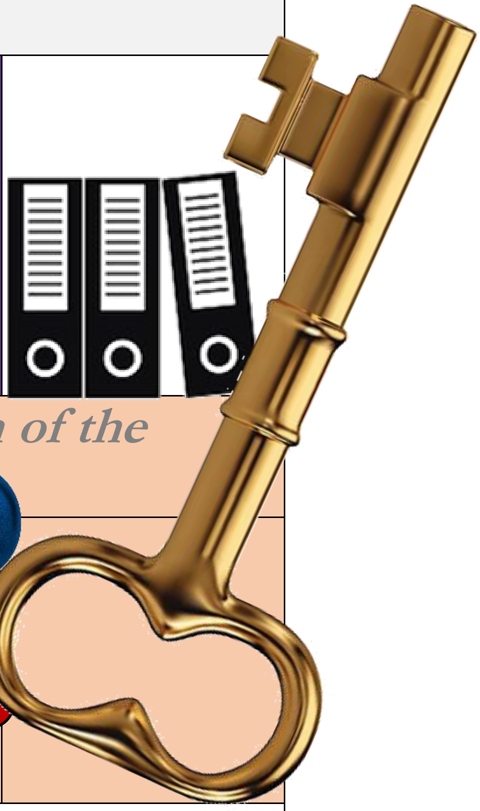
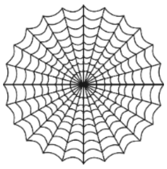
Aggressive Expansion

Contents



WorkMail	 Budgets	Elastic File System Also discussed are NFS, NTFS.	CodeCommit
CodePipeline	API Gateway	Device Farm	Service Catalog
 Web Application Firewall			

Announced on Wednesday 7th October 2015:

Inspector	QuickSight	Database Migration Service	Snowball Family

This Wednesday, in one expression: “I *don’t* mind serving Inspector Snowball quickly”. The term “don’t mind serving” involves the letters DMS.

Elasticsearch Service	 IoT Core	 Cloud Adoption Framework (CAF)	
Announced in Dec 2015:			
AWS Cost and Usage Report	Elastic Container Registry		

Image credits

The cover image shows the Space Needle in Seattle, which is the home to the Amazon headquarters. It is from Getty Images.

It is found, for example, in:

> Notaro, Vicki (2024). A holiday in Seattle: coffee, craft beer and spectacular views. *The Times* [Online].

Regarding the title of this book, Andy Troutman, head of internal deployment services at Amazon, said "today CodeDeploy is in five AWS Regions—we're aggressively expanding it".

Hans Zimmer also named one of his tracks for *The Dark Knight* (2008) as Aggressive Expansion. Principal photography for this film took place in Chicago in 2007.

Cost Explorer

Cost Explorer

Bibliography

I. Official

II. Unofficial

[Bueno 2023]

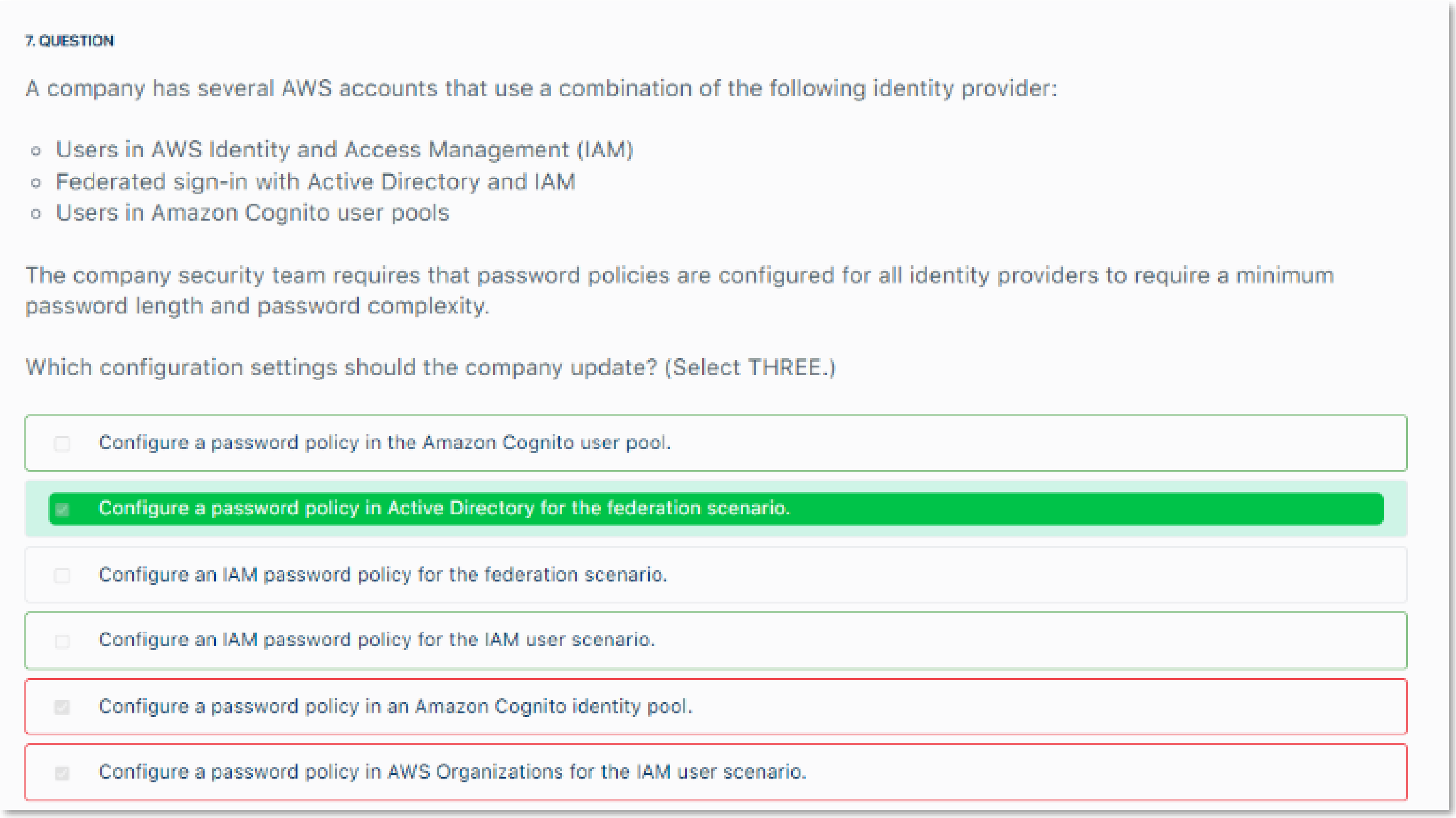
Carlos, Bueno (2023). A Dismal Guide to AWS Billing. Available at: <https://carlos.bueno.org/2023/03/aws-dismal-guide.html?ck_subscriber_id=1560524742>

III. Critical

IV. General

Amazon Mobile Analytics

Amazon Cognito



7. QUESTION

A company has several AWS accounts that use a combination of the following identity provider:

- Users in AWS Identity and Access Management (IAM)
- Federated sign-in with Active Directory and IAM
- Users in Amazon Cognito user pools

The company security team requires that password policies are configured for all identity providers to require a minimum password length and password complexity.

Which configuration settings should the company update? (Select THREE.)

☐ Configure a password policy in the Amazon Cognito user pool.

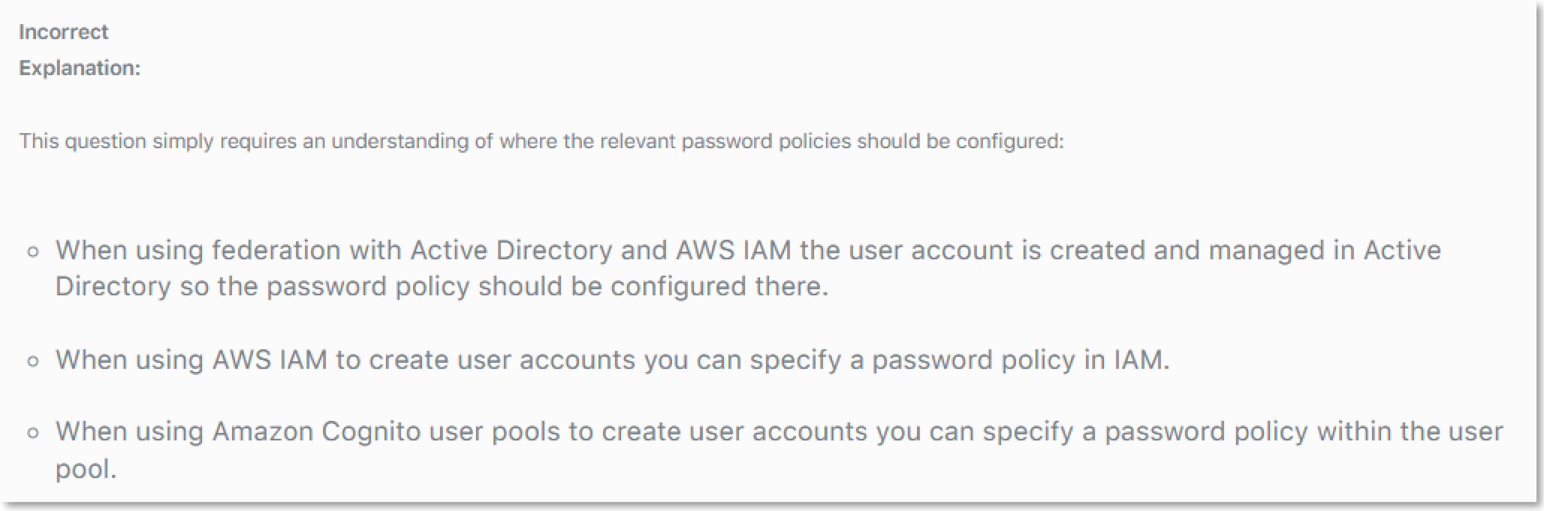
☑ Configure a password policy in Active Directory for the federation scenario.

☐ Configure an IAM password policy for the federation scenario.

☐ Configure an IAM password policy for the IAM user scenario.

☑ Configure a password policy in an Amazon Cognito identity pool.

☑ Configure a password policy in AWS Organizations for the IAM user scenario.

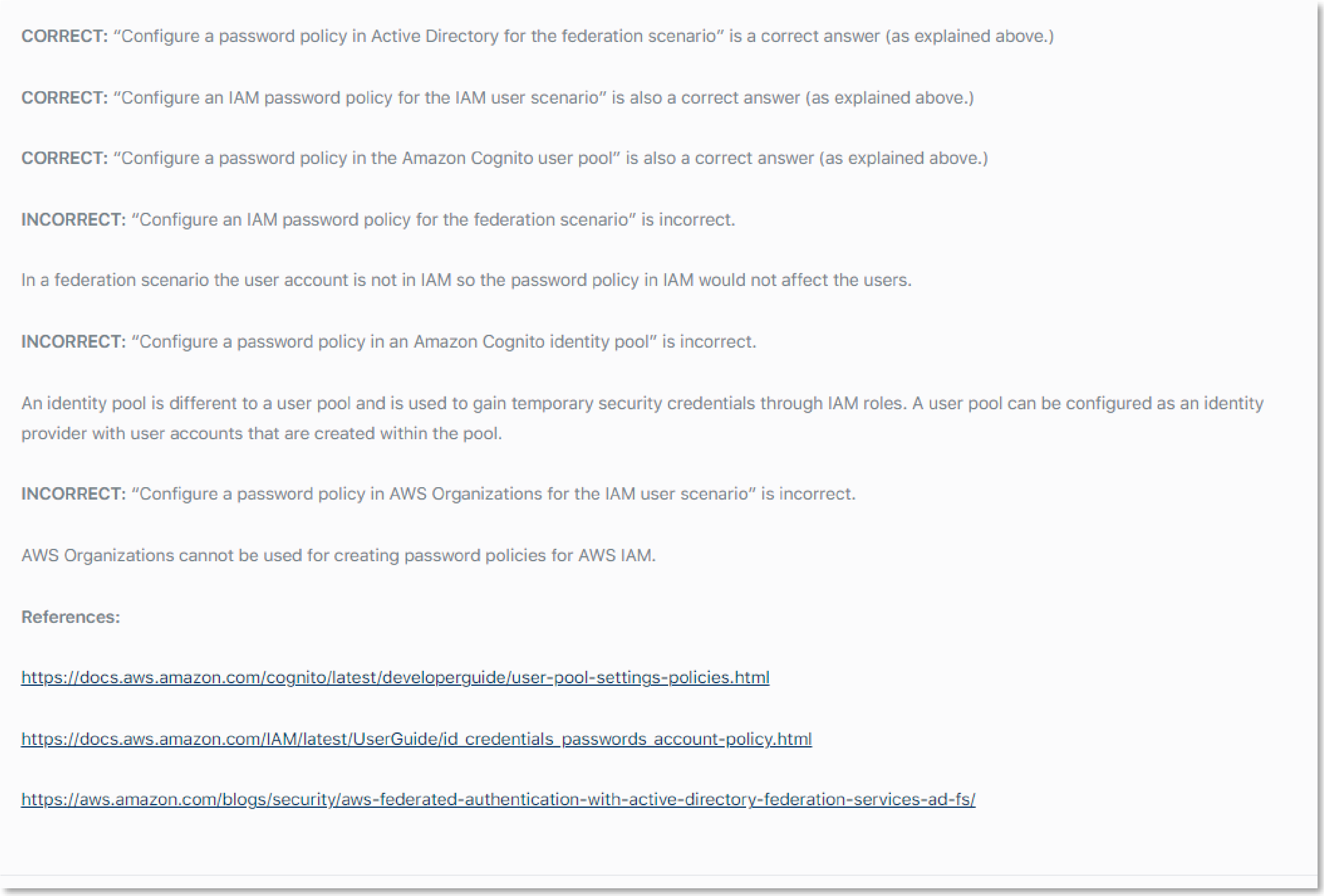


Incorrect

Explanation:

This question simply requires an understanding of where the relevant password policies should be configured:

- When using federation with Active Directory and AWS IAM the user account is created and managed in Active Directory so the password policy should be configured there.
- When using AWS IAM to create user accounts you can specify a password policy in IAM.
- When using Amazon Cognito user pools to create user accounts you can specify a password policy within the user pool.



