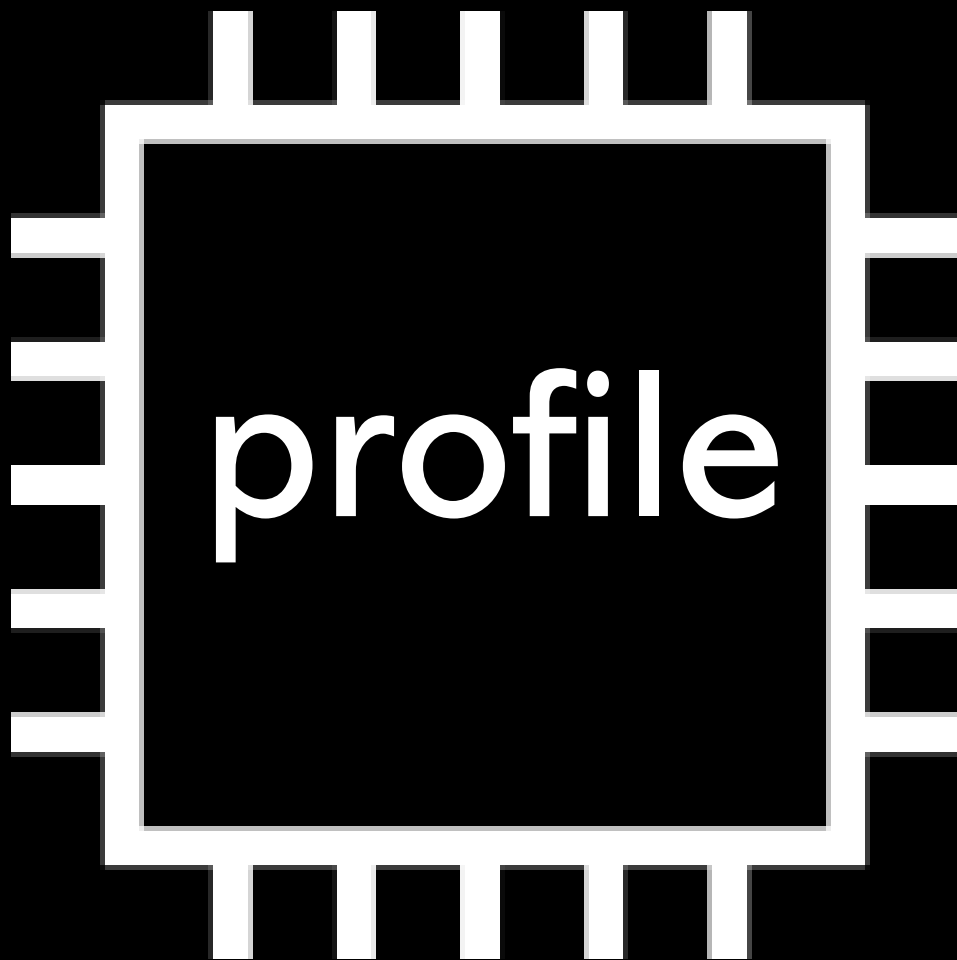
section called User Data and there are two radio buttons. One says “as text” and the other “as file”.

The text or file we provide is going to automate certain things. There is often talk of “User Data *Scripts*”. [Here](#) is Stephane Maarek.

When working with instance user data, keep the following in mind:

- User data must be base64-encoded. The Amazon EC2 console can perform the base64-encoding for you or accept base64-encoded input.
- User data is limited to 16 KB, in raw form, before it is base64-encoded. The size of a string of length  $n$  after base64-encoding is  $\text{ceil}(n/3)*4$ .
- User data must be base64-decoded when you retrieve it. If you retrieve the data using instance met'data or the console, it's decoded for you automatically.
- User data is treated as opaque data: what you give is what you get back. It is up to the instance to be able to interpret it.
- If you stop an instance, modify its user data, and start the instance, the updated user data is not run when you start the instance.

# Instance





# What on **earth** is an “instance profile”?

AWS are very subtle in providing the definition of an instance profile. They tell you lots of things about instance profile (formal characteristics), but never tell you what an instance profile actually *is*.

Here are how they’re described in the EC2 documentation:

Amazon EC2 uses an instance profile as a container for an IAM role.

That’s a start, but it’ll need some elaboration. We’re further told “an instance profile can contain only one IAM role. This limit cannot be increased”.



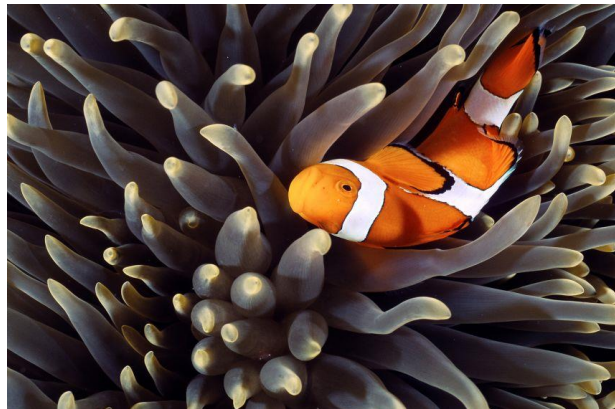
Two sources have helped me here. The first is a 2020 Wordpress article. See [Kichik]. The second is an article published on Medium. See **[Medium 1]**.

He writes:

EC2 **instance profiles** allow you to attach an IAM role to an EC2 instance.

This allows any application running on the instance to access certain resources defined in the role policies. Instance profiles are usually recommended over configuring a static access key as they are considered more secure and easier to maintain.

It's clear that instance profiles are things assumed by EC2 instances. Strictly, EC2 instances are *not* IAM roles. If they were, they would be called IAM roles. However, they do share similar properties (they are assumed, and can be discarded etc). Instance profiles and IAM roles have a symbiotic relationship. In other words, they help each other out. Specifically, the instance profile *enables* (allows you) to attach an IAM role to an EC2 instance.



*Symbiosis* – Instance profiles work closely with IAM roles

The question is now raised about how, exactly, an instance role (possessed by an instance) enables an IAM role to be assumed. To understand the mechanism, I refer back to X who poses the right questions and then answers them:

But how does an application running on EC2 use this **instance profile**? Where do the credentials come from? How does this work without any application configuration change?

EC2 shares the credentials with the application through the **metadata service**. Each instance can access [this service](#) through `http://169.254.169.254` (unless disabled) and EC2 will expose instance-specific information there. The exposed information includes AMI id, user-data, instance id and IPs, and more.

Hhhhh

The instance profile credentials are exposed on <http://169.254.169.254/latest/meta-data/iam/security-credentials/>.

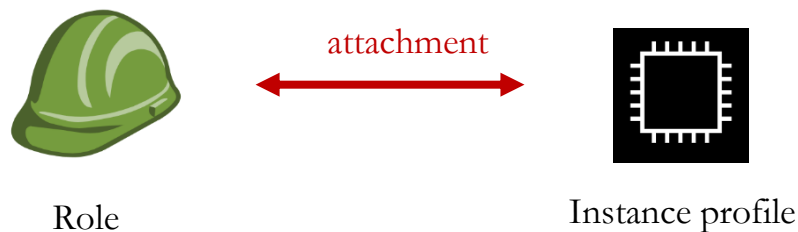
When you `curl` this URL on an EC2 instance, you will get the name of the instance profile attached to the instance.

dgsdfgds

When you `curl` the same URL with the instance profile name at the end, you get the temporary credentials as JSON. The metadata service will return access key id, secret access key, a token, and the expiration date of the temporary credentials. Behind the scenes it is using [STS AssumeRole](#).

```
[ec2-user ~]$ curl -w "\n" http://169.254.169.254/latest/meta-data/iam/security-credentials/SomeRole
SomeRole
[ec2-user ~]$ curl -w "\n" http://169.254.169.254/latest/meta-data/iam/security-credentials/SomeRole
{
  "Code" : "Success",
  "LastUpdated" : "2020-09-04T21:42:41Z",
  "Type" : "AWS-HMAC",
  "AccessKeyId" : "AS....",
  "SecretAccessKey" : "pkqh....",
  "Token" : "IQoJb...",
  "Expiration" : "2020-09-05T04:18:05Z"
}
[ec2-user ~]$
```

It's as if there is a role **attached** to the instance profile:



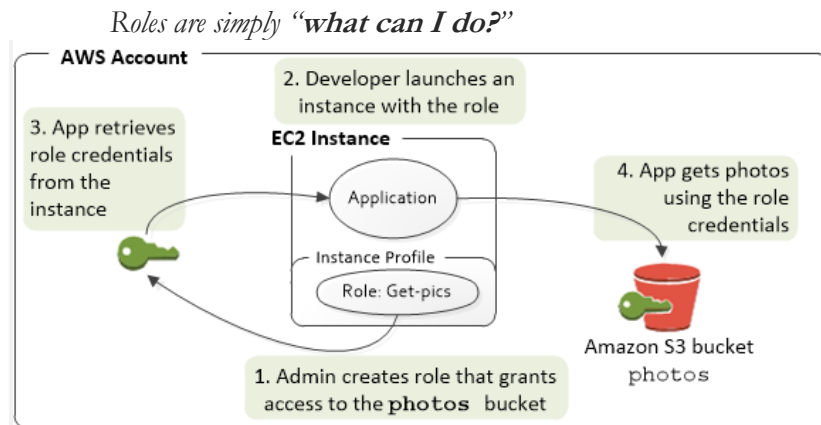
I hope you are now have a somewhat clearer conception of instance profiles. We are going to try and clarify things further.

In 2020, an article was published by a DevOps professional, which explains the nature of EC2 instance profiles in a clear manner. See [Medium 1] in the bibliography. I want to provide an extended extract, in which the distinction between roles and instance profiles is made clear. We break authentication down into two elements: “who am I?” and “What am I permitted to do?” The author states that instance profiles focus on the “who am I?”. In fact, instance profiles are *limited* to the “who am I?” element.

There are two key parts of any authentication system, not just IAM:

- Who am I?
- What am I permitted to do?

When you create an IAM user, those two questions are mixed into a single principle: the **IAM user has both properties**. It has credentials in which someone can “be” the user, and it has permissions attached to allow the user to perform actions.



They provide a mechanism to define a collection of permissions. You assign Managed Policies and inline policies to the role to give it permissions to act. But it, in and of itself, isn't a particular person or thing. It does not define “who am I?”

As you can see, IAM roles do not focus on the “who am I?” element. The whole point is that they can be applied to one user and then another, and so on. With instance profiles, on the other hand, they do answer the question “who am I?”. The author states:

An instance profile, on the other hand, defines “who am I?” Just like an IAM user represents a person, **an instance profile represents EC2 instances**.

The only permissions an EC2 instance profile has is the power to assume a role.

So the EC2 instance runs under the EC2 instance profile, defining “who” the instance is. It then “assumes” the IAM role, which ultimately gives it any real power.

In the entity known as an IAM user, the two questions are mixed into one and cannot be untangled. In other words, the IAM user concerns itself with both “who am I?” as well as “what am I permitted to do?”. The

instance profile is similar to the IAM user in that it provides an answer to “who am I?”. However, the instance profile is unlike the IAM user in this respect: the instance profile provides no information about “what am I permitted to do?”

The instance profile has not made IAM roles redundant. EC2 instances still require IAM roles to be able to do things. We can say, however that:

A necessary condition of an EC2 instance assuming an IAM role is that the EC2 instance has an instance profile.

IAM roles are necessarily silent on the question “who am I?” This is their strength—roles are supposed to be portable. This is precisely why it’s not enough for an instance to merely have an IAM role. An EC2 instance must also have an instance profile to provide that information about “who am I?”

Now that we have the concept of an instance profile clear, let’s look at some of the dirty practicalities about how instance profiles are assigned.

## Dirty Details

Things work slightly differently depending on whether you’re using the Management Console or the CLI.



When you create an IAM Role for EC2 using the AWS Management Console, it creates both an EC2 instance profile as well as an IAM role.

However, if you are using the AWS CLI, SDKs, or CloudFormation, you will need to explicitly define both:

- An IAM role with policies and permissions, and
- An EC2 instance profile specifying which roles it can assume

[Medium 1]

## Commands related to instance profiles

There are a few commands, which you can use on the command line interface (CLI), which are likely to be useful in this context. We have:

**Create-instance-profile**

**Add-role-to-instance-profile**

**Associate-iam-instance-profile**

## Order of attachment

A helpful official AWS video to watch is found [here](#). In this brief tutorial, an instance profile is created on the console and then using the CLI.

On the CLI, first, an instance profile is created. What's I find interesting is that it isn't immediately attached to the EC2 instance. Rather, we attach a role to the instance profile. Only *then* do we attach the instance profile (with role attached to it) to the instance profile.

### Instance profiles

Amazon EC2 uses an *instance profile* as a container for an IAM role. When you create an IAM role using the IAM console, the console creates an instance profile automatically and gives it the same name as the role to which it corresponds. If you use the Amazon EC2 console to launch an instance with an IAM role or to attach an IAM role to an instance, you choose the role based on a list of instance profile names.

If you use the AWS CLI, API, or an AWS SDK to create a role, you create the role and instance profile as separate actions, with potentially different names. If you then use the AWS CLI, API, or an AWS SDK to launch an instance with an IAM role or to attach an IAM role to an instance, specify the instance profile name.

An instance profile can contain only one IAM role. This limit cannot be increased.

For more information, see [Instance Profiles](#) in the *IAM User Guide*.



# Using an instance profile to create a presigned URL

What's interesting is that you sort of can do some things with an instance profile. For example, the instance profile is a suitable credential for creating a presigned URL to access an S3 bucket. The documentation on S3 writes:

The following are credentials that can be used to create a presigned URL:

- IAM instance profile: Valid up to 6 hours.
- AWS Security Token Service: Valid up to 36 hours when signed with permanent credentials, such as the credentials of the AWS account root user or an IAM user.
- IAM user: Valid up to 7 days when using AWS Signature Version 4.

To create a presigned URL that's valid for up to 7 days, first designate IAM user credentials (the access key and secret key) to the SDK that you're using. Then, generate a presigned URL using AWS Signature Version 4.

## 21. QUESTION

An application runs on Amazon EC2 instances that use an Amazon SQS queue and an Amazon DynamoDB table. The application processes highly confidential information and the connectivity between these AWS services should be private.

Which combination of steps should the security engineer take to meet this requirement? (Select THREE.)

☒ Create an interface VPC endpoint for Amazon SQS.

☐ Configure a connection to Amazon S3 through an AWS Managed VPN.

☒ Create a gateway VPC endpoint for Amazon DynamoDB.

☒ Modify the IAM role applied to the EC2 instance profile to allow outbound traffic to the interface endpoints.

☐ Configure a connection to Amazon S3 through AWS Direct Connect.

☐ Modify the endpoint policies on all VPC endpoints. Specify the SQS and DynamoDB resources that the application uses.

## Incorrect

## Explanation:

A VPC endpoint enables connections between a virtual private cloud (VPC) and supported services, without requiring that you use an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Therefore, your VPC is not exposed to the public internet.

For Amazon S3 and DynamoDB you must use a gateway VPC endpoint and for other AWS services you create an interface VPC endpoint. In both cases you can apply policies to control access to the service(s) to which you are connecting.

**CORRECT:** "Create an interface VPC endpoint for Amazon SQS" is a correct answer (as explained above.)

**CORRECT:** "Create a gateway VPC endpoint for Amazon DynamoDB" is also a correct answer (as explained above.)

**CORRECT:** "Modify the endpoint policies on all VPC endpoints. Specify the SQS and DynamoDB resources that the application uses" is also a correct answer (as explained above.)

**INCORRECT:** "Configure a connection to Amazon S3 through AWS Direct Connect" is incorrect.

You cannot connect from a VPC to S3 using Direct Connect (DX). You create DX connections between on-premises data centers and a VPC.

**INCORRECT:** "Configure a connection to Amazon S3 through an AWS Managed VPN" is incorrect.

You cannot create a VPN between S3 and a VPC. You must use VPC endpoints instead.

**INCORRECT:** "Modify the IAM role applied to the EC2 instance profile to allow outbound traffic to the interface endpoints" is incorrect.

The IAM role attached to an instance profile specifies the permissions the instance has for other AWS services. For controlling traffic across a VPC endpoint you should instead use endpoint policies.

What on earth is  
“**SR-IOV**”?

Elastic Fabric  
Adapter

# Placement groups

There are three kinds of placement group:

1. Cluster groups
2. Spread groups
3. Partition groups

## Placement groups

AWS have this notion of “placement groups”. There are three different ways of “placing” your EC2 instance. They are known as CLUSTERED, SPREAD, and PARTITION. Each has advantages and disadvantages.



**CLUSTER**



**SPREAD**



**PARTITION**

I will admit, the differences between these groups are not always made precise, but commentators. CLUSTERING instances involves grouping them into a single AZ (Availability Zone). This has advantages: latencies for communication between instances start to become small. If the majority of communication is going to be between instances, then choose the CLUSTER placement group. The “cluster” placement group can be easily distinguished from the other placement groups. Where many AWS educators struggle is distinguishing SPREAD from PARTITION, so let’s try to improve this. We’ll start by looking at the official documentation, which states:

When you launch a new EC2 instance, the EC2 service attempts to place the instance in such a way that all of your instances are spread out across underlying hardware to minimize correlated failures. You can use *placement groups* to influence the placement of a group of *interdependent* instances to meet the needs of your workload. Depending on the type of workload, you can create a placement group using one of the following placement strategies:

The above reminds us that these PLACEMENT GROUPS are really for *interdependent* instances. Multi-tier architectures seem to be a good example of instances that are dependent on one another. Data is inputted in one tier (perhaps the web tier), then passed to another tier for processing. Perhaps, finally, an instance is depended on to be used to write data to a database. Can you think of cases where, although multiple instances are used, they are not dependent on one another?

They then go on to list the three PLACEMENT GROUPS:

When you launch a new EC2 instance, the EC2 service attempts to place the instance in such a way that all of your instances are spread out across underlying hardware to minimize correlated failures. You can use *placement groups* to influence the placement of a group of *interdependent* instances to meet the needs of your workload. Depending on the type of workload, you can create a placement group using one of the following placement strategies:

- *Cluster* – packs instances close together inside an Availability Zone. This strategy enables workloads to achieve the low-latency network performance necessary for tightly-coupled node-to-node communication that is typical of HPC applications.
- *Partition* – spreads your instances across logical partitions such that groups of instances in one partition do not share the underlying hardware with groups of instances in different partitions. This strategy is typically used by large distributed and replicated workloads, such as Hadoop, Cassandra, and Kafka.
- *Spread* – strictly places a small group of instances across distinct underlying hardware to reduce correlated failures.

There is no charge for creating a placement group.

The “cluster” placement group is easy enough to understand. However, SPREAD and PARTITION blur into one another. For “spread”, they write:

Strictly places a small group of instances across distinct **underlying hardware** to reduce correlated failures.

The idea is this: if hardware in one portion fails, another portion is not affected. This is indicated by the words “distinct” and “hardware”. It is therefore problematic that AWS choose to characterise the “partition” placement group by saying the instances:

Do not share the **underlying hardware**.

Clearly, the documentation writers have chosen to use the same, single idea—that of using distinct hardware—to characterise both placement groups. It is for this reason that these two placement groups are hard to differentiate. Not surprisingly, others have noted this:

## AWS EC2 Placement Groups: Partition vs Spread

Asked 3 years, 3 months ago   Modified 4 months ago   Viewed 7k times



Is there a good comparison table for the two types of AWS EC2 Placement Groups?

19


- Partition
- Spread




I have read the [AWS Documentation](#) but I am still a bit confused.

5



 amazon-web-services

 amazon-ec2





41



Firstly, a Rack server, is a computer dedicated to be used as a server and designed to be installed in a framework called a rack. Each rack has its own network and power source.

In **Cluster Placement group**, all instances are placed within a rack. If the rack fails (hardware failure), all instances fails at the same time. Hence, this is not suitable for High Availability or mission critical applications. But ideal for High Performance applications, as all the instances are in very close proximity to each other.

In **Spread Placement group**, each instance is placed in its own distinct rack. Each rack has at most one instance. A rack failure (hardware failure) will not affect more than one instance. Hence, this is ideal for High Availability or mission critical applications. But not really suitable for High Performance applications, as the instances are spread much further apart.

In **Partition Placement group**, each partition represents a rack. If a rack fails (hardware failure), it may affect multiple instances on that rack, but only within that partition. This way, failure of one partition is isolated from the rest of the partitions. So, if you have replication in other partitions, then your data will be safe. This placement group strikes a balance between High Performance and High Availability. This will be good for Big data applications like HDFS, HBase, Cassandra, Kafka, etc. which needs High Performance, but must also be Fault Tolerant at the same time.

Share Improve this answer Follow

edited Sep 12, 2021 at 2:36

answered Jan 16, 2020 at 4:28



Vishnu Vivek

1,479 ● 1 ● 17 ● 27

This makes thing clearer to me. It will be helpful to think of SPREAD as at the extreme end. Fault tolerance is the absolute priority. No thought is given to the performance cost which follows from having instances spread in far apart corners of the data centre. PARTITION is a kind of middle-ground. It balances a concern for tolerance to faults with a concern for performance. SPREAD—it seems to me—puts fault tolerance on a high alter. Let me explain this further.

Let us define a partition in the following way. A partition is

a unit which fails completely, and independently.

This informal definition will suffice for this discussion. With the “partition” placement group, these units can contain multiple instances. With the “spread” placement group, these units can



contain, *at most*, one instance. This is the difference between placement and spread. AWS confirms this:

## Partition placement groups

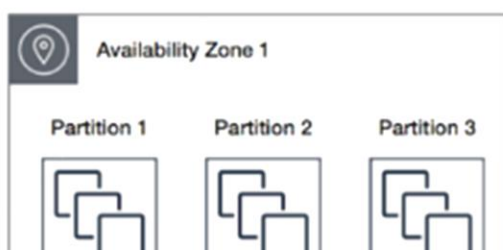
Partition placement groups help reduce the likelihood of correlated hardware failures for your application. When using partition placement groups, Amazon EC2 divides each group into logical segments called partitions. Amazon EC2 ensures that each partition within a placement group has its own set of racks. Each rack has its own network and power source. No two partitions within a placement group share the same racks, allowing you to isolate the impact of hardware failure within your application.

So, we create this artificial entity: the PARTITION. This is nothing more than a *logical* grouping of instances. It directs how we may place them. One rule is as follows:

- (i) Each partition must have its own set of racks.

Of course, a partition might contain multiple instances. So, it is perfectly in line with the above rule that multiple instances *share* a rack. Clearly, hardware failures are contained to partitions:

The following image is a simple visual representation of a partition placement group in a single Availability Zone. It shows instances that are placed into a partition placement group with three partitions—**Partition 1**, **Partition 2**, and **Partition 3**. Each partition comprises multiple instances. The instances in a partition do not share racks with the instances in the other partitions, allowing you to contain the impact of a single hardware failure to only the associated partition.



It's important to note that these entities—partitions—can be placed in different AZs. AWS tell us:

A partition placement group can have partitions in multiple Availability Zones in the same Region. A partition placement group can have a maximum of seven partitions per Availability Zone. The number of instances that can be launched into a partition placement group is limited only by the limits of your account.

The above seems to imply that, although these PARTITIONS can be in different AZs, they must be in the same region. It seems that AWS will try to evenly distribute the instances across your partitions. Yet at the same time, you have the capacity can specify that an instance be placed into a particular partition. It's not clear whether the aforementioned capacity can be used to cause instances to be placed unevenly (can you have one partition group with *one* instance within it, and another partition with *seven* within it?):

Partition placement groups can be used to deploy large distributed and replicated workloads, such as HDFS, HBase, and Cassandra, across distinct racks. When you launch instances into a partition placement group, Amazon EC2 tries to distribute the instances evenly across the number of partitions that you specify. You can also launch instances into a specific partition to have more control over where the instances are placed.

With the SPREAD placement group, single instances constitute the partition, as it were. We're told:

### Spread placement groups

A spread placement group is a group of instances that are each placed on distinct hardware.

Spread placement groups are recommended for applications that have a small number of critical instances that should be kept separate from each other. Launching instances in a spread level placement group reduces the risk of simultaneous failures that might occur when instances share the same equipment. Spread level placement groups provide access to distinct hardware, and are therefore suitable for mixing instance types or launching instances over time.

That first sentence (in the above) is just dire, because that could equally describe the partition placement group. Anyhow, it seems that PARTITION is for large numbers of instances, while SPREAD is for *small* numbers of instances. We were told, above, after all, that the PARTITION placement group is helpful for “large distributed and replicated workloads, such as HDFS, HBase and Cassandra”. How can I remember that the SPREAD placement group is for small numbers of instances (and not the PARTITION)?

# SPREAD

# Small

# numbers of instances

## 3. QUESTION

A security team is concerned about a possible vulnerability affecting the instance metadata service. The team requires that all existing and new Amazon EC2 instances must use version 2 of the instance metadata service (IMDSV2).

Which combination of steps should the security team take to complete the migration to IMDSV2 in the AWS environment? (Select TWO.)

- ☐ Update Network ACLs to deny all inbound traffic to the 169.254.169.254 IP address for the HTTP protocol.
- ☐ When using the `ec2:runinstances` API action set the `--metadata-options HttpTokens` option to `"required"`.
- ☐ Update existing instances using the `"ec2 modify-instance-metadata-options"` commands from the AWS CLI with the `"HttpEndpoint disabled"` option.
- ☐ When using the `ec2:runinstances` API action set the `--metadata-options HttpTokens` option to `"disabled"`.
- ☐ Update existing instances using the `"ec2 modify-instance-metadata-options"` commands from the AWS CLI with the `"HttpTokens required"` option.

A mock exam question on the Instance Metadata Service (IMDS). This is from the Security Speciality exam—courtesy of the training course by Neal Davis. I found this question very difficult.

### 3. QUESTION

A security team is concerned about a possible vulnerability affecting the instance metadata service. The team requires that all existing and new Amazon EC2 instances must use version 2 of the instance metadata service (IMDSV2).

Which combination of steps should the security team take to complete the migration to IMDSV2 in the AWS environment? (Select TWO.)

- ☐ Update Network ACLs to deny all inbound traffic to the 169.254.169.254 IP address for the HTTP protocol.
- ☐ When using the `ec2:runinstances` API action set the `--metadata-options HttpTokens` option to `"required"`.
- ☐ Update existing instances using the `"ec2 modify-instance-metadata-options"` commands from the AWS CLI with the `"HttpEndpoint disabled"` option.
- ☐ When using the `ec2:runinstances` API action set the `--metadata-options HttpTokens` option to `"disabled"`.
- ☒ Update existing instances using the `"ec2 modify-instance-metadata-options"` commands from the AWS CLI with the `"HttpTokens required"` option.

Incorrect

Explanation:

You can access instance metadata from a running instance using one of the following methods:

- Instance Metadata Service Version 1 (IMDSv1) – a request/response method
- Instance Metadata Service Version 2 (IMDSv2) – a session-oriented method

By default, you can use either IMDSv1 or IMDSv2, or both. The instance metadata service distinguishes between IMDSv1 and IMDSv2 requests based on whether, for any given request, either the PUT or GET headers, which are unique to IMDSv2, are present in that request.

You can configure the instance metadata service on each instance such that local code or users must use IMDSv2. When you specify that IMDSv2 must be used, IMDSv1 no longer works.

IMDSv2 uses session-oriented requests. With session-oriented requests, you create a session token that defines the session duration, which can be a minimum of one second and a maximum of six hours.

You can require the use of IMDSv2 by setting the `HttpTokens` parameter to `"required"` for the `ec2 run instances` action and by modifying existing instances using the CLI. Please check the reference link below for more detailed information on transitioning to IMDSv2.



**CORRECT:** "When using the ec2:runinstances API action set the "--metadata-options HttpTokens" option to "required"" is a correct answer (as explained above.)

**CORRECT:** "Update existing instances using the "ec2 modify-instance-metadata-options" commands from the AWS CLI with the "HttpTokens required" option" is also a correct answer (as explained above.)

**INCORRECT:** "Update existing instances using the "ec2 modify-instance-metadata-options" commands from the AWS CLI with the "HttpEndpoint disabled" option" is incorrect.

Setting the endpoint to disabled will prevent the instance metadata service from working completely.

**INCORRECT:** "Update Network ACLs to deny all inbound traffic to the 169.254.169.254 IP address for the HTTP protocol" is incorrect.

This is the IP address on which the instance metadata service runs and is a local address used from within the instance itself. Therefore, updating security groups and network ACLs will not achieve anything.

**INCORRECT:** "When using the ec2:runinstances API action set the "--metadata-options HttpTokens" option to "disabled" is incorrect.

This is not a valid option and does not achieve the stated objectives.



API action

Tokens are required!

**REQUIRED**



**Modify-instance-metadata-options**

# Configure the instance metadata options

[PDF](#) | [RSS](#)

Instance metadata options allow you to configure new or existing instances to do the following:

- Require the use of IMDSv2 when requesting instance metadata
- Specify the PUT response hop limit
- Turn off access to instance metadata

You can also use IAM condition keys in an IAM policy or SCP to do the following:

- Allow an instance to launch only if it's configured to require the use of IMDSv2
- Restrict the number of allowed hops
- Turn off access to instance metadata

## Note

You should proceed cautiously and conduct careful testing before making any changes. Take note of the following:

- If you enforce the use of IMDSv2, applications or agents that use IMDSv1 for instance metadata access will break.
- If you turn off all access to instance metadata, applications or agents that rely on instance metadata access to function will break.
- For IMDSv2, you must use `/latest/api/token` when retrieving the token.

## Topics

- [Configure instance metadata options for new instances](#)
- [Modify instance metadata options for existing instances](#)

## Configure instance metadata options for new instances

You can require the use of IMDSv2 on an instance when you launch it. You can also create an IAM policy that prevents users from launching new instances unless they require IMDSv2 on the new instance.

[New console](#)

[Old console](#)

[AWS CLI](#)

[AWS CloudFormation](#)

### To require the use of IMDSv2 on a new instance

- When launching a new instance in the Amazon EC2 console, expand **Advanced details**, and do the following:
  - For **Metadata accessible**, choose **Enabled**.
  - For **Metadata version**, choose **V2 only (token required)**.

For more information, see [Advanced details](#).



### To enforce the use of IMDSv2 on all new instances

To ensure that IAM users can only launch instances that require the use of IMDSv2 when requesting instance metadata, you can specify that the condition to require IMDSv2 must be met before an instance can be launched. For the example IAM policy, see [Work with instance metadata](#).

### Configure IPv4 and IPv6 endpoints

By default, the IPv6 endpoint is disabled. This is true even if you are launching an instance into an IPv6-only subnet. You can choose to enable this endpoint at instance launch. The IPv6 endpoint for IMDS is only accessible on [Instances built on the Nitro System](#). For more information about the metadata options, see [run-instances](#) in the *AWS CLI command reference*. The following example shows you how to enable the IPv6 endpoint for IMDS.

```
aws ec2 run-instances
--image-id ami-0abcdef1234567890
--instance-type t3.Large
...
--metadata-options "HttpEndpoint=enabled,HttpProtocolIpv6=enabled"
```



New console

Old console

AWS CLI

AWS CloudFormation

### To turn off access to instance metadata

- To ensure that access to your instance metadata is turned off, regardless of which version of the instance metadata service you are using, launch the instance in the Amazon EC2 console with the following specified under **Advanced details**:
  - For **Metadata accessible**, choose **Disabled**.

For more information, see [Advanced details](#).

## Modify instance metadata options for existing instances

You can require the use of IMDSv2 on an existing instance. You can also change the PUT response hop limit and turn off access to instance metadata on an existing instance. You can also create an IAM policy that prevents users from modifying the instance metadata options on an existing instance.

Currently only the AWS SDK or AWS CLI support modifying the instance metadata options on existing instances. You can't use the Amazon EC2 console for modifying instance metadata options.

### To require the use of IMDSv2

You can opt in to require that IMDSv2 is used when requesting instance metadata. Use the [modify-instance-metadata-options](#) CLI command and set the `http-tokens` parameter to `required`. When you specify a value for `http-tokens`, you must also set `http-endpoint` to `enabled`.

```
aws ec2 modify-instance-metadata-options \
  --instance-id i-1234567898abcdef0 \
  --http-tokens required \
  --http-endpoint enabled
```



### To change the PUT response hop limit

For existing instances, you can change the settings of the PUT response hop limit. Use the [modify-instance-metadata-options](#) CLI command and set the `http-put-response-hop-limit` parameter to the required number of hops. In the following example, the hop limit is set to `3`. Note that when specifying a value for `http-put-response-hop-limit`, you must also set `http-endpoint` to `enabled`.

```
aws ec2 modify-instance-metadata-options \
  --instance-id i-1234567898abcdef0 \
  --http-put-response-hop-limit 3 \
  --http-endpoint enabled
```



### To restore the use of IMDSv1 on an instance using IMDSv2

You can use the [modify-instance-metadata-options](#) CLI command with `http-tokens` set to `optional` to restore the use of IMDSv1 when requesting instance metadata.

```
aws ec2 modify-instance-metadata-options \  
  --instance-id i-1234567898abcdef0 \  
  --http-tokens optional \  
  --http-endpoint enabled
```



### To turn on the IPv6 endpoint for your instance

By default, the IPv6 endpoint is disabled. This is true even if you have launched an instance into an IPv6-only subnet. The IPv6 endpoint for IMDS is only accessible on [Instances built on the Nitro System](#). For more information about the metadata options, see [modify-instance-metadata-options](#) in the *AWS CLI command reference*. The following example shows you how to turn on the IPv6 endpoint for the instance metadata service.

```
aws ec2 modify-instance-metadata-options \  
  --instance-id i-1234567898abcdef0 \  
  --http-protocol-ipv6 enabled \  
  --http-endpoint enabled
```



### To turn off access to instance metadata

You can turn off access to your instance metadata by disabling the HTTP endpoint of the instance metadata service, regardless of which version of the instance metadata service you are using. You can reverse this change at any time by enabling the HTTP endpoint. Use the [modify-instance-metadata-options](#) CLI command and set the `http-endpoint` parameter to `disabled`.

```
aws ec2 modify-instance-metadata-options \  
  --instance-id i-1234567898abcdef0 \  
  --http-endpoint disabled
```



### To control the use of modify-instance-metadata-options

To control which IAM users can modify the instance metadata options, specify a policy that prevents all users other than users with a specified role to use the [ModifyInstanceMetadataOptions](#) API. For the example IAM policy, see [Work with instance metadata](#).



# TPN

- 53. *Phenomenon1* – the tendency of X to Y.
- 54. *Phen2* – the tendency of X to Y.
- 55. *Phen3* – the tendency of X to Y.
- 56. *Phen4* – the tendency of X to Y.
- 57. *Phen5* – the tendency of X to Y.
- 58. *Phen6* – the tendency of X to Y.
- 59. *Phen7* – the tendency of X to Y.
- 60. *Phen8* – the tendency of X to Y.
- 61. *Phen9* – the tendency of X to Y.

62.     *Phen10* – the tendency of X to Y.

# Glossary

## Term1

Description of what term means here.

## Term2

Description of what term means here.

## Term3

Description of what term means here.

# Bibliography

XXV.   Official  
XXVI.  Unofficial  
XXVII. Critical  
XXVIII.General

XXV.   Official

**[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

**[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

## XXVI. Unofficial

**[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

**[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

**[Medium 1]**

“The difference between an AWS role and an instance profile”.  
*Medium*. 5<sup>th</sup> [May 2020](#). Available at: <https://medium.com/devops-dudes/the-difference-between-an-aws-role-and-an-instance-profile-ae81abd700d>

**[Kichik]**

“How do EC2 Instance Profiles Work?” Available at:  
<https://kichik.com/2020/09/08/how-does-ec2-instance-profile-work/>

**[ExamPro]**

ExamPro. “EC2 – Instance [Profile](#)”. Available at:  
[https://www.youtube.com/watch?v=t-uZa7FI9mk&ab\\_channel=ExamPro](https://www.youtube.com/watch?v=t-uZa7FI9mk&ab_channel=ExamPro)

[https://www.youtube.com/watch?v=7hSvFa9R9D0&ab\\_channel=AmazonWebServices](https://www.youtube.com/watch?v=7hSvFa9R9D0&ab_channel=AmazonWebServices)

## XXVII. Critical

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

## XXVIII. General

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

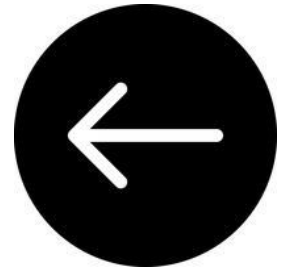
### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

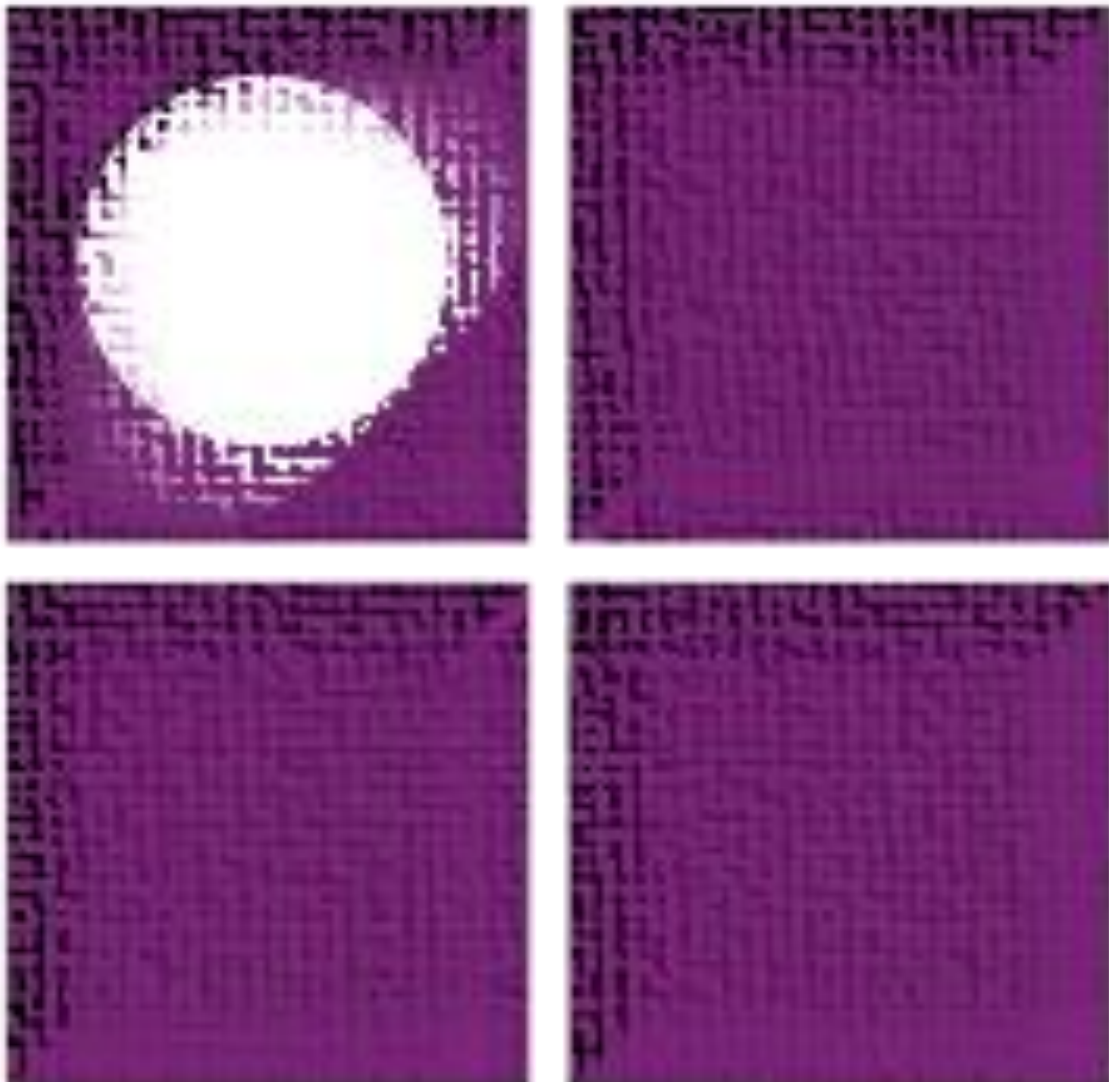




# SimpleDB



*RETURN TO CONTENTS*

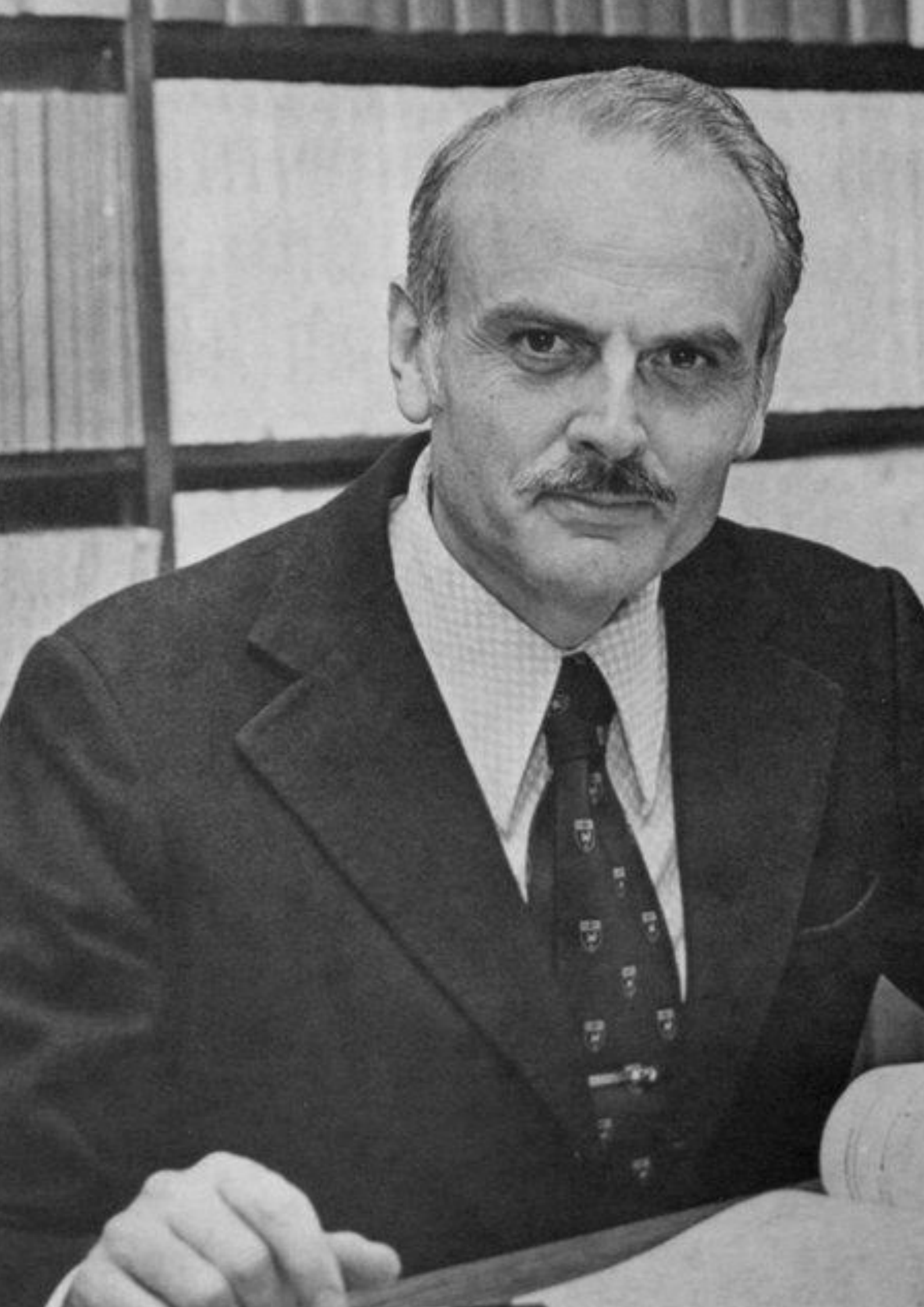


AWS used to have a service called SimpleDB. This has been superseded now. A number of superior services have taken its place, and AWS do not recommend that you use SimpleDB.

A service called RDS will be announced in 2009 (the beginning of FIRMAMENT era) followed by DynamoDB in 2012.

I think it's important to mention services such as SimpleDB, which have been discontinued. If I didn't, the student would worry when the encounter it.

I will use this chapter to provide notes on databases. I want to introduce you to some important people and give you forceful and vivid impressions of certain things. This will prime you for when we get to the AWS database services.







# SQL

*Structured Query Language*



**Don  
Chamberlin**

# Raymond F Boyce

Boyce was born in 1946. He would have been thirty years old in 1976. Sadly, Boyce did not reach this age, suffering an aneurysm aged 28.

Raymond Boyce was employed by IBM in the state of New York when he died. It was while at IBM that he developed SQL, along with Don Chamberlin. Boyce was also managing a database group for IBM in San Jose.

The original language was called SEQUEL. The SE stands for “Structured English”, the QUE is short for query, and the final “L” stands for language. In 1974, Boyce and Chamberlin published their paper, entitled “SEQUEL: A Structured English Query Language”.

## SEQUEL: A STRUCTURED ENGLISH QUERY LANGUAGE

by

Donald D. Chamberlin  
Raymond F. Boyce

IBM Research Laboratory  
San Jose, California

**ABSTRACT:** In this paper we present the data manipulation facility for a structured English query language (SEQUEL) which can be used for accessing data in an integrated relational data base. Without resorting to the concepts of bound variables and quantifiers SEQUEL identifies a set of simple operations on tabular structures, which can be shown to be of equivalent power to the first order predicate calculus. A SEQUEL user is presented with a consistent set of keyword English templates which reflect how people use tables to obtain information. Moreover, the SEQUEL user is able to compose these basic templates in a structured manner in order to form more complex queries.



Raymond Boyce grew up in the state of New York. He earned his PhD in Computer Science at Purdue. This university is in the US state of Indiana—can be found below Chicago on a map of the USA.



**R. F. Boyce**  
**(1947-1974)**

It is quite hard to find images of Raymond Boyce. The above is from [this](#) presentation.



# 1974

This year was important. Boyce didn't just collaborate with Don Chamberlin. He also worked with Edgar Codd himself. In 1974, they produced Boyce-Codd normal form (BCNF).

## Boyce-Codd normal form [\[edit\]](#)

[Boyce–Codd normal form](#) (or BCNF) was developed in 1974 by Boyce and [Edgar F. Codd](#). It is a type of [normal form](#) that is used in [database normalization](#). The goal of relational database design is to generate a set of [database schemas](#) that store information without unnecessary redundancy. Boyce-Codd accomplishes this and allows users to retrieve information easily. Using BCNF, databases will have all redundancy removed based on functional dependencies. It is a slightly stronger version of the [third normal form](#).

What on earth is  
“[normalisation](#)”?

The following example from Wikipedia gives you a hint about what this is all about. Try to remain calm.

## Designs that violate 1NF [\[ edit \]](#)

This table over customers' credit card transactions does not conform to first normal form:

| Customer | Customer ID | Transactions   |             |        |
|----------|-------------|----------------|-------------|--------|
| Abraham  | 1           | Transaction ID | Date        | Amount |
|          |             | 12890          | 14-Oct-2003 | -87    |
|          |             | 12904          | 15-Oct-2003 | -50    |
| Isaac    | 2           | Transaction ID | Date        | Amount |
|          |             | 12898          | 14-Oct-2003 | -21    |
| Jacob    | 3           | Transaction ID | Date        | Amount |
|          |             | 12907          | 15-Oct-2003 | -18    |
|          |             | 14920          | 20-Nov-2003 | -70    |
|          |             | 15003          | 27-Nov-2003 | -60    |

To each customer corresponds a 'repeating group' of transactions. Such a design can be represented in a [Hierarchical database](#) but not a SQL database, since SQL does not support nested tables.

The automated evaluation of any query relating to customers' transactions would broadly involve two stages:

1. Unpacking one or more customers' groups of transactions allowing the individual transactions in a group to be examined, and
2. Deriving a query result based on the results of the first stage

For example, in order to find out the monetary sum of all transactions that occurred in October 2003 for all customers, the system would have to know that it must first unpack the *Transactions* group of each customer, then sum the *Amounts* of all transactions thus obtained where the *Date* of the transaction falls in October 2003.

One of Codd's important insights was that structural complexity can be reduced. Reduced structural complexity gives users, applications, and DBMSs more power and flexibility to formulate and evaluate the queries. A more normalized equivalent of the structure above might look like this:

## Designs that comply with 1NF [\[ edit \]](#)

To bring the model into the first normal form, we can perform normalization. Normalization (to first normal form) is a process where attributes with non-simple domains are extracted to separate stand-alone relations. The extracted relations are amended with foreign keys referring to the primary key of the relation which contained it. The process can be applied recursively to non-simple domains nested in multiple levels.<sup>[4]</sup>

In this example, *Customer ID* is the primary key of the containing relations and will therefore be appended as foreign key to the new relation:

## Designs that comply with 1NF [\[ edit \]](#)

To bring the model into the first normal form, we can perform normalization. Normalization (to first normal form) is a process where attributes with non-simple domains are extracted to separate stand-alone relations. The extracted relations are amended with foreign keys referring to the primary key of the relation which contained it. The process can be applied recursively to non-simple domains nested in multiple levels.<sup>[4]</sup>

In this example, *Customer ID* is the primary key of the containing relations and will therefore be appended as foreign key to the new relation:

| Customer | Customer ID |
|----------|-------------|
| Abraham  | 1           |
| Isaac    | 2           |
| Jacob    | 3           |

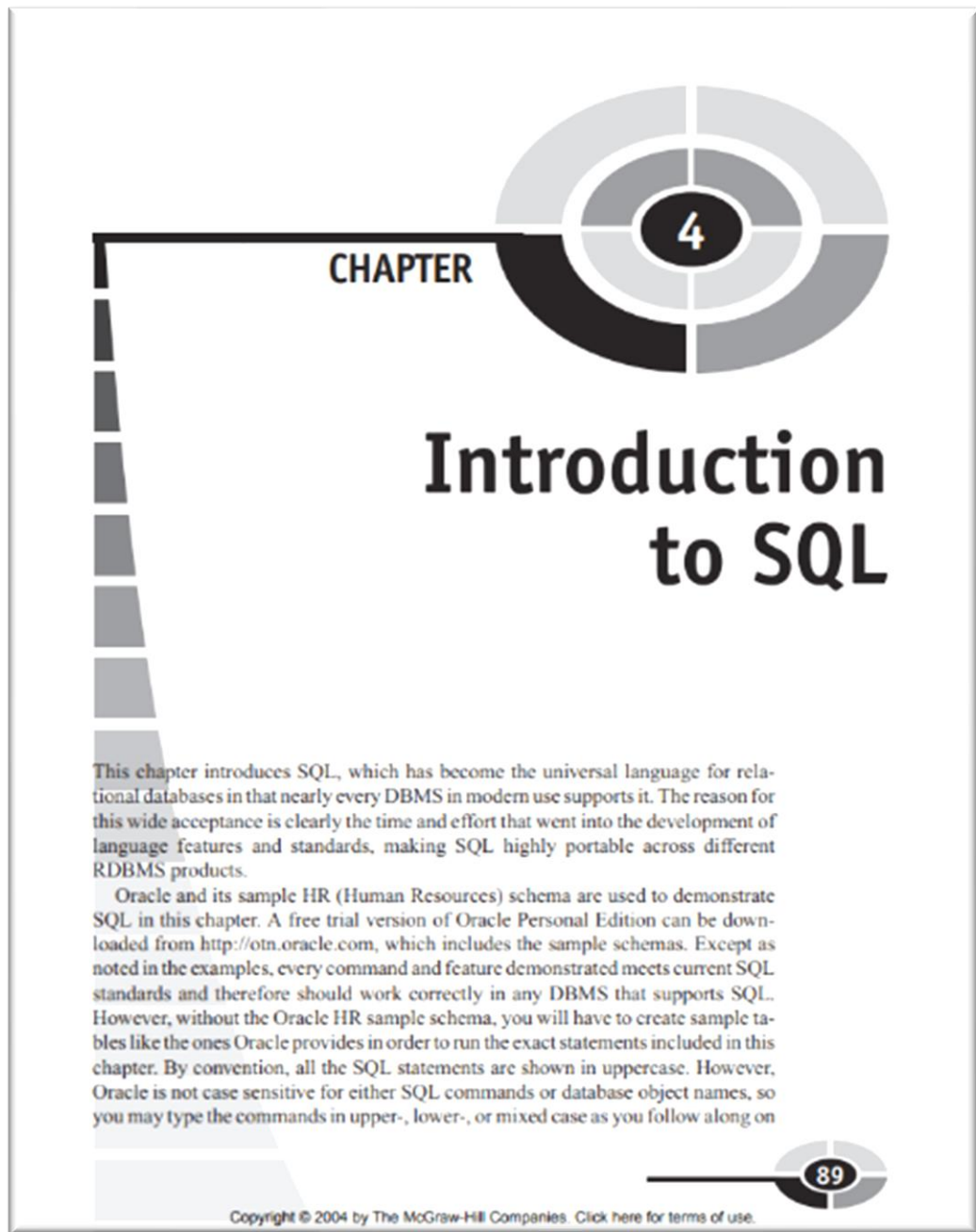
| Customer ID | Transaction ID | Date        | Amount |
|-------------|----------------|-------------|--------|
| 1           | 12890          | 14-Oct-2003 | −87    |
| 1           | 12904          | 15-Oct-2003 | −50    |
| 2           | 12898          | 14-Oct-2003 | −21    |
| 3           | 12907          | 15-Oct-2003 | −18    |
| 3           | 14920          | 20-Nov-2003 | −70    |
| 3           | 15003          | 27-Nov-2003 | −60    |

In the modified structure, the [primary key](#) is {Customer ID} in the first relation, {Customer ID, Transaction ID} in the second relation.

Now each row represents an individual credit card transaction, and the DBMS can obtain the answer of interest, simply by finding all rows with a Date falling in October, and summing their Amounts. The data structure places all of the values on an equal footing, exposing each to the DBMS directly, so each can potentially participate directly in queries; whereas in the previous situation some values were embedded in lower-level structures that had to be handled specially. Accordingly, the normalized design lends itself to general-purpose query processing, whereas the unnormalized design does not.

It is worth noting that this design meets the additional requirements for [second](#) and [third normal form](#).

Prepare a cup of tea and read the fantastic chapter below.



What do I need to tell you about SQL?

## Publication Venues for DB Research

39

- Conferences:
  - **SIGMOD**: ACM Special Interest Group on Management of Data (since 1975)
  - **PODS**: ACM Symp. on Principles of Database Systems (since 1982)
  - **VLDB**: Intl. Conf. on Very Large Databases (since 1975)
  - **ICDE**: IEEE Intl. Conf. on Data Engineering (since 1984)
  - **ICDT**: Intl. Conference on Database Theory (since 1986)
  - **EDBT**: Intl. Conference on Extending Database Technology (since 1988)
- Journals:
  - **TODS**: ACM Transactions on Database Systems (since 1976)
  - **VLDBJ**: The VLDB Journal (since 1992)
  - **SIGMOD REC**: ACM SIGMOD Record (since 1969)

What should we call  
the vertical thing?



# ATTRIBUTE

1 : to explain (something) by indicating a cause

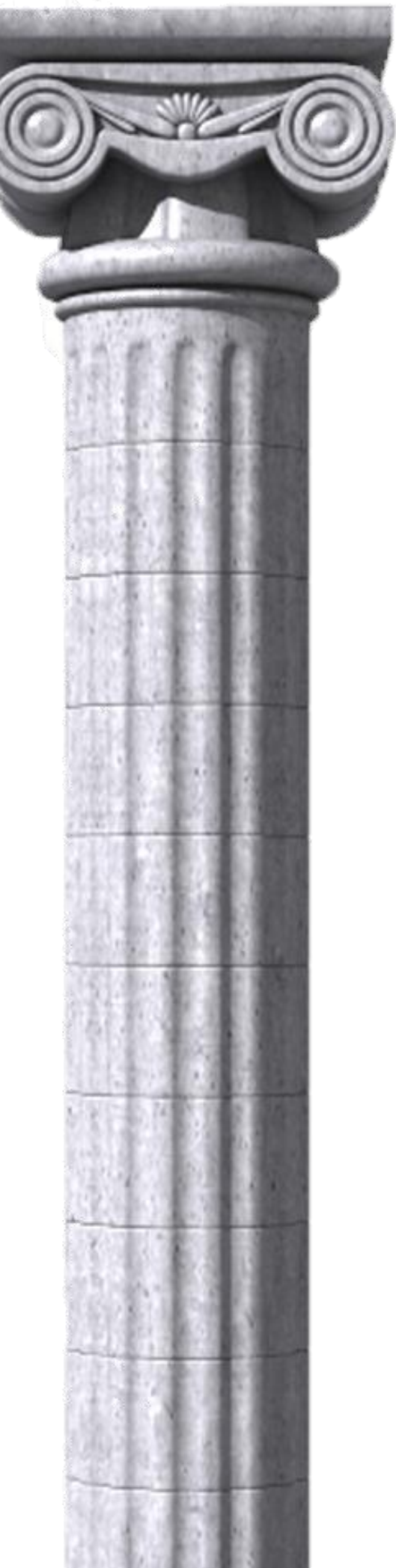
// He *attributed* his success to hard work.

2 a : to regard as a characteristic of a person or thing

// should *not attribute* adult reasoning to children







# COLUMN

FROM LATIN, *COLUMNA*

FIELD (No)



THE  WOR



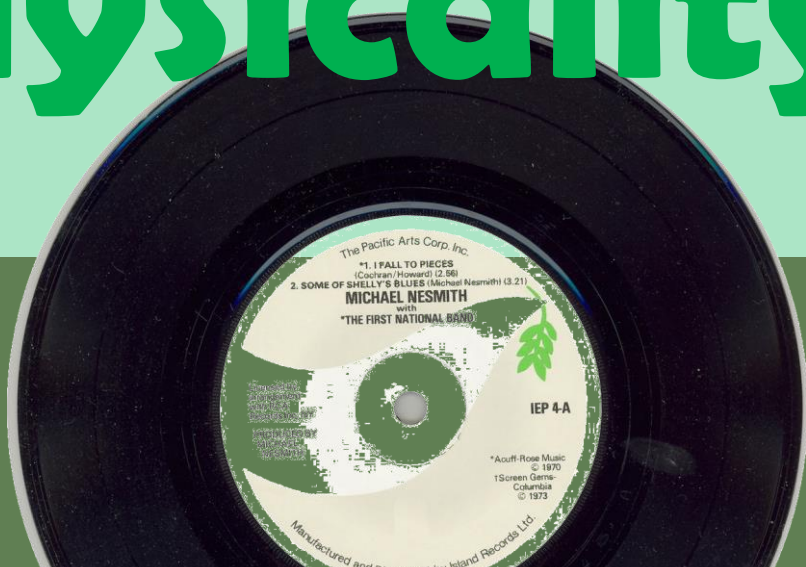


What should we call the  
horizontal thing?

# record

## Sordid

## physicality



# ROW





# Tuple



# ITEM



*(Term used in no-SQL  
databases)*



## So, what's all the fuss about?

Although your question is already answered very well, I would like to add my points too. *May be you find it helpful upto some extend. Also my answer is not specific to SQL Server*

11

These words are used interchangeably.



| 1      | 2 | 3      | 4 |           |   |                             |
|--------|---|--------|---|-----------|---|-----------------------------|
| Row    | = | Record | = | Tuple     | = | Entity                      |
| Column | = | Field  | = | Attribute | = | Attribute                   |
| table  | = | File   | = | Relation  | = | Entity Types(or Entity Set) |

- 4 terminology good to use when we learn ER-Modules
- 3 use when Relational Model
- 2 in- general scene, **DataBase books start with these terminology** because these are much commonly used by people in real life, also in file-system.

**Record** is the basic unit in storage system that have implicit meaning. In DBMS the word **record** use in chapter describes how database tables stores on disk blocks. In DBMS a **record-oriented file-system** is a file system where files are stored as collections of records.

Share Improve this answer Follow

answered Jan 13 2013 at 4:41



Grijesh Chauhan

571 3 5 18

1 also, you can say that TABLES, ROWs, and COLUMNS are logical representation in the "DBMS" while FILES, RECORDs, and FIELDs are a physical representation of the file-system – [Ahmed Boutaraa](#) Feb 25 2021 at 22:48

@AhmedBoutaraa valid statment. – [Grijesh Chauhan](#) Mar 2 2021 at 9:00

Add a comment



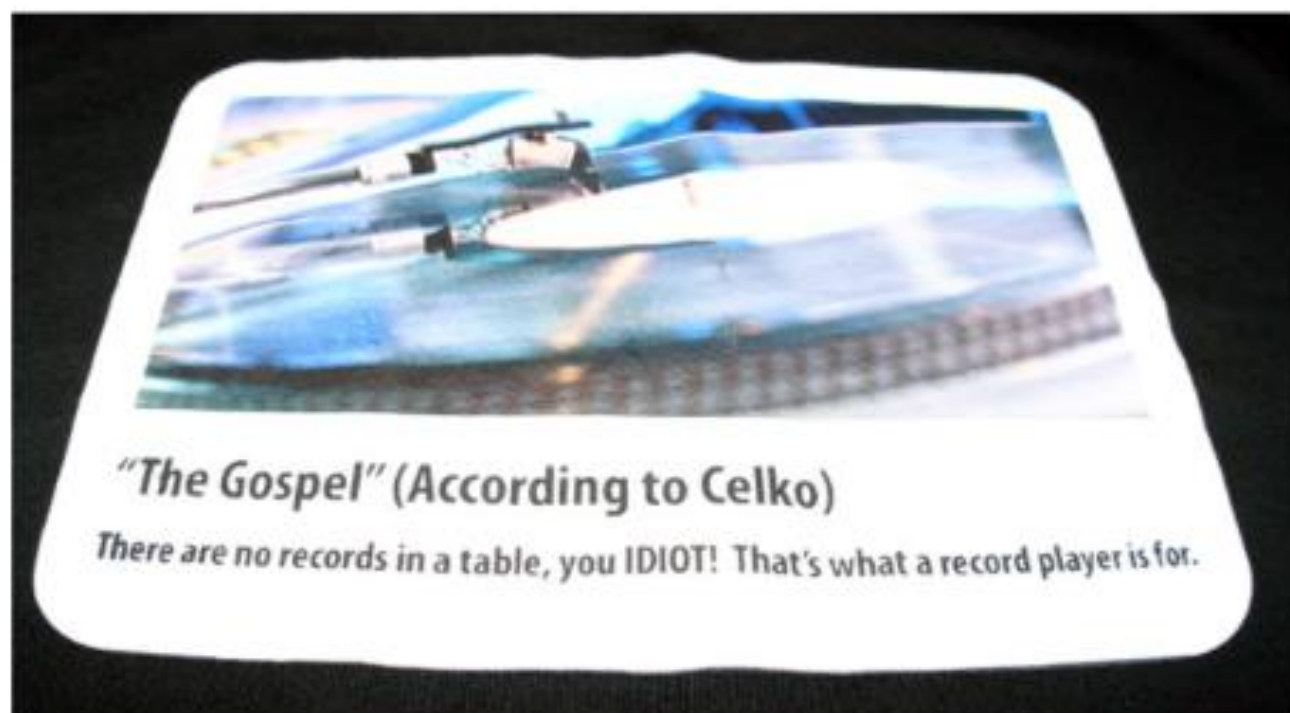
75



To quote Joe Celko (not only can you find this reference all over the web and in [his Wikipedia entry](#), but you will even see it on T-shirts at some conferences):

Rows are not records.