CORRECT: "Configure a password policy in Active Directory for the federation scenario" is a correct answer (as explained above.)

CORRECT: "Configure an IAM password policy for the IAM user scenario" is also a correct answer (as explained above.)

CORRECT: "Configure a password policy in the Amazon Cognito user pool" is also a correct answer (as explained above.)

INCORRECT: "Configure an IAM password policy for the federation scenario" is incorrect.

In a federation scenario the user account is not in IAM so the password policy in IAM would not affect the users.

INCORRECT: "Configure a password policy in an Amazon Cognito identity pool" is incorrect.

An identity pool is different to a user pool and is used to gain temporary security credentials through IAM roles. A user pool can be configured as an identity provider with user accounts that are created within the pool.

INCORRECT: "Configure a password policy in AWS Organizations for the IAM user scenario" is incorrect.

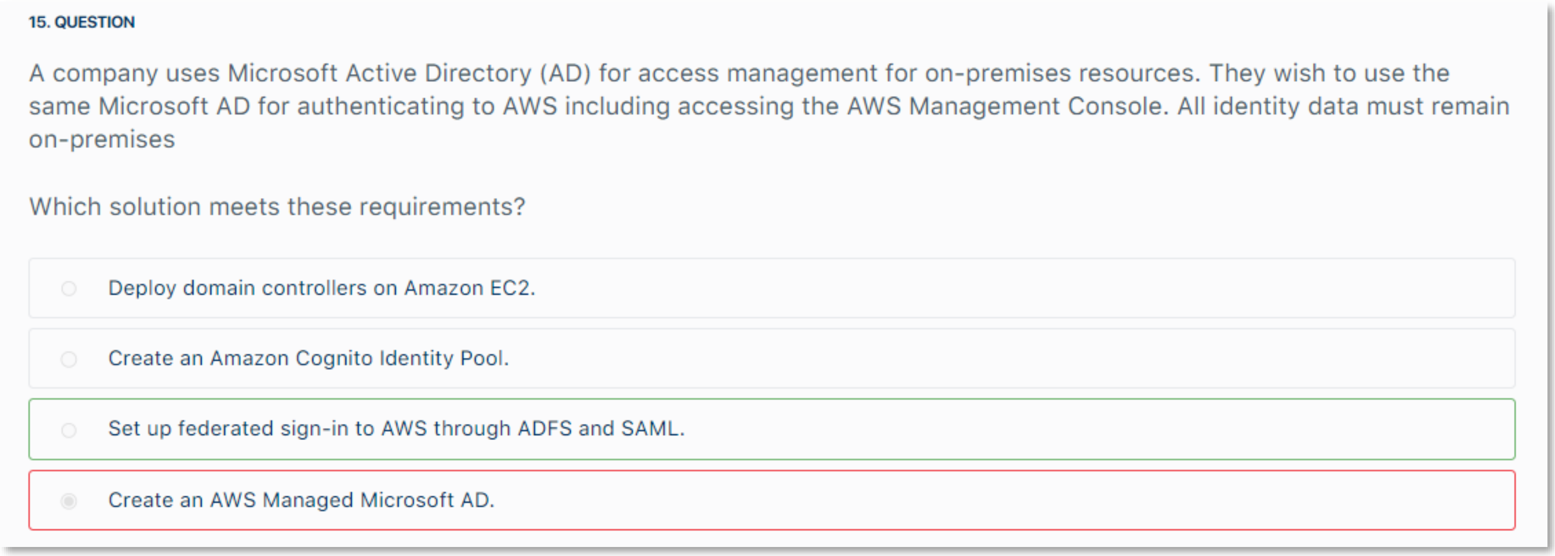
AWS Organizations cannot be used for creating password policies for AWS IAM.

References:

https://docs.aws.amazon.com/cognito/latest/developerguide/user-pool-settings-policies.html

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_passwords_account-policy.html

https://aws.amazon.com/blogs/security/aws-federated-authentication-with-active-directory-federation-services-ad-fs/



15. QUESTION

A company uses Microsoft Active Directory (AD) for access management for on-premises resources. They wish to use the same Microsoft AD for authenticating to AWS including accessing the AWS Management Console. All identity data must remain on-premises

Which solution meets these requirements?

○ Deploy domain controllers on Amazon EC2.

○ Create an Amazon Cognito Identity Pool.

○ Set up federated sign-in to AWS through ADFS and SAML.

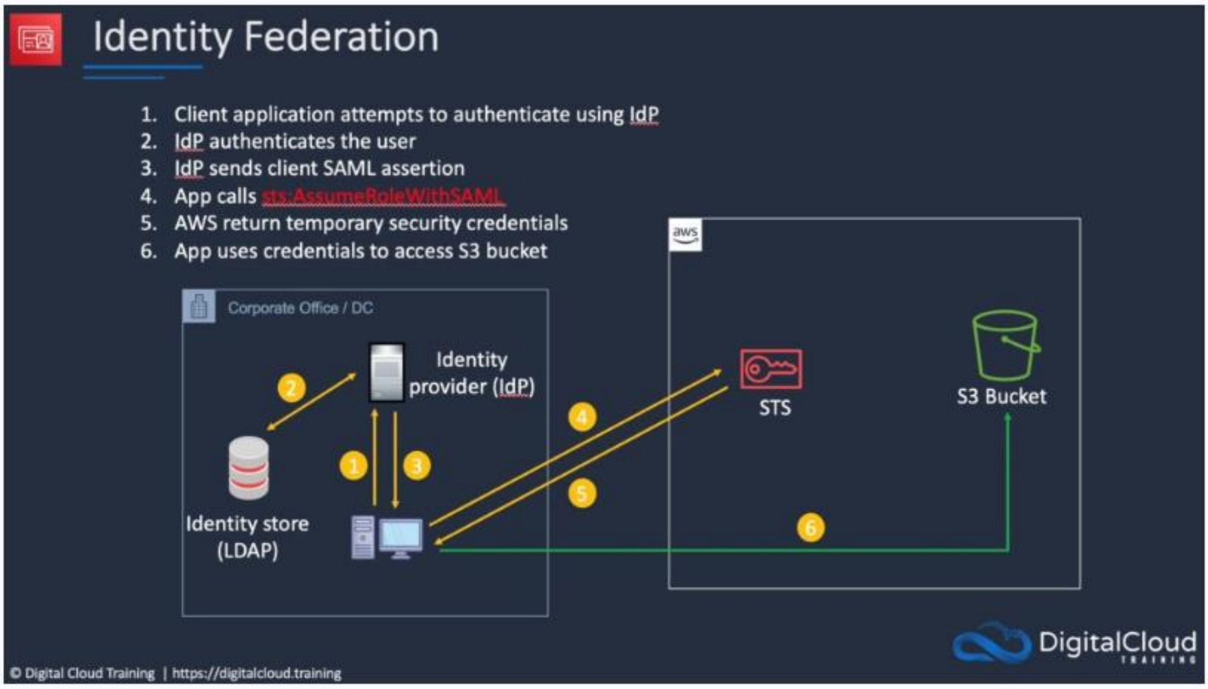
● Create an AWS Managed Microsoft AD.

Incorrect

Explanation:

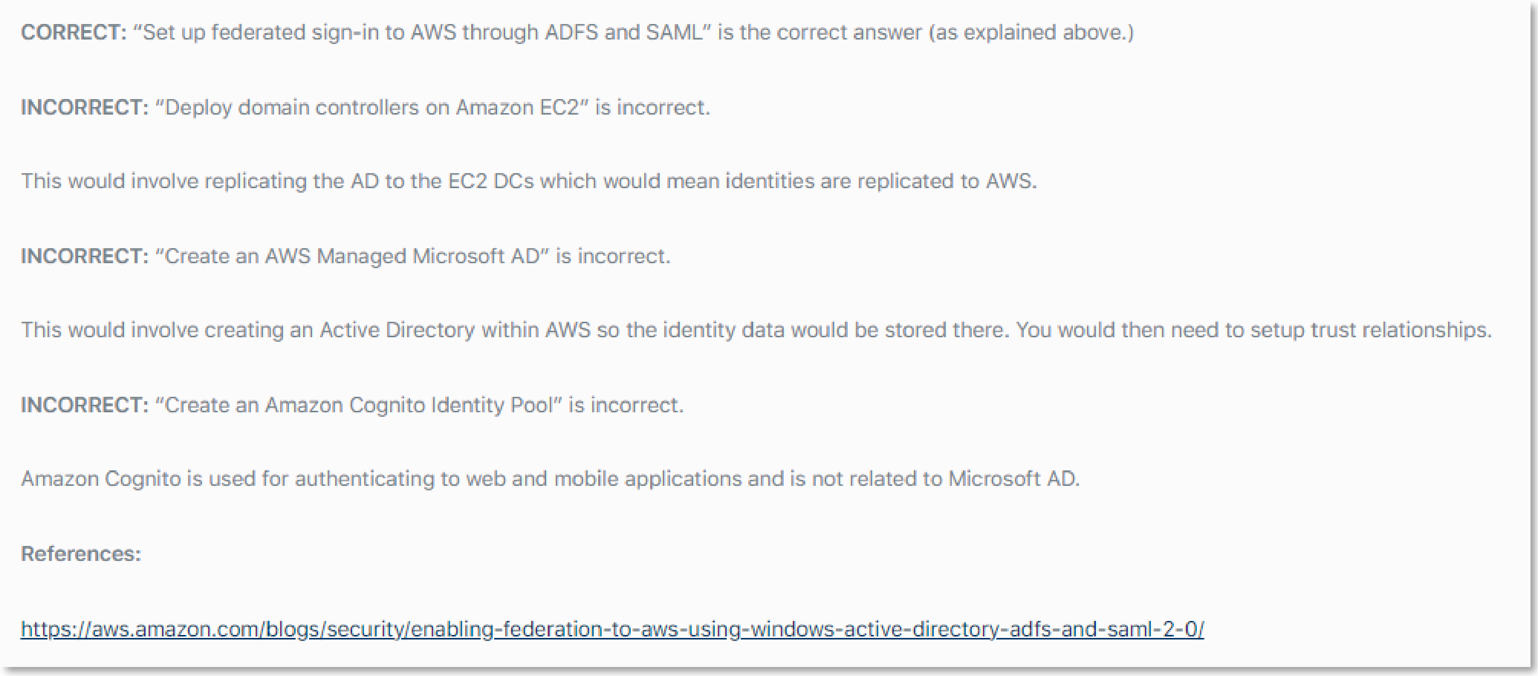
This solution uses federated single sign-on (SSO), which lets users sign into the AWS Management Console or make programmatic calls to AWS APIs by using assertions from a SAML-compliant identity provider (IdP) like ADFS. With this solution all identity data is stored in the on-premises directory and only authentication tokens are used within AWS. This meets the security requirements for the authentication solution.

The following image depicts the steps involved in identity federation:



Identity Federation

1. Client application attempts to authenticate using IdP
2. IdP authenticates the user
3. IdP sends client SAML assertion
4. App calls sts:AssumeRoleWithSAML
5. AWS return temporary security credentials
6. App uses credentials to access S3 bucket



CORRECT: “Set up federated sign-in to AWS through ADFS and SAML” is the correct answer (as explained above.)

INCORRECT: “Deploy domain controllers on Amazon EC2” is incorrect.

This would involve replicating the AD to the EC2 DCs which would mean identities are replicated to AWS.

INCORRECT: “Create an AWS Managed Microsoft AD” is incorrect.

This would involve creating an Active Directory within AWS so the identity data would be stored there. You would then need to setup trust relationships.

INCORRECT: “Create an Amazon Cognito Identity Pool” is incorrect.

Amazon Cognito is used for authenticating to web and mobile applications and is not related to Microsoft AD.