A lot of people point him out as a pedantic jerk who just likes to humble and verbally abuse newbies, and I will admit that is how he comes across. But I have also met him in person - even shared a meal with him - and I can't tell you how different his real-life persona is from his online front. I even once caught him calling rows records, and he was very embarrassed ([full backstory here](#)).



In any case, say what you will about the guy's online character, but he **wrote the standard**, and the fact that such an authority dictates that there is a distinction should tell you something. And as much as he cringes when someone calls a row a record, so do many of my colleagues - who are also experts in the SQL Server world. And those of us in that camp believe he is right.

For example, Itzik Ben-Gan, an obvious SQL Server guru. Here is a quote from the very first lesson in his [Training Kit \(Exam 70-461\): Querying Microsoft SQL Server 2012](#):

As an example of incorrect terms in T-SQL, people often use the terms "field" and "record" to refer to what T-SQL calls "column" and "row," respectively. Fields and records are

For example, Itzik Ben-Gan, an obvious SQL Server guru. Here is a quote from the very first lesson in his [Training Kit \(Exam 70-461\): Querying Microsoft SQL Server 2012](#):

As an example of incorrect terms in T-SQL, people often use the terms "field" and "record" to refer to what T-SQL calls "column" and "row," respectively. Fields and records are physical. Fields are what you have in user interfaces in client applications, and records are what you have in files and cursors. Tables are logical, and they have logical rows and columns.





Eric Brewer proposed CAP theorem at a conference in 2000. It was the keynote at the conference called *Principles of Distributed Computing*. When it comes to shared-data systems, only two of the following three properties can be achieved:

- Data consistency
- System availability
- Tolerance to network partition

A more formal confirmation can be found in a 2002 paper by Seth Gilbert and Nancy Lynch.



**Non-relational databases like to use the term “item”. *How can I remember this fact?***

Well, non-relational databases tend to be used by large retailers. When I think of retailing, I think of the automated checkouts in shops. They usually say “unexpected ITEM in bagging area”. So, I can make a link from non-relational, to retail (to the word ITEM). The till doesn’t announce “unexpected ROW in bagging area”, does it?



# A simple timeline

1979

Oracle database released



1989

Microsoft SQL Server released



1995

MySQL released



1997

PostgreSQL released



I've found that there are four database engines that get mentioned a lot with AWS databases. In my head, I have them forming a square. The top row has the two "big corporate guys". These are the major companies. These databases are not open source. On the right-most column the theme is "M". M for Microsoft and for MySQL. Water drains to the bottom, so on the bottom row we have the blue water, splashed about by the trunk of the PostgreSQL elephant and swam in by the MySQL dolphin. Oracle is top left because it came first. We can add to this basic schema as we see necessary. (For example, we will eventually append MariaDB—another engine—to MySQL in the bottom right). I will introduce these four in turn.



Beginning with  
"M"





# ORACLE®

## “Big Red”

### History [\[edit\]](#)

[Larry Ellison](#) and his two friends and former co-workers, [Bob Miner](#) and [Ed Oates](#), started a consultancy called Software Development Laboratories (SDL) in 1977. SDL developed the original version of the Oracle software. The name *Oracle* comes from the code-name of a CIA-funded project Ellison had worked on while formerly employed by [Ampex](#).<sup>[6]</sup>





My Widenius





## The beginning

### Early days

- Created by Dr. Heikki Tuuri - 1994 - architecture based on G&R book
- Closed source
- All development on a Pentium Pro Windows using Visual C++
  - Original comments referring to the Pentium Pro still preserved :-)
- Single table space for all data and the undo logs
- Separate files for circular redo
- Stand alone ACID database with Transactions/SQL/FK/Row locks etc.

[https://www.youtube.com/watch?v=vWpfi5ryO1s&list=PLn2pda68RIHWJrE8-hA8TmdzNZczDvLHu&index=1&ab\\_channel=PingCAP](https://www.youtube.com/watch?v=vWpfi5ryO1s&list=PLn2pda68RIHWJrE8-hA8TmdzNZczDvLHu&index=1&ab_channel=PingCAP)

# PostgreSQL



# SCHEMA



Amazon SimpleDB is simpler than a traditional, relational database as it requires no schema, automatically indexes data, and provides simple APIs for storage and access. Amazon SimpleDB allows businesses and developers to easily scale database processing and frees developers from many of the complexities of capacity planning while transforming large capital expenditures into much smaller operating costs. Amazon SimpleDB also integrates seamlessly with other AWS services, such as Amazon S3 and Amazon EC2, to provide the infrastructure for web-scale applications.

MyMiniLife.com is a social networking site allowing anyone to fully express their physical life on the web. "MyMiniLife processes millions of transactions per day and we were looking for a highly scalable system to support the load," said Joel Poloney, Co-Founder of MyMiniLife.com. "After looking at a couple of possibilities, we realized that the current relational database model would not be able to scale updates and inserts as well as Amazon SimpleDB. We jumped on to SimpleDB when it was in Private Beta and we've been extremely impressed with its scalability, low cost, and support team responsiveness."

Issuu is an online publishing service where people and businesses alike can upload and share documents. "SimpleDB serves as our main database for storing information about users and their publications," said Mikkel Jensen, Founding Partner and Head of Operations for Issuu. "When we first started out we knew that the ability to scale would be a central criterion for success and that off-the-shelf RDBMS systems just wouldn't cut it. With its scalable infrastructure Amazon SimpleDB was a natural choice for us. The low barrier to entry and amount of maintenance required means we don't need to have a staff of DBAs employed. Instead we can concentrate on our main goal of creating great services."



AWS News Blog

## A Place for Everything – Amazon SimpleDB

by [Jeff Barr](#) | on 14 DEC 2007 | [Permalink](#) | [Share](#)



Voiced by **Amazon Polly**

We are now accepting applications for the limited public beta of [Amazon SimpleDB](#)!

Amazon SimpleDB makes it really easy and straightforward to store and to retrieve structured data. You no longer need to worry about creating, maintaining, or migrating database schemas, monitoring and tuning the performance of your

**Welcome, Jeff @ AWS.**  
(Not you? [Click here.](#))

[Your Web Services Account](#)

[Sign Up For This Web Service](#)



## Summary of the Amazon SimpleDB Service Disruption

We wanted to share what we've learned from our investigation of the June 13 SimpleDB disruption in the US East Region. The service was unavailable to all API calls (except a fraction of the eventually consistent read calls) from 9:16 PM to 11:16 PM (PDT). From 11:16 PM to 1:30 AM, we continued to have elevated error rates for CreateDomain and DeleteDomain API calls.

SimpleDB is a distributed datastore that replicates customer data across multiple data centers. The service employs servers in various roles. Some servers are responsible for the storage of user data ("storage nodes"), with each customer Domain replicated across a group of storage nodes. Other nodes store metadata about each customer Domain ("metadata nodes"), such as which storage nodes it is located on. SimpleDB uses an internal lock service to determine which set of nodes are responsible for a given Domain. This lock service itself is replicated across multiple data centers. Each node handshakes with the lock service periodically to verify it still has responsibility for the data or metadata it hosts.

## What's the point of using Amazon SimpleDB?

Asked 13 years, 6 months ago   Modified 2 years, 8 months ago   Viewed 16k times

▲  
51 I thought that I could use SimpleDB to take care of the most challenging area of my application (as far as scaling goes) - twitter-like comments, but with location on top - till the point when I sat down to actually start implementing it with SDB.

▼  
28 First thing, SDB has a 1000 bytes limitation per attribute value, which is not enough even for comments (probably need to break down longer values into multiple attributes).

Then, maximum domain size is 10GB. The promise was that you could scale up without worrying about database sharding etc., since SDB will not degrade with increasing loads of data. But if I understand correctly, with domains I would have exactly the same problem as with sharding, ie. at some point need to implement data records' distribution and queries across domains on application level.

Even for the simplest objects that I have in the whole application, ie. atomic user ratings, SDB is not an option, because it cannot calculate an average within the query (everything is string based). So to calculate average user rating for an object, I would have to load all records - 250 at a time - and calculate it on application level.

Am I missing something about SDB? Is 10GB really that much of a database to get over all SDB limitations? I was honestly enthusiastic about taking advantage of SDB, since I use S3 and EC2 already, but now I simply don't see a use case.

database

aws amazon-web-services

amazon-simplifiedb

### 9 Answers

Sorted by: Highest score (default)

▲  
35 I use SDB on a couple of large-ish applications. The 10 GB limit per domain does worry me, but we are gambling on Amazon allowing this to be extended if we need it. They have a request form on their site if you want more space.

▼  
✓ As far as cross domain joins, don't think of SDB as a traditional database. During the migration of my data to SDB, I had to denormalize some of it so I could manually do the cross domain joins.

Then  
⌚ The 1000 byte per attribute limitation was tough to work around also. One of the applications I have is a blog service which stores posts and comments in the database. While porting it over to SDB, I ran into this limitation. I ended up storing the posts and comments as files in S3, and read that in my code. Since this server is on EC2, the traffic to S3 isn't costing anything extra.



Michael S. Fischer

Mar 7, 2017 · 8 min read ·  Listen



## Resurrecting Amazon SimpleDB

*While working on a recent problem, I tested out a number of alternative Databases-As-A-Service (DBaaS) on the Amazon Web Services (AWS) platform. The best fit for my particular use case ended up being an often-overlooked one: **SimpleDB**. Surprised? Me, too.*

## The SimpleDB Solution

It turns out that there's a database service in the AWS product family that nearly everyone forgets about: Amazon SimpleDB. It's not super fancy or built for massive amounts of data, but it has exactly the properties we need:

- Arbitrary "columns" (actually, attributes) to store and search for our key-value pairs
- Easily accessible, but protected using IAM (Identity and Access Management).
- A query language that reads like SQL, but doesn't require us to construct cross-products of our tag table;
- Dirt-cheap to run: 25 machine-hours per month *for free*; followed by 14 cents per machine-hour thereafter (not including transfer costs, which are miniscule as well)

We tried it out and it worked perfectly. Remember our complex query builder, above? It looks like this now (again, Ruby):



Eden Hare

Oct 19, 2020 · 10 min read · ✨ Member-only ·



## A Five Minute Overview of Amazon SimpleDB

Sometimes we are working on a project where we need a data store, but the complexities of Relational Database Service (RDS), DynamoDB, DocumentDB, et al are more than what is needed. This is where Amazon SimpleDB becomes a valuable resource.

# TPN

- 63. ***Phenomenon1*** – the tendency of X to Y.
- 64. ***Phen2*** – the tendency of X to Y.
- 65. ***Phen3*** – the tendency of X to Y.
- 66. ***Phen4*** – the tendency of X to Y.
- 67. ***Phen5*** – the tendency of X to Y.
- 68. ***Phen6*** – the tendency of X to Y.
- 69. ***Phen7*** – the tendency of X to Y.
- 70. ***Phen8*** – the tendency of X to Y.
- 71. ***Phen9*** – the tendency of X to Y.
- 72. ***Phen10*** – the tendency of X to Y.

# Glossary

### **Term1**

Description of what term means here.

### **Term2**

Description of what term means here.

### **Term3**

Description of what term means here.

# Bibliography

- XXIX. Official
- XXX. Unofficial
- XXXI. Critical
- XXXII. General

XXIX. Official

<https://aws.amazon.com/blogs/aws/amazon-simplifiedb-management-in-eclipse/>



<https://aws.amazon.com/blogs/aws/a-place-for-eve/>

<https://aws.amazon.com/message/65649/>

<https://medium.com/swlh/a-five-minute-overview-of-amazon-simplifiedb-4823a829d99>

**[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

**[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

XXX. Unofficial

**[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

**[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

<https://stackoverflow.com/questions/648270/whats-the-point-of-using-amazon-simplifiedb>

<https://zendesk.engineering/resurrecting-amazon-simplydb-9404034ec506>

## XXXI. Critical

### [Surname1]

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

### [Surname1]

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

## XXXII. General

### [Surname1]

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

### [Surname1]

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

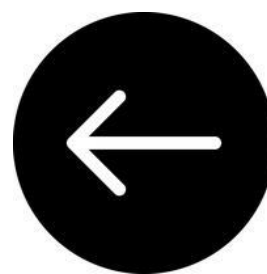
### [Date 2003]

Date, Chris J (2003). An Overview of Database Management. In *An Introduction to Database Systems*, 8<sup>th</sup> edition.





# EBS



[RETURN TO CONTENTS](#)

This topic is so large that it must be dealt with in five modules.

|  |                 |                        |
|--|-----------------|------------------------|
|  | <b>Module 1</b> |                        |
|  | 1               | Taxonomy               |
|  | 2               | Magnetic drives        |
|  | 3               | Adjectives for storage |
|  | 4               | RAID                   |
|  | <b>Module 2</b> |                        |
|  | 1               | Bursting               |
|  | 2               | Partition schemes      |
|  | 3               |                        |
|  | <b>Module 3</b> |                        |
|  | 1               | EBS Multi-Attach       |
|  | 2               | The Nitro System       |
|  | 3               |                        |
|  | <b>Module 4</b> |                        |
|  | 1               | Snapshots              |
|  | 2               | Encryption in EBS      |
|  |                 |                        |
|  | <b>Module 5</b> |                        |
|  | 1               | The instance store     |
|  | 2               | Root device volume     |
|  | 3               | Root device mappings   |



# EBS Module 1

Types of EBS volume; words used to adjectives for storage; RAID.

[RETURN TO EBS](#)

In this first module, I introduce a taxonomy of different types of EBS volumes. The basic choice is between EBS volumes backed by solid-state drives, and those backed by hard disk drives. This first module is also an appropriate place to introduce RAID. You ought to know about RAID; it will come up in various contexts relating to storage, not only EBS.

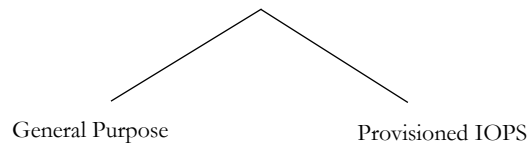
## Types of EBS volume

It's necessary to rattle through this taxonomy. At the top level, there are three types of EBS volume:

1. Solid-state drives
2. Hard disk drives
3. Previous generation

Now, there are species within each category, which, believe me, will be given the full treatment in a moment. Allow me to lay the ground for some later some abbreviations. When I say “s.t.”, you yell “super throughput”. And when I chant “s.c.”, I want you to yell “super cold”. Finally, HDD stands for *hard disk drive*.

You might opt for EBS volumes which are backed by solid-state drives (SSD). AWS say “we recommend these volumes for most workloads”. There are two kinds, within the SSD category. They are known as *general-purpose* and *provisioned IOPS*.



Basically, the *provisioned IOPS* volumes are for higher performance. AWS write:

They are the highest performance Amazon EBS storage volumes designed for critical, IOPS-intensive, and throughput-intensive workloads that require low latency.

From the name of this category—Provisioned IOPS—it’s clear that these EBS volumes are going to excel in terms of IOPS. But they also should be chosen for “throughput intensive workloads that require low latency”. These three things, IOPS, throughput, and latency, are not the same. See the BLITZ acronym, in the appendix. However, it seems that Provisioned IOPS should be the choice when you need to excel at the L, I, and P of BLITZ.

We now move to the third level of the taxonomy, and examine three types of *Provisioned IOPS*. The three types are known as:

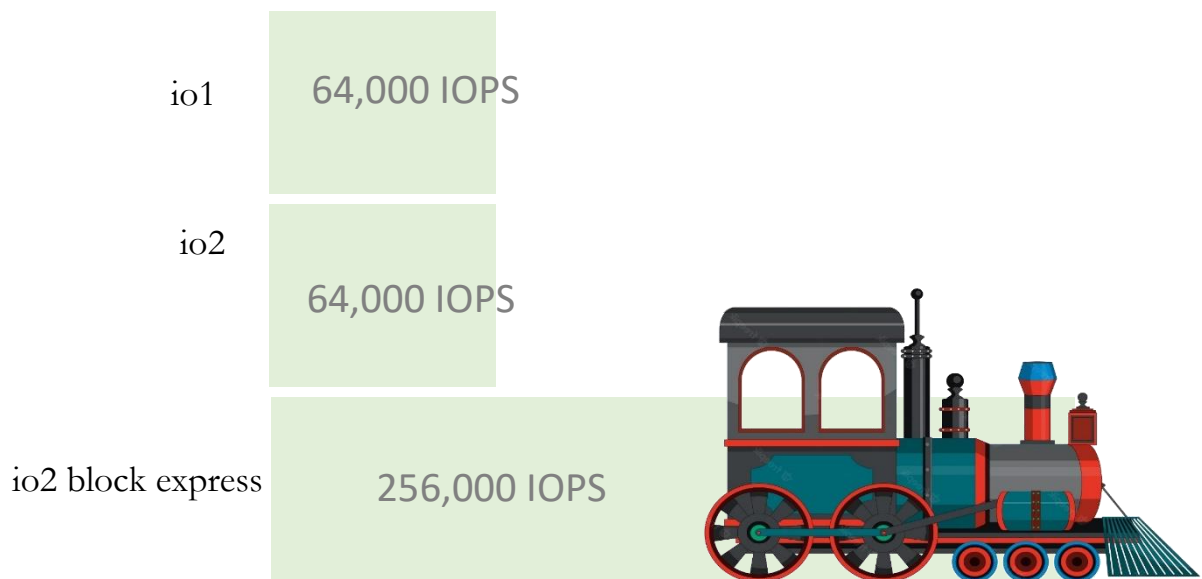
io1  
io2  
io2 block express.

It’s “io” because this is “provisioned IOPS” after all. The O within IOPS stands for *operation*, not output. Each successive one slightly improves things. For example, the durability of io1 is stated to be 99.8-99.9%. Durability is about the integrity of the data *in itself*. It’s completely different from availability. Availability is about *accessibility*. To use an analogy, carrots are highly available in the supermarkets because I can access them year-round. But being vegetables, they degrade quickly, so are not very durable. In contrast, this massive plush easter bunny toy, will stick around for years, sadly. It’s only available in some shops, only at easter. So it has low levels of availability. Yet it is durable. If you can’t ping a server, it’s not available.

The durability of io1 is 99.8-99.9. What io2 does is step this up, and fix durability at “five nines”, 99.999%. This is quite hard to beat, so io2 block express also has five nines. Remember, these expressions—“five nines”, “three nines”—have what we’ll call **EXPRESSION INCLUSIVITY**. Everything is included in the count, digits before and after the decimal point. “Five nines” does *not* refer to 99.99999.

There's basically four things we can talk about with these species of EBS volume. They are (1) the durability (2) the IOPS achievable (3) throughput achievable (4) the size of the volume.

The "io3 block express" proves its worth in the figure for the maximum IOPS achievable. Both io1 and io2 can achieve 64,000 IOPS. This gets quadrupled for io2 block express: it can achieve 256,000 IOPS.



So, this quadrupling is about the number I/O operations achievable in a second. Every time the hand on your watch ticks, four times *more* operations occur with io2 block express (compared with io1 and io2).

If each operation made a sound, such as the faint sound made by your shoes when you stand to attention, io1 and io2 would be the kerfuffle of a church meeting getting up to leave. In contrast, *io2 Block Express* would be the intimidating thud of a number of soldiers—enough to pack out a school sports hall—standing to attention.

I/O Block Express *also* quadruples the **throughput**. The units of throughput are bytes per second. With io1 and io2, they achieve 1,000 mebibytes per second. With Express, it's 4,000. (AWS state it this way. Why they don't say "1 gibibyte per second" I don't know).

Finally, let's talk about the size of the volume. For io1 and io2, the EBS volume must be at least 4 GiB, or gibibytes. The maximum size of the volume is 16 tebibytes. What io2 block express does is push up the maximum. I invite you to take a guess, regarding how many times the upper limit (16 tebibytes) is multiplied.

It's quadrupled. With io2 Block Express, the upper limit is 64 tebibytes. We've now seen how io2 Block Express improved upon io1 and io2 by quadrupling three things:

1. the operations occurring every second;
2. how much data (in bytes) passes *through* every second and;
3. the upper limit on the size of a volume.

*Choo choo choo!*



---

## Provisioned IOPS and volume size

The size of a volume and the IOPS that can be achieved are not completely unrelated. In fact:

The Maximum Ratio Principle (my name)

the maximum ratio of provisioned IOPS to requested volume size (in GiB) is 50:1 for io1 volumes, and 500:1 for io2 volumes.

# “maximum ratio”

Recall that for io1 and io2 volumes must be at least 4 GiB. And they can climb up to 16 TiB. What is being said above? I will take a stab at it.

We can relate [the provisioned IOPS] and [the volume size] in a *ratio*. For example, if the ratio was 2:1 and the volume size was the minimum, 4 GiB, then [the provisioned IOPS] would be 8 IOPS. Thus:

|          |       |
|----------|-------|
| 2 :      | 1     |
| 8 IOPS : | 4 GiB |

I don't find talk of "making the ratio larger" very natural (how can a ratio, a relation between two numbers, have a magnitude?) but let's try and do this, because AWS talk about a "maximum ratio". So, we might make the ratio above "bigger" by making it 4:1. Thus:

|           |       |
|-----------|-------|
| 4 :       | 1     |
| 16 IOPS : | 4 GiB |

And let's keep going:

|           |       |
|-----------|-------|
| 10 :      | 1     |
| 40 IOPS : | 4 GiB |

|           |       |
|-----------|-------|
| 20 :      | 1     |
| 80 IOPS : | 4 GiB |

|            |       |
|------------|-------|
| 40 :       | 1     |
| 160 IOPS : | 4 GiB |

|            |       |
|------------|-------|
| 50 :       | 1     |
| 200 IOPS : | 4 GiB |

|            |       |
|------------|-------|
| 60 :       | 1     |
| 240 IOPS : | 4 GiB |



But in fact, as we kept "increasing the ratio", AWS would have interrupted us". Once we reached "50:1", Dr Vogels would have gently patted us on the shoulder and said "uh, uh, you've have enough". This is for **io1 only**.

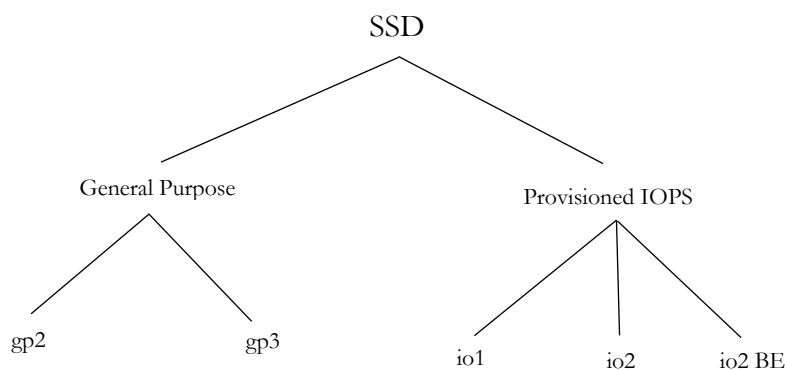
For io2, things are considerably more relaxed. We can keep going until we reach 500:1.



500 : 1  
2000 IOPS : 4 GiB

Have I interpreted the Maximum Ratio Principle correctly?

Now let's look at another sort of "Provisioned IOPS",  
general purpose.



A special thanks go to the team who name these things. Now, we have to make a special effort to remember whether it's the io's or the gp's which decide to skip the number 1. Perhaps someone in AWS was taunted at school for their resemblance to a military 4-by-4, and colleagues constantly mentioning "Jeepy One" would bring it all back, so we don't talk about gp1.

AWS say that gp2 and gp3 are good for most things:

These include virtual desktops, medium-sized single instance databases, latency sensitive interactive applications, development and test environments, and boot volumes. We recommend these volumes for most workloads.

Like io1, the durability of these volumes is 99.8-99.9%. The size of these volumes has an upper limit of 16 gibibytes. However, they can start off small: they can be just 1 gibibyte, rather than 4, as with all of the *Provisioned IOPS* categories.

For gp2 and gp3, the maximum IOPS per volume is a measly 16,000. Remember that even for io1, it is 64,000.

The only other thing which we need to talk about is the throughput which can be achieved. For gp2, it is 250 mebibytes per second. This gets quadrupled when we move to gp3. For gp3,

it is 1000 mebibytes per second. Look, clearly, there's a lot of figures here and to get on top of them you'll have to "work out" a bit—as Eddie Murphy says regarding stand up comedy, namely, get into it and test yourself.

## HDD

Now we move completely away from solid-state drive and into the world of hard-disk drive. There are two options here: st1 and sc1. AWS put it in terms of *low* and *lowest* cost:

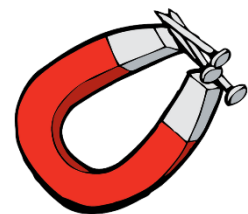
- Throughput Optimized HDD — A low-cost HDD designed for frequently accessed, throughput-intensive workloads.
- Cold HDD — The lowest-cost HDD design for less frequently accessed workloads.

I imagine that "sc1" stands for "super cold 1". This is the category called *Cold HDD*. The category called *Throughput optimized HDD* is known as st1. For both of these, the durability is 99.8-99.9%. This is by far the most common value for durability. Cold sc1 can achieve 250 mebibytes per second in terms of throughput—the same as gp2. The st1 volumes achieve 500 mebibytes per second.

I want to remind you that both gp2 and gp3 achieve 16,000 IOPS per volume. Well, sc1 achieves just 250 IOPS. Not 250 *thousand*, just 250. It's throughput optimized sibling is better: st1 achieves 500 IOPS.

We're told that use cases for st1 include Big data, Data warehouses and Log processing

|                                 | Throughput Optimized HDD  | Cold HDD   |
|---------------------------------|---|--|
| Volume type                     | st1   | sc1  |
| Durability                      | 99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)  | 99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)   |
| Use cases                       | <ul style="list-style-type: none"> <li>• Big data</li> <li>• Data warehouses</li> <li>• Log processing</li> </ul> | <ul style="list-style-type: none"> <li>• Throughput-oriented storage for data that is infrequently accessed</li> <li>• Scenarios where the lowest storage cost is important</li> </ul> |
| Volume size                     | 125 GiB - 16 TiB  | 125 GiB - 16 TiB   |
| Max IOPS per volume (1 MiB I/O) | 500   | 250  |
| Max throughput per volume       | 500 MiB/s   | 250 MiB/s  |
| Amazon EBS Multi-attach         | Not supported   | Not supported  |
| Boot volume                     | Not supported   | Not supported  |



## Magnetic

AWS want to discourage you from using magnetic-backed EBS volumes. However, at time of writing, this table is still in the documentation:

|                           | Magnetic                                      |
|---------------------------|---|
| Volume type               | standard                                      |
| Use cases                 | Workloads where data is infrequently accessed |
| Volume size               | 1 GiB-1 TiB                                   |
| Max IOPS per volume       | 40-200  |
| Max throughput per volume | 40-90 MiB/s                                   |
| Boot volume               | Supported                                     |



# Adjectives for storage

As you go into this area (EBS and storage in general), you're going to come across a few different adjectives used to describe storage. Roughly, there are two camps: permanent and temporary. But we should just be cautious about any finer differences. I think I've collected all of them here:



## Potted Plants

P ersistent  
T ransient  
T emporary  
E phemeral  
D urable  
  
P ermanent



There also “volatile”, a term commonly used in computer science literature. Volatile memory is lost when the machine powers down. It is temporary. You’ll often hear of “non-volatile memory” (NVM).

# RAID



This image is a still from a [YouTube video](#) entitled “Above the Cloud – A Berkeley View of Cloud Computing” (2009). From left to right is: Armando Fox, Anthony Joseph, Randy Katz, and David Patterson.

RAID stands for Redundant Array of Independent Disks. Sometimes the “I” is considered to stand for Inexpensive. The

whole idea of RAID was introduced in a 1988 paper by David Patterson, Garth Gibson and Randy Katz.

## Above the Clouds: A Berkeley View of Cloud Computing

Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz,  
Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia  
(Comments should be addressed to [abovetheclouds@cs.berkeley.edu](mailto:abovetheclouds@cs.berkeley.edu))

UC Berkeley Reliable Adaptive Distributed Systems Laboratory \*  
<http://radlab.cs.berkeley.edu/>

February 10, 2009

KEYWORDS: Cloud Computing, Utility Computing, Internet Datacenters, Distributed System Economics

### 1 Executive Summary

Cloud Computing, the long-held dream of computing as a utility, has the potential to transform a large part of the IT industry, making software even more attractive as a service and shaping the way IT hardware is designed and purchased. Developers with innovative ideas for new Internet services no longer require the large capital outlays in hardware to deploy their service or the human expense to operate it. They need not be concerned about over-provisioning for a service whose popularity does not meet their predictions, thus wasting costly resources, or under-

So, what the idea?

## The different types of RAID

It has

## How to remember the different types of RAID

It has



### 3. QUESTION

A persistent database must be migrated from an on-premises server to an Amazon EC2 instances. The database requires 64,000 IOPS and, if possible, should be stored on a single Amazon EBS volume.

Which solution should a Solutions Architect recommend?

- ☒ Create a Nitro-based Amazon EC2 instance with an Amazon EBS Provisioned IOPS SSD (i01) volume attached. Provision 64,000 IOPS for the volume.
- ☐ Create an Amazon EC2 instance with two Amazon EBS Provisioned IOPS SSD (i01) volumes attached. Provision 32,000 IOPS per volume and create a logical volume using the OS that aggregates the capacity.
- ☐ Create an Amazon EC2 instance with four Amazon EBS General Purpose SSD (gp2) volumes attached. Max out the IOPS on each volume and use a RAID 0 stripe set.
- ☐ Use an instance from the I3 I/O optimized family and leverage instance store storage to achieve the IOPS requirement.

Correct

#### Explanation:

Amazon EC2 Nitro-based systems are not required for this solution but do offer advantages in performance that will help to maximize the usage of the EBS volume. For the data storage volume an i01 volume can support up to 64,000 IOPS so a single volume with sufficient capacity (50 IOPS per GiB) can be deliver the requirements.

The current list of EBS volume types is in the table below:

|             | General Purpose SSD  |  | Provisioned IOPS SSD   |     |   |
|-------------|--|--|--|-----|---|
| Volume type | gp3  | gp2  | io2 Block Express ‡  | io2 | io1   |
| Durability  | 99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)   | 99.8% - 99.9% durability (0.1% - 0.2% annual failure rate) | 99.999% durability (0.001% annual failure rate)  |     | 99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)  |
| Use cases   | <ul style="list-style-type: none"><li>Low-latency interactive apps</li><li>Development and test environments</li></ul> |  | Workloads that require sub-millisecond latency, and sustained IOPS performance or more than 64,000 IOPS or 1,000 MiB/s of throughput |     | <ul style="list-style-type: none"><li>Workloads that require sustained IOPS performance or more than 16,000 IOPS</li><li>I/O-intensive database workloads</li></ul> |

#### 7. QUESTION

A company runs an application on an Amazon EC2 instance that requires 250 GB of storage space. The application is not used often and has small spikes in usage on weekday mornings and afternoons. The disk I/O can vary with peaks hitting a maximum of 3,000 IOPS. A Solutions Architect must recommend the most cost-effective storage solution that delivers the performance required.

Which configuration should the Solutions Architect recommend?

Which solution should the solutions architect recommend?

- ☐ Amazon EBS Throughput Optimized HDD (st1)
- ☐ Amazon EBS Provisioned IOPS SSD (io1)
- ☒ Amazon EBS General Purpose SSD (gp2)
- ☐ Amazon EBS Cold HDD (sc1)

Incorrect

Explanation:

General Purpose SSD (gp2) volumes offer cost-effective storage that is ideal for a broad range of workloads. These volumes deliver single-digit millisecond latencies and the ability to burst to 3,000 IOPS for extended periods of time.