References:

https://aws.amazon.com/blogs/security/enabling-federation-to-aws-using-windows-active-directory-adfs-and-saml-2-0/

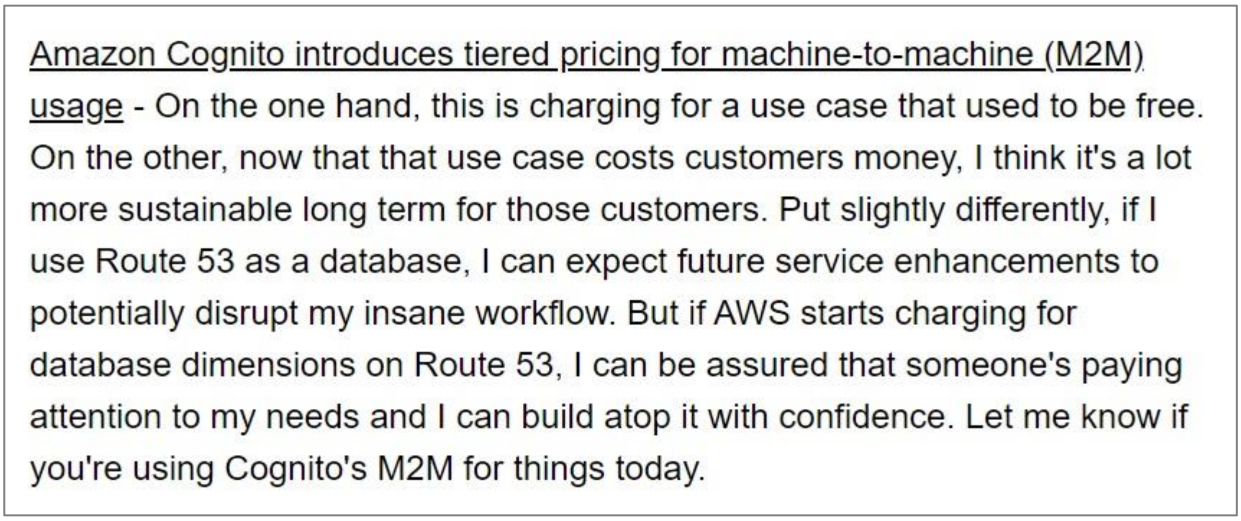
Amazon Cognito introduces tiered pricing for machine-to-machine (M2M) usage

Posted On: May 9, 2024

Amazon Cognito introduces pricing for machine-to-machine (M2M) authentication to better support continued growth and expand capabilities. There is no change to Amazon Cognito's user based pricing (monthly active users or MAUs). Customer accounts currently using Amazon Cognito for M2M use cases will be exempt from pricing for 12 months. M2M pricing is based on the number of application clients configured for M2M authentication and the number of tokens requested for them. You can find details on our pricing page.

Amazon Cognito makes it easier to add authentication, authorization, and identity management to your web and mobile apps. In addition to supporting human identities, Cognito's M2M authentication enables developers to leverage machine identities to secure interactions between their services or across organizations. Developers can define machine identities and generate OAuth 2.0 tokens to authenticate them using Cognito user pools that are configured with the OAuth 2.0 client credentials grant. This pricing change applies to only to user pools configured in this way and is not applicable to any other OAuth 2.0 flows.

Amazon Cognito is available in 29 AWS Regions globally. To learn more about Amazon Cognito’s support for OAuth 2.0 standards, visit the product documentation page. To get started, visit the Amazon Cognito home page.



Amazon Cognito introduces tiered pricing for machine-to-machine (M2M) usage - On the one hand, this is charging for a use case that used to be free. On the other, now that that use case costs customers money, I think it's a lot more sustainable long term for those customers. Put slightly differently, if I use Route 53 as a database, I can expect future service enhancements to potentially disrupt my insane workflow. But if AWS starts charging for database dimensions on Route 53, I can be assured that someone's paying attention to my needs and I can build atop it with confidence. Let me know if you're using Cognito's M2M for things today.

Corey Quinn in an email on May 13th 2024 [Quinn 2024]

Bibliography

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

I. Official

[Edward 2023]

Sun, Edward (2023). *AWS Security Blog*. Approaches for migrating users to Amazon Cognito user pools. Available at: <https://aws.amazon.com/blogs/security/approaches-for-migrating-users-to-amazon-cognito-user-pools/>

[AWS 2024]

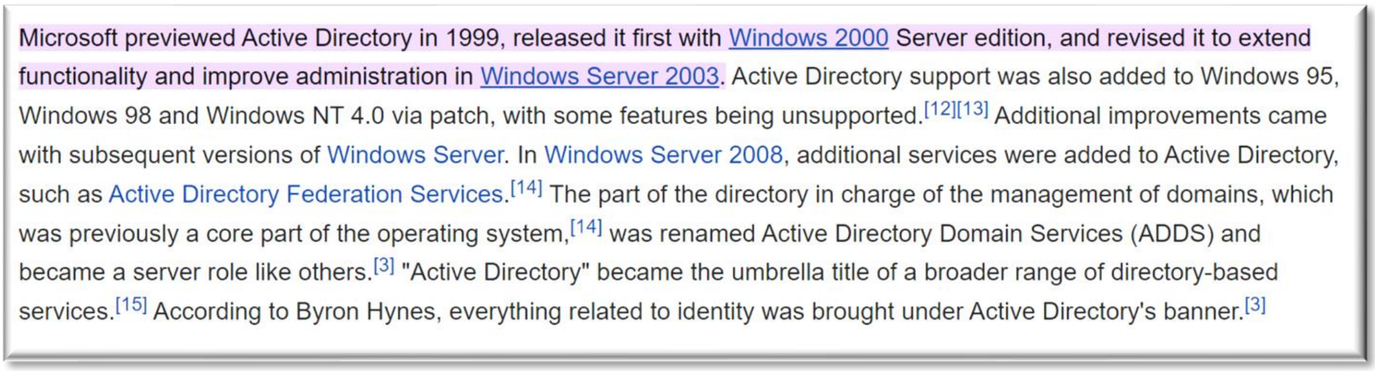
Amazon Cognito introduces tiered pricing for machine-to-machine (M2M) usage [Announcement]. Available at: < https://aws.amazon.com/about-aws/whats-new/2024/05/amazon-cognito-tiered-pricing-m2m-usage >

II. Unofficial

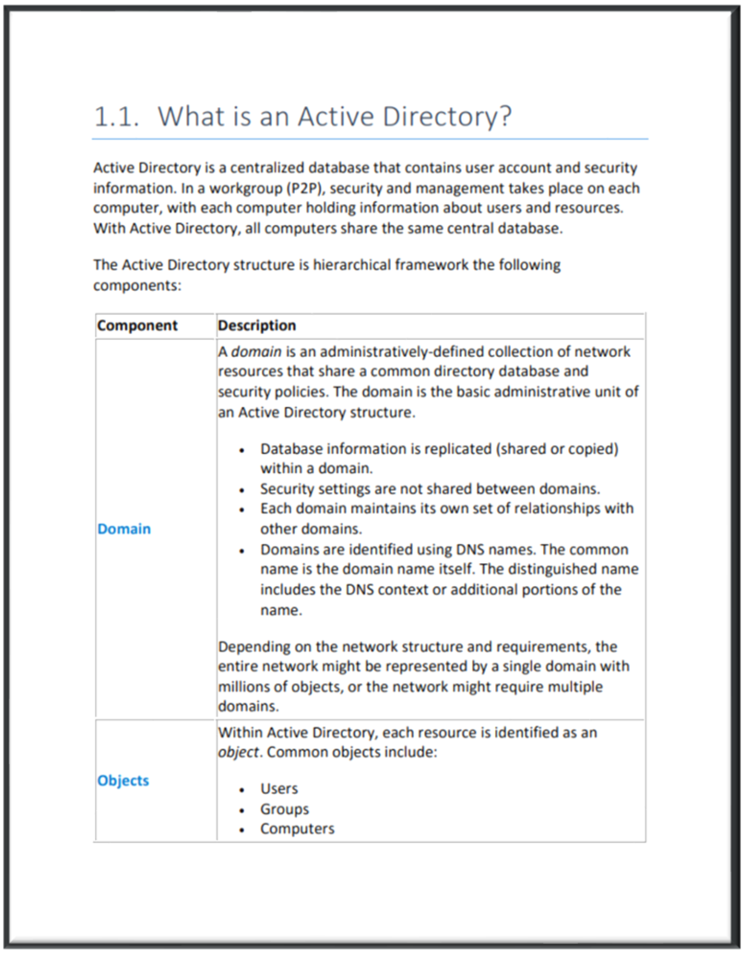
[Quinn 2024]

Quinn, Corey (2024). *Last Week in AWS* [Newsletter]. May 13th 2024.

AWS Directory Service



Microsoft previewed Active Directory in 1999, released it first with Windows 2000 Server edition, and revised it to extend functionality and improve administration in Windows Server 2003. Active Directory support was also added to Windows 95, Windows 98 and Windows NT 4.0 via patch, with some features being unsupported.[12][13] Additional improvements came with subsequent versions of Windows Server. In Windows Server 2008, additional services were added to Active Directory, such as Active Directory Federation Services.[14] The part of the directory in charge of the management of domains, which was previously a core part of the operating system,[14] was renamed Active Directory Domain Services (ADDS) and became a server role like others.[3] "Active Directory" became the umbrella title of a broader range of directory-based services.[15] According to Byron Hynes, everything related to identity was brought under Active Directory's banner.[3]



1.1. What is an Active Directory?

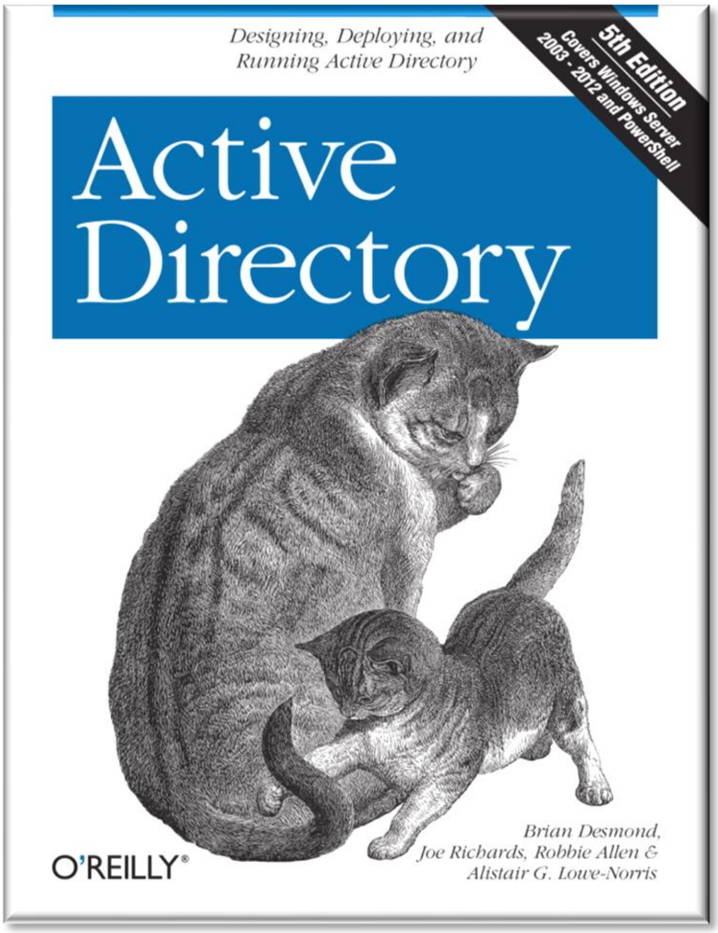
Active Directory is a centralized database that contains user account and security information. In a workgroup (P2P), security and management takes place on each computer, with each computer holding information about users and resources. With Active Directory, all computers share the same central database.

The Active Directory structure is hierarchical framework the following components:

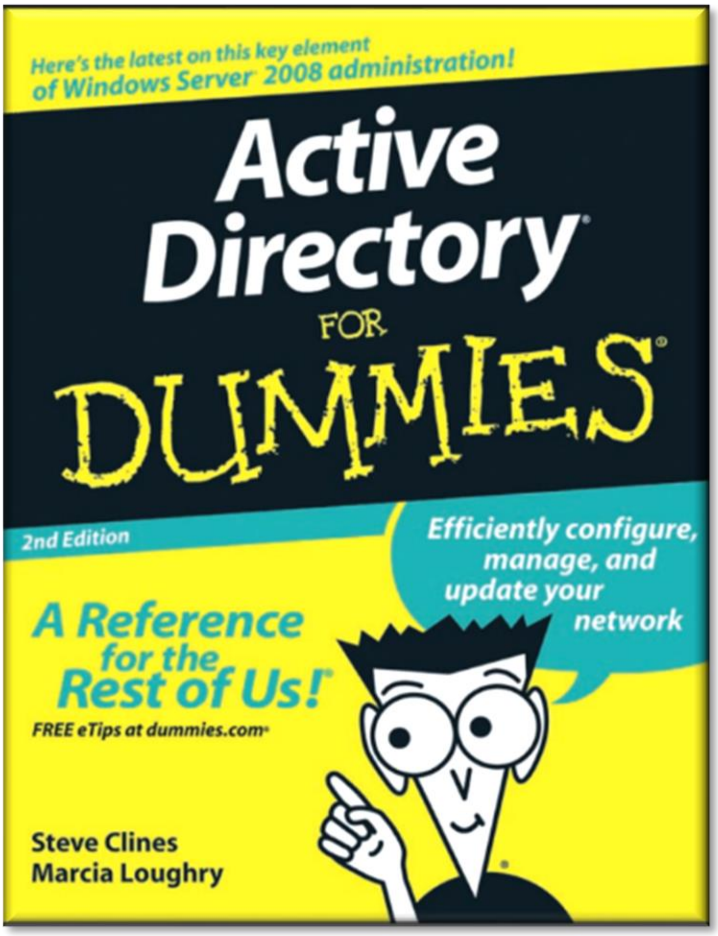
Component	Description
Domain	A *domain* is an administratively-defined collection of network resources that share a common directory database and security policies. The domain is the basic administrative unit of an Active Directory structure. • Database information is replicated (shared or copied) within a domain. • Security settings are not shared between domains. • Each domain maintains its own set of relationships with other domains. • Domains are identified using DNS names. The common name is the domain name itself. The distinguished name includes the DNS context or additional portions of the name. Depending on the network structure and requirements, the entire network might be represented by a single domain with millions of objects, or the network might require multiple domains.
Objects	Within Active Directory, each resource is identified as an *object*. Common objects include: • Users • Groups • Computers

This PDF provides an overview of the key terms.

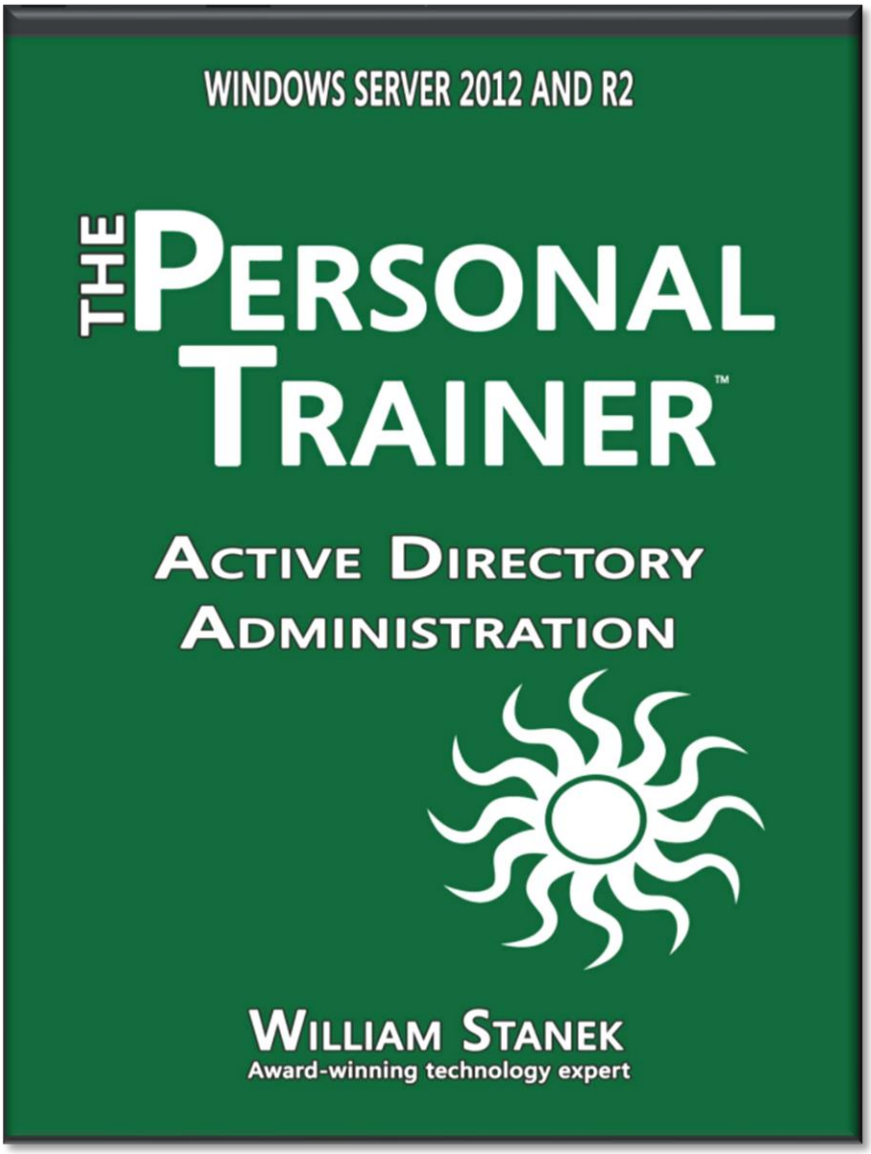
There are some excellent textbooks on Microsoft Active Directory available…



Designing, Deploying, and Running Active Directory
5th Edition
Covers Windows Server 2003 - 2012 and PowerShell
Active Directory
Brian Desmond, Joe Richards, Robbie Allen & Alistair G. Lowe-Norris
O'REILLY



Here's the latest on this key element of Windows Server 2008 administration!
Active Directory FOR DUMMIES
2nd Edition
Efficiently configure, manage, and update your network
A Reference for the Rest of Us!
FREE eTips at dummies.com
Steve Clines
Marcia Loughry



WINDOWS SERVER 2012 AND R2

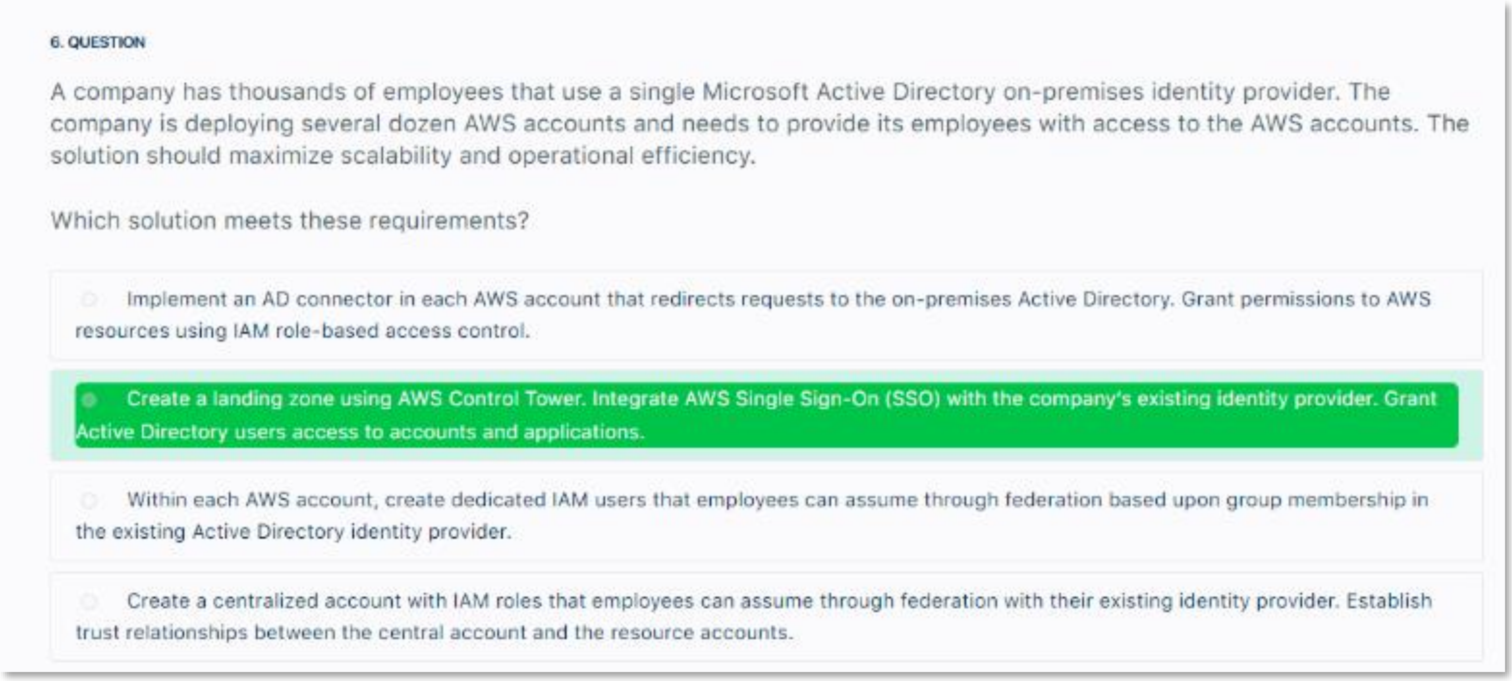
THE PERSONAL TRAINER™

ACTIVE DIRECTORY ADMINISTRATION

WILLIAM STANEK
Award-winning technology expert

“Eli the Computer Guy” introducing Microsoft Active Directory

Ron Cully explaining how AWS Directory Service works in a 2018 video released by Amazon Web Services



6. QUESTION

A company has thousands of employees that use a single Microsoft Active Directory on-premises identity provider. The company is deploying several dozen AWS accounts and needs to provide its employees with access to the AWS accounts. The solution should maximize scalability and operational efficiency.

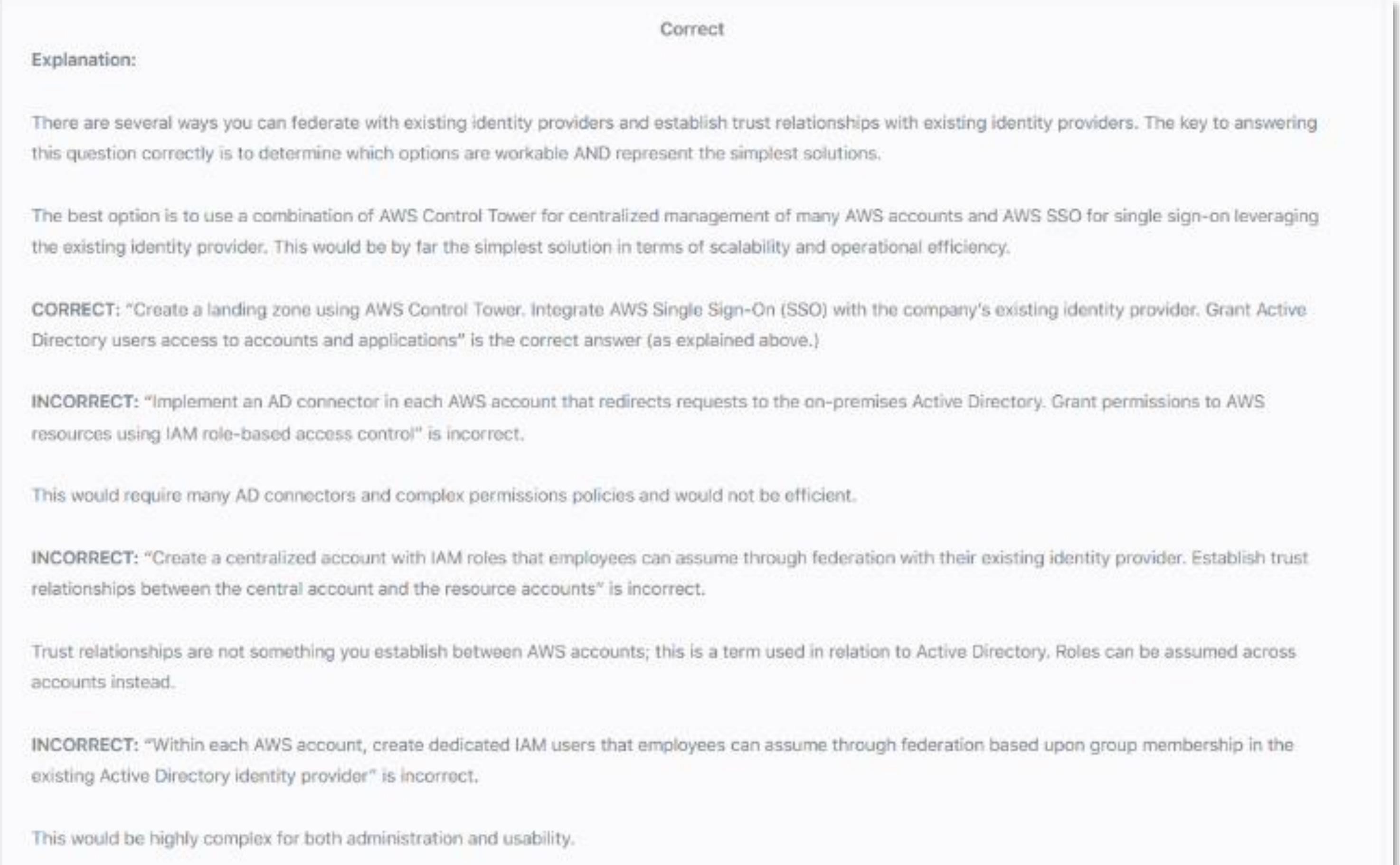
Which solution meets these requirements?

○ Implement an AD connector in each AWS account that redirects requests to the on-premises Active Directory. Grant permissions to AWS resources using IAM role-based access control.

● Create a landing zone using AWS Control Tower. Integrate AWS Single Sign-On (SSO) with the company's existing identity provider. Grant Active Directory users access to accounts and applications.

○ Within each AWS account, create dedicated IAM users that employees can assume through federation based upon group membership in the existing Active Directory identity provider.

○ Create a centralized account with IAM roles that employees can assume through federation with their existing identity provider. Establish trust relationships between the central account and the resource accounts.



Correct

Explanation:

There are several ways you can federate with existing identity providers and establish trust relationships with existing identity providers. The key to answering this question correctly is to determine which options are workable AND represent the simplest solutions.

The best option is to use a combination of AWS Control Tower for centralized management of many AWS accounts and AWS SSO for single sign-on leveraging the existing identity provider. This would be by far the simplest solution in terms of scalability and operational efficiency.