Between a minimum of 100 IOPS (at 33.33 GiB and below) and a maximum of 16,000 IOPS (at 5,334 GiB and above), baseline performance scales linearly at 3 IOPS per GiB of volume size. AWS designs gp2 volumes to deliver their provisioned performance 99% of the time. A gp2 volume can range in size from 1 GiB to 16 TiB.

In this configuration the volume will provide a baseline performance of 750 IOPS but will always be able to burst to the required 3,000 IOPS during periods of increased traffic.

**CORRECT:** "Amazon EBS General Purpose SSD (gp2)" is the correct answer.

**INCORRECT:** "Amazon EBS Provisioned IOPS SSD (io1)" is incorrect. The io1 volume type will be more expensive and is not necessary for the performance levels required.

**INCORRECT:** "Amazon EBS Cold HDD (sc1)" is incorrect. The sc1 volume type is not going to deliver the performance requirements as it cannot burst to 3,000 IOPS.

**INCORRECT:** "Amazon EBS Throughput Optimized HDD (st1)" is incorrect. The st1 volume type is not going to deliver the performance requirements as it cannot burst to 3,000 IOPS.

References:

# TPN

## *Ten Phenomena, Named!*

1. ***Multiplicity*** – the ability to attach multiple EBS volumes to an EC2 instance.
2. ***Devotion*** – the inability of an EBS volume to be attached to more than one EC2 instance.
3. ***Independence*** – the ability of an EBS volume to be attached to zero EC2 instances.
4. ***Proximity*** – the requirement that an EBS volume is in the same AZ as the EC2 instance it is attached to.
5. ***Default deletion*** – the tendency of root EBS volumes to be deleted upon termination of the EC2 instance they are attached to.
6. ***Non deletion*** – the tendency of extra, non-boot volumes to *not* be deleted on termination.
7. ***Ongoing upgrade*** – the ability to upgrade the size and type of an EBS volume without downtime (does not apply to magnetic volumes).
8. ***Unshrinkability*** – the inability to decrease the size of an EBS volume.
9. ***Feeding the Five Thousand*** – the inability to have more than 5,000 EBS volumes.
10. ***Launch only*** – the inability to specify instance store volumes for an instance at any time *other* than when the instance is launched.
  
11. ***Multiplicity*** – the ability to attach multiple EBS volumes to an EC2 instance.
12. ***Devotion*** – the inability of an EBS volume to be attached to more than one EC2 instance.
13. ***Independence*** – the ability of an EBS volume to be attached to zero EC2 instances.
14. ***Proximity*** – the requirement that an EBS volume is in the same AZ as the EC2 instance it is attached to.
15. ***Default deletion*** – the tendency of root EBS volumes to be deleted upon termination of the EC2 instance they are attached to.

16. ***Non deletion***– the tendency of extra, non-boot volumes to *not* be deleted on termination.
17. ***Ongoing upgrade*** – the ability to upgrade the size and type of an EBS volume without downtime (does not apply to magnetic volumes).
18. ***Unshrinkability*** – the inability to decrease the size of an EBS volume.
19. ***Feeding the Five Thousand*** – the inability to have more than 5,000 EBS volumes.
20. ***Launch only*** – the inability to specify instance store volumes for an instance at any time *other* than when the instance is launched.

## Review questions

# Glossary

### **Data volume**

Description of what term means here.

### **Burst bucket**

### **Root volume**

Generally the word “root” means fundamental, or foundational.

## **Boot volume**

Description of what term means here.

## **Gibibyte**

Description of what term means here.

## **MBR**

Master boot record.

## **GPT**

GUID partition table.

## **Bootable volume**

Bootable `st1` volumes are not supported.

## **NVMe**

Description of what term means here.

## **SSD**

Solid-state drive.

## **SATA**

Solid-state drive.

## **Block device mapping**

Description of what term means here.

## **Attachment**

Description of what term means here.

## **Instance store**

This storage is located on disks that are physically attached to the host computer. Offers high performance and low latency. The

cost of instance stores is included in the instance charges, so it can be more cost effective than EBS-provisioned IOPS.

## **Snapshot**

Roughly, an image of a volume.

Used for backups. Neal Davis writes “snapshots capture a point in time of an instance” (p.33). Snapshots are stored on S3.

Snapshots are useful for the following:

1. Sharing data with other users or accounts
2. Migrate a system to a new AZ or Region
3. Convert an unencrypted volume to an encrypted volume

## **IOPS**

The first letter in this acronym stands for “input/output”. The second letter stands for “operations”. The PS at the end is “per second”.

## **RAID**

Redundant Array of Inexpensive Disks.

## **EBS-backed**

Adjective applied to EC2 instances. As in, *this EC2 instance is EBS-backed*. An instance is EBS backed if and only if: the root volume of the instance is an EBS volume.

## **Image**

Elas

## **Persistent**

Elas

## **MTTR**

Stands for “mean time to repair”. In other words, the average time taken to repair the system such that it is online or available.

## **MTTF**

Stands for “mean time to failure”. The time until a failure in a system. *How long does the system take to fail?*

## **HDD**

Stands for “Hard Disk Drive”.

## **Block device**

Answer

## **ACM TOS**

The *ACM Transactions on Storage*. This is a scholarly journal for publishing advancements in storage research and practice. ACM stands for Association for Computing Machinery. At present, the most cited authors are Ramesh Govindan and Deborah Estrin. It *appears* to have begun in 2005. View the journal’s website [here](#).

## **Ephemeral**

Adjective, usually used in front of the word ‘storage’ or the word ‘volume’ to denote that it is in some way temporary. AWS’s Instance Store provides ephemeral storage.

It’s generally held that: If storage is ephemeral, then it is *not* persistent. Similarly, If storage is persistent, then it is not ephemeral.

# Bibliography

XXXIII. Official

XXXIV. Unofficial

XXXV. Critical

XXXVI. General

## I. Official

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

[https://www.youtube.com/watch?v=gUYa7RzrNhM&ab\\_channel=AmazonWebServices](https://www.youtube.com/watch?v=gUYa7RzrNhM&ab_channel=AmazonWebServices)

## II. Unofficial

### **[Patterson 1988]**

Patterson, David A, Garth Gibson and Randy H Katz (1988). A Case for Redundant Array of Inexpensive Disks (RAID). ACM SIGMOD Conference. Available at:  
<https://www2.eecs.berkeley.edu/Pubs/TechRpts/1987/CSD-87-391.pdf>

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

## III. Critical



### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

## V. General

### **[Gibson 1999]**

Gibson, Garth A, David F Nagel, William Courtright II, Nat Lanza, Paul Mazanitis, Marc Unangst and Jim Zelenka (1999). “NASD Scalable Storage Systems”. Proceedings of USENIX 1999, Linux Workshop, Monterey, CA. June 9-11 1999. Available at:  
<https://www.pdl.cmu.edu/PDL-FTP/NASD/elixir.pdf>

### **[Gibson 2005]**

Gibson, Garth (2005). “The Path from Physical RAID to Virtual Object Storage”. 17<sup>th</sup> June 2005. YouTube Channel: infraNET Project. Available at:  
[https://www.youtube.com/watch?v=kDQJTWmBQE4&ab\\_channel=infraNETProject](https://www.youtube.com/watch?v=kDQJTWmBQE4&ab_channel=infraNETProject)

### **[Gibson 1998]**

Gibson et al. “A cost-effective, high-bandwidth storage architecture”. ACM SIGOPS Operating Systems Review, vol 32, Issue 5 pp. 92-103. Available at: <https://dl.acm.org/doi/abs/10.1145/384265.291029>

### **[Hulen 2002]**

Hulen, Harry, Otis Graf, Keith Fitzgerald and Richard Watson (2002). “Storage Area Networks and the High Performance Storage System”. Tenth NASA Goddard Conference on Mass Storage Systems. Available at:  
<https://storageconference.us/2002/papers/d03ap-hhu.pdf>

### **[Wikipedia]**

“Direct-Attached Storage”. *Wikipedia*. Available at:  
[https://en.wikipedia.org/wiki/Direct-attached\\_storage](https://en.wikipedia.org/wiki/Direct-attached_storage)

### **[Bandulet 2007]**

Bandulet, Christian (2007). “The Storage Evolution: From Blocks, Files and Objects to Object Storage Systems”. Available at:  
[https://www.snia.org/sites/default/education/tutorials/2007/spring/storage/The\\_Storage\\_Evolution.pdf](https://www.snia.org/sites/default/education/tutorials/2007/spring/storage/The_Storage_Evolution.pdf)

### **[Factor 2005]**

Factor, Michael et al. (2005). “Object Storage: the Future Building Block for Storage Systems”. Available at:  
<https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.122.3959&rep=rep1&type=pdf>

### **[Buchholz 1962]**

Computer System: Project Stretch. Werner Buchholz (Ed.). New York: McGraw-Hill Book Company, Inc. Available at:  
[http://archive.computerhistory.org/resources/text/IBM/Stretch/pdfs/Buchholz\\_102636426.pdf](http://archive.computerhistory.org/resources/text/IBM/Stretch/pdfs/Buchholz_102636426.pdf)

### **[Mesnier 2003]**

Available at:  
[https://www.cse.psu.edu/~buu1/teaching/spring07/598d/\\_assoc/45906ADE1F30470392B0017DE0965C08/object.pdf](https://www.cse.psu.edu/~buu1/teaching/spring07/598d/_assoc/45906ADE1F30470392B0017DE0965C08/object.pdf)

<https://www.doc-developpement-durable.org/file/Projets-informatiques/cours-&-manuels-informatiques/stockage/Introduction%20to%20Storage%20Area%20Networks%20and%20System%20Networking.pdf>

[https://nscpolteksby.ac.id/ebook/files/Ebook/Computer%20Engineering/EMC%20Information%20Storage%20and%20Management%20\(2009\)/11.%20Chapter%206%20-%20Storage%20Area%20Networks.pdf](https://nscpolteksby.ac.id/ebook/files/Ebook/Computer%20Engineering/EMC%20Information%20Storage%20and%20Management%20(2009)/11.%20Chapter%206%20-%20Storage%20Area%20Networks.pdf)

<https://www.ibm.com/cloud/blog/san-vs-nas>

<https://www.oreilly.com/library/view/using-sans-and/0596001533/>

# EBS Module 2

Bursting; partition schemes

[RETURN TO EBS](#)

In this module, we will discuss the capacity of EBS volumes to **burst**. We also introduce partition schemes. We will concern ourselves with two partition schemes: **MBR** and **GPT**.

## What do we mean by “bursting”?

Your EBS volume in fact has IO credit balance. In the words of Dougal Ballantyne “every single gp2 volume which is created... comes with a 5.4 million I/O credit bucket. So, it’s automatically ready to go with 5.4 million IOPS” (2015).

What are we to make of this? Well, let’s think about what “credit” means. We seem to be using it in this sense:

money that a bank or business will allow a person to use and then pay back in the future

Similarly, I/O credits *allow* a volume to use a certain number of input/output operations per second. To talk about how many I/O credits a volume has is to talk about the operations per second it’s capable of.

Now let's think about bursting, in general. In cloud computing, this refers to a temporary increase in capacity.

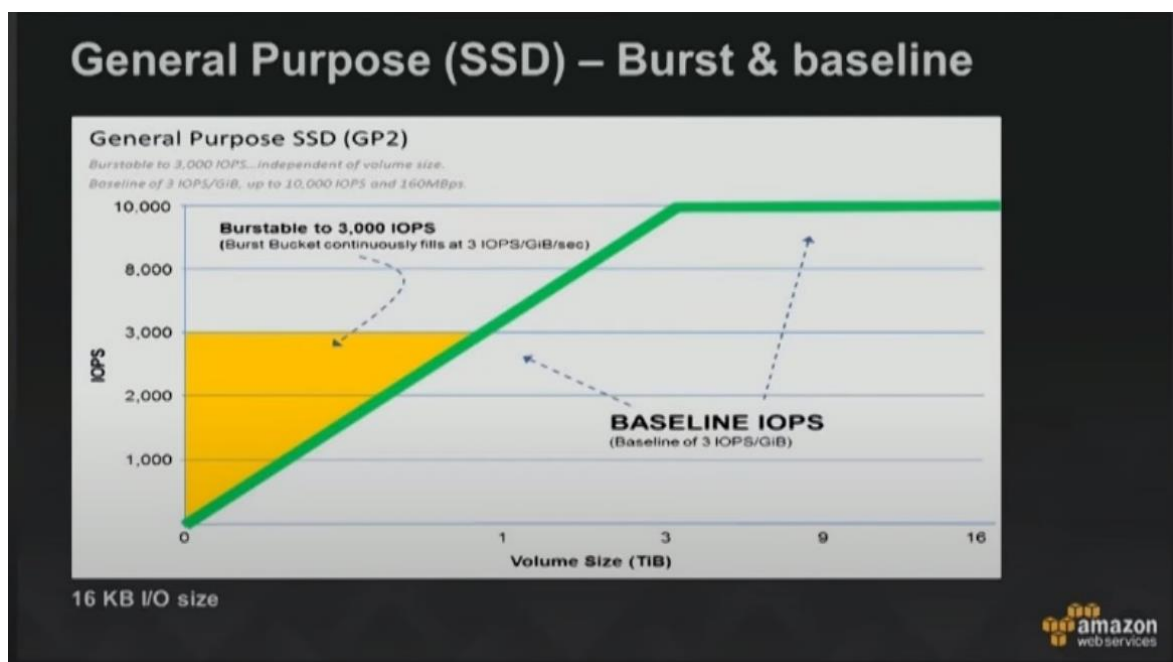
In 2015, Ballantyne stated:

every volume in gp2 can burst up to 3,000 IOPS,  
or go higher if the baseline performance allows it

What does this mean? I in fact think this statement is ambiguous. In other words, it has two meanings. It could mean (1) a gp2 volume can extend its IOPS by 3,000. So, if it was 1,000 IOPS, it can become 4,000. If it was 2,000 IOPS, it can become 5,000.

Alternatively, it could mean (2) that the volume can increase (by a variable amount) such that it achieves 3,000 IOPS. So, if it was 1,000 IOPS, it can extend to 3,000. If it was 2,000, it can also extend to 3,000. And *this*, I think, is Ballantyne's meaning. When we say a volume "can burst up to X", X represents the final amount achieved, *not* the IOPS added in order to achieve the final amount.

Indeed, we're presented with the following graph:



Allow a moment for the graph to speak to you. On the horizontal is the size of the volume, and on the vertical the number of operations (input/output, specifically) being achieved every second.

Because the green line gradually increases from the origin (it climbs, from left-to-right), it's clear that:

Larger volumes can achieve a greater number of IOPS.

This is only true up to a point. Eventually the green line becomes horizontal (on the right-hand side of the image). This indicates that once your volumes are a certain size, increasing their size further will do nothing for the IOPS.

What does the yellow area mean? Well, consider the edge of this area which is bounded by the green line. Any point on this green line maps a particular size of volume to a particular number of volumes.

Well, imagine that you have the tip of a pencil, pointed at one point on this line. You could move the tip upwards, into the yellow area. As you move it upwards, the IOPS would be increasing. The effect of putting a yellow, solid area on the graph is this: particular sizes of volume are no longer mapped to one value for IOPS, but a whole *range* of values of IOPS.

For example, if you started on the horizontal axis, at half a tebibyte. You can see that this maps to about 1,500 IOPS (because of the point on the green line). However, the yellow area is above this point, effectively stating that:

A volume size of half a tebibyte gets you 1,500 IOPS, or any IOPS above this, up to 3,000.

And that's the idea of bursting, in this context: a particular sized volume achieving more IOPS than it's size gives it "by default".

Clearly, for some points on the green line, there is no yellow area above them. For gp2 volumes that have this size, bursting is not possible. Here is how Duncan Ballantyne puts it:

As we start to pass the 1 [tebibyte] volume size, the green line—our baseline performance—exceeds 3,000 IOPS. So at that point, our baseline performance is higher than our burst performance.

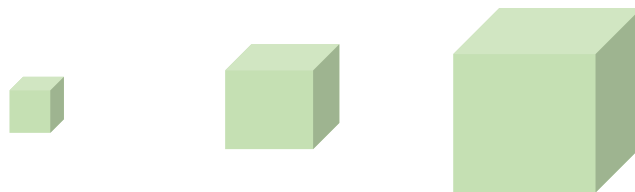
So, “baseline performance” is Ballantyne’s term for the IOPS that a particular size of volume “naturally” achieves. The “baseline performance” is the IOPS that a volume achieves when it is *not* bursting.

## What is a burst bucket?

This is the term we use for the collection of I/O credits that an EBS volume has. The bucket is constantly accumulating credits. How much it accumulates every second depends on the size of the volume. Larger volumes accumulate more per second. To be more precise:

The bucket accumulates 3 IOPS  
Per gibibyte  
Per second

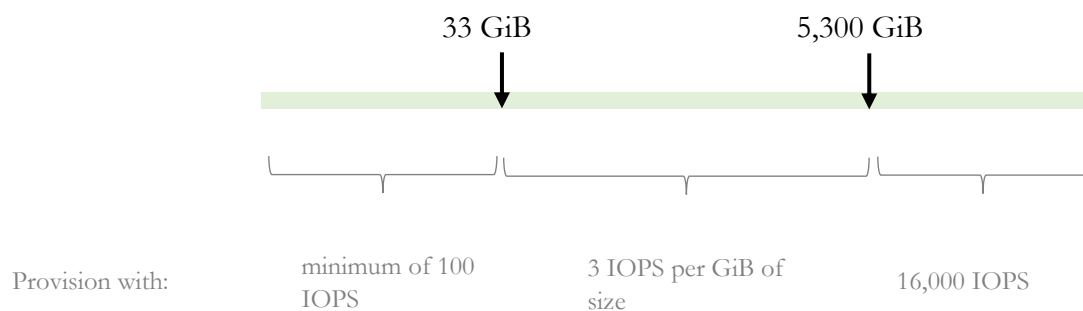
In the AWS documentation, we’re told how things differ for differently sized volumes (small, medium, and huge).



AWS tell us how they decide how many IOPS to provision these differently-sized volumes:

1. Volumes 33.33 GiB and smaller are provisioned with
  - the minimum of 100 IOPS.
2. Volumes larger than 33.33 GiB are provisioned with
  - 3 IOPS per GiB of volume size...
    - up to the maximum of 16,000 IOPS, (which is reached at 5,334 GiB (3 X 5,334).)
3. Volumes 5,334 GiB and larger are provisioned with
  - 16,000 IOPS.

That's all quite convoluted, and particularly in (2) there are a lot of conditions and rejoinders. If you visualise a spectrum of volume sizes (from 0 up to several gigabytes), these are the significant points:





The documentation continues:

A volume's ability to burst is governed by I/O credits.

They then provide us with this equation:

$$\text{Burst duration} = \frac{(\text{I/O credit balance})}{(\text{Burst IOPS}) - (\text{Baseline IOPS})}$$

Let's try and make sense of each of the elements in this equation. On the left is the "burst duration". This is easy enough to make sense out of. Presumably it is in seconds.

On the right-hand side is the "I/O credit balance". Again, this makes sense. Some values might be 2 million credits or 3 million credits.

But what is meant by "burst IOPS"? Are we talking about the number of IOPS you *arrive* at, having bursted? Or are we talking about the size of the burst (i.e. the number of IOPS which is *added* to the base amount)?

Let's think about credits for a moment. It seems reasonable that you are only ever doing one of three things: *earning* credits, *spending* credits, or just *maintaining* your amount of credits. Of course, if you earn credits at the same rate that you spend them, then you will essentially be *maintaining* the amount. In other words,

there are two ways to maintain your credits: (1) doing nothing at all or (2) spending at the rate at which you earn. We're told about the situations in which you *spend* credits:

When I/O demand is greater than baseline performance, the volume **spends I/O credits** to burst to the required performance level (up to 3,000 IOPS).

While bursting, I/O credits are spent at a rate of 3 I/O credits per GiB of volume size per second.

We are also told about situations in which you *earn* credits:

When I/O demand drops to baseline performance level or lower, the volume starts to **earn I/O credits** at a rate of 3 I/O credits per GiB of volume size per second.

## Constraints on size

AWS tell us:

The size of an Amazon EBS volume is constrained by the physics and arithmetic of block data storage, as well as by the implementation decisions of operating system (OS) and file system designers.

One of the things which constrains the size of an EBS volume is the partitioning scheme. Now, what on earth is a partitioning scheme? Well, there are really only two which we will discuss here:

1. Master boot record (MBR)
2. GUID Partition Table (GPT)

Let's explain what a master boot record is.

### Master Boot Record (MBR)

MBR is much older than GPT. Master Boot Record (MBR) was first introduced on PCs in 1983. Since then, it has become the de facto standard.

As the name suggests, Master Boot Record, commonly known as MBR, is the first (main) sector of a hard disk and determines

the location of the operating system (OS) to complete the execution of the booting process.

[Scaler topics]

So, even though it is called a “record”, the master boot record is really a *sector*. It is a sector on the hard disk. Wikipedia tells us that a master boot record is a special type of **boot sector** [Wikipedia 1].

What is a boot sector? A boot sector is the sector of a persistent storage device which contains machine code. This code is loaded into Random Access Memory (RAM) and then executed by a computer’s system’s built-in firmware (usually the BIOS).

Usually, the very first sector of the hard disk is the boot sector, regardless of sector size (512 or 4096 bytes).

Why do we define one particular sector as the *boot sector*? Well, we achieve interoperability between firmware and various operating systems.

A **master boot record** is a *special type* of boot sector found at the very beginning of partitioned mass storage devices. The concept of MBRs was publicly introduced in 1983 with PC DOS 2.0.

The MBR holds information regarding how the disk’s sectors are divided into partitions, each partition notionally containing a file system. The MBR also contains executable code to function as a loader for the installed operating system. This MBR code is usually referred to as a *boot loader*.

Here is a definition of master boot record:

A special area on a computer’s main hard disk that gives the location of the disk’s boot block or bootable partition where the operating system is installed.

[FOLDOC]

MBRs are not present on non-partitioned media such as floppies, superfloppies or other storage devices configured to behave as such.

## A bit of history

The master boot record (MBR) partitioning scheme, widely used since the early 1980s, imposes limitation for the use of modern hardware. The available size for block addresses and related information is limited to 32 bits.

For hard disks with 512-byte sectors, the MBR partition table entries allow a maximum size of 2 TiB ( $2^{32} \times 512$  bytes) or 2.20 TB ( $2.20 \times 10^{12}$  bytes).

In the late 1990s, Intel developed a new partition table format as part of what eventually became the Unified Extensible Firmware Interface (UEFI). The GUID Partition Table is specified in Chapter 5 of the UEFI 2.8 specification.

GPT uses 64 bits for logical block addresses, allowing a maximum disk size of  $2^{64}$  sectors.

For disks with 512-bytes sectors, the maximum size is 8 ZiB or 9.44 ZB.

For disks with 4,096-byte sectors, the maximum size is 64 ZiB or 75.6 ZB.

In 2010, hard-disk manufacturers introduced drives with 4,096-byte sectors (Advanced Format). For compatibility with legacy hardware and software, those drives include an emulation technology (512e) that presents 512-byte sectors to the entity accessing the hard drive, despite their underlying 4,096-byte physical sectors.

## Features

Like MBR, GPTs use logical block addressing (LBA) in place of the historical cylinder-head-sector (CHS) addressing.

The protective MBR is stored at LBA 0 and the GPT header is in LBA 1. The GPT header has a pointer to the partition table (Partition Entry Array), which is typically at LBA 2.

Each entry on the partition table has a size of 128 bytes.

The UEFI specification stipulates that a minimum of 16,384 bytes, regardless of sector size, are allocated for the Partition Entry Array.

Thus, on a disk with 512-byte sectors, at least 32 sectors are used for the Partition Entry Array, and the first usable block is at LBA 34 or higher, while on a 4,096-byte sectors disk, at least 4 sectors are used for the

Partition Entry Array, and the first usable block is at LBA 6 or higher.

## The GUID Partition Table

The GUID Partition Table (GPT) is a standard for the layout of partition tables of a physical computer storage device, such as a hard disk drive or solid-state drive, using universally unique identifiers.

Universally unique identifiers are also known **globally unique identifiers** (GUIDs).

Forming a part of the Unified Extensible Firmware Interface (UEFI) standard, it is nevertheless also used for some BIOS systems, because of the limitations of master boot record (MBR) partition tables, which use 32 bits for logical block addressing (LBA) of traditional 512-byte disk sectors.

All modern personal computer operating systems support GPT. Some, including macOS and Microsoft Windows on the x86 architecture, support booting from GPT partitions only on systems with EFI firmware. Free BSD and most Linux distributions can boot from GPT partitions on systems with either the BIOS or the EFI firmware interface.

## How does all this relate to EBS?

# TPN

*Ten Phenomena, Named!*

1.     ***Phenomenon1*** – the tendency of X to Y.
2.     ***Phen2*** – the tendency of X to Y.
3.     ***Phen3*** – the tendency of X to Y.
4.     ***Phen4*** – the tendency of X to Y.
5.     ***Phen5*** – the tendency of X to Y.
6.     ***Phen6*** – the tendency of X to Y.
7.     ***Phen7*** – the tendency of X to Y.
8.     ***Phen8*** – the tendency of X to Y.
9.     ***Phen9*** – the tendency of X to Y.
10.    ***Phen10*** – the tendency of X to Y.

## Review questions

# Glossary

### Partition

Description of what term means here.

## **MBR**

Master boot record.

## **GPT**

Stands for GUID Partition Table.

## **Partition entry array**

Stands for GUID Partition Table.

## **LBA**

Stands for Logical block addressing.

## **CHS**

Cylinder-head-sector. CHS is an early method for giving addresses to each physical block of data on a hard disk drive.

## **Partition scheme**

Description of what term means here.

## **Partition table**

Description of what term means here.

# Bibliography

XXXVII. Official

XXXVIII. Unofficial

XXXIX. Critical

## XL. General

### I. Official

### II. Unofficial

#### [Wikipedia 1]

“Boot Sector”. Available at:  
[https://en.wikipedia.org/wiki/Boot\\_sector](https://en.wikipedia.org/wiki/Boot_sector)

#### [Wikipedia 2]

“Master boot record”. Available at:  
[https://en.wikipedia.org/wiki/Master\\_boot\\_record](https://en.wikipedia.org/wiki/Master_boot_record)

#### [CodeCamp]

<https://www.freecodecamp.org/news/mbr-vs-gpt-whats-the-difference-between-an-mbr-partition-and-a-gpt-partition-solved/>

#### [FOLDOC]

Free Online Dictionary of Computing (FOLDOC). “Master Boot Record”. [Archived] Available at:  
<https://web.archive.org/web/20170824002628/https://foldoc.org/master%20boot%20record>

#### [Wikipedia 3]

Wikipedia. “Cylinder-head-sector”. Available at:  
<https://en.wikipedia.org/wiki/Cylinder-head-sector>



## [Unknown 1]

“The Master Boot Record and Why is it necessary?” Available at:  
[https://eecs.wsu.edu/~sshaikot/docs/Other/master\\_boot\\_record.pdf](https://eecs.wsu.edu/~sshaikot/docs/Other/master_boot_record.pdf)

## [GreenHornLinux]

“12\_Learning about MBR (Master Boot Record)”. 20<sup>th</sup> Aug 2012.  
YouTube Channel: GreenHornLinux. Available at:  
[https://www.youtube.com/watch?v=PM91jOnLyrM&ab\\_channel=GreenHornLinux](https://www.youtube.com/watch?v=PM91jOnLyrM&ab_channel=GreenHornLinux)

## [YouTube 2014]

“MBR - Master Boot Record”. 17<sup>th</sup> Dec 2014. YouTube channel:  
ReclaiMe Data Recovery. Available at:  
[https://www.youtube.com/watch?v=CcYgyvCFd\\_w&ab\\_channel=ReclaiMeDataRecovery](https://www.youtube.com/watch?v=CcYgyvCFd_w&ab_channel=ReclaiMeDataRecovery)

## [FreeTraining 2013]

“MBR and GPT Partition Tables”. YouTube channel: itfreetraining.  
18<sup>th</sup> Dec 2013. Available at:  
[https://www.youtube.com/watch?v=vMB8uyosdOA&ab\\_channel=itfreetraining](https://www.youtube.com/watch?v=vMB8uyosdOA&ab_channel=itfreetraining)

## [Naik 2022]

Naik, Vritika (2022). “What is Master Boot Record?”. *Scaler Topics*.  
Available at: <https://www.scaler.com/topics/operating-system/master-boot-record/>



# EBS Module 3

EBS Multi-Attach; the Nitro System; the Nitro hypervisor; Nitro enclaves.

[RETURN TO EBS](#)

In this module, I want to talk about EBS Multi-Attach. Then, I will move onto the “Nitro System”. This will involve a brief discussion of the Nitro hypervisor and Nitro enclaves.

## EBS Multi-Attach

On Valentine’s Day in 2020, AWS announced a new EBS feature, *Multi-Attach*:



## Amazon EBS Multi-Attach now available on Provisioned IOPS io1 volumes

Posted On: Feb 14, 2020

Today we are announcing general availability of Multi-Attach on Amazon Elastic Block Store (Amazon EBS) volumes. You can now enable Multi-Attach on Amazon EBS Provisioned IOPS io1 volumes to allow a single volume to be concurrently attached to up to sixteen AWS Nitro System-based Amazon Elastic Compute Cloud (Amazon EC2) instances within the same Availability Zone. Each attached instance has full read and write permission to the shared volume. For applications that manage storage consistency from multiple writers, Multi-Attach makes it easier to achieve higher application availability.

# The Nitro System

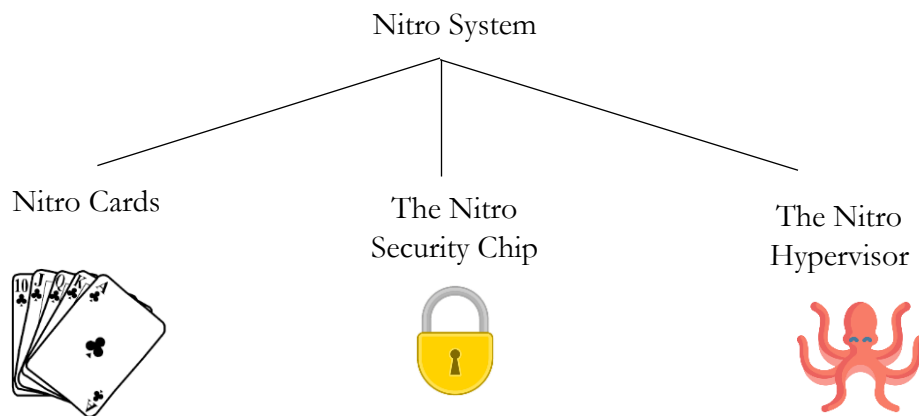
What is the AWS Nitro System? The “SYSTEM” consists of three parts:

The Nitro System is comprised of three main parts: the Nitro Cards, the Nitro Security Chip, and the Nitro Hypervisor.

[Vogels 2020]

The most important element is perhaps the Nitro hypervisor.

*AWS have invented they're own hypervisor!*



Here is some background from the Chief Technology Officer at AWS:

In the early days of EC2, we used the **Xen hypervisor**, which is purely software-based, to protect the physical hardware and system firmware; virtualize the CPU, storage, and networking; and provide a rich set of management capabilities.

But with this architecture, as much as 30% of the resources in an instance were allocated to the hypervisor and operational management for network, storage, and monitoring.

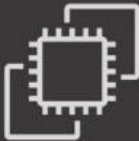
[Vogels 2020]

## When was the Nitro System launched?

Ravi Murty said: “we announced the Nitro system in 2017, with the launch of our C5 instances. But actually, we’ve been working on the Nitro System for many years now.” [Murty 2019].