CORRECT: "Create a landing zone using AWS Control Tower. Integrate AWS Single Sign-On (SSO) with the company's existing identity provider. Grant Active Directory users access to accounts and applications" is the correct answer (as explained above.)

INCORRECT: "Implement an AD connector in each AWS account that redirects requests to the on-premises Active Directory. Grant permissions to AWS resources using IAM role-based access control" is incorrect.

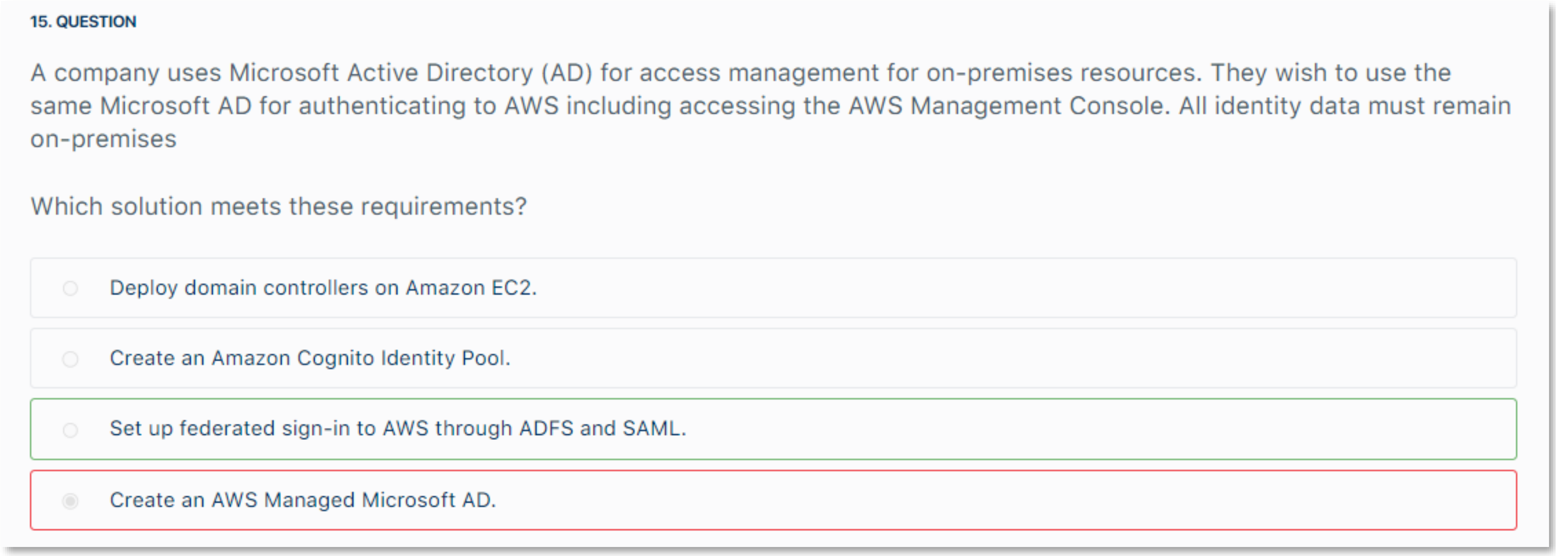
This would require many AD connectors and complex permissions policies and would not be efficient.

INCORRECT: "Create a centralized account with IAM roles that employees can assume through federation with their existing identity provider. Establish trust relationships between the central account and the resource accounts" is incorrect.

Trust relationships are not something you establish between AWS accounts; this is a term used in relation to Active Directory. Roles can be assumed across accounts instead.

INCORRECT: "Within each AWS account, create dedicated IAM users that employees can assume through federation based upon group membership in the existing Active Directory identity provider" is incorrect.

This would be highly complex for both administration and usability.



15. QUESTION

A company uses Microsoft Active Directory (AD) for access management for on-premises resources. They wish to use the same Microsoft AD for authenticating to AWS including accessing the AWS Management Console. All identity data must remain on-premises

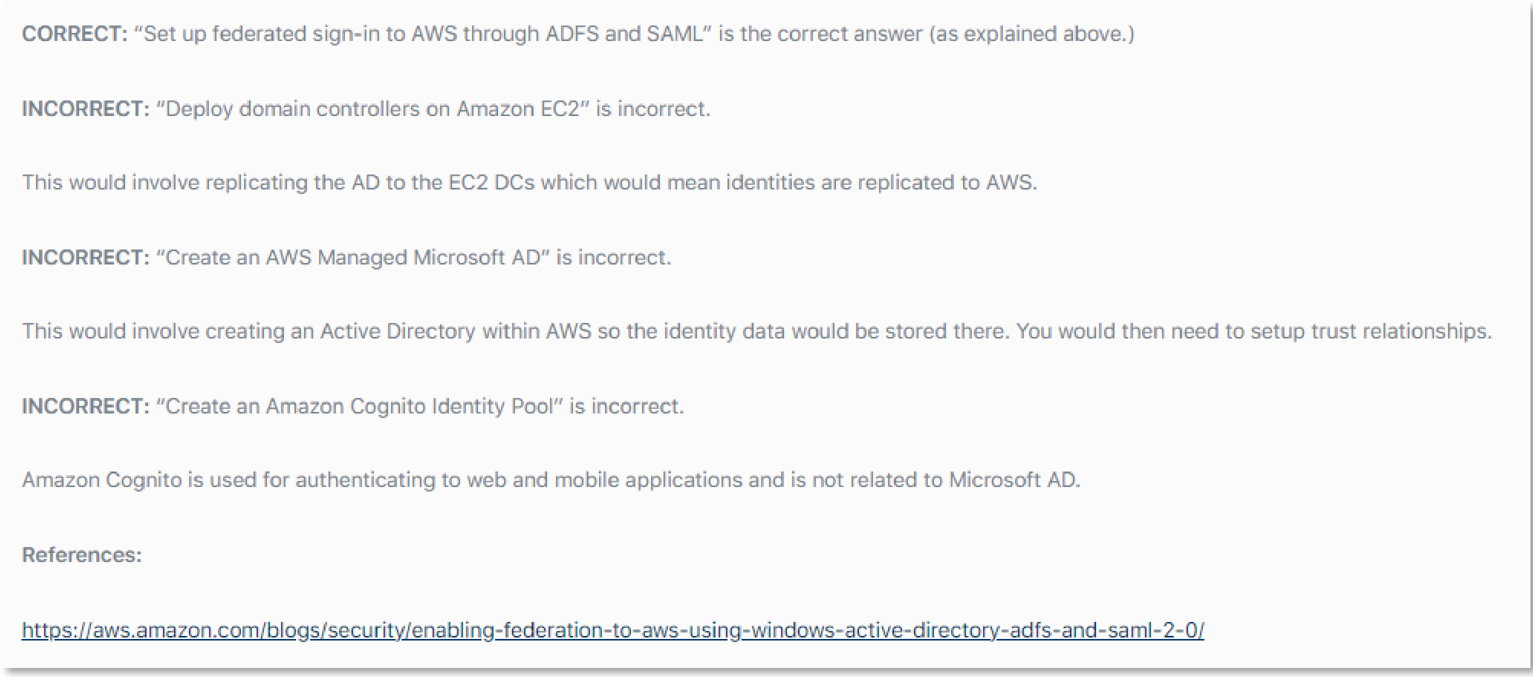
Which solution meets these requirements?

○ Deploy domain controllers on Amazon EC2.

○ Create an Amazon Cognito Identity Pool.

○ Set up federated sign-in to AWS through ADFS and SAML.

● Create an AWS Managed Microsoft AD.



CORRECT: "Set up federated sign-in to AWS through ADFS and SAML" is the correct answer (as explained above.)

INCORRECT: "Deploy domain controllers on Amazon EC2" is incorrect.

This would involve replicating the AD to the EC2 DCs which would mean identities are replicated to AWS.

INCORRECT: "Create an AWS Managed Microsoft AD" is incorrect.

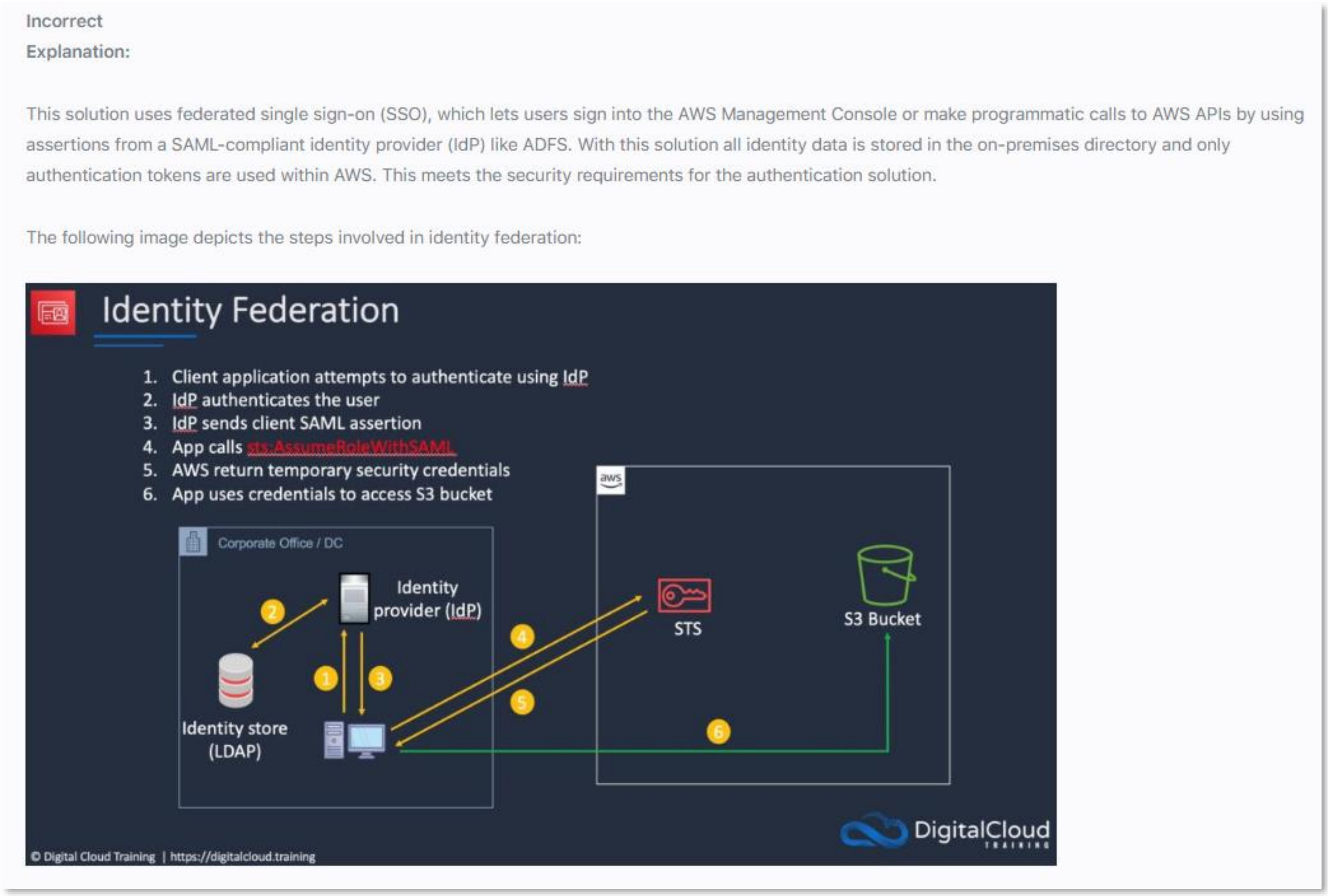
This would involve creating an Active Directory within AWS so the identity data would be stored there. You would then need to setup trust relationships.

INCORRECT: "Create an Amazon Cognito Identity Pool" is incorrect.

Amazon Cognito is used for authenticating to web and mobile applications and is not related to Microsoft AD.

References:

https://aws.amazon.com/blogs/security/enabling-federation-to-aws-using-windows-active-directory-adfs-and-saml-2-0/



Incorrect

Explanation:

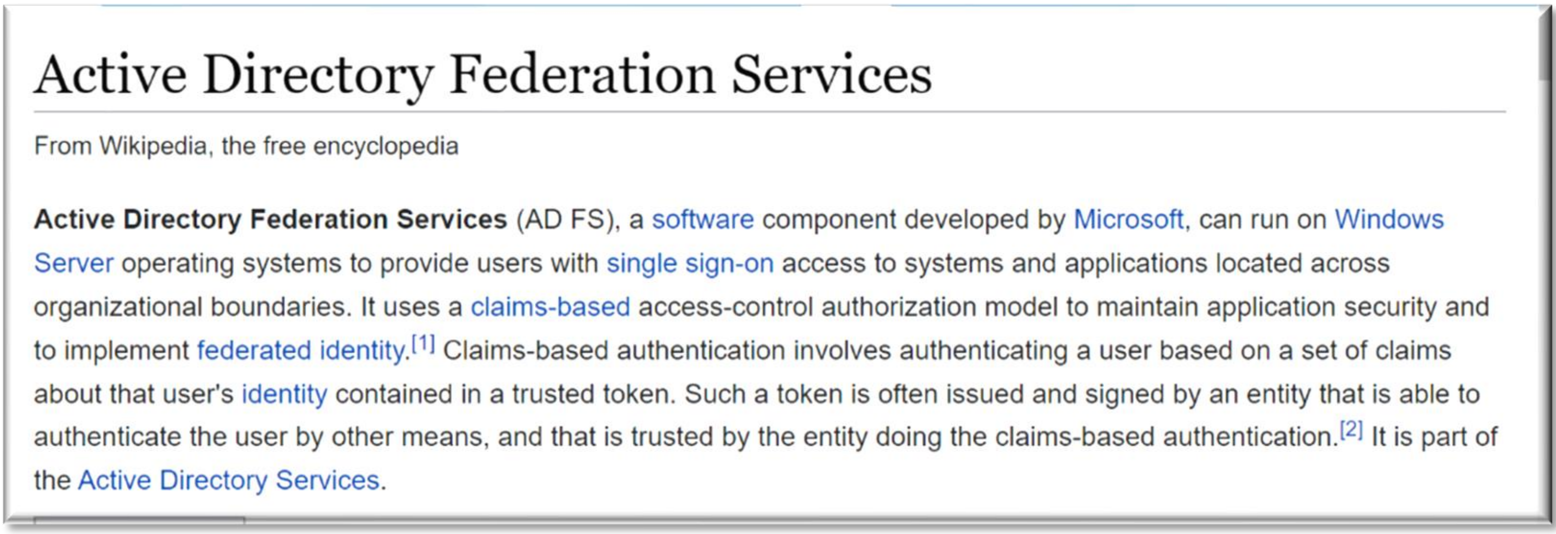
This solution uses federated single sign-on (SSO), which lets users sign into the AWS Management Console or make programmatic calls to AWS APIs by using assertions from a SAML-compliant identity provider (IdP) like ADFS. With this solution all identity data is stored in the on-premises directory and only authentication tokens are used within AWS. This meets the security requirements for the authentication solution.

The following image depicts the steps involved in identity federation:

Identity Federation

1. Client application attempts to authenticate using IdP
2. IdP authenticates the user
3. IdP sends client SAML assertion
4. App calls sts:AssumeRoleWithSAML
5. AWS return temporary security credentials
6. App uses credentials to access S3 bucket

© Digital Cloud Training | https://digitalcloud.training



Active Directory Federation Services

From Wikipedia, the free encyclopedia

Active Directory Federation Services (AD FS), a software component developed by Microsoft, can run on Windows Server operating systems to provide users with single sign-on access to systems and applications located across organizational boundaries. It uses a claims-based access-control authorization model to maintain application security and to implement federated identity.[1] Claims-based authentication involves authenticating a user based on a set of claims about that user's identity contained in a trusted token. Such a token is often issued and signed by an entity that is able to authenticate the user by other means, and that is trusted by the entity doing the claims-based authentication.[2] It is part of the Active Directory Services.

Issue with AWS Directory Service EnableRoleAccess

Initial Publication Date: 06/14/2023 4:30PM PDT

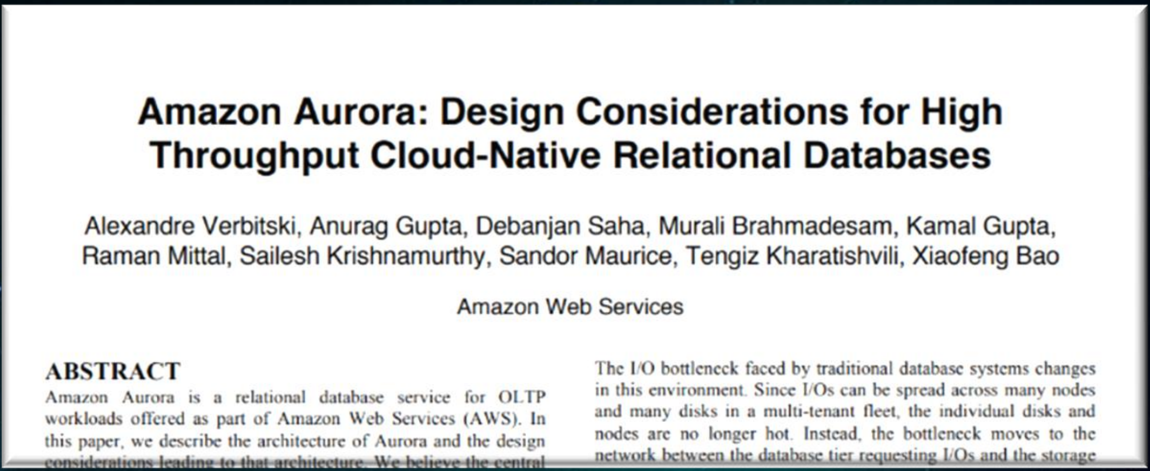
A researcher recently reported an issue in AWS Directory Service which would have enabled customer’s IAM principals, who are allowed to call the “EnableRoleAccess” API, to enable role access on the directory user even if that IAM principal did not have the “iam:passrole” permission. This specific issue would only occur if the calling IAM principal had permissions to call “EnableRoleAccess” API and would be limited to the customer’s account.

The issue has been remediated by enforcing the requirement to have IAM “iam:passrole“ permission in order to enable role access in addition to having IAM permissions to call the ”EnableRoleAccess“ API. Customers using the recommended policy for the feature would not have been impacted by this issue and no customer action is required.

We like to thank Cloudar Security for responsibly disclosing this issue and working with us on its resolution. Security-related questions or concerns can be brought to our attention via aws-security@amazon.com.

Aurora

The engine that ROARS



Amazon Aurora: Design Considerations for High Throughput Cloud-Native Relational Databases

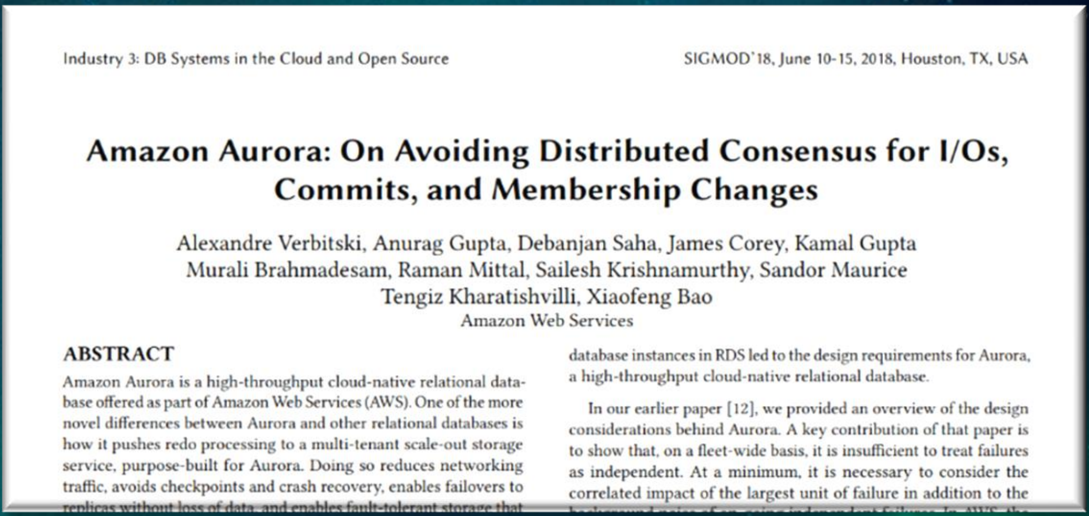
Alexandre Verbitski, Anurag Gupta, Debanjan Saha, Murali Brahmadesam, Kamal Gupta, Raman Mittal, Sailesh Krishnamurthy, Sandor Maurice, Tengiz Kharatishvili, Xiaofeng Bao

Amazon Web Services

ABSTRACT

Amazon Aurora is a relational database service for OLTP workloads offered as part of Amazon Web Services (AWS). In this paper, we describe the architecture of Aurora and the design considerations leading to that architecture. We believe the central

The I/O bottleneck faced by traditional database systems changes in this environment. Since I/Os can be spread across many nodes and many disks in a multi-tenant fleet, the individual disks and nodes are no longer hot. Instead, the bottleneck moves to the network between the database tier requesting I/Os and the storage



Amazon Aurora: On Avoiding Distributed Consensus for I/Os, Commits, and Membership Changes

Alexandre Verbitski, Anurag Gupta, Debanjan Saha, James Corey, Kamal Gupta
Murali Brahmadesam, Raman Mittal, Sailesh Krishnamurthy, Sandor Maurice
Tengiz Kharatishvilli, Xiaofeng Bao
Amazon Web Services

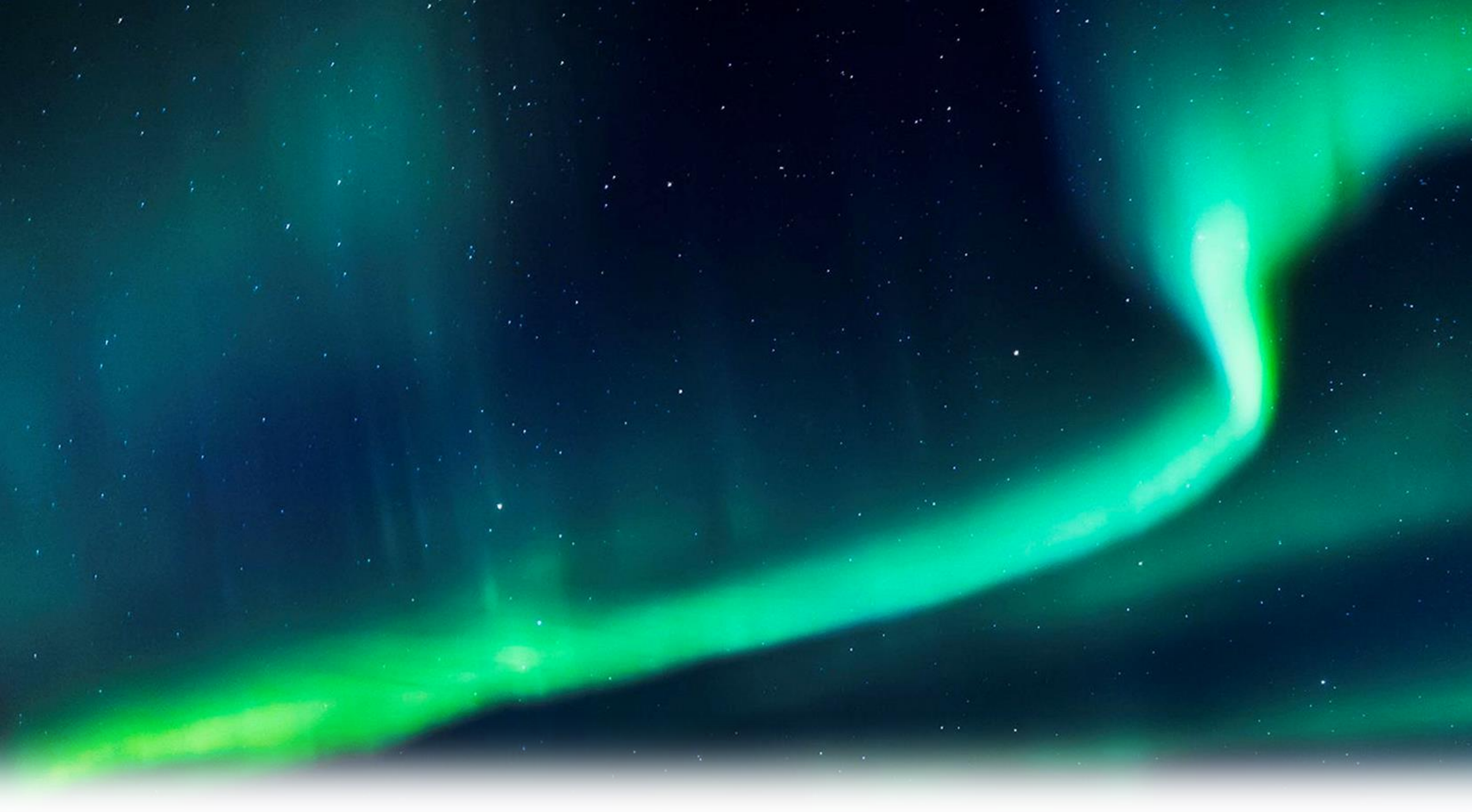
ABSTRACT

Amazon Aurora is a high-throughput cloud-native relational database offered as part of Amazon Web Services (AWS). One of the more novel differences between Aurora and other relational databases is how it pushes redo processing to a multi-tenant scale-out storage service, purpose-built for Aurora. Doing so reduces networking traffic, avoids checkpoints and crash recovery, enables failovers to replicas without loss of data, and enables fault-tolerant storage that

database instances in RDS led to the design requirements for Aurora, a high-throughput cloud-native relational database.

In our earlier paper [12], we provided an overview of the design considerations behind Aurora. A key contribution of that paper is to show that, on a fleet-wide basis, it is insufficient to treat failures as independent. At a minimum, it is necessary to consider the correlated impact of the largest unit of failure in addition to the

Amazon Aurora



Aurora featured in Roman mythology. She was a goddess intended to represent the Dawn, or Sun. Think of Ancient Rome. The Ancient Roman Empire was on an immensely large scale. Individual soldiers were often dispensable. Roman law was one of the great legal systems. Votes would be taken—but they had to reach quorum. In other words, a certain proportion of those present had to support the proposal.