Murty’s slide suggests that the Nitro System has been in development since 2013:

Nitro: Two years later



**AWS Nitro**

- Launched in November 2017
- In development since 2013
- Purpose-built hardware/software
- Hypervisor built for AWS

All new instance launches use the Nitro System

## What is *Nitro Enclaves*?

Generally, an *enclave* (pronounced ON-clave) is a portion or territory surrounded by a larger territory. The inner area tends to have a population that is culturally distinct from the surrounding area.

On 28<sup>th</sup> October 2020, a video emerges from AWS, in which Colm MacCartheigh provides this definition of a “nitro enclave”:

An isolated, hardened, and highly-constrained virtual machine.

[AWS 2]

The video lists the following features of Nitro Enclaves, which I have numbered:

1. Not a container
2. No persistent storage
3. No administrator or operator access
4. Communication between your instance and your enclave is over a secure local channel
5. Lightweight Linux kernel
6. Independent kernel
7. Own encryption keys

#### 4. QUESTION

A company is deploying a fleet of Amazon EC2 instances running Linux across multiple Availability Zones within an AWS Region. The application requires a data storage solution that can be accessed by all of the EC2 instances simultaneously. The solution must be highly scalable and easy to implement. The storage must be mounted using the NFS protocol.

Which solution meets these requirements?

- ☐ Create an Amazon EBS volume and use EBS Multi-Attach to mount the volume to all EC2 instances across each Availability Zone.
- ☐ Create an Amazon RDS database and store the data in a BLOB format. Point the application instances to the RDS endpoint.
- ☒ Create an Amazon EFS file system with mount targets in each Availability Zone. Configure the application instances to mount the file system.
- ☐ Create an Amazon S3 bucket and create an S3 gateway endpoint to allow access to the file system using the NFS protocol.

Correct

#### Explanation:

Amazon EFS provides scalable file storage for use with Amazon EC2. You can use an EFS file system as a common data source for workloads and applications running on multiple instances. The EC2 instances can run in multiple AZs within a Region and the NFS protocol is used to mount the file system.

With EFS you can create mount targets in each AZ for lower latency. The application instances in each AZ will mount the file system using the local mount target.

**CORRECT:** "Create an Amazon EFS file system with mount targets in each Availability Zone. Configure the application instances to mount the file system" is the correct answer.

**INCORRECT:** "Create an Amazon S3 bucket and create an S3 gateway endpoint to allow access to the file system using the NFS protocol" is incorrect. You cannot use NFS with S3 or with gateway endpoints.

**INCORRECT:** "Create an Amazon EBS volume and use EBS Multi-Attach to mount the volume to all EC2 instances across each Availability Zone" is incorrect. You cannot use Amazon EBS Multi-Attach across multiple AZs.

**INCORRECT:** "Create an Amazon RDS database and store the data in a BLOB format. Point the application instances to the RDS endpoint" is incorrect. This is not a suitable storage solution for a file system that is mounted over NFS.

# Glossary

## ASIC

Stands for Application Specific Integrated Circuit.

## **Nitro System**

Term used to describe the combination of (1) the Nitro security chip (2) Nitro cards, and (3) the Nitro hypervisor.

## **Xen**

Gggggggg

## **Nitro Enclave**

Gggggggg

## **Nitro hypervisor**

gggggggg

# Bibliography

- I. Official
- II. Unofficial
- III. Critical
- IV. General

## I. Official



## [AWS 1]

Available at: <https://aws.amazon.com/about-aws/whats-new/2020/02/ebs-multi-attach-available-provisioned-iops-ssd-volumes/>

## [AWS 2]

“Introducing AWS Nitro Enclaves”. 28<sup>th</sup> Oct 2020. Short video featuring Colm MacCarthaigh. Available at: [https://www.youtube.com/watch?v=PZTtJu1QtBE&ab\\_channel=AmazonWebServices](https://www.youtube.com/watch?v=PZTtJu1QtBE&ab_channel=AmazonWebServices)

## [Roberts 2020]

Roberts, Steve (2020). “New – Multi-Attach for Provisioned IOPS (io1) Amazon EBS Volumes” *AWS Blog*. 14<sup>th</sup> Feb 2020. Available at: <https://aws.amazon.com/blogs/aws/new-multi-attach-for-provisioned-iops-io1-amazon-ebs-volumes/>

## [Liguori 2018]

Liguori, Anthony (2018). Powering Next-Gen EC2 Instances.

Available at:

[https://www.youtube.com/watch?v=e8DVmwj3OEs&ab\\_channel=AmazonWebServices](https://www.youtube.com/watch?v=e8DVmwj3OEs&ab_channel=AmazonWebServices)

## [Jassy 2018]

Jassy, Andy (2018). Available at:

[https://www.youtube.com/watch?v=ZOIkOnW640A&ab\\_channel=AmazonWebServices](https://www.youtube.com/watch?v=ZOIkOnW640A&ab_channel=AmazonWebServices)

## [Murty 2019]

Murty, Ravi (2019). Powering next-gen Amazon EC2: Deep dive into the Nitro System. Available at:

[https://www.youtube.com/watch?v=rUY-00yFIE4&t=1s&ab\\_channel=AWSEvents](https://www.youtube.com/watch?v=rUY-00yFIE4&t=1s&ab_channel=AWSEvents)

# II. Unofficial

## [Vogels 2020]

Vogels, Werner (2020). Reinventing virtualization with the AWS Nitro System. Available at:  
<https://www.allthingsdistributed.com/2020/09/reinventing-virtualization-with-nitro.html>

[Hamilton 2018]

“AWS Nitro System”. Blog. Available at:  
<https://perspectives.mvdirona.com/2019/02/aws-nitro-system/>

### III. Critical

### IV. General



# EBS Module

## 4

[RETURN TO EBS](#)

### Snapshots



In computer systems, a snapshot is the state of a system at a particular point in time. The term comes from photography and has been used in storage contexts for a while now. AWS provides a snapshot facility with EBS.

These things—snapshots—have an enormous list of properties. It is important to plod through them and learn the details of the *behaviour* of snapshots. I will do my best to systematise this enormous list.

First, note that snapshots can be thought of as images because they represent things. The thing which snapshots represent are *EBS volumes*. Snapshots are not EBS volumes. Snapshots are a different sort of thing than EBS volumes.

Why do AWS describe snapshots as “point-in-time”? *Surely everything exists in time*. To respond, we can conceive of a hard drive that *continuously* mirrors some other storage. Snapshots don’t do this. A point in time is different from, say, a period of time. Snapshots are static things, like photographs on a piece of paper. They don’t track the state of an EBS volume, as its data is altered.

However, we can take *lots* of snapshots, to the extent that we’re *continuously taking snapshots* (for example, think of stop-motion animations). When we take multiple snapshots of a single EBS volume, an interesting feature is revealed. The EC2 documentation writes:

Snapshots are *incremental* backups, which means that only the blocks on the device that have changed after your most recent snapshot are saved. This minimizes the time required to create the snapshot and saves on storage costs by not duplicating data.

The “device” they talk about is a block device. This is the thing which breaks up your “file” into blocks and spreads them about. You can suppose they simply wrote “EBS volume” in that paragraph, for now.

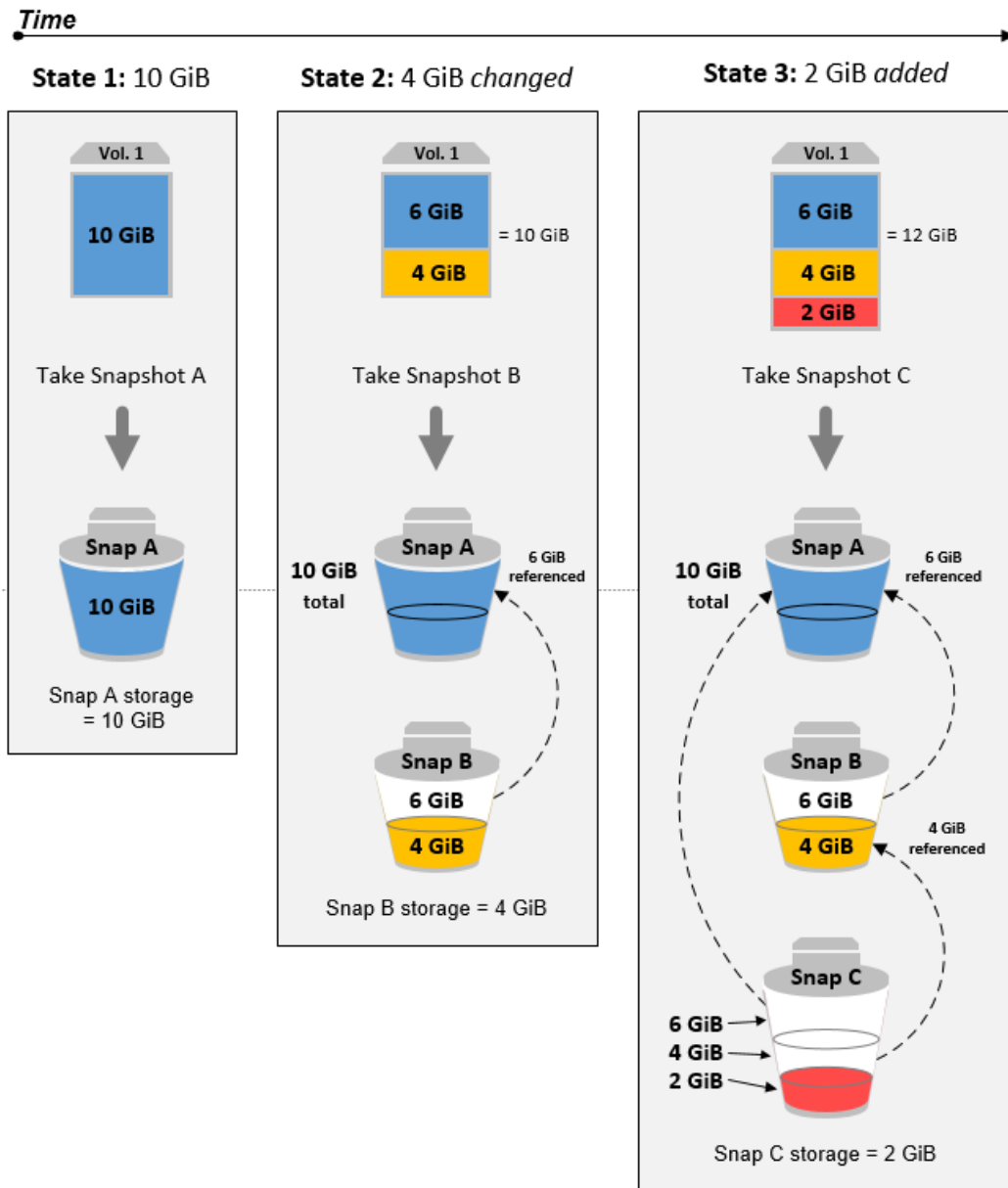
What does it mean for snapshots to be *incremental* backups? It means that a particular snapshot is only going to record things that have changed since the last snapshot. It’s not going to bother “capturing” the whole EBS volume if only a part of it has changed.

Think of the cliché in movies, when two people are being consulted for their opinion. The first character spits out some impressive, intelligent suggestion. The second character, for comedic effect, haplessly goes *What he said*.<sup>1</sup>

Just as these characters refer to one another, snapshots refer to one another. They are lazy. Some snapshot will make a record of what’s changed. But if any data has *already* been recorded, they will simply refer to the snapshot that was responsible (for that part, at least). Study this diagram:

---

<sup>1</sup> I refer you to British film *Hot Fuzz* (2007). Strategizing before storming the supermarket, Sergeant Angel listens to Fisher’s suggestion and then says “yeah, what he said”. In *Boom Town* (2005), an episode of Doctor Who, the Doctor says “Like he said”.



The documentation continues:

Each snapshot contains all of the information that is needed to restore your data (from the moment when the snapshot was taken) to a new EBS volume.

So, snapshots can be used to create EBS volumes (again and again). Keep in mind that we have three things in play: EC2

instances, EBS volumes, and snapshots. It's possible to conflate instances and EBS volumes because they're both in *some* sense flexible:

4. A snapshot can be used to create one volume, and then another etc.
5. An EBS volume can be detached from one instance and attached to another.

Clearly, these statements are not the same. Snapshots behave like templates; EBS volumes are portable. Snapshots are stored in S3:

You can back up the data on your Amazon EBS volumes to Amazon S3 by taking point-in-time snapshots.

I have so far told you about three properties. Snapshots provide point-in-time, incremental backups; they reside in S3; and give volumes vigour (contain everything needed to create an EBS volume). I want to make progress now and add five properties to those three. Snapshots are capable of being:

4.       **Shared**  
You can share a snapshot across accounts. You do this by altering the access permissions.

5.       **Copied**  
The documentation tells us that "You can make copies of your own snapshots as well as snapshots that have been shared with you." You can copy a snapshot from one region to another (this will become important later).

6.       **Encrypted**  
AWS *keenly* encrypt. I say this for two reasons. First, if you create a snapshot of a volume that is encrypted, that snapshot will be automatically encrypted (fantastic). Second, if you create a volume from an encrypted snapshot, that volume will be encrypted. So, it's infectious encryption, in both directions.

I previously said that snapshots can give volumes vigour. There's a constraint here, though. AWS write that:

A snapshot is constrained to the AWS Region where it was created.

This is the seventh property of snapshots. Snapshots are *Region-bound*, like patients who are bed-bound. What do you mean "constrained to"? What this means is that when a particular snapshot is used to create EBS volumes, that EBS volume must be in the same region as the snapshot. The assumption is that

snapshots and EBS volumes are regional animals: they're always sat in particular Regions. And they are.

To use a snapshot to create an EBS volume which is in another Region, you must *copy* the snapshot. So, snapshots cannot be moved, but they can be copied.

# TPN

11. ***Phenomenon1*** – the tendency of X to Y.
12. ***Phen2*** – the tendency of X to Y.
13. ***Phen3*** – the tendency of X to Y.
14. ***Phen4*** – the tendency of X to Y.
15. ***Phen5*** – the tendency of X to Y.
16. ***Phen6*** – the tendency of X to Y.
17. ***Phen7*** – the tendency of X to Y.
18. ***Phen8*** – the tendency of X to Y.
19. ***Phen9*** – the tendency of X to Y.
20. ***Phen10*** – the tendency of X to Y.

# Glossary

## Term1

Description of what term means here.



## Term2

Description of what term means here.

## Term3

Description of what term means here.

# Bibliography

- V. Official
- VI. Unofficial
- VII. Critical
- VIII. General

## I. Official

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

## II. Unofficial

Difference between incremental and differential backups.

[https://www.youtube.com/watch?v=o-83E6levzM&ab\\_channel=PowerCertAnimatedVideos](https://www.youtube.com/watch?v=o-83E6levzM&ab_channel=PowerCertAnimatedVideos)

[https://www.youtube.com/watch?v=07EHsPuKXc0&ab\\_channel=IBMTechnology](https://www.youtube.com/watch?v=07EHsPuKXc0&ab_channel=IBMTechnology)

**[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

**[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

## III. Critical

**[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

**[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

# EBS Module

## 5

The Instance Store; concept of a *Root Device* volume; block device mappings (and associated concepts); Physalia; the re-mirroring storm of 2011;

**[RETURN TO EBS](#)**

What on earth is the  
“instance store”?



Tibetan monks at work on a mandala sand painting

To begin to answer this question, let's look at the documentation. AWS write:

An *instance store* provides **temporary** block-level storage for your instance.

Notice how it is “temporary” storage. We can contrast *temporary* storage with *persistent* storage. EBS volumes provide persistent storage. Data stored in the instance store is *not* data stored in EBS volumes—these are two distinct things. However, as the sentence above indicates, it's still true that the instance store provides “block-level” storage. AWS continue:

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

I'm quite certain EBS volumes are *not* physically attached to the host computer. Instead, they are reached over a network. Either way, instance store storage *is* physically connected. AWS go on to tell us what it is good for:

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

Let's break that down. At least four things were mentioned there:



Buffers



Caches



Scratch  
data



fleet of web  
servers

An instance store consists of one or more instance store volumes exposed as block devices. The size of an instance store as well as the number of devices available varies by instance type.

In the documentation, we're told lots of different things about data in the instance store. I've tried to reduce this to a list of ten facts.

### **1. Four ways to lose your instance store data**

AWS are clear that you don't even have to TERMINATE an EC2 instance for the data in the instance store volumes to be lost. You merely have to STOP the instance (STOPPING and EC2 instance is not the same as TERMINATING it). However, somewhat bizarrely, REBOOTING an instance is the big exception here. They write:

The data in an instance store persists only during the lifetime of its associated instance. If an instance reboots (intentionally or unintentionally), data in the instance store persists. However, data in the instance store is lost under any of the following circumstances:

- The underlying disk drive fails
- The instance stops
- The instance hibernates
- The instance terminates

### **2. Data cannot be included in an AMI**

It is possible to create AMIs from instances. If you do, then when you use the created AMI, it won't contain the data that was in the instance store of the original instance. AWS write:

If you create an AMI from an instance, the data on its instance store volumes isn't preserved and

isn't present on the instance store volumes of the instances that you launch from the AMI.

### 3. Changing instance type

AWS write: If you change the instance type, an instance store will not be attached to the new instance type. For more information, see [Change the instance type](#).

### 4. Size matters

Well, this is determined by your instance type. We're told:

The instance type determines the size of the instance store available and the type of hardware used for the instance store volumes

### 5. Not a distinct cost

You don't pay for two distinct things, the instance and the instance store. Rather, "Instance store volumes are included as part of the instance's usage cost." (AWS).

### 6. Specify which to use at launch

You must specify the instance store volumes that you'd like to use when you launch the instance (except for NVMe instance store volumes, which are available by default). Then format and mount the instance store volumes before using them.

### 7. [Instance store] + [SSD]

We've talked about EBS volumes which are backed by SSDs. In this passage, AWS seem to be suggesting that the instance store can *also* use SSDs. And just as the *size*



of the instance store is determined by the instance type, whether SSD is used or not is determined by the instance type.

They write:

Some instance types use NVMe or SATA-based solid state drives (SSD) to deliver high random I/O performance. This is a good option when you need storage with very low latency, but you don't need the data to persist when the instance terminates or you can take advantage of fault-tolerant architectures.

## 8. Encrypted at rest

We're told: "The data on NVMe instance store volumes and some HDD instance store volumes is encrypted at rest."

Well, we get encryption with NVMe and HDD volumes—what hasn't been mentioned here? SSD. Does this mean that SSD volumes are *not* encrypted? Extremely unhelpful statement, AWS.

## 9. The Tremendous Table

We're provided with a tremendous table, which tells you what sort of instance store you're going to get for a particular instance type. It's very large, so I'll provide only a part of it here:

| Instance type | Instance store volumes | Type     | Needs initialization* | TRIM support** |
|---------------|------------------------|----------|-----------------------|----------------|
| c1.medium     | 1 x 350 GB†            | HDD      | ✓                     |                |
| c1.xlarge     | 4 x 420 GB (1.6 TB)    | HDD      | ✓                     |                |
| c3.large      | 2 x 16 GB (32 GB)      | SSD      | ✓                     |                |
| c3.xlarge     | 2 x 40 GB (80 GB)      | SSD      | ✓                     |                |
| c3.2xlarge    | 2 x 80 GB (160 GB)     | SSD      | ✓                     |                |
| c3.4xlarge    | 2 x 160 GB (320 GB)    | SSD      | ✓                     |                |
| c3.8xlarge    | 2 x 320 GB (640 GB)    | SSD      | ✓                     |                |
| c5ad.large    | 1 x 75 GB              | NVMe SSD |                       | ✓              |
| c5ad.xlarge   | 1 x 150 GB             | NVMe SSD |                       | ✓              |
| c5ad.2xlarge  | 1 x 300 GB             | NVMe SSD |                       | ✓              |
| c5ad.4xlarge  | 2 x 300 GB (600 GB)    | NVMe SSD |                       | ✓              |
| c5ad.8xlarge  | 2 x 600 GB (1.2 TB)    | NVMe SSD |                       | ✓              |
| c5ad.12xlarge | 2 x 900 GB (1.8 TB)    | NVMe SSD |                       | ✓              |
| c5ad.16xlarge | 2 x 1,200 GB (2.4 TB)  | NVMe SSD |                       | ✓              |

10.

11. Can instance store volumes be used as root device volumes?

We are told:

**⚠ Important**

Only the following instance types support an instance store volume as the root device: C3, D2, G2, I2, M3, and R3.

## “Root device”



In their documentation on EBS, AWS frequently mention the “root device”. For example, they write:

What is the “root device”? Unfortunately, they do not give a definition of “root device” in the documentation.

Somebody called Tom [asked](#) a question about the nature of the root device recently, on StackExchange. Ian Wilson writes:

Tom: The root device is the virtual device that houses the partition where your filesystem is stored -- ephemeral devices have it running on the same physical host at the server, and EBS devices have it mounted using iSCSI.

– [Ian Wilson](#)

[Jul 6, 2012 at 6:14](#)

So, the root device “houses the partition where your filesystem is store”. That’s *somewhat* understandable. Someone adds:

I agree with what Ian wrote. I would add that the “root device” in EC2 is analogous to the operating system partition in a personal computer. It is where the filesystem of your OS resides.

It is clear that if we are to understand the nature of a “root device”, we must understand the nature of the “operating system partition in a personal computer”.

On [this](#) old page from 2013, we’re told:

Amazon Web Services provides EC2 instances with two types of root devices.: “EBS-backed” and “instance store”.

Does anybody else use this expression “root device” in the way that AWS do? It’s hard to find examples. I have not been able to find any dictionary entries for “root device”. I found the expression used in a Microsoft [article](#) its “device tree”. This reminds me that ROOT is a term used in the contexts of TREES. Think of DNS, for example. There are child and parent nodes. Some nodes (with no children) will be leaf nodes. One node (with no parents) will be the root node. This tells us something about this AWS expression: “root device”.



The root device is perhaps part of a tree of devices, and it has no parents. The *root* device is fundamental.

## Why is it a root device?

I reckon the expression “root device” is helpful precisely because “device” is a general term.

It leaves open when an EBS volume or the Instance Store is being used. If we said “root volume”, instead of “root device”, this would suggest we are using an EBS *volume*. However, we could be using the instance store.

Other things I am less sure about. The “root device” is obviously not a hand-held device such as a mobile phone or laptop. This is one sense of “device”. However, perhaps it is still a physical thing (perhaps an EBS volume).

But maybe not. After all, “device” *can* be [used](#) in a more abstract sense, to simply denote something suited to a particular purpose. For example, Paul Daniels tells us how playing cards became a gambling device in the 1400s.

Early on the EC2 documentation, when they describe AMIs, AWS write:

*All AMIs are categorized as either backed by Amazon EBS or backed by instance store.*

- Amazon EBS-backed AMI – The root device for an instance launched from the AMI is an Amazon Elastic Block Store (Amazon EBS) volume created from an Amazon EBS snapshot.
- Amazon instance store-backed AMI – The root device for an instance launched from the

AMI is an instance store volume created from a template stored in Amazon S3.

[AWS documentation]

We're given two examples of "root devices" here: an EBS volume and an instance store volume. So, EBS volumes are not necessarily "root devices" but they can be. We can ask: what must we add to a plain old EBS volume to make it a "root device"?

## The words "root" and "boot"

It seems that some AWS educators simply interchange *root* and *boot*. I really don't advise this.

The fact is that words are never exactly the same. And these words *do* denote different things. And despite AWS not helping the situation by not coming forth with any definition, "boot" and "root" have distinct usage patterns and connotations.



The word **BOOT** is related to the bootstrap loader: a small programme executed when you physically press the button on a computing device (see the history [here](#)).

This comes up in expressions such as "**boot disk**", "**boot floppy**" or even "**boot drive**". [Biersdorfer 1998].

Thus, the words "boot" and "root" are entirely different. The adjective "root" is for items occupying the **supreme** position in a tree. The adjective "boot" is for items which allow a system to start up.

I'm able to bring up the documentation on EBS (which is within the EC2 user guide) and analyse the uses of "boot" and "root".

I've tried to enumerate all the senses in which "boot" is used:

1. As a noun, denoting the event that is: booting a server.

“Use the `chkconfig` command to configure the Apache web server to start at each system boot”

“If you want the MySQL Server to start at every boot, type the following command”

“UEFI Boot”

“UEFI secure boot”

2. As a verb, denoting something *done*.

“...set it to start each time the **system** boots”.

“The main differences between **PV and HVM** are the way in which they **boot** and whether they can take of special hardware extensions (CPU, network, and storage) for better performance.”

“When a **computer** boots...”

“After the **VM** has booted”.

“Having a fallback kernel enables the **instance** to boot even if the new kernel isn’t found”.

3. As an adjective,
  - i. BOOT MODES
  - ii. BOOT TIME  
“boot time **for an instance**”
  - iii. BOOT CYCLE
  - iv. BOOT IMAGES

What about the word “root”? It is most frequently used in the expression “**root device volume**”. Sometimes,

the expression “root volume” is used. I believe this is just an abbreviation of “root device volume”.

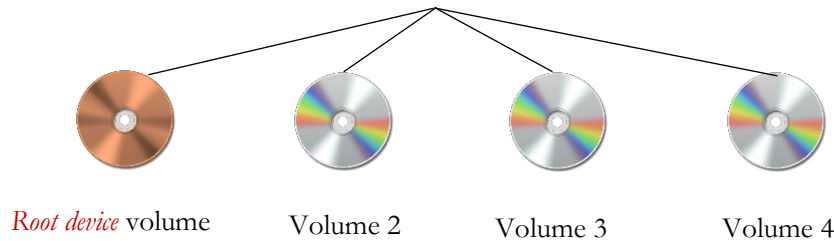
You can spot “root device volume” in the table below. Clearly, it can denote *either* an EBS volume or an instance store volume:

| Characteristic               | Amazon EBS-backed AMI   | Amazon instance store-backed AMI  |
|------------------------------|---|---|
| Boot time for an instance    | Usually less than 1 minute  | Usually less than 5 minutes   |
| Size limit for a root device | 64 TiB**  | 10 GiB  |
| Root device volume           | EBS volume  | Instance store volume   |
| Data persistence             | By default, the root volume is deleted when the instance terminates.* Data on any other EBS volumes persists after instance termination by default. | Data on any instance store volumes persists only during the life of the instance. |
| Modifications                | The instance type, kernel, RAM disk, and user data can be changed while the instance is stopped.  | Instance attributes are fixed for the life of an instance.                        |
| Charges                      | You're charged for instance usage, EBS volume usage, and storing your AMI as an EBS snapshot.   | You're charged for instance usage and storing your AMI in Amazon S3.              |
| AMI creation/bundling        | Uses a single command/call  | Requires installation and use of AMI tools  |
| Stopped state                | Can be in a stopped state. Even when the instance is stopped and not running, the root volume is persisted in Amazon EBS                            | Cannot be in a stopped state; instances are running or terminated                 |

So, the picture is as follows. An EC2 instance might be associated with a number of EBS volumes. One of these will play the role of the “root device”. We will therefore call it the “*root device* volume”. For short, you can just call it the “root volume”:



EC2 instance



(Needless to say, the above is just a silly sketch. EBS volumes are *not* CDs.)

## What is a “block device mapping”?

The first thing to note is that AWS count two things as block devices: EBS volumes and instance store volumes. Those two things are the only examples we are ever given of “block devices”. It would be really helpful if AWS told us the sufficient and necessary conditions for being a “block device”. They give us this definition of “block device”:

*A block device* is a storage device that moves data in sequences of bytes or bits (blocks).

These devices support random access and generally use buffered I/O.

Examples include hard disks, CD-ROM drives, and flash drives.

A block device can be physically attached to a computer or accessed remotely as if it were physically attached to the computer.

Anyhow, now that we know that a “block device” is basically going to be an EBS- or instance store volume, we are ready to look at what a block device *mapping* is:

*A block device mapping* defines the block devices (instance store volumes and EBS volumes) to attach to an instance.

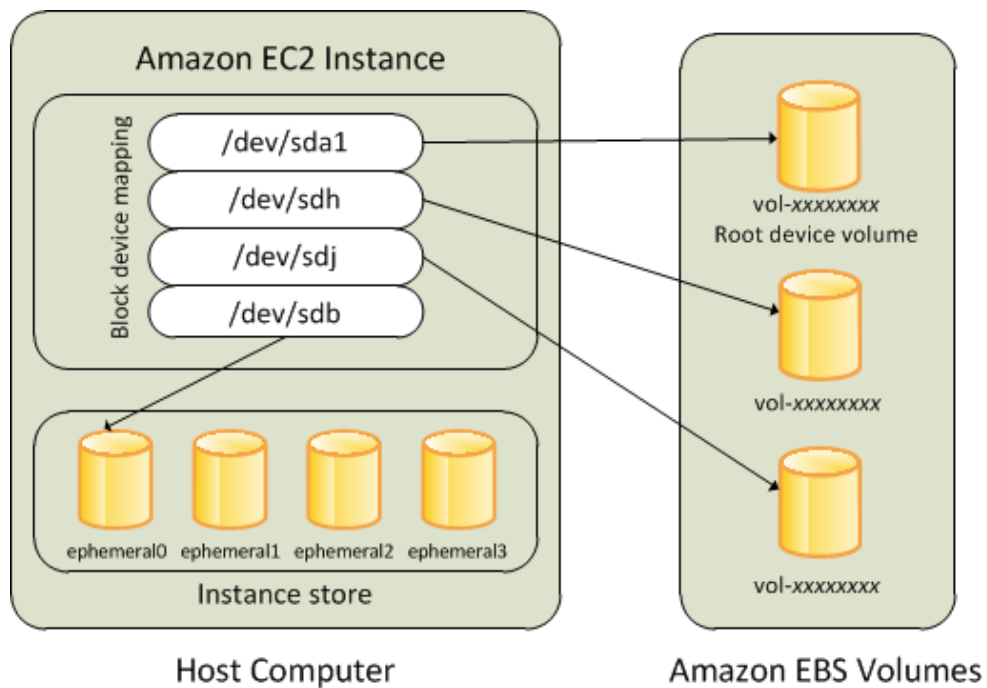
You can specify a block device mapping as part of creating an AMI so that the mapping is used by all instances launched from the AMI.

So, when we say “mapping”, we’re talking about the relationship (the mapping) between an EC2 instance and its volumes.



## Example mapping

AWS give us an example of a block device mapping.



Notice how one of the EBS volumes is the “root device volume”. Notice how the rounded box which has within it “Block device mappings” is not actually a list of the names of the volumes.

AWS tell us that they give these “block devices” certain names:

Device names like `/dev/sdh` and `xvdh` are used by Amazon EC2 to describe block devices.

The block device mapping is used by Amazon EC2 to specify the block devices to attach to an EC2 instance.

Get the following four letters tattooed on your arm:



We should note that we are depending on the OS to mount the block device. This must happen before it can be accessed. AWS write:

After a **block device** is attached to an instance, it must be mounted by the operating system before you can access the storage device.

When a block device is detached from an instance, it is unmounted by the operating system and you can no longer access the storage device.

As you can see, *detaching a block device from an instance* is not the same as *the OS unmounting the block device*. However, they occur together.

## The Important of the Instance Type

As strange as this may be, the instance type plays a role in determining how many volumes can be attached. AWS write:

The instance type determines which instance store volumes are formatted and mounted by default.

You can mount additional instance store volumes at launch, as long as you don't exceed the number of instance store volumes available for your instance type.

So, this is for the *instance store* volumes, not the EBS volumes. So, if you find yourself wondering “which instance store volumes should I attach?”, then this is answered for you by the instance type.

## The Re-mirroring storm of 2011







# Physalia



## 2. QUESTION

A video production company is planning to move some of its workloads to the AWS Cloud. The company will require around 5 TB of storage for video processing with the maximum possible I/O performance. They also require over 400 TB of extremely durable storage for storing video files and 800 TB of storage for long-term archival.

Which combinations of services should a Solutions Architect use to meet these requirements?