Aurora was mentioned by the Roman poet called Virgil, and also the Roman poet Ovid, who came later.

The most famous story of Aurora is in fact adopted from the Greeks.

Being a goddess, Aurora was immortal. However, she fell in love with a man called Tithonus, who was mortal. Tithonus was Prince of Troy. Now, wanting to be with Tithonus for ever, Aurora tried to arrange things so that Tithonus lived forever. She simply went to Jupiter, the chief god in the Roman state religion, and asked it to be so. However, Aurora forgot to ask that Tithonus have eternal youth. Tithonus therefore continued to age, forever. Eventually, Aurora decided to turn the aged man into a cicada.

Because Aurora represents the dawn, she is sometimes used in literature as a metaphor for the early morning Sun. Shakespeare mentions Aurora in his play Romeo and Juliet:

> But all so soon as the all-cheering sun
> Should in the furthest east begin to draw
> The shady curtains from Aurora's bed,
> Away from the light steals home my heavy son...

Aurora Global Database

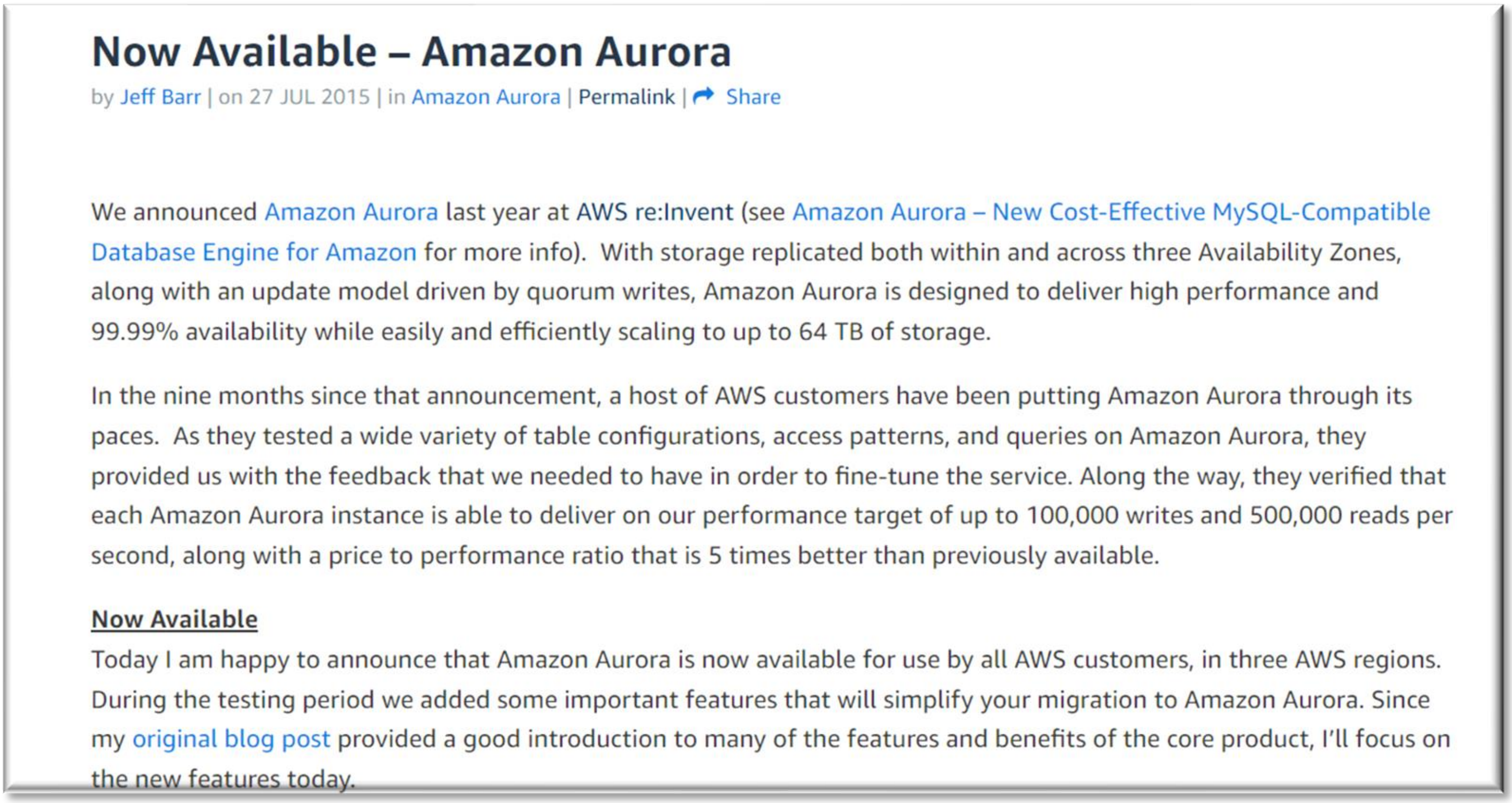
MySQL Compatibility mode

Aurora

Asynchronous

Amazon Aurora and MySQL

It's fair to say that Amazon Aurora has a special relationship with MySQL. In the original announcement post, from Jeff Barr, he mentions only MySQL. There is no mention of Oracle, MicrosoftSQLServer, or PostgreSQL.



Now Available – Amazon Aurora

by Jeff Barr | on 27 JUL 2015 | in Amazon Aurora | Permalink | Share

We announced Amazon Aurora last year at AWS re:Invent (see Amazon Aurora – New Cost-Effective MySQL-Compatible Database Engine for Amazon for more info). With storage replicated both within and across three Availability Zones, along with an update model driven by quorum writes, Amazon Aurora is designed to deliver high performance and 99.99% availability while easily and efficiently scaling to up to 64 TB of storage.

In the nine months since that announcement, a host of AWS customers have been putting Amazon Aurora through its paces. As they tested a wide variety of table configurations, access patterns, and queries on Amazon Aurora, they provided us with the feedback that we needed to have in order to fine-tune the service. Along the way, they verified that each Amazon Aurora instance is able to deliver on our performance target of up to 100,000 writes and 500,000 reads per second, along with a price to performance ratio that is 5 times better than previously available.

Now Available

Today I am happy to announce that Amazon Aurora is now available for use by all AWS customers, in three AWS regions. During the testing period we added some important features that will simplify your migration to Amazon Aurora. Since my original blog post provided a good introduction to many of the features and benefits of the core product, I'll focus on the new features today.

Amazon Aurora
MySQL

`

Single Master

Master Vs Primary

There is a confusion which you can sometimes get into. Let me first try to describe it. The thought runs that there are two kinds of terms. There are terms that describe the archetype, or original. Then, there are other terms which describe copies, or things which are in some way subservient to the original. So, we have just two things: ORIGINAL and SUBSERVIENT.

Examples of the ORIGINAL sort of terms include: primary, master. Examples of the SUBSERVIENT sort of term include: replica, standby. Sometimes you get view only these two categories—master and subservient—and pay no attention to which particular term is used. For example, whether you opt for replica or standby doesn't really matter.

However, this distinction is too coarse. In fact, one should pay attention to which particular term is chosen. Let's start with the word "primary". This word is used in discussions about the "Multi-AZ" feature available in RDS. It denotes the main instances, of course, and is contrasted with the standby instances. Notice how the word "primary" is not used in discussions of "read replicas", even though this discussion has the contrast between some original and many subservient instances. So, it's helpful to think of "primary" as having connotations related to failover. To call an instance the "primary" is to say it is the "thing replaced upon failure". We might be replicating the data on this primary to elsewhere, but if we were making a point about *replication*, we would probably opt for "master" instead of "primary".

How am I to remember this? How am I to remember that the word "primary" is about failover? We *could*, after all, have chosen "primary" to denote the whole idea of improving performance by having multiple read replicas. But we didn't—we use the term "master" for that. So, how do I remember this arbitrary fact?

Well, the prim**ary** instance is w**ary**. To be wary is to feel or show caution about possible dangers or problems. The fact that "primary" and "wary" both end with "ary" has proved a helpful heuristic. When we denote an instance the primary, we mean it is not the *standby*. In this context, we are thinking about failing over, to maintain the availability of our architecture, when things fail. For this reason, I associated "PRIMARY" with being wary, and thinking about how we will failover. Discussions of the "multi-AZ" feature of RDS use the term PRIMARY.

To call something a "standby" is to say that it is ready to step in. In military contexts, we might say that a battalion is on standby. This means that the battalion is ready to assist if called upon. Similarly, standby instances are ready to step in if the primary instance fails. To be clear, you will see the term "STANDBY" in discussions about failover (and maintaining availability); it is *not* chiefly used in optimizing performance (i.e. making things better, when nothing has failed). As Piper and Clinton note on page 147:

> the standby instance is not a read replica and cannot serve read traffic.

What follows from calling something a replica? Generally, to be a replica is to be an exact copy of something. In the context of Amazon RDS, to be a replica is to exactly have all the capabilities and habits of the original. The replica is really running, and really raising the game. Remove the idea that the replica is off-stage somewhere, like soldiers hidden in a garrison on standby. Replicas are *r*eally *r*unning. In RDS, a specific type of replicas called READ replicas, improve the performance of the database. They can only deal with read operations, and cannot deal with write operations, and this is why they are called "read replicas".

Now, we should accept that these terms—replica, standby—are somewhat arbitrary. It is true that if you are a soldier stationed in a garrison, ready to step in, you need to be capable. You must have so many capabilities, that you are essentially a replica of the original whose aid you are called to. So, calling the entity that we failover to a "replica" is not entirely unreasonable. Similarly, the "read replicas" within Amazon RDS clearly improve the original in some way, and they do stand by it. So, to call these things "standbys" would not be entirely unreasonable. Our designations are therefore somewhat arbitrary.

For this reason, I emphasise that **replicas are running**, readily reaching new performance heights. They help with performance. They are *not* the guys to turn to when things go wrong. The saints who step in are the standbys. They are on standby for when things fail. Standbys are things which can be failed over *to*.

The term we use for an entity (server etc) capable of being replaced immediately upon its failure is PRIMARY. We are W**ARY** of the PRIM**ARY** failing. When we're interested in making things go FASTER, we talk of the MASTER. In summary, my thesis is that we have these two contexts, performance (or replication) and failover. Each context has a pair of terms appropriate to it:

	Discussion about **PERFORMANCE** 	Discussion about **FAILOVER**
Original	Master	Primary
Subservient	(Read) replica	Standby

This thesis finds support from Piper and Clinton's (2021) Sybex textbook. On page 145, there is a page entitled "Read Replicas". The words "primary" or "standby" do not appear once on this page, because this is a discussion about optimizing performance, not failover.

They talk about masters, writing:

> A read replica takes some of the query load off the *master database instance*, which remains solely responsible for writing data to the database.

p.145

And also:

> If the master database instance fails before the replication is complete, you'll lose the unsynchronized data.

Ibid.

Later, the discussion turns to the "Multi-AZ" feature. Now, the terms "primary" and "standby" are used:

> In a multi-AZ deployment, you have a primary database instance in one availability zone that handles reads and writes to the database, and you have a standby database instance in a different availability zone.

p. 146

The "Multi-AZ" feature helps with *things going wrong*. The emphasis is not so much on optimization when *things are going well*.

I'm sure there is some crude way to remember this vividly—such as picturing a soldier in a garrison eating a bowl of "Multi-hoops" cereal as he waits to be called into action—but I will not attempt that here.

My thesis does not always stand. For example, there are two ways to implement this "Multi-AZ" feature with Amazon Aurora. The first is called "Single-Master" and the second "Multi-Master". This is problematic, because my thesis stated that "master" was a term reserved for performance discussions. And yet here it is, in a discussion about *failover* (specifically, the Multi-AZ feature).

How do we respond to this? First, note the idea here. With Single-Master, there is just one instance capable of performing writes. If it fails, we must failover to some other instance. With Multi-Master, multiple instances are capable of performing writes. If one of them fails, well, this doesn't really matter. It's not really the case that we have to fail over. We still have at least one master. To failover, there needs to be a failure, and there hasn't been a failure here. So, that is the idea behind these two options: "Single-Master" and "Multi-Master". To be a master, it is necessary to be capable of performing writes.

Consider this thought experiment. Suppose AWS called it "Single-Primary" and "Multi-Primary". Why would this be problematic? Firstly, the notion of "multi-primary" is incoherent. If something is primary, it is a unitary thing. It is incoherent to talk of multiple primaries. Prime numbers cannot be divided (except by themselves and 1) and Prime Ministers are single things. It follows that "Single Primary" is also a silly expression, akin to "three-sided triangle".

The above thought experiment reveals that the notion of PRIMARY can be analysed into two ideas. First, the idea of a singular unit. Second, that its failure demands that another entity carry on the work (that we failover). For an instance to be PRIMARY is for it to be capable of failing as a unit, such that the unit needs to be replaced. It is a singular unit and we ought to ensure it can fail over to another thing.

With "Single-Master" mode on Amazon Aurora, we have a master (i.e. something which is used as an original, in a replication process). Recall the two elements involved in being a PRIMARY: (1) singularity and (2) failure of a kind which demands another entity carry on the work. Well, these two elements are present with our single master (our lone instance capable of writing). For it is a unit, and its failure demands that a distinct entity carry on its work.