- ☒ Amazon EBS for maximum performance, Amazon EFS for durable data storage, and Amazon S3 Glacier for archival storage.
- ☐ Amazon EC2 instance store for maximum performance, Amazon EFS for durable data storage, and Amazon S3 for archival storage.
- ☐ Amazon EBS for maximum performance, Amazon S3 for durable data storage, and Amazon S3 Glacier for archival storage.
- ☐ Amazon EC2 instance store for maximum performance, Amazon S3 for durable data storage, and Amazon S3 Glacier for archival storage.

Incorrect

Explanation:

The best I/O performance can be achieved by using instance store volumes for the video processing. This is safe to use for use cases where the data can be recreated from the source files so this is a good use case.

For storing data durably Amazon S3 is a good fit as it provides 99.999999999% of durability. For archival the video files can then be moved to Amazon S3 Glacier which is a low cost storage option that is ideal for long-term archival.

**CORRECT:** "Amazon EC2 instance store for maximum performance, Amazon S3 for durable data storage, and Amazon S3 Glacier for archival storage" is the correct answer.

**INCORRECT:** "Amazon EBS for maximum performance, Amazon S3 for durable data storage, and Amazon S3 Glacier for archival storage" is incorrect. EBS is not going to provide as much I/O performance as an instance store volume so is not the best choice for this use case.

**INCORRECT:** "Amazon EC2 instance store for maximum performance, Amazon EFS for durable data storage, and Amazon S3 for archival storage" is

I thought that the question above, from Neal Davis, was very difficult. The reason I chose an option involving EBS is that I thought this carried performance advantages over the instance store, particularly in terms of I/O performance. I also made a decision to select an option involving EFS (Elastic File System) because the question explicitly mentioned “file”.

I must admit I do not fully understand the justification provided for using the instance store. We were supposed to choose the instance store, S3 and S3 Glacier.

# Glossary

## **Instance store**

Put description here.

## **xvda**

You will see these four letters a lot. This is modelled on “sda”, which was used in SCSI devices. The “x” stands for Xen. The “vd” stands for Virtual Disk. As for the “a”, I *believe* it is just the first letter of the alphabet, such that you might have “xvdb” as the next item. See [here](#).

## **Root device**

AWS never tell us what the necessary and sufficient conditions are, for something being a “root device”. However, it is clear that two things can play the role of a “root device”: an EBS volume and an instance store volume.

## **Root device volume**

This is a volume (either EBS- or instance store volume) which is playing the role of the root device. (See *Root device*).

## **Block device**

AWS talk as if there are just two examples of block devices: instance store volumes and EBS volumes. We’re told:

*A block device* is a storage device that moves data in sequences of bytes or bits (blocks).

These devices support random access and generally use buffered I/O.

Examples include hard disks, CD-ROM drives, and flash drives.

A block device can be physically attached to a computer or accessed remotely as if it were physically attached to the computer.

AWS also tell us that: “An instance store consists of one or more instance store volumes *exposed as block devices*.” (see Instance Store)

## **Block device mapping**

Gggggggg

## **NVMe**

NVM stands for “non-volatile memory”. The “e” stands for “express”.

## SATA

Stands for “Serial AT Attachment”. It was announced on 22<sup>nd</sup> Aug 2000, at a conference in San Jose, CA. It connects the [host bus adapter] to a [mass storage device]. For now, a [bus] is like jelly in a computer: it’s a communication system that transfers data between all the components within a computer (or between computers).

Note: *ALL* the components. “Bus” comes from “omnibus”. You know *Eastenders Omnibus* on television: they show *ALL* the episodes in one go...

# Bibliography

- IX. Official
- X. Unofficial
- XI. Critical
- XII. General

## V. Official

[AWS 1]



Available at: <https://aws.amazon.com/about-aws/whats-new/2020/02/ebs-multi-attach-available-provisioned-iops-ssd-volumes/>

<https://stackoverflow.com/questions/11347768/what-means-to-be-a-root-device-on-ec2>

<https://serverfault.com/questions/1042539/aws-device-names-dev-xvda>

## VI. Unofficial

### [Vogels 2020]

Vogels, Werner (2020). Reinventing virtualization with the AWS Nitro System. Available at:  
<https://www.allthingsdistributed.com/2020/09/reinventing-virtualization-with-nitro.html>

<https://web.archive.org/web/20140601070714/https://skeddly.desk.com/customer/portal/articles/1346918-ebs-backed-versus-instance-store>

### [Biersdorfer 1998]

<https://www.nytimes.com/1998/10/08/technology/q-a-emergency-preparedness.html>

[https://en.wikipedia.org/wiki/Device\\_mapper](https://en.wikipedia.org/wiki/Device_mapper)

### [Wikipedia X]

“Boot disk”. Available at: [https://en.wikipedia.org/wiki/Boot\\_disk](https://en.wikipedia.org/wiki/Boot_disk)

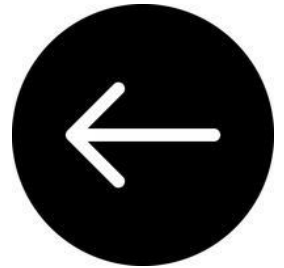
## VII. Critical

## VIII. General


### Image credits


**Tibetan monks:** <https://www.houstoniamag.com/arts-and-culture/2019/08/mandala-sand-painting-asia-society>

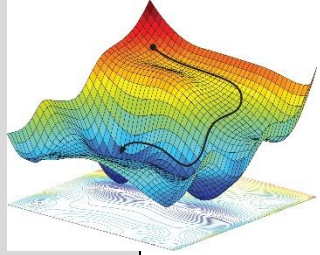

# CloudFront



[RETURN TO CONTENTS](#)

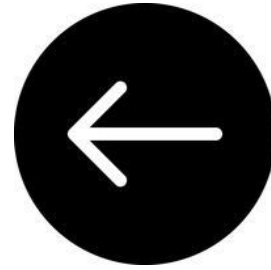
|  |   |  |
|--|---|--|
|  |   |  |
|  |   | <b><u>MODULE ONE</u></b>   |
|  |   | The features of a CDN  |
|  |   | The features of a cache  |
|  |   | The history of CDNs  |
|  |   | Cloudfront pricing   |
|  |   | Savings bundle   |
|  |   | Price classes  |
|  |   | <b><u>MODULE TWO</u></b> – bread and butter  |
|  |   |  |
|  | 1 | Working with distributions   |
|  |   | Overview of distributions  |
|  |   | Creating and updating distributions  |
|  |   |  |
|  |   |  |
|  |   |  |
|  | 3 | Three sorts of policy  |

|  |  |  |
|--|--|--|
|  |  | Cache policy   |
|  |  | Origin request policy  |
|  |  | Response headers policy  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | 4  | “Adding, removing, or replacing content that CloudFront distributes” |
|  |  |  |
|  |  |  |
|  | <p><b><u>MODULE THREE</u></b> -<br/>securing access</p>  |  |
|  | 1  | Using HTTPS with CloudFront  |
|  | 2  | Alternate Domain names and HTTPS                                     |
|  | 3  | <b>Signed URLs</b> and Signed cookies                                |
|  | 4  | Restricting access to S3 buckets                                     |
|  |  | OAI (Origin Access Identity)   |
|  |  |  |
|  |  |  |
|  |  |  |
|  | 5  | Restricting access to ALBs   |
|  | 6  |  |
|  | 7  | Using AWS WAF  |
|  | 8  | Restricting the geographic distribution                              |
|  | 9  | Field Level encryption   |
|  |  |  |

|   |  |   |
|---|--|---|
| <div> <div> <b><u>MODULE FOUR</u></b> </div> <div>  </div> </div> <div> <h1>Optimization</h1> </div> |  |   |
| 1   | Optimizing caching and availability            |   |
|   |  | Increasing the cache hit ratio                    |
|   |  | Using Origin Shield                               |
|   |  | CloudFront Origin Failover                        |
|   |  | Managing expiration                               |
|   |  | Managing content based on query string parameters |
|   |  |   |
|   |  |   |
|   |  |   |
| 2   | VOD (Video on demand) and live streaming video |   |
|   |  |   |
|   |  |   |
| <div> <div> <b><u>MODULE FIVE</u></b> </div> <div>  </div> </div>                                    |  |   |
| 1   | Customizing at the edge with functions         |   |
|   |  | CloudFront Functions                              |
|   |  | Lambda@edge                                       |
|   |  |   |
|   |  |   |
|   |  |   |
| 2   | Reports, metrics, and logs                     |   |
|   |  | Billing and usage reports                         |
|   |  | Monitoring with Amazon CloudWatch                 |
|   |  | Edge function logging                             |
|   |  | Lambda@Edge                                       |
| 4   | Security in CloudFront                         |   |
|   |  |   |
|   |  |   |



# CloudFront1



[BACK TO CLOUDFRONT](#)

## The History of CDNs

### The Akamai Network: A Platform for High-Performance Internet Applications

Erik Nygren<sup>†</sup>

Ramesh K. Sitaraman<sup>†‡</sup>

Jennifer Sun<sup>†</sup>

<sup>†</sup>Akamai Technologies, 8 Cambridge Center, Cambridge, MA 02142  
{nygren, ramesh}@akamai.com, jennifer\_sun@post.harvard.edu

<sup>‡</sup>Department of Computer Science, University of Massachusetts, Amherst, MA 01002  
ramesh@cs.umass.edu

#### ABSTRACT

Comprising more than 61,000 servers located across nearly 1,000 networks in 70 countries worldwide, the Akamai platform delivers hundreds of billions of Internet interactions daily, helping thousands of enterprises boost the performance and reliability of their Internet applications. In this paper, we give an overview of the components and capabilities of this large-scale distributed computing platform, and offer some insight into its architecture, design principles, operation, and management.

and architectural approaches used to achieve its results. We hope to offer insight into the richness of the platform and the breadth of technological research and innovation needed to make a system of this scale work.

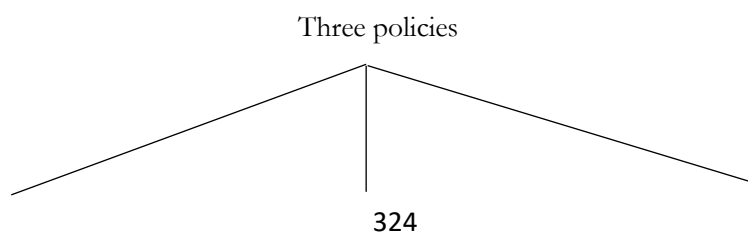
The paper is organized as follows. We first present the problem space and look at the motivations for creating such a platform. Next, an overview of the Akamai platform is followed by an examination of how it overcomes the Internet's inherent limitations for delivering web content, media streams, and dynamic applications. We present the case that a highly distributed network is the most effective architecture for these

## Multihoming, Content Delivery Networks, and the Market for Internet Connectivity

## Working with distributions

What is a distribution? This is the core concept within AWS CloudFront. AWS tell us that “You create a CloudFront distribution to tell CloudFront where you want content to be delivered from, and the details about how to track and manage content delivery.”

## Working with policies





Cache policy

Origin request  
policy

Response  
headers policy

The *cache policy* can be used to specify

1. The HTTP headers, cookies, query strings that CloudFront includes within the cache key.
2. The TTL for objects in the cache
3. Whether CloudFront requests and caches compressed objects

Including fewer values in the cache key increases the likelihood of a cache hit.

Why is this?

### 3. QUESTION

A company offers an online product brochure that is delivered from a static website running on Amazon S3. The company's customers are mainly in the United States, Canada, and Europe. The company is looking to cost-effectively reduce the latency for users in these regions.

What is the most cost-effective solution to these requirements?

- ☐ Create an Amazon CloudFront distribution that uses origins in U.S, Canada and Europe.

Correct

Explanation:

With Amazon CloudFront you can set the price class to determine where in the world the content will be cached. One of the price classes is "U.S, Canada and Europe" and this is where the company's users are located. Choosing this price class will result in lower costs and better performance for the company's users.

**CORRECT:** "Create an Amazon CloudFront distribution and set the price class to use only U.S, Canada and Europe." is the correct answer.

**INCORRECT:** "Create an Amazon CloudFront distribution and set the price class to use all Edge Locations for best performance" is incorrect. This will be more expensive as it will cache content in Edge Locations all over the world.

**INCORRECT:** "Create an Amazon CloudFront distribution that uses origins in U.S, Canada and Europe" is incorrect. The origin can be in one place, there's no need to add origins in different Regions. The price class should be used to limit the caching of the content to reduce cost.

**INCORRECT:** "Create an Amazon CloudFront distribution and use Lambda@Edge to run the website's data processing closer to the users" is incorrect. Lambda@Edge will not assist in this situation as there is no data processing required, the content from the static website must simply be cached at an edge location.

# A Walk through Content Delivery Networks

[Novella Bartolini](#), [Emiliano Casalicchio](#) & [Salvatore Tucci](#)

Conference paper

**1101** Accesses | **11** [Citations](#)

Part of the [Lecture Notes in Computer Science](#) book series (LNCS, volume 2965)

# An Overview of Cloud Based Content Delivery Networks: Research Dimensions and State-of-the-Art

[Meisong Wang](#), [Prem Prakash Jayaraman](#), [Rajiv Ranjan](#), [Karan Mitra](#), [Miranda Zhang](#), [Eddie Li](#), [Samee Khan](#), [Mukkaddim Pathan](#) & [Dimitrios Georgeakopoulos](#) 

Chapter | [First Online: 01 January 2015](#)






**1957** Accesses | **31** Citations

Part of the [Lecture Notes in Computer Science](#) book series (TLDKS,volume 9070)

## A Comprehensive Survey on Fog Computing: State-of-the-Art and Research Challenges

[Carla Mouradian](#), [Diala Naboulsi](#), [Sami Yangui](#), [R. Giltho](#), [M. Morrow](#), [P. Polakos](#) · Computer Science · IEEE Communications Surveys & Tutorials · 2018

**TLDR** A comprehensive survey on fog computing is presented, critically reviews the state of the art in the light of a concise set of evaluation criteria and covers both the architectures and the algorithms that make fog systems. [Expand](#)

 **614**   View 1 excerpt, cites background  Save  Alert

## Efficient virtual machine allocation and optimization of response time in cloud delivery network

[D. Shah](#) · Computer Science · 2019

**TLDR** This research project proposes a novel framework for Cloud Delivery Networks (CDNs), which provides the system architecture and optimization for the



### 23. QUESTION

A company runs a web application that serves weather updates. The application runs on a fleet of Amazon EC2 instances in a Multi-AZ Auto scaling group behind an Application Load Balancer (ALB). The instances store data in an Amazon Aurora database. A solutions architect needs to make the application more resilient to sporadic increases in request rates.

Which architecture should the solutions architect implement? (Select TWO.)

☐ Add an AWS Transit Gateway to the Availability Zones

☒ Add Amazon Aurora Replicas

☐ Add an AWS Global Accelerator endpoint

☐ Add and AWS WAF in front of the ALB

Explanation:

The architecture is already highly resilient but the may be subject to performance degradation if there are sudden increases in request rates. To resolve this situation Amazon Aurora Read Replicas can be used to serve read traffic which offloads requests from the main database. On the frontend an Amazon CloudFront distribution can be placed in front of the ALB and this will cache content for better performance and also offloads requests from the backend.

**CORRECT:** "Add Amazon Aurora Replicas" is the correct answer.

**CORRECT:** "Add an Amazon CloudFront distribution in front of the ALB" is the correct answer.

**INCORRECT:** "Add and AWS WAF in front of the ALB" is incorrect. A web application firewall protects applications from malicious attacks. It does not improve performance.

**INCORRECT:** "Add an AWS Transit Gateway to the Availability Zones" is incorrect as this is used to connect on-premises networks to VPCs.

**INCORRECT:** "Add an AWS Global Accelerator endpoint" is incorrect as this service is used for directing users to different instances of the application in different regions based on latency.

# TPN

21. ***Phenomenon1*** – the tendency of X to Y.
22. ***Phen2*** – the tendency of X to Y.
23. ***Phen3*** – the tendency of X to Y.
24. ***Phen4*** – the tendency of X to Y.
25. ***Phen5*** – the tendency of X to Y.
26. ***Phen6*** – the tendency of X to Y.
27. ***Phen7*** – the tendency of X to Y.
28. ***Phen8*** – the tendency of X to Y.

- 29. ***Phen9*** – the tendency of X to Y.
- 30. ***Phen10*** – the tendency of X to Y.

# Glossary

## **Static content**

Description of what term means here.

## **Perfect forward secrecy**

Description of what term means here.

## **SSL**

Description of what term means here.

## **WebSockets**

Description of what term means here.

## **Viewer**

Term used by CloudFront to denote web browsers or other clients.

## **Signed URL**

bbb

## **HTTP**

Stands for Hypertext Transfer Protocol. HTTP is a stateless application-level protocol. It operates at Layer 7 of the OSI Model. It requires a reliable network transport connection in order to transmit data between client and server.

HTTP [RFC 7230].

In HTTP, TCP/IP connections are used. It uses well-known ports; HTTP uses port 80.

Tim Berners-Lee and his team at CERN are credited with creating the original HTTP, in 1989. This was to help with his project, the WorldWideWeb.

## HTTP header

When an **HTTP request** is made, and an **HTTP response** given by a web server, the request or response is usually accompanied by additional information, contained in a so-called "HTTP header."

The additional information in the **HTTP headers** help to ensure that the data pulled from the web server can be properly displayed in the browser window.

All the headers are case-insensitive, headers fields are separated by colon, key-value pairs in clear-text string format. The end of the header section denoted by an empty field header.

Here are some examples of headers:

| Header     | Description  |
|------------|--|
| Connection | It is a general type header that allows the sender or client to specify options that are desired for that particular connection. |
| Keep-Alive | It is a general-type header used to inform that how long a persistent connection should stay open.                               |

| Header          | Description   |
|-----------------|---|
| Age             | It is a response header. It defines the times in seconds of the object that have been in the proxy cache.           |
| Cache-Control   | It is a general type header used to specify directives for caching mechanisms.                                      |
| Clear-Site-Data | It is a response-type header. This header is used in deleting the browsing data which is in the requesting website. |
| Expires         | It is a response-type header, it is used to define date/time after after that time that will be vanished.           |
| Pragma          | It is general-type header, but response behavior is not specified and thus implementation-specific.                 |
| Warnings        | It is a general type header that is used to inform possible problems to the client.                                 |

## HTTP Method

The "method" indicates what kind of request this is.

The most common methods are GET, POST, and HEAD.

## Path

Description of what term means here.

## Query string

A query string is a part of a Uniform Resource Locator (URL) that assigns values to specific parameters.

Typical URL containing a query string is as follows:

```
https://example.com/over/there?name=ferret
```

The query string, in the above, is “name=ferret”. The question mark is used as a separator, and is not part of the query string. [RFC 3986]

Web frameworks may provide methods for parsing multiple parameters in the query string, separated by some delimiter. In the example URL below, multiple query parameters are separated by the ampersand, “&”:

```
https://example.com/path/to/page?name=ferret&color=
purple
```

## Cache policy

### Origin request policy

With a CloudFront *origin request policy*, you can specify the HTTP headers, cookies, and query strings that CloudFront includes in *origin requests*. These are the requests that CloudFront sends to the origin when there’s a cache miss.

### Cache key

AWS write: “The cache key is the unique identifier for every object in the cache, and it **determines** whether a viewer request results in a *cache hit*.”

A cache hit occurs when a viewer request generates the same cache key as a prior request, and the object for that cache key is in the edge location’s cache and valid.”

It is a KEY, in the sense that it can take multiple VALUES (you may have seen KEY-VALUE pairs elsewhere). AWS talk about including values “in” the cache key:

Including fewer values in the cache key increases the likelihood of a cache hit.

## **Cache hit**

Description of what term means here.

## **Signed cookies**

Description of what term means here.

## **Origin access identity (OAI)**

Description of what term means here.

## **Distribution**

Term used in CloudFront to denote a set-up, or configuration, for distributing content. To be using CloudFront, you must *have* a distribution.

There are two types of distribution: web distribution and **RTMP** (stands for).

## **Zone apex**

Description of what term means here.

## **CNAME record**

Description of what term means here.

## **Signed URL**

Description of what term means here.

## **TTL**

Stands for Time to Live. In CloudFront, it is recorded in seconds. The default TTL value is 24 hours.

## **Field-level encryption**

Description of what term means here.

## **Custom Origin**

Description of what term means here.



## **Match viewer**

Description of what term means here.

## **PCI DSS**

Description of what term means here.

## **HIPAA**

Description of what term means here.

## **AWS WAF**

Description of what term means here.

## **Smooth Streaming**

Description of what term means here.

## **Object invalidation**

Description of what term means here.

## **RTMP**

One of two types of Distribution (see Distribution). Use RTMP to distribute streaming media files using Adobe Flash Media Server's RTMP protocol.

## **CDN**

Description of what term means here.

## **Origin**

Term used to denote the source of the content, which is to be distributed. Examples of things which can play the role of the origin are: (1) S3 bucket (2) an EC2 instance (3) an Elastic Load Balancer (4) Route 53 (5) something external to AWS.

## **Origin request**

These are the requests that CloudFront sends to the origin when there's a cache miss. (see *Origin*)

### **Cache**

Description of what term means here.

### **Regional Edge Cache (REC)**

Description of what term means here.

### **Edge locations (EL)**

Description of what term means here.

### **Dynamic content**

Description of what term means here.

### **Streaming content**

Description of what term means here.

### **Interactive content**

Description of what term means here.

### **Term3**

Description of what term means here.

# Bibliography

- XIII. Official
- XIV. Unofficial
- XV. Critical

## XVI. General

### IV. Official

#### [Surname1]

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

#### [Surname1]

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

### V. Unofficial

#### [Surname1]

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

#### [Surname1]

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

<https://code.tutsplus.com/tutorials/http-headers-for-dummies--net-8039>

### VI. Critical

**[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

**[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

Chiu, Chi-Huang, Hsien-Tang Lin, and Shyan-Ming Yuan.  
"CloudEdge: a content delivery system for storage service in cloud  
environment." *International Journal of Ad Hoc and Ubiquitous  
Computing* 6, no. 4 (2010): 252-262.

## VII. General

**[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

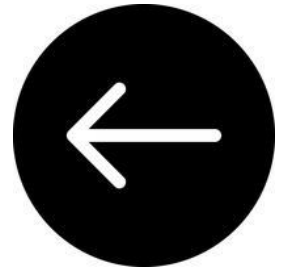
**[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

Stefan Podlipnig and Laszlo Böszörményi. 2003. A survey of Web  
cache replacement strategies. *ACM Comput. Surv.* 35, 4  
(December 2003), 374–398.  
<https://doi.org/10.1145/954339.954341>



# CloudFront2 – bread and butter



[BACK TO CLOUDFRONT](#)



## NOTE

### CloudFront or CloudFormation?

Some people get confused between the terms “CloudFormation” and “CloudFront”. Cloudfor-**mation** is about auto-**mation**.

Cloud*Front* bears the *brunt* of those incoming requests (using caching and so on). CloudFront’s edge locations get out there and do the dirty work of bringing content close to customers. CloudFormation –to its elation—is well away from the front line.





CUSTOM ORIGIN



# What on earth do AWS mean by “custom origin”?

They use this expression a lot. Below, they explain what they mean. If it's just an HTTP server, why not just call it an HTTP server?

## Using Amazon EC2 (or another custom origin)

A custom origin is an HTTP server, for example, a web server. The HTTP server can be an Amazon EC2 instance or an HTTP server that you host somewhere else. An Amazon S3 origin configured as a website endpoint is also considered a custom origin.

When you use your own HTTP server as a custom origin, you specify the DNS name of the server, along with the HTTP and HTTPS ports and the protocol that you want CloudFront to use when fetching objects from your origin.

Note how your “custom origin” can even be an EC2 instance! Instead of chucking out random examples, of “custom origins”, perhaps the documentation authors could explain what *makes* something a custom origin. I want to be told what all these things have in common. Sometimes you cannot help becoming infuriated at the documentation writers.

Perhaps they do indeed do this in the second paragraph. Perhaps the essential property (of “custom origins”) is that you are specifying the DNS name of the server. If you do this, it's a “custom origin” and if you don't, it's not.

I know what “custom” means. If something is “custom”, then it has been made to order. It is tailored for a specific “use case”. So, it's obviously something to do with that. (But note how S3 buckets do not seem to count as “custom origins”).

AWS should acknowledge why they chose this term to be—which denotes HTTP servers—“custom origin”. To me (stupid reader who cannot understand AWS's incredibly simple products) the connection between [HTTP] and [custom] is completely arbitrary.



Most CloudFront features are supported when you use a custom origin with the exception of private content. Although you can use a signed URL to distribute content from a custom origin, for CloudFront to access the custom origin, the origin must remain publicly accessible. For more information, see [Serving private content with signed URLs and signed cookies](#).

Follow these guidelines for using Amazon EC2 instances and other custom origins with CloudFront.

- Host and serve the same content on all servers that are serving content for the same CloudFront origin. For more information, see [Origin settings](#) in the [Values that you specify when you create or update a distribution](#) topic.
- Log the X-Amz-Cf-Id header entries on all servers in case you need AWS Support or CloudFront to use this value for debugging.
- Restrict requests to the HTTP and HTTPS ports that your custom origin listens on.
- Synchronize the clocks of all servers in your implementation. Note that CloudFront uses Coordinated Universal Time (UTC) for signed URLs and signed cookies, for logs, and reports. In addition, if you monitor CloudFront activity using CloudWatch metrics, note that CloudWatch also uses UTC.
- Use redundant servers to handle failures.
- For information about using a custom origin to serve private content, see [Restricting access to files on custom origins](#).
- For information about request and response behavior and about supported HTTP status codes, see [Request and response behavior](#).

If you use Amazon EC2 for a custom origin, we recommend that you do the following:

- Use an Amazon Machine Image that automatically installs the software for a web server. For more information, see the [Amazon EC2 documentation](#).
- Use an Elastic Load Balancing load balancer to handle traffic across multiple Amazon EC2 instances and to isolate your application from changes to Amazon EC2 instances. For example, if you use a load balancer, you can add and delete Amazon EC2 instances without changing your application. For more information, see the [Elastic Load Balancing documentation](#).
- When you create your CloudFront distribution, specify the URL of the load balancer for the domain name of your origin server. For more information, see [Creating a distribution](#).

## Using CloudFront origin groups

You can specify an origin group for your CloudFront origin if, for example, you want to configure origin failover for scenarios when you need high availability. Use origin failover to designate a primary origin for CloudFront plus a second origin that CloudFront automatically switches to when the primary origin returns specific HTTP status code failure responses.



#### 16. QUESTION

A company runs a dynamic website that is hosted on an on-premises server in the United States. The company is expanding to Europe and is investigating how they can optimize the performance of the website for European users. The website's backed must remain in the United States. The company requires a solution that can be implemented within a few days.

What should a Solutions Architect recommend?

- ☐ Migrate the website to Amazon S3. Use cross-Region replication between Regions and a latency-based Route 53 policy.
- ☐ Launch an Amazon EC2 instance in an AWS Region in the United States and migrate the website to it.
- ☒ Use Amazon CloudFront with a custom origin pointing to the on-premises servers.
- ☐ Use Amazon CloudFront with Lambda@Edge to direct traffic to an on-premises origin.



Correct

Explanation:

A custom origin can point to an on-premises server and CloudFront is able to cache content for dynamic websites. CloudFront can provide performance optimizations for custom origins even if they are running on on-premises servers. These include persistent TCP connections to the origin, SSL enhancements such as Session tickets and OCSP stapling.

Additionally, connections are routed from the nearest Edge Location to the user across the AWS global network. If the on-premises server is connected via a Direct Connect (DX) link this can further improve performance.

**CORRECT:** "Use Amazon CloudFront with a custom origin pointing to the on-premises servers" is the correct answer.

**INCORRECT:** "Use Amazon CloudFront with Lambda@Edge to direct traffic to an on-premises origin" is incorrect. Lambda@Edge is not used to direct traffic to on-premises origins.

**INCORRECT:** "Launch an Amazon EC2 instance in an AWS Region in the United States and migrate the website to it" is incorrect. This would not necessarily improve performance for European users.

**INCORRECT:** "Migrate the website to Amazon S3. Use cross-Region replication between Regions and a latency-based Route 53 policy" is incorrect. You cannot host dynamic websites on Amazon S3 (static only).



#### 12. QUESTION

A company hosts an application on Amazon EC2 instances behind Application Load Balancers in several AWS Regions. Distribution rights for the content require that users in different geographies must be served content from specific regions.

Which configuration meets these requirements?

- ☒ Create Amazon Route 53 records with a geolocation routing policy.
- ☐ Create Amazon Route 53 records with a geoproximity routing policy.
- ☐ Configure Application Load Balancers with multi-Region routing.
- ☐ Configure Amazon CloudFront with multiple origins and AWS WAF.



#### Explanation:

To protect the distribution rights of the content and ensure that users are directed to the appropriate AWS Region based on the location of the user, the geolocation routing policy can be used with Amazon Route 53.

Geolocation routing lets you choose the resources that serve your traffic based on the geographic location of your users, meaning the location that DNS queries originate from.

When you use geolocation routing, you can localize your content and present some or all of your website in the language of your users. You can also use geolocation routing to restrict distribution of content to only the locations in which you have distribution rights.

**CORRECT:** "Create Amazon Route 53 records with a geolocation routing policy" is the correct answer.

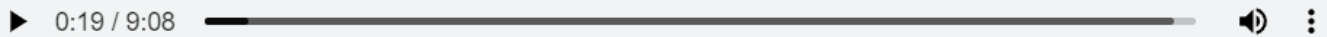
**INCORRECT:** "Create Amazon Route 53 records with a geoproximity routing policy" is incorrect. Use this routing policy when you want to route traffic based on the location of your resources and, optionally, shift traffic from resources in one location to resources in another.

**INCORRECT:** "Configure Amazon CloudFront with multiple origins and AWS WAF" is incorrect. AWS WAF protects against web exploits but will not assist with directing users to different content (from different origins).

**INCORRECT:** "Configure Application Load Balancers with multi-Region routing" is incorrect. There is no such thing as multi-Region routing for ALBs.

## Amazon CloudFront – Support for Dynamic Content

by Jeff Barr | on 14 MAY 2012 | in [Amazon EC2](#) | [Permalink](#) | [Share](#)



Voiced by [Amazon Polly](#)

Post updated on May 11, 2020.

### Introduction

[Amazon CloudFront](#)'s network of edge locations ([currently 30](#), with more in the works) gives you the ability to distribute static and streaming content to your users at high speed with low latency.

Today we are introducing a set of features that, taken together, allow you to use CloudFront to serve dynamic, personalized content more quickly.

### What is Dynamic Personalized Content?

5. Specify [cache behavior](#) settings for the distribution, as shown in the following screenshot. You can configure each URL path pattern with a set of associated cache behaviors. For dynamic web applications, set the **Minimum TTL** to **0** so that CloudFront will make a GET request with an **If-Modified-Since** header back to the origin. When CloudFront proxies traffic to the origin from edge locations and back, multiple concurrent requests for the same object are collapsed into a single request. The request is sent over a persistent connection from the edge location to the region over networks monitored by AWS. The use of a [large initial TCP window size](#) in CloudFront maximizes the available bandwidth, and [TCP Fast Open](#) (TFO) reduces latency.

Holly Willey, [writing](#) in March 2017, about features within CloudFront



Internet Engineering Task Force (IETF)  
Request for Comments: 7413  
Category: Experimental  
ISSN: 2070-1721

Y. Cheng  
J. Chu  
S. Radhakrishnan  
A. Jain  
Google  
December 2014

## TCP Fast Open

### Abstract

This document describes an experimental TCP mechanism called TCP Fast Open (TFO). TFO allows data to be carried in the SYN and SYN-ACK packets and consumed by the receiving end during the initial connection handshake, and saves up to one full round-trip time (RTT) compared to the standard TCP, which requires a three-way handshake (3WHS) to complete before data can be exchanged. However, TFO deviates from the standard TCP semantics, since the data in the SYN could be replayed to an application in some rare circumstances. Applications should not use TFO unless they can tolerate this issue, as detailed in the Applicability section.

Status of This Memo

# TPN

73. ***Phenomenon1*** – the tendency of X to Y.
74. ***Phen2*** – the tendency of X to Y.
75. ***Phen3*** – the tendency of X to Y.
76. ***Phen4*** – the tendency of X to Y.
77. ***Phen5*** – the tendency of X to Y.
78. ***Phen6*** – the tendency of X to Y.

- 79. ***Phen7*** – the tendency of X to Y.
- 80. ***Phen8*** – the tendency of X to Y.
- 81. ***Phen9*** – the tendency of X to Y.
- 82. ***Phen10*** – the tendency of X to Y.

# Glossary

## Smooth Streaming

Description of what term means here.

# Bibliography

- V. Official
- VI. Unofficial

- VII. Critical
- VIII. General

## VI. Official

### [Surname1]

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

### [Surname1]

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

### [AWS 2004]

“Introducing the Amazon Simple Queue Service”. AWS.  
3<sup>rd</sup> Nov 2004. Available at:  
<https://aws.amazon.com/about-aws/whats-new/2004/11/03/introducing-the-amazon-simple-queue-service/>

### [Barr 1]

Barr, Jeff (2006). “Amazon Simple Queue Service Released”. *AWS News Blog*. 13<sup>th</sup> July 2006. Available at:  
[https://aws.amazon.com/blogs/aws/amazon\\_simple\\_q/](https://aws.amazon.com/blogs/aws/amazon_simple_q/)

## VII. Unofficial

### [Levitt 2005]

Levitt, Jason (2005). Fun with Amazon’s simple queue service. Available at:  
<https://www.xml.com/pub/a/2005/01/05/sqs.html>

### [Barr 2]



Barr, Jeff (year). My first 12 years at Amazon.com. Jeff-barr.com. Available at: <http://jeff-barr.com/2014/08/19/my-first-12-years-at-amazon-dot-com/>

## VIII. Critical

## IX. General

Elastic message queues. Ahmed El Rheddane and Noel De Palma

MSMQ is dead. David Boike.

Chapter 18 – Distributed computing – models and methods. Leslie Lamport and Nancy Lynch.

### **[Wikipedia 1]**

Message queue. Wikipedia. Available at: [https://en.wikipedia.org/wiki/Message\\_queue](https://en.wikipedia.org/wiki/Message_queue)

### **[Bajantri 1986]**

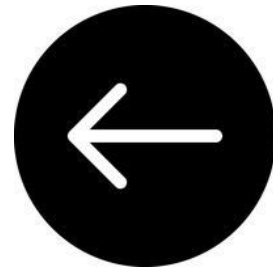
Bajantri, M. and David B Skillicorn (1986). A fast multiprocessor message passing implementation. Information Processing Letters, 24 (6): 381-389. Available at:

<https://www.sciencedirect.com/science/article/abs/pii/0020019087901153>

**[Christopher 1988]**

Christopher, Thomas (1988). Message driven computing and its relationship to actors. OOPSLA/ECOOP '88: Proceedings of the 1988 ACM SIGPLAN workshop on Object-based concurrent programming.  
<https://doi.org/10.1145/67386.67405>.

# CloudFront3 – securing access



**[BACK TO CLOUDFRONT](#)**

The agenda:

1. [Using HTTPS with CloudFront](#)
2. [Alternate domain names and HTTPS](#)
3. [Signed URLs and signed cookies](#)
4. [Restricting access to S3 buckets](#)
5. Restricting access to ALBs
6. Using AWS WAF
7. [Restricting the geographic distribution](#)
8. Field level encryption

Part 1 – Using HTTPS with CloudFront

## Part 2 – Alternate domain names and HTTPS

### Choosing how CloudFront serves HTTPS requests

[PDF](#) | [RSS](#)

If you want your viewers to use HTTPS and to use alternate domain names for your files, choose one of the following options for how CloudFront serves HTTPS requests:

- Use [Server Name Indication \(SNI\)](#) [🔗](#) – Recommended
- Use a dedicated IP address in each edge location

This section explains how each option works.

## Part 3 – Signed URLs and Signed cookies

### 31. QUESTION

A company hosts video files for a website in an Amazon S3 bucket that is configured as an origin for an Amazon CloudFront distribution. The company was recently notified that the videos were being accessed from unauthorized countries.

Which actions should a security engineer take to limit the distribution of the video files? (Select TWO.)

- ☒ Update the distribution settings in CloudFront and configure restrictions based on the geography of the request.
- ☒ Configure the Restrict Viewer Access option in CloudFront and specify a deny list of unauthorized countries.
- ☐ Configure a query string whitelist in CloudFront and specify a list of countries that should be denied access using query string parameters.
- ☐ Update the S3 bucket policy with condition statements that deny access based on the source IP addresses of users.
- ☐ Create an origin access identity (OAI) for the CloudFront distribution and update the S3 bucket policy to restrict access to the OAI.

This question (which is quite cruel in my opinion) is from Neal Davis's Digital Cloud Training course, preparing students for the AWS Security Speciality exam

**Incorrect**

**Explanation:**

You can use *geo restriction*, also known as *geo blocking*, to prevent users in specific geographic locations from accessing content that you're distributing through a CloudFront distribution. The CloudFront geo restriction feature can be used to restrict access to all the files that are associated with a distribution and to restrict access at the country level.

To ensure users in the restricted countries cannot bypass CloudFront and go straight to the Amazon S3 bucket, an origin access identity can be configured in the CloudFront distribution. This identity is then granted access in the S3 bucket policy, and all other connections are denied.

**CORRECT:** "Create an origin access identity (OAI) for the CloudFront distribution and update the S3 bucket policy to restrict access to the OAI" is a correct answer (as explained above.)

**CORRECT:** "Update the distribution settings in CloudFront and configure restrictions based on the geography of the request" is also a correct answer (as explained above.)

**INCORRECT:** "Configure the Restrict Viewer Access option in CloudFront and specify a deny list of unauthorized countries" is incorrect.

This feature is used to configure signed URLs and signed cookies.

**INCORRECT:** "Update the S3 bucket policy with condition statements that deny access based on the source IP addresses of users" is incorrect.

This would be hard to manage as the addresses may change and be hard to identify. The users should always go through the CloudFront distribution as well.

**INCORRECT:** "Configure a query string whitelist in CloudFront and specify a list of countries that should be denied access using query string parameters" is incorrect.

This is not the correct usage for the query string whitelist feature which is used for determining the query string parameters that you want CloudFront to use as a basis for caching.

**References:**

“Configure the **restrict viewer access** option...  
this feature is used to  
configure signed URLs and  
cookies”

Neal Davis

## Part 4 – Restricting Access to S3 buckets

### Amazon CloudFront launches Origin Access Control (OAC)

Posted On: Aug 25, 2022

Amazon CloudFront now offers Origin Access Control, a new feature that enables CloudFront customers to easily secure their S3 origins by permitting only designated CloudFront distributions to access their S3 buckets. Customers can now enable [AWS Signature Version 4](#) (SigV4) on CloudFront requests to S3 buckets with the ability to set when and if CloudFront should sign requests. Additionally, customers can now use [SSE-KMS](#) when performing uploads and downloads through CloudFront.

Until now, customers were limited to using Origin Access Identity to restrict access to their S3 origins to CloudFront. Origin Access Control improves upon Origin Access Identity by strengthening security and deepening feature integrations. Origin Access Control provides stronger security posture with short term credentials, and more frequent credential rotations as compared to Origin Access Identity. With Origin Access Control, customers can create granular policy configurations through resource-based policies, which provides better protection against [confused deputy attacks](#). Customers can use Origin Access Control to fetch and put data into S3 origins in [regions that require SigV4](#). Also, Origin Access Control allows customer to use SSE-KMS with their S3 origins, which was not possible using Origin Access Identity.

CloudFront supports both the new Origin Access Control and legacy Origin Access Identity. If you have a distribution configured to use Origin Access Identity, you can easily migrate the distribution to Origin Access Control with few simple clicks. Any distributions using Origin Access Identity will continue to work and you can continue to use Origin Access Identity for new distributions. Refer to [CloudFront origin access migration documentation](#) for upcoming region restrictions.

CloudFront Origin Access Control is now available worldwide except for AWS China regions. You can start using Origin Access Control through the CloudFront console, APIs, SDK, or CLI. There is no additional fee to use Origin Access Control. To learn about how to configure Origin Access Control, refer to the [CloudFront origin access control documentation](#). To get started with CloudFront, visit the [CloudFront product page](#).

### 13. QUESTION

A developer is deploying a website hosted in an Amazon S3 bucket. An Amazon CloudFront distribution will be deployed in front of the S3 bucket to cache the content. The developer requires that users may only access the website using the CloudFront distribution and should not be able to access the website directly by using the S3 URL.

Which configurations should a security engineer make to support these requirements? (Select TWO.)

- ☒ Configure the S3 bucket permissions so that only the origin access identity can access the bucket contents.
- ☒ Create an origin access identity (OAI) and associate it with the CloudFront distribution.
- ☐ Implement a condition in the S3 bucket policy that limits access to "Principal": "cloudfront.amazonaws.com".
- ☐ Configure a gateway endpoint for the S3 bucket and attach the CloudFront distribution to the VPC.
- ☐ Configure CloudFront to add a custom HTTP header to requests for the S3 bucket and configure the bucket to only accept requests with the custom header.

Screenshot taken in 2022. This is a mock exam question from Neal Davis, on his course for the AWS Security Specialty.



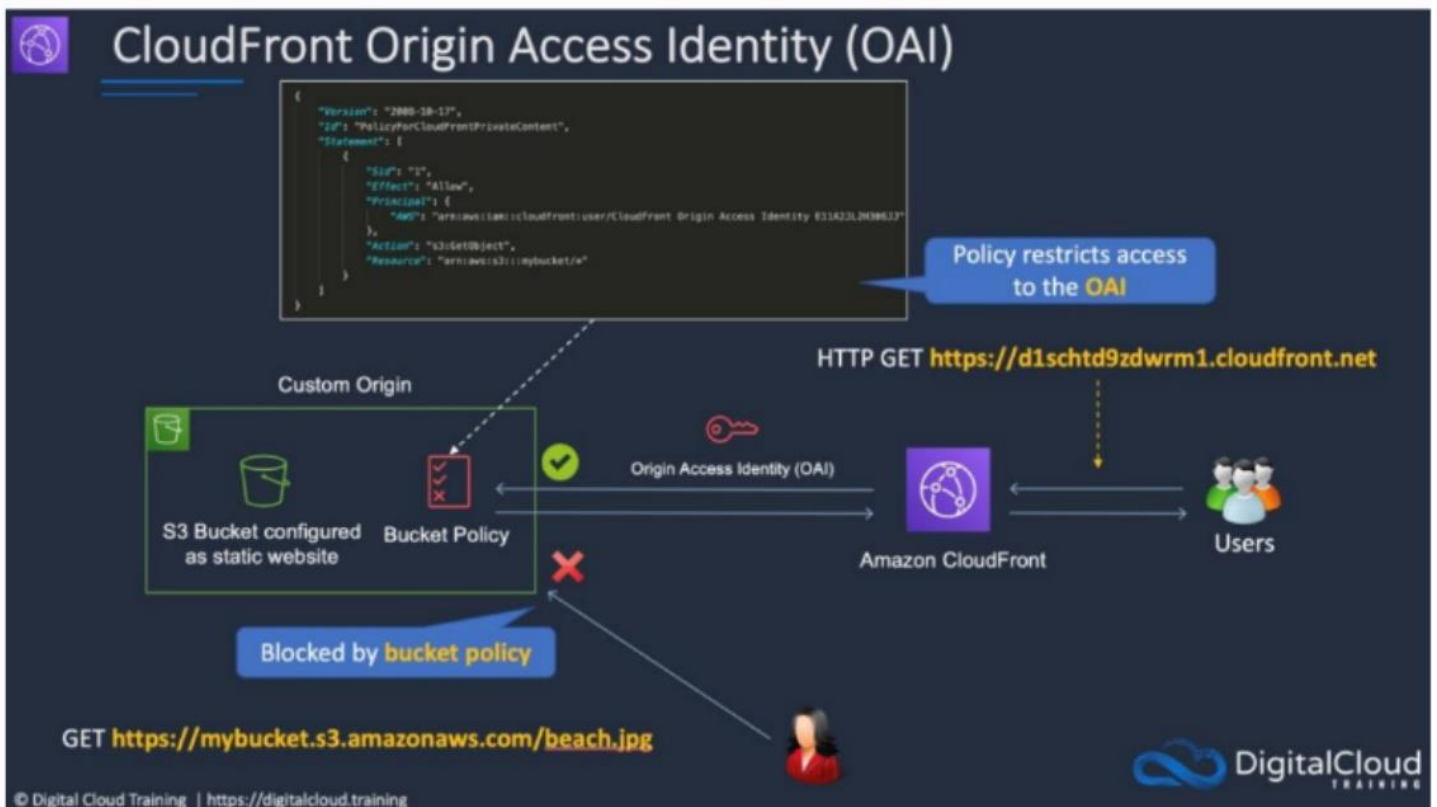
Correct

#### Explanation:

To restrict access to content that you serve from Amazon S3 buckets you can create a special CloudFront user called an origin access identity (OAI) and associate it with your distribution.

You must then configure your S3 bucket permissions so that CloudFront can use the OAI to access the files in your bucket and serve them to your users. This also ensures that users can't use a direct URL to the S3 bucket to access its contents.

The diagram below depicts this scenario.



**CORRECT:** "Create an origin access identity (OAI) and associate it with the CloudFront distribution" is a correct answer (as explained above.)

**CORRECT:** "Configure the S3 bucket permissions so that only the origin access identity can access the bucket contents" is also a correct answer (as explained above.)

**INCORRECT:** "Implement a condition in the S3 bucket policy that limits access to "Principal": "cloudfront.amazonaws.com" is incorrect.

An OAI should be used and referenced in the policy condition rather than the principal for the AWS service.

**INCORRECT:** "Configure a gateway endpoint for the S3 bucket and attach the CloudFront distribution to the VPC" is incorrect.

You cannot attach a CloudFront distribution to a VPC, so this does not work.

**INCORRECT:** "Configure CloudFront to add a custom HTTP header to requests for the S3 bucket and configure the bucket to only accept requests with the custom header" is incorrect.

This technique should be used with elastic load balancers and does not work with S3.

Even though it is now legacy, the question below helps me to understand origin access identity (OAI). The expression had never really been explained to me. You're saying that if you want to access a particular resource (such as an S3 bucket), then you have to have a certain identity—like the identities within IAM! And what we're

granting this identity access *to* is the origin. Furthermore, though, the question below emphasises the fact that you are—by using OAIs—ensuring that the CloudFront distribution is used.

QUESTION 31 OF 33

31. QUESTION

A company hosts video files for a website in an Amazon S3 bucket that is configured as an origin for an Amazon CloudFront distribution. The company was recently notified that the videos were being accessed from unauthorized countries.

Which actions should a security engineer take to limit the distribution of the video files? (Select TWO.)

- ☒ Update the distribution settings in CloudFront and configure restrictions based on the geography of the request.
- ☒ Configure the Restrict Viewer Access option in CloudFront and specify a deny list of unauthorized countries.
- ☐ Configure a query string whitelist in CloudFront and specify a list of countries that should be denied access using query string parameters.
- ☐ Update the S3 bucket policy with condition statements that deny access based on the source IP addresses of users.
- ☐ Create an origin access identity (OAI) for the CloudFront distribution and update the S3 bucket policy to restrict access to the OAI.

**Incorrect**

**Explanation:**

You can use *geo restriction*, also known as *geo blocking*, to prevent users in specific geographic locations from accessing content that you're distributing through a CloudFront distribution. The CloudFront geo restriction feature can be used to restrict access to all the files that are associated with a distribution and to restrict access at the country level.

To ensure users in the restricted countries cannot bypass CloudFront and go straight to the Amazon S3 bucket, an origin access identity can be configured in the CloudFront distribution. This identity is then granted access in the S3 bucket policy, and all other connections are denied.

**CORRECT:** "Create an origin access identity (OAI) for the CloudFront distribution and update the S3 bucket policy to restrict access to the OAI" is a correct answer (as explained above.)

**CORRECT:** "Update the distribution settings in CloudFront and configure restrictions based on the geography of the request" is also a correct answer (as explained above.)

**INCORRECT:** "Configure the Restrict Viewer Access option in CloudFront and specify a deny list of unauthorized countries" is incorrect.

This feature is used to configure signed URLs and signed cookies.

**INCORRECT:** "Update the S3 bucket policy with condition statements that deny access based on the source IP addresses of users" is incorrect.

This would be hard to manage as the addresses may change and be hard to identify. The users should always go through the CloudFront distribution as well.

**INCORRECT:** "Configure a query string whitelist in CloudFront and specify a list of countries that should be denied access using query string parameters" is incorrect.

This is not the correct usage for the query string whitelist feature which is used for determining the query string parameters that you want CloudFront to use as a basis for caching.

**References:**

## Amazon CloudFront

Developer Guide



- ▶ What is Amazon CloudFront?
  - Setting up
- ▶ Getting started
- ▶ Working with distributions
- ▶ Working with policies
- ▶ Adding, removing, or replacing content
- ▼ Configuring secure access and restricting access to content
  - ▶ Using HTTPS with CloudFront
  - ▶ Using alternate domain names and HTTPS
  - ▶ Restricting content with signed URLs and signed cookies
    - Restricting access to an Amazon S3 origin
    - Restricting access to Application Load Balancers
    - Using AWS WAF to control access to your content
  - Geographically restricting content**
    - Using field-level encryption to help protect sensitive data
- ▶ Optimizing caching and availability
- ▶ Troubleshooting
- ▶ Request and response behavior
- ▶ Video on demand (VOD) and live streaming video
- ▶ Customizing with edge functions
- ▶ Reports, metrics, and logs

# Restricting the geographic distribution of your content

[PDF](#) | [RSS](#)

You can use *geographic restrictions*, sometimes known as *geo blocking*, to prevent users in specific geographic locations from accessing content that you're distributing through a CloudFront distribution. To use geographic restrictions, you have two options:

- Use the CloudFront geographic restrictions feature. Use this option to restrict access to all of the files that are associated with a distribution and to restrict access at the country level.
- Use a third-party geolocation service. Use this option to restrict access to a subset of the files that are associated with a distribution or to restrict access at a finer granularity than the country level.

## Topics

- [Using CloudFront geographic restrictions](#)
- [Using a third-party geolocation service](#)

## Using CloudFront geographic restrictions

When a user requests your content, CloudFront typically serves the requested content regardless of where the user is located. If you need to prevent users in specific countries from accessing your content, you can use the CloudFront geographic restrictions feature to do one of the following:

- Allow your users to access your content only if they're in one of the approved countries on your allow list.
- Prevent your users from accessing your content if they're in one of the banned countries on your block list.

For example, if a request comes from a country where you are not authorized to distribute your content, you can use CloudFront geographic restrictions to block the request.

### Note

CloudFront determines the location of your users by using a third-party database. The accuracy of the mapping between IP addresses and countries varies by Region. Based on recent tests, the overall accuracy is 99.8%. If CloudFront can't determine a user's location, CloudFront serves the content that the user has requested.

Here's how geographic restrictions work:

1. Suppose you have rights to distribute your content only in Liechtenstein. You update your CloudFront distribution to add an allow list that contains only Liechtenstein. (Alternatively, you could add a block list that contains every country except Liechtenstein.)
2. A user in Monaco requests your content, and DNS routes the request to a CloudFront edge location in Milan, Italy.
3. The edge location in Milan looks up your distribution and determines that the user in Monaco is not allowed to download your content.
4. CloudFront returns an HTTP status code `403` (Forbidden) to the user.

You can optionally configure CloudFront to return a custom error message to the user, and you can specify how long you want CloudFront to cache the error response for the requested file. The default value is 10 seconds. For more information, see [Creating a custom error page for specific HTTP status codes](#).

Geographic restrictions apply to an entire distribution. If you need to apply one restriction to part of your content and a different restriction (or no restriction) to another part of your content, you must either create separate CloudFront distributions or [use a third-party geolocation service](#).

If you enable CloudFront [standard logs](#) (access logs), you can identify the requests that CloudFront rejected by searching for the log entries in which the value of `sc-status` (the HTTP status code) is `403`. However, using only the standard logs, you can't distinguish a request that CloudFront rejected based on the location of the user from a request that CloudFront rejected because the user didn't have permission to access the file for another reason. If you have a third-party geolocation service such as Digital Element or MaxMind, you can identify the location of requests based on the IP address in the `c-ip` (client IP) column in the access logs. For more information about CloudFront standard logs, see [Configuring and using standard logs \(access logs\)](#).

The following procedure explains how to use the CloudFront console to add geographic restrictions to an existing distribution. For information about how to use the console to create a distribution, see [Creating a distribution](#).

## Part 7 – Restricting the geographic distribution of your content



# Restricting the geographic distribution of your content

[PDF](#) | [RSS](#)

You can use *geographic restrictions*, sometimes known as *geo blocking*, to prevent users in specific geographic locations from accessing content that you're distributing through a CloudFront distribution. To use geographic restrictions, you have two options:

- Use the CloudFront geographic restrictions feature. Use this option to restrict access to all of the files that are associated with a distribution and to restrict access at the country level.
- Use a third-party geolocation service. Use this option to restrict access to a subset of the files that are associated with a distribution or to restrict access at a finer granularity than the country level.

## Topics

- [Using CloudFront geographic restrictions](#)
- [Using a third-party geolocation service](#)

### 6. QUESTION

A company delivers content to subscribers distributed globally from an application running on AWS. The application uses a fleet of Amazon EC2 instance in a private subnet behind an Application Load Balancer (ALB). Due to an update in copyright restrictions, it is necessary to block access for specific countries.

What is the EASIEST method to meet this requirement?

- ☐ Modify the ALB security group to deny incoming traffic from blocked countries
- ☐ Modify the security group for EC2 instances to deny incoming traffic from blocked countries
- ☒ Use Amazon CloudFront to serve the application and deny access to blocked countries
- ☐ Use a network ACL to block the IP address ranges associated with the specific countries

Correct



Correct

**Explanation:**

When a user requests your content, CloudFront typically serves the requested content regardless of where the user is located. If you need to prevent users in specific countries from accessing your content, you can use the CloudFront geo restriction feature to do one of the following:

- Allow your users to access your content only if they're in one of the countries on a whitelist of approved countries.
- Prevent your users from accessing your content if they're in one of the countries on a blacklist of banned countries.

For example, if a request comes from a country where, for copyright reasons, you are not authorized to distribute your content, you can use CloudFront geo restriction to block the request.

This is the easiest and most effective way to implement a geographic restriction for the delivery of content.

**CORRECT:** "Use Amazon CloudFront to serve the application and deny access to blocked countries" is the correct answer.

**INCORRECT:** "Use a Network ACL to block the IP address ranges associated with the specific countries" is incorrect as this would be extremely difficult to manage.

## Part 8 – Field level encryption



# Field level encryption

# Introducing Field-Level Encryption on Amazon CloudFront

Posted On: Dec 14, 2017

Starting today, you can use a new [Amazon CloudFront](#) capability called Field-Level Encryption to further enhance the security of sensitive data, such as credit card numbers or personally identifiable information (PII) like social security numbers. CloudFront's field-level encryption further encrypts sensitive data in an HTTPS form using field-specific encryption keys (which you supply) before a POST request is forwarded to your origin. This ensures that sensitive data can only be decrypted and viewed by certain components or services



16. QUESTION

A company runs a dynamic website that is hosted on an on-premises server in the United States. The company is expanding to Europe and is investigating how they can optimize the performance of the website for European users. The website's backed must remain in the United States. The company requires a solution that can be implemented within a few days.

What should a Solutions Architect recommend?

- ☐ Migrate the website to Amazon S3. Use cross-Region replication between Regions and a latency-based Route 53 policy.

# TPN

- 83. ***Phenomenon1*** – the tendency of X to Y.
- 84. ***Phen2*** – the tendency of X to Y.
- 85. ***Phen3*** – the tendency of X to Y.
- 86. ***Phen4*** – the tendency of X to Y.
- 87. ***Phen5*** – the tendency of X to Y.
- 88. ***Phen6*** – the tendency of X to Y.
- 89. ***Phen7*** – the tendency of X to Y.
- 90. ***Phen8*** – the tendency of X to Y.
- 91. ***Phen9*** – the tendency of X to Y.
- 92. ***Phen10*** – the tendency of X to Y.

# Glossary

## Signed URL

AWS write:

“ You can distribute private content using a signed URL that is valid for only a short time—possibly for as little as a few minutes. Signed URLs that are valid for such a short period are good for distributing content on-the-fly to a user for a specific purpose, such as distributing movie rentals or music downloads to customers on demand. You can also distribute private content using a signed URL that is valid for a longer time, possibly for years. “

# Bibliography

- IX.      Official
- X.       Unofficial
- XI.     Critical
- XII.    General

## X.      Official

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:

<URL here>.

## [AWS 2004]

“Introducing the Amazon Simple Queue Service”. AWS. 3<sup>rd</sup> Nov 2004. Available at:  
<https://aws.amazon.com/about-aws/whats-new/2004/11/03/introducing-the-amazon-simple-queue-service/>

## [Barr 1]

Barr, Jeff (2006). “Amazon Simple Queue Service Released”. *AWS News Blog*. 13<sup>th</sup> July 2006. Available at:  
[https://aws.amazon.com/blogs/aws/amazon\\_simple\\_q/](https://aws.amazon.com/blogs/aws/amazon_simple_q/)

# XI. Unofficial

“What is difference between Pre-Signed Url and Signed Url?”

**Available at:**

<https://stackoverflow.com/questions/20862195/what-is-difference-between-pre-signed-url-and-signed-url>

[https://en.wikipedia.org/wiki/Column\\_level\\_encryption](https://en.wikipedia.org/wiki/Column_level_encryption)

[https://link.springer.com/chapter/10.1007/978-1-4684-4730-9\\_19](https://link.springer.com/chapter/10.1007/978-1-4684-4730-9_19)

<https://stackoverflow.com/questions/41929534/relationship-between-origin-access-identities-oais-and-cloudfront-signed-urls>

Boyd, Greg. Database Encryption. *Mainframe Crypto*. Available at:  
[newera.com/INFO/Database\\_Encryption.pdf](http://newera.com/INFO/Database_Encryption.pdf)

## XII. Critical

## XIII. General

Elastic message queues. Ahmed El Rheddane and Noel De Palma

MSMQ is dead. David Boike.

Chapter 18 – Distributed computing – models and methods. Leslie Lamport and Nancy Lynch.

### **[Wikipedia 1]**

Message queue. Wikipedia. Available at:  
[https://en.wikipedia.org/wiki/Message\\_queue](https://en.wikipedia.org/wiki/Message_queue)

### **[Bajantri 1986]**

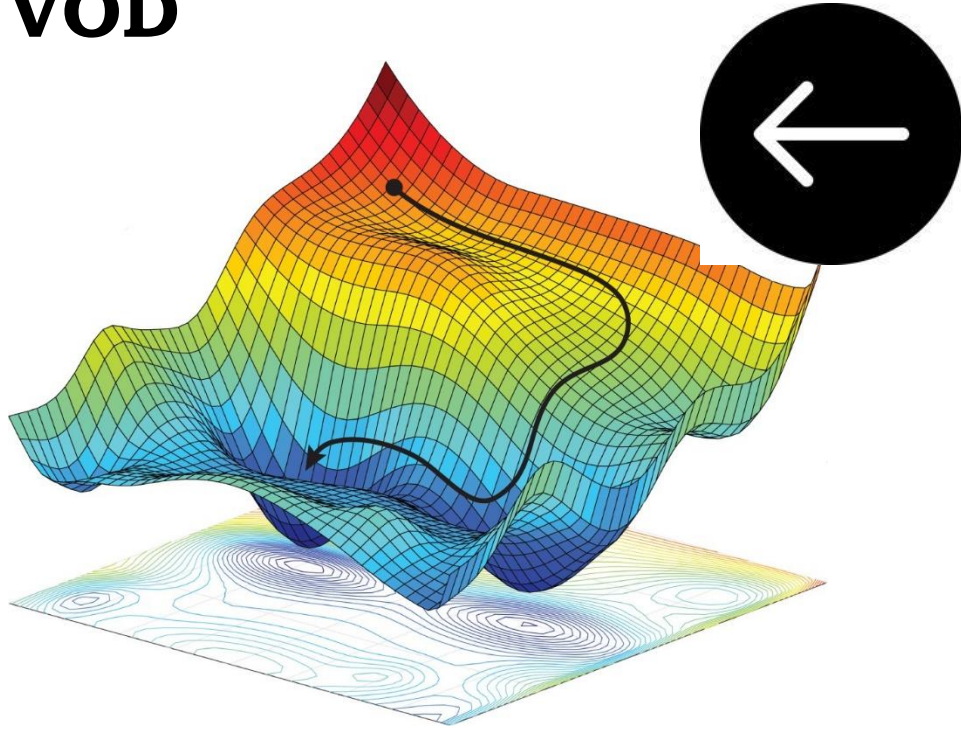
Bajantri, M. and David B Skillicorn (1986). A fast multiprocessor message passing implementation. Information Processing Letters, 24 (6): 381-389. Available at:  
<https://www.sciencedirect.com/science/article/abs/pii/0020019087901153>

### **[Christopher 1988]**

Christopher, Thomas (1988). Message driven computing and its relationship to actors. OOPSLA/ECOOP '88: Proceedings of the 1988 ACM SIGPLAN workshop on Object-based concurrent programming.  
<https://doi.org/10.1145/67386.67405>.



# CloudFront4 – optimization and VOD



[BACK TO CLOUDFRONT](#)

## Optimizing high availability with CloudFront origin failover

[PDF](#) | [RSS](#)

You can set up CloudFront with origin failover for scenarios that require high availability. To get started, you create an *origin group* with two origins: a primary and a secondary. If the primary origin is unavailable, or returns specific HTTP response status codes that indicate a failure, CloudFront automatically switches to the secondary origin.

To set up origin failover, you must have a distribution with at least two origins. Next, you create an origin group for your distribution that includes two origins, setting one as the primary. Finally, you create or update a cache behavior to use the origin group.

To see the steps for setting up origin groups and configuring specific origin failover options, see [Creating an origin group](#).



### 1. QUESTION

An Amazon S3 bucket in the us-east-1 Region hosts the static website content of a company. The content is made available through an Amazon CloudFront origin pointing to that bucket. A second copy of the bucket is created in the ap-southeast-1 Region using cross-region replication. The chief solutions architect wants a solution that provides greater availability for the website.

Which combination of actions should a solutions architect take to increase availability? (Select TWO.)

- ☐ Point Amazon Route 53 to the replica bucket by creating a record.
- ☒ Using us-east-1 bucket as the primary bucket and ap-southeast-1 bucket as the secondary bucket, create a CloudFront origin group.
- ☐ Add an origin for ap-southeast-1 to CloudFront.
- ☒ Set up failover routing in Amazon Route 53.
- ☐ Create an origin for CloudFront for both buckets.

Incorrect

Explanation:

You can set up CloudFront with origin failover for scenarios that require high availability. To get started, you create an *origin group* with two origins: a primary and a secondary. If the primary origin is unavailable or returns specific HTTP response status codes that indicate a failure, CloudFront automatically switches to the secondary origin.

**CORRECT:** "Add an origin for ap-southeast-1 to CloudFront" is the correct answer (as explained above.)

**CORRECT:** "Using us-east-1 bucket as the primary bucket and ap-southeast-1 bucket as the secondary bucket, create a CloudFront origin group" is also a correct answer (as explained above.)

**INCORRECT:** "Create an origin for CloudFront for both buckets" is incorrect. This would not increase the availability of the solution on its own.

**INCORRECT:** "Set up failover routing in Amazon Route 53" is incorrect as we are trying to enable failover in CloudFront and using Route 53 is for routing domain names.

**INCORRECT:** "Create a record in Amazon Route 53 pointing to the replica bucket" is incorrect as we are trying to enable failover in CloudFront and using Route 53 is for routing domain names.

# TPN

- 93. ***Phenomenon1*** – the tendency of X to Y.
- 94. ***Phen2*** – the tendency of X to Y.
- 95. ***Phen3*** – the tendency of X to Y.
- 96. ***Phen4*** – the tendency of X to Y.
- 97. ***Phen5*** – the tendency of X to Y.
- 98. ***Phen6*** – the tendency of X to Y.
- 99. ***Phen7*** – the tendency of X to Y.
- 100. ***Phen8*** – the tendency of X to Y.
- 101. ***Phen9*** – the tendency of X to Y.
- 102. ***Phen10*** – the tendency of X to Y.

# Glossary

## Smooth Streaming

Description of what term means here.

# Bibliography

- XIII.      Official
- XIV.      Unofficial
- XV.      Critical
- XVI.      General

## XIV. Official

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

### **[Surname1]**

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

### **[AWS 2004]**

“Introducing the Amazon Simple Queue Service”. AWS.  
3<sup>rd</sup> Nov 2004. Available at:  
<https://aws.amazon.com/about-aws/whats-new/2004/11/03/introducing-the-amazon-simple-queue-service/>

#### [Barr 1]

Barr, Jeff (2006). “Amazon Simple Queue Service Released”. *AWS News Blog*. 13<sup>th</sup> July 2006. Available at:  
[https://aws.amazon.com/blogs/aws/amazon\\_simple\\_q/](https://aws.amazon.com/blogs/aws/amazon_simple_q/)

## XV. Unofficial

#### [Levitt 2005]

Levitt, Jason (2005). Fun with Amazon’s simple queue service. Available at:  
<https://www.xml.com/pub/a/2005/01/05/sqs.html>

#### [Barr 2]

Barr, Jeff (year). My first 12 years at Amazon.com. Jeff-barr.com. Available at: <http://jeff-barr.com/2014/08/19/my-first-12-years-at-amazon-dot-com/>

## XVI. Critical

## XVII. General

Elastic message queues. Ahmed El Rheddane and Noel De Palma

MSMQ is dead. David Boike.

Chapter 18 – Distributed computing – models and methods. Leslie Lamport and Nancy Lynch.

### **[Wikipedia 1]**

Message queue. Wikipedia. Available at:  
[https://en.wikipedia.org/wiki/Message\\_queue](https://en.wikipedia.org/wiki/Message_queue)

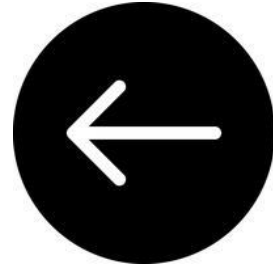
### **[Bajantri 1986]**

Bajantri, M. and David B Skillicorn (1986). A fast multiprocessor message passing implementation. Information Processing Letters, 24 (6): 381-389. Available at:  
<https://www.sciencedirect.com/science/article/abs/pii/0020019087901153>

### **[Christopher 1988]**

Christopher, Thomas (1988). Message driven computing and its relationship to actors. OOPSLA/ECOOP '88: Proceedings of the 1988 ACM SIGPLAN workshop on Object-based concurrent programming.  
<https://doi.org/10.1145/67386.67405>.

# CloudFront5 – Lambda@edge and monitoring



[BACK TO CLOUDFRONT](#)



QUESTION 6 OF 65

6. QUESTION

A company delivers content to subscribers distributed globally from an application running on AWS. The application uses a fleet of Amazon EC2 instance in a private subnet behind an Application Load Balancer (ALB). Due to an update in copyright restrictions, it is necessary to block access for specific countries.

What is the EASIEST method to meet this requirement?

- ☐ Modify the ALB security group to deny incoming traffic from blocked countries
- ☐ Modify the security group for EC2 instances to deny incoming traffic from blocked countries
- ☒ Use Amazon CloudFront to serve the application and deny access to blocked countries
- ☐ Use a network ACL to block the IP address ranges associated with the specific countries

Correct

2. QUESTION

An application runs on Amazon EC2 instances behind an Application Load Balancer (ALB). Amazon CloudFront is used as the front-end of the application and AWS WAF is used to protect the front-end with the AWS Managed Rules rule group.

A security architect is concerned that the infrastructure is vulnerable to layer 7 DDoS attacks. What improvements can be made to the solution to protect against this type of attack?

- ☐ Configure an IP set match rule on AWS WAF that blocks web requests based on the IP address of the web request origin.
- ☒ Configure a rate-based rule on AWS WAF that puts a temporary block on requests from IP addresses that send excessive requests.
- ☐ Configure field-level encryption for the distribution and upload an SSL/TLS certificate from Amazon Certificate Manager (ACM).
- ☐ Configure a Lambda@Edge function that imposes a rate limit on CloudFront viewer requests and blocks traffic that exceeds the limits.

Correct

**Explanation:**

A rate-based rule tracks the rate of requests for each originating IP address and triggers the rule action on IPs with rates that go over a limit. You set the limit as the number of requests per 5-minute time span. You can use this type of rule to put a temporary block on requests from an IP address that's sending excessive requests. By default, AWS WAF aggregates requests based on the IP address from the web request origin, but you can configure the rule to use an IP address from an HTTP header, like X-Forwarded-For, instead.

This rule can help prevent layer 7 DDoS attacks as the IP addresses of bots would be automatically blocked once they exceed the rate defined.

**CORRECT:** "Configure a rate-based rule on AWS WAF that puts a temporary block on requests from IP addresses that send excessive requests" is the correct answer (as explained above.)

**INCORRECT:** "Configure a Lambda@Edge function that imposes a rate limit on CloudFront viewer requests and blocks traffic that exceeds the limits" is incorrect.

AWS WAF can do this using rate-based rules and is much better suited to the job than writing your own custom code and running it using Lambda.

**INCORRECT:** "Configure an IP set match rule on AWS WAF that blocks web requests based on the IP address of the web request origin" is incorrect.

An IP set match rule uses a list of known IP addresses. With a DDoS attack you don't know the IP addresses of the bots ahead of time so this would not be effective.

**INCORRECT:** "Configure field-level encryption for the distribution and upload an SSL/TLS certificate from Amazon Certificate Manager (ACM)" is incorrect.

Field level encryption adds protection for certain data in transit and is not useful for protecting against DDoS attacks.

**References:**

# TPN

- 103. ***Phenomenon1*** – the tendency of X to Y.
- 104. ***Phen2*** – the tendency of X to Y.
- 105. ***Phen3*** – the tendency of X to Y.
- 106. ***Phen4*** – the tendency of X to Y.
- 107. ***Phen5*** – the tendency of X to Y.
- 108. ***Phen6*** – the tendency of X to Y.
- 109. ***Phen7*** – the tendency of X to Y.
- 110. ***Phen8*** – the tendency of X to Y.
- 111. ***Phen9*** – the tendency of X to Y.
- 112. ***Phen10*** – the tendency of X to Y.

QUESTION 9 OF 65

## 9. QUESTION

A company hosts an application on Amazon EC2 instances behind Application Load Balancers in several AWS Regions. Distribution rights for the content require that users in different geographies must be served content from specific regions.

Which configuration meets these requirements?

- ☐ Configure Amazon CloudFront with multiple origins and AWS WAF.
- ☐ Create Amazon Route 53 records with a geoproximity routing policy.
- ☒ Create Amazon Route 53 records with a geolocation routing policy.
- ☐ Configure Application Load Balancers with multi-Region routing.

Correct

**Explanation:**

To protect the distribution rights of the content and ensure that users are directed to the appropriate AWS Region based on the location of the user, the geolocation routing policy can be used with Amazon Route 53.

Geolocation routing lets you choose the resources that serve your traffic based on the geographic location of your users, meaning the location that DNS queries originate from.

When you use geolocation routing, you can localize your content and present some or all of your website in the language of your users. You can also use geolocation routing to restrict distribution of content to only the locations in which you have distribution rights.

**CORRECT:** "Create Amazon Route 53 records with a geolocation routing policy" is the correct answer.

**INCORRECT:** "Create Amazon Route 53 records with a geoproximity routing policy" is incorrect. Use this routing policy when you want to route traffic based on the location of your resources and, optionally, shift traffic from resources in one location to resources in another.

**INCORRECT:** "Configure Amazon CloudFront with multiple origins and AWS WAF" is incorrect. AWS WAF protects against web exploits but will not assist with directing users to different content (from different origins).

**INCORRECT:** "Configure Application Load Balancers with multi-Region routing" is incorrect. There is no such thing as multi-Region routing for ALBs.

**References:**

<https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-policy.html>

Save time with our AWS cheat sheets:

🍁 SALE 🍁 Check out the Special FALL DEALS and get 40% OFF

**14. QUESTION**

A company runs a web application that serves weather updates. The application runs on a fleet of Amazon EC2 instances in a Multi-AZ Auto scaling group behind an Application Load Balancer (ALB). The instances store data in an Amazon Aurora database. A solutions architect needs to make the application more resilient to sporadic increases in request rates.

Which architecture should the solutions architect implement? (Select TWO.)

☒ Add an Amazon CloudFront distribution in front of the ALB

☐ Add an AWS Global Accelerator endpoint

☐ Add an AWS Transit Gateway to the Availability Zones

☒ Add Amazon Aurora Replicas

☐ Add and AWS WAF in front of the ALB

Correct

Explanation:

The architecture is already highly resilient but it may be subject to performance degradation if there are sudden increases in request rates. To resolve this situation Amazon Aurora Read Replicas can be used to serve read traffic which offloads requests from the main database. On the frontend an Amazon CloudFront distribution can be placed in front of the ALB and this will cache content for better performance and also offloads requests from the backend.

**CORRECT:** "Add Amazon Aurora Replicas" is the correct answer.

**CORRECT:** "Add an Amazon CloudFront distribution in front of the ALB" is the correct answer.

**INCORRECT:** "Add and AWS WAF in front of the ALB" is incorrect. A web application firewall protects applications from malicious attacks. It does not improve performance.

**INCORRECT:** "Add an AWS Transit Gateway to the Availability Zones" is incorrect as this is used to connect on-premises networks to VPCs.

**INCORRECT:** "Add an AWS Global Accelerator endpoint" is incorrect as this service is used for directing users to different instances of the application in different regions based on latency.

References:

<https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Aurora.Replication.html>

<https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Introduction.html>

See also with our AWS cheat sheet:

5. QUESTION

A company offers an online product brochure that is delivered from a static website running on Amazon S3. The company's customers are mainly in the United States, Canada, and Europe. The company is looking to cost-effectively reduce the latency for users in these regions.

What is the most cost-effective solution to these requirements?

- ☐ Create an Amazon CloudFront distribution and set the price class to use all Edge Locations for best performance.
- ☐ Create an Amazon CloudFront distribution and use Lambda@Edge to run the website's data processing closer to the users.
- ☒ Create an Amazon CloudFront distribution and set the price class to use only U.S., Canada and Europe.
- ☐ Create an Amazon CloudFront distribution that uses origins in U.S., Canada and Europe.

Correct

**Explanation:**

With Amazon CloudFront you can set the price class to determine where in the world the content will be cached. One of the price classes is "U.S, Canada and Europe" and this is where the company's users are located. Choosing this price class will result in lower costs and better performance for the company's users.

**CORRECT:** "Create an Amazon CloudFront distribution and set the price class to use only U.S, Canada and Europe." is the correct answer.

**INCORRECT:** "Create an Amazon CloudFront distribution and set the price class to use all Edge Locations for best performance" is incorrect. This will be more expensive as it will cache content in Edge Locations all over the world.

**INCORRECT:** "Create an Amazon CloudFront distribution that uses origins in U.S, Canada and Europe" is incorrect. The origin can be in one place, there's no need to add origins in different Regions. The price class should be used to limit the caching of the content to reduce cost.

**INCORRECT:** "Create an Amazon CloudFront distribution and use Lambda@Edge to run the website's data processing closer to the users" is incorrect. Lambda@Edge will not assist in this situation as there is no data processing required, the content from the static website must simply be cached at an edge location.

**References:**

# Glossary

## Smooth Streaming

Description of what term means here.

# Bibliography

- XVII. Official
- XVIII. Unofficial
- XIX. Critical
- XX. General

## XVIII. Official

### [Surname1]

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

### [Surname1]

Smith, David (year). Title of Work Here. 1<sup>st</sup> Jan 2022. City: Publisher.  
Available at:  
<URL here>.

### [AWS 2004]

“Introducing the Amazon Simple Queue Service”. AWS.  
3<sup>rd</sup> Nov 2004. Available at:  
<https://aws.amazon.com/about-aws/whats-new/2004/11/03/introducing-the-amazon-simple-queue-service/>

### [Barr 1]

Barr, Jeff (2006). “Amazon Simple Queue Service Released”. *AWS News Blog*. 13<sup>th</sup> July 2006. Available at:  
[https://aws.amazon.com/blogs/aws/amazon\\_simple\\_q/](https://aws.amazon.com/blogs/aws/amazon_simple_q/)

## XIX. Unofficial

### [Levitt 2005]

Levitt, Jason (2005). Fun with Amazon's simple queue service. Available at:  
<https://www.xml.com/pub/a/2005/01/05/sqs.html>

### **[Barr 2]**

Barr, Jeff (year). My first 12 years at Amazon.com. Jeff-barr.com. Available at: <http://jeff-barr.com/2014/08/19/my-first-12-years-at-amazon-dot-com/>

## XX. Critical

## XXI. General

Elastic message queues. Ahmed El Rheddane and Noel De Palma

MSMQ is dead. David Boike.

Chapter 18 – Distributed computing – models and methods. Leslie Lamport and Nancy Lynch.

### **[Wikipedia 1]**

Message queue. Wikipedia. Available at:  
[https://en.wikipedia.org/wiki/Message\\_queue](https://en.wikipedia.org/wiki/Message_queue)

### **[Bajantri 1986]**

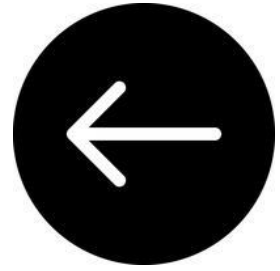


Bajantri, M. and David B Skillicorn (1986). A fast multiprocessor message passing implementation. Information Processing Letters, 24 (6): 381-389. Available at:  
<https://www.sciencedirect.com/science/article/abs/pii/0020019087901153>

### **[Christopher 1988]**

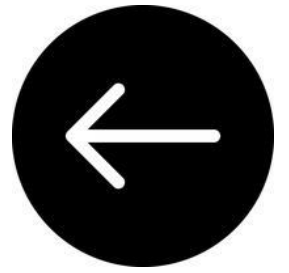
Christopher, Thomas (1988). Message driven computing and its relationship to actors. OOPSLA/ECOOP '88: Proceedings of the 1988 ACM SIGPLAN workshop on Object-based concurrent programming.  
<https://doi.org/10.1145/67386.67405>.

# Appendices



1. [Performance metrics](#)
2. [Availability versus durability](#)
3. [ACM's Special Interest Groups](#)
4. [Slippery Strings](#)
5. [You RASQAL](#)
6. [Slash Direction](#)
7. [APP](#)
8. [Geography](#)
9. [Disaster Recovery](#)

# I. The performance metrics



B andwidth

L atency

I OPS

T hroughput

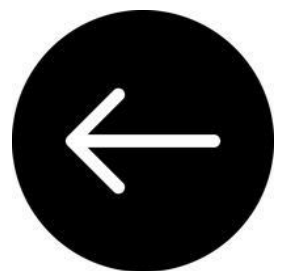
S peed

Some of these terms are used in our discussion of EBS. Dougal Ballantyne explains [here](#).



Armando Fox gives a generic talk on cloud computing

## II. Availability and durability



<https://aws.amazon.com/blogs/publicsector/achieving-five-nines-cloud-justice-public-safety/#:~:text=The%20accepted%20availability%20standard%20for,system%20must%20work%20seamlessly%20together.>

## *Chapter 2*

---

# Reliability and Availability

---

The two concepts reliability and availability are talked about, written about, equated with each other, and given star status but, in the main, remain somewhat one-dimensional concepts. In this chapter, and throughout this book, I hope to show that these concepts, particularly availability, have other dimensions and interpretations as well.

### **Introduction to Reliability, Availability, and Serviceability**

*Reliability* represents the probability of components, parts, and systems to perform their required

Critchley



Some of our other favourite  
characters:





There are ways in which reliability and resilience come apart. Boris Johnson may have been resilient, but he was not able to be relied upon. Night time reliably follows day time, but there's nothing resilient about this period of darkness.

Reliability is often cited as a virtue of scientific measurements. It denotes that they are repeatable and, literally, are able to be relied upon.

Resiliency denotes the stable, continuing functioning of something. These terms are of course similar. Someone will probably try to tell you there is some crucial difference between them at some point or another.

In this book, we focus on three concerns that are important in most software systems:

#### *Reliability*

The system should continue to work *correctly* (performing the correct function at the desired level of performance) even in the face of *adversity* (hardware or software faults, and even human error). See “**Reliability**” on page 6.

#### *Scalability*

As the system *grows* (in data volume, traffic volume, or complexity), there should be reasonable ways of dealing with that growth. See “**Scalability**” on page 10.

#### *Maintainability*

Over time, many different people will work on the system (engineering and operations, both maintaining current behavior and adapting the system to new use cases), and they should all be able to work on it *productively*. See “**Maintainability**” on page 18.

These words are often cast around without a clear understanding of what they mean. In the interest of thoughtful engineering, we will spend the rest of this chapter

Martin Kleppmann (in his *Designing Data Intensive Applications*) defines three virtues of software systems.

A Boeing 747 can fly relatively safely on a single engine so with four it enjoys a comfortable degree of redundancy. This is why Boeing puts them there. Under normal circumstances, engineers consider even a single engine failure highly unlikely, so, until British Airways inadvertently proved otherwise, they considered the likelihood of all four failing during the same flight to be negligible – as near to ‘impossible’ as to make no difference. In fact, the aviation industry considered a quadruple engine failure so unlikely that they taught pilots to treat indications of it as an instrumentation failure. Nobody had considered the possibility of volcanic ash.



The above is an extract from a fascinating paper on redundancy and how it can mislead engineers. The author is John Downer and the title is “When Failure is an option”.

# The coming day of fault tolerance

By Wilbur H. Highleyman

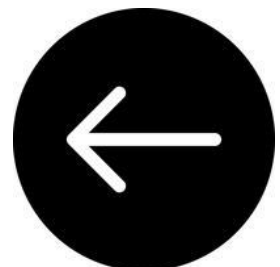
A little more than four years ago, my In Depth series, "Survivable Systems" [CW, Feb. 4-Feb. 25, 1980], appeared in *Computerworld*. At that time, Tandem Computers, Inc. offered the only commercially available fault-tolerant system, the Nonstop. Now that there are about two dozen offerings from domestic and foreign manufacturers, it is clear that fault tolerance has

---

*As with memory and languages and*

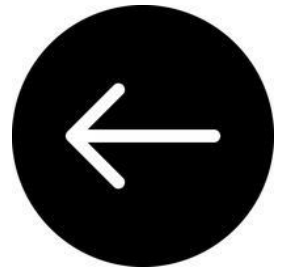


## III. Special Interest Groups





## IV. Slippery Strings



### The History of the URL

05/03/2020



Zack Bloom

On the [11th of January 1982](#) twenty-two computer scientists met to discuss an issue with 'computer mail' (now known as email). Attendees included [the guy who would create Sun Microsystems](#), [the guy who made Zork](#), [the NTP guy](#), and [the guy who convinced the government to pay for Unix](#). The problem was simple: there were 455 hosts on the ARPANET and the situation was getting out of control.

# amazonaws





#### 5.2.1.1 Resources and Resource Identifiers

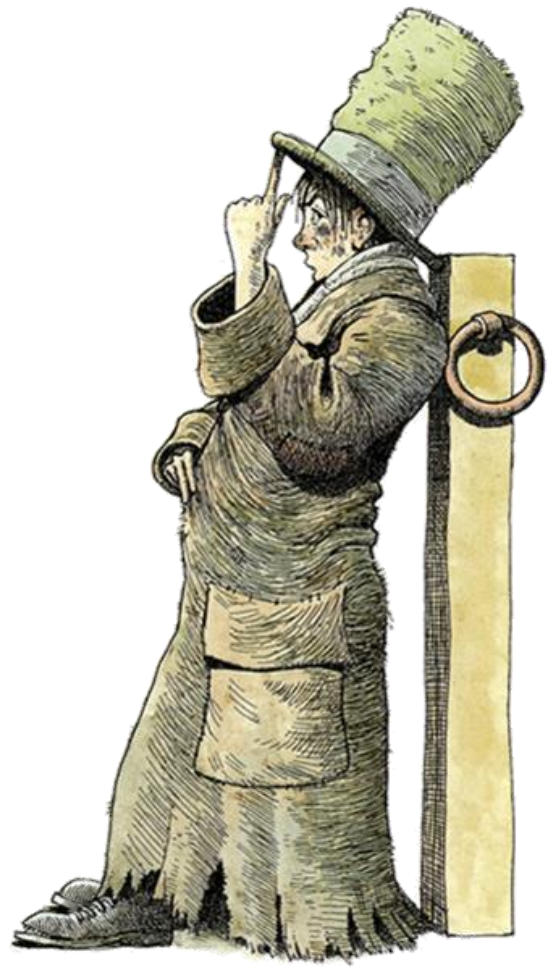
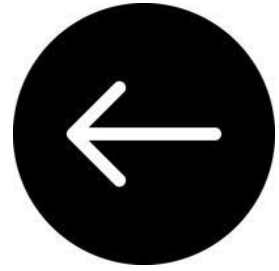
The key abstraction of information in REST is a *resource*. Any information that can be named can be a resource: a document or image, a temporal service (e.g. “today’s weather in Los Angeles”), a collection of other resources, a non-virtual object (e.g. a person), and so on. In other words, any concept that might be the target of an author’s hypertext reference must fit within the definition of a resource. A resource is a conceptual mapping to a set of entities, not the entity that corresponds to the mapping at any particular point in time.

More precisely, a resource  $R$  is a temporally varying membership function  $M_R(t)$ , which for time  $t$  maps to a set of entities, or values, which are equivalent. The values in the set may be *resource representations* and/or *resource identifiers*. A resource can map to the

empty set, which allows references to be made to a concept before any realization of that concept exists — a notion that was foreign to most hypertext systems prior to the Web [61]. Some resources are static in the sense that, when examined at any time after their creation, they always correspond to the same value set. Others have a high degree of variance in their value over time. The only thing that is required to be static for a resource is the semantics of the mapping, since the semantics is what distinguishes one resource from another.

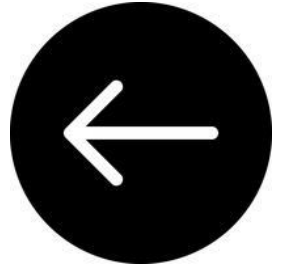
For example, the “authors’ preferred version” of an academic paper is a mapping whose value changes over time, whereas a mapping to “the paper published in the proceedings of conference X” is static. These are two distinct resources, even if they both map to the same value at some point in time. The distinction is necessary so that both resources can be identified and referenced independently. A similar example from software engineering is the separate identification of a version-controlled source code file when referring to the “latest revision”, “revision number 1.2.7”, or “revision included with the Orange release.”

V. RASQAL





## VI. Slash direction



In Windows systems, for example, the backslash is used to separate elements of a file path, for example: C:\Documents\User\File. In [C](#), [Perl](#) and [Unix](#) scripting, the backslash indicates that the following character must be treated in some special way. Within the TeX typesetting markup system, the backslash starts [tags](#).

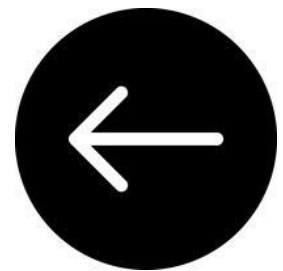


The forward slash (or simply slash) character (/) is the divide symbol in programming and on calculator keyboards. For example, **10 / 7** means 10 divided by 7. The slash is also often used in command line syntax to indicate a switch. For example, in the DOS/Windows Xcopy statement **xcopy \*.\* d: /s**, the **/s** is a switch that tells the program to copy all subfolders. In Unix paths, which have become popular due to Internet addresses, the slash separates the elements of the path as in **www.company.com/news/previous/abc.html**.

### **It Used to Be Just a Slash**

Before computers became ubiquitous, the forward slash was simply a "slash." Since the days of DOS, which introduced the horrid backslash, many people refer to a regular slash as a forward slash to avoid confusion. See [backslash](#).

## VII. App



AppMesh  
AppSync  
AppStream  
AppFlow

In chronological order

AppStream (announced 2013)

AppSync (announced 2017)

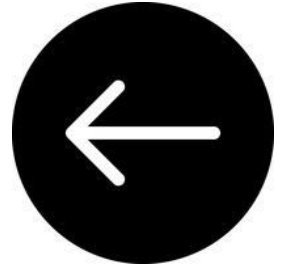
App Mesh (announced 2018)

AppFlow (announced 2020)

AWS App Runner (announced 2021)



## VIII. Geography



- We talk about CloudFront Distributions being *in front* of load balancers.

### 14. QUESTION

A company runs a web application that serves weather updates. The application runs on a fleet of Amazon EC2 instances in a Multi-AZ Auto scaling group behind an Application Load Balancer (ALB). The instances store data in an Amazon Aurora database. A solutions architect needs to make the application more resilient to sporadic increases in request rates.

Which architecture should the solutions architect implement? (Select TWO.)

- ☐ Add an Amazon CloudFront distribution in front of the ALB

- We talk about auto scaling groups being *behind* load balancers.
- We talk about CloudFront distributions being *in front* of S3 buckets.

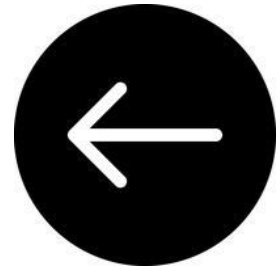
13. QUESTION

A developer is deploying a website hosted in an Amazon S3 bucket. An Amazon CloudFront distribution will be deployed in front of the S3 bucket to cache the content. The developer requires that users may only access the website using the CloudFront distribution and should not be able to access the website directly by using the S3 URL.

Which configurations should a security engineer make to support these requirements? (Select TWO.)

- ☐ Configure the S3 bucket permissions so that only the origin access identity can access the bucket contents.
- ☐ Create an origin access identity (OAI) and associate it with the CloudFront distribution.
- ☐ Implement a condition in the S3 bucket policy that limits access to "Principal": "cloudfront.amazonaws.com".
- ☐ Configure a gateway endpoint for the S3 bucket and attach the CloudFront distribution to the VPC.
- ☐ Configure CloudFront to add a custom HTTP header to requests for the S3 bucket and configure the bucket to only accept requests with the custom header.

# IX. Disaster Recovery



---

## Disaster Recovery of Workloads on AWS: Recovery in the Cloud

### **AWS Well-Architected Framework**

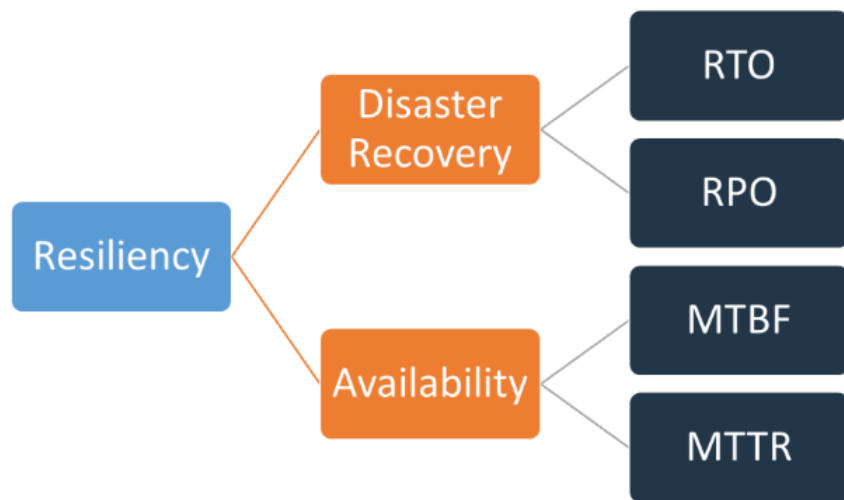
## Contributors

Contributors to this document include:

- Alex Livingstone, Practice Lead Cloud Operations, AWS Enterprise Support
- Seth Eliot, Principal Reliability Solutions Architect, Amazon Web Services

## Disaster recovery and availability

Disaster recovery can be compared to *availability*, which is another important component of your resiliency strategy. Whereas disaster recovery measures objectives for one-time events, availability objectives measure mean values over a period of time.



## Section 5

**High availability** is  
*not* **disaster recovery**



## Pilot light

From Wikipedia, the free encyclopedia

*For an assist light in flash photography, see Modeling light.*

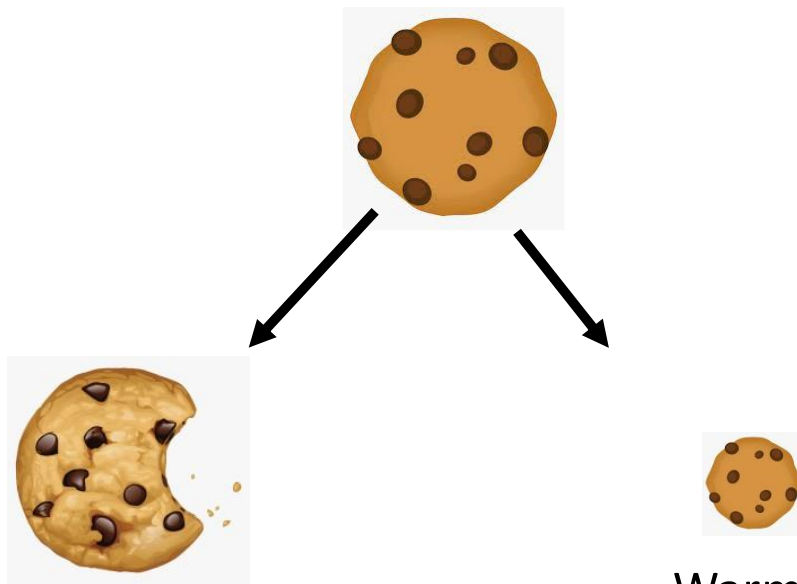
A **pilot light** is a small [gas flame](#), usually [natural gas](#) or [liquefied petroleum gas](#), which serves as an ignition source for a more powerful gas burner. Originally a pilot light was kept permanently alight, but this wastes gas. Now it is more common to light a burner electrically, but gas pilot lights are still used when a high energy ignition source is necessary, as in when lighting a large burner.

The term "pilot light" is also used occasionally for an electrical indicator light that illuminates to show that electrical power is available, or that an electrical device is operating. Such indicators were originally [incandescent lamps](#) or [neon lamps](#), but now are usually [LEDs](#).

## Pilot light

With the *pilot light* approach, you replicate your data from one Region to another and provision a copy of your core workload infrastructure. Resources required to support data replication and backup, such as databases and object storage, are always on. Other elements, such as application servers, are loaded with application code and configurations, but are "switched off" and are only used during testing or when disaster recovery failover is invoked. In the cloud, you have the flexibility to deprovision resources when you do not need them, and provision them when you do. A best practice for "switched off" is to not deploy the resource, and then create the configuration and capabilities to deploy it ("switch on") when needed. Unlike the backup and restore approach, your core infrastructure is always available and you always have the option to quickly provision a full scale production environment by switching on and scaling out your application servers.





Pilot light

Warm standby