It therefore turns out that the "Single-Master" found within Aurora, *is* a primary instance. It is perfectly acceptable that something be a master and a primary instance at the same time. This explains why Piper and Clinton write:

> An Amazon Aurora single-master cluster **consists of** a primary instance.

What they're basically saying there is that this master (thing-from-which-we-replicate) is both singular and demanding that someone step in upon its failure. Again, it's not appropriate to call it "single-primary" because singularity (being unit) is what it *means* to be primary.

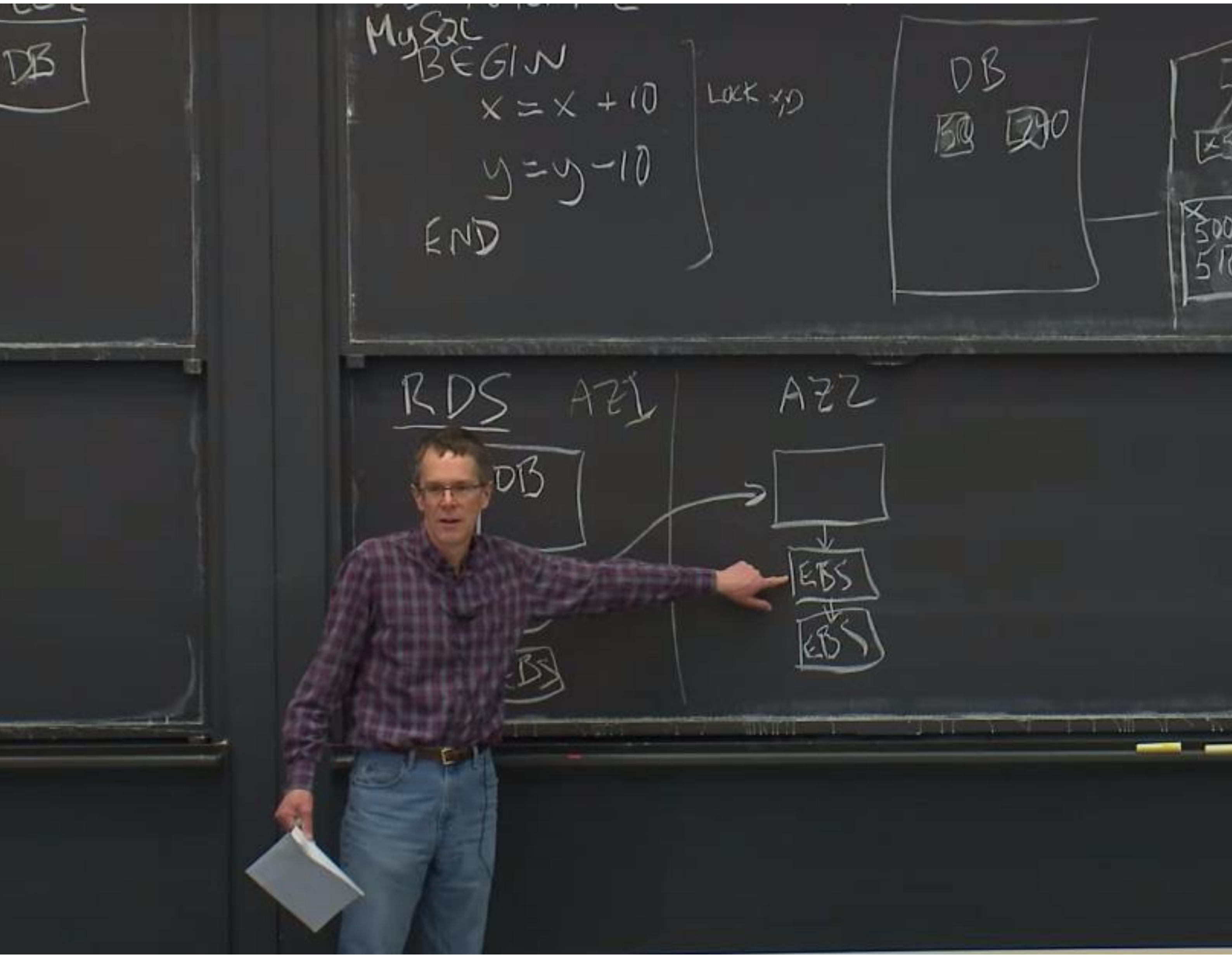
In "multi-master" mode, we have multiple masters. You *could* say that collectively, these masters constitute a primary, but this would be misleading, because they do not fail as a unit. Each master fails independently. (Also, it is simply misleading since "primary" conventionally denotes a individual instance). The important point is that the two elements involved in being PRIMARY are not present here. Certainly, masters within a "multi-master" deployment are capable of failing, in one go. So the first element is present. However, because there are multiple masters, their failure does not demand that another instance carry on its work. So, the second element is *absent.* Therefore, we should not point to an individual instance, within a "multi-master" deployment, and say "primary instance". It is not primary.

Piper and Clinton make this point in a similar way:

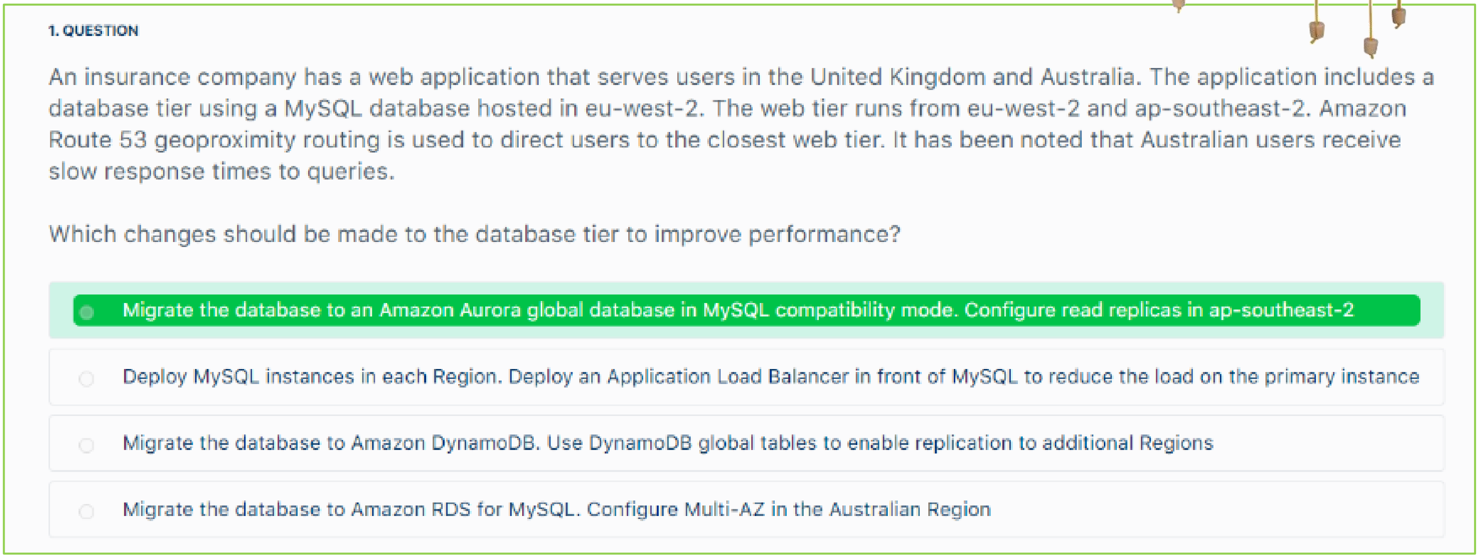
> In a multi-master cluster, all instances can write to the database. Thus, when one instance fails, **no failover occurs** because all instances can continue to write to the shared cluster volume that stores the database.
>
> Amazon refers to this as continuous availability, rather than high availability, because as long as at least one database instance is running, you can read from and write to the database.

p. 147

My thesis in fact remains intact. "Single-Master" *does* involve a primary instance. The only reason we avoid the actual word "primary" is that talking of a "Single-Primary" would be repetitive, akin to a "three-sided triangle". The fact is that an entity can be both a master and a primary. It can be an original, which is replicated, *and also* fail in a way that demands another entity step in. In fact the "Single-Master" instance which is found in a “Single-Master” Aurora deployment is of this nature. We call this primary “Single-*Master*” to contrast it with “Multi-Master”. The aggregation of masters in an Aurora “Multi-Master” deployment do not fail as a unit. They are not unit. Their multiplicity provides redundancy. When one of these masters fails, it’s not correct to say that failover occurs. Therefore, no failover occurs, and so we do not use the term “primary”. My thesis stands.



Robert Morris—famous for the Morris worm—explaining how Amazon Aurora works in one of his lectures at MIT



1. QUESTION

An insurance company has a web application that serves users in the United Kingdom and Australia. The application includes a database tier using a MySQL database hosted in eu-west-2. The web tier runs from eu-west-2 and ap-southeast-2. Amazon Route 53 geoproximity routing is used to direct users to the closest web tier. It has been noted that Australian users receive slow response times to queries.

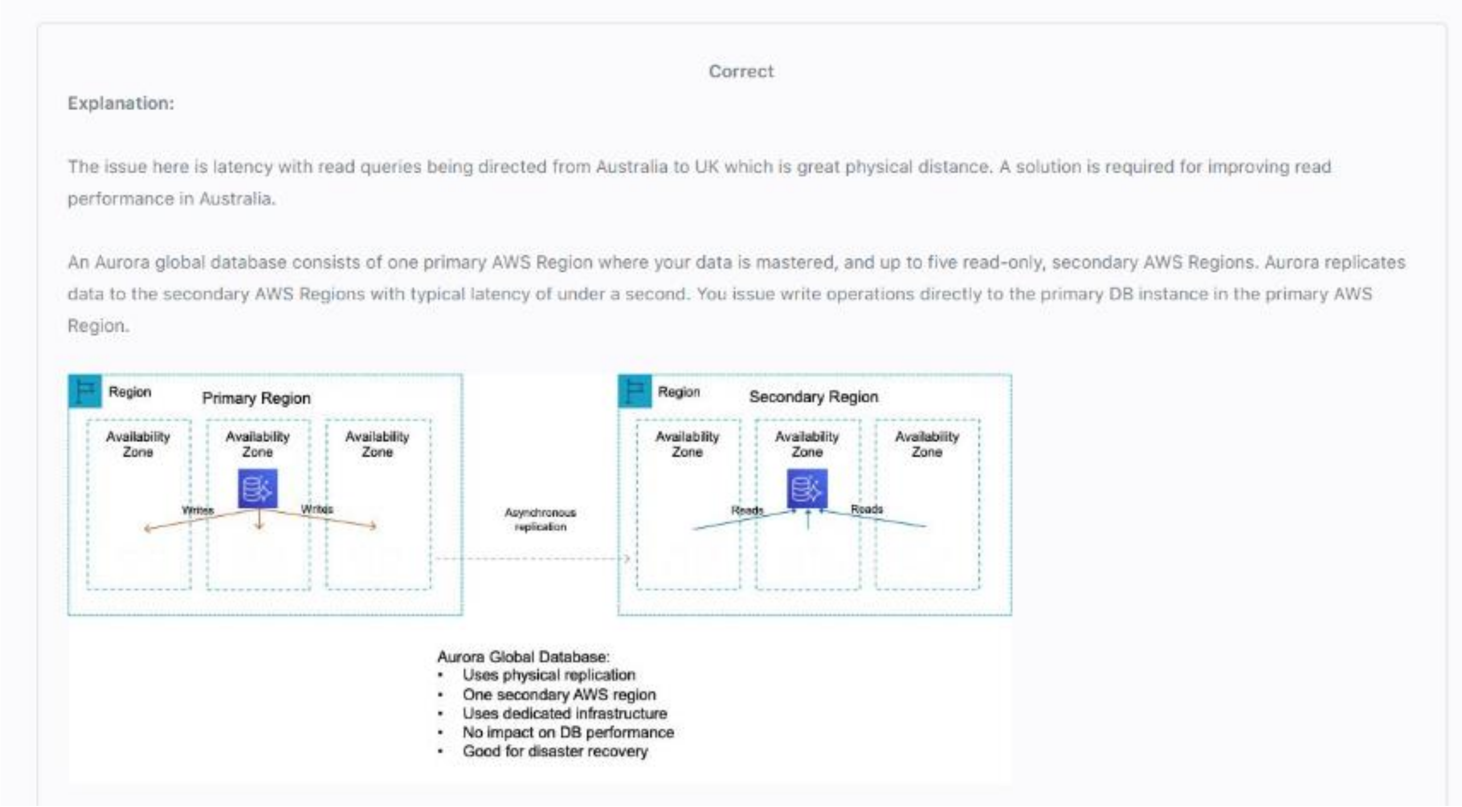
Which changes should be made to the database tier to improve performance?

● Migrate the database to an Amazon Aurora global database in MySQL compatibility mode. Configure read replicas in ap-southeast-2

○ Deploy MySQL instances in each Region. Deploy an Application Load Balancer in front of MySQL to reduce the load on the primary instance

○ Migrate the database to Amazon DynamoDB. Use DynamoDB global tables to enable replication to additional Regions

○ Migrate the database to Amazon RDS for MySQL. Configure Multi-AZ in the Australian Region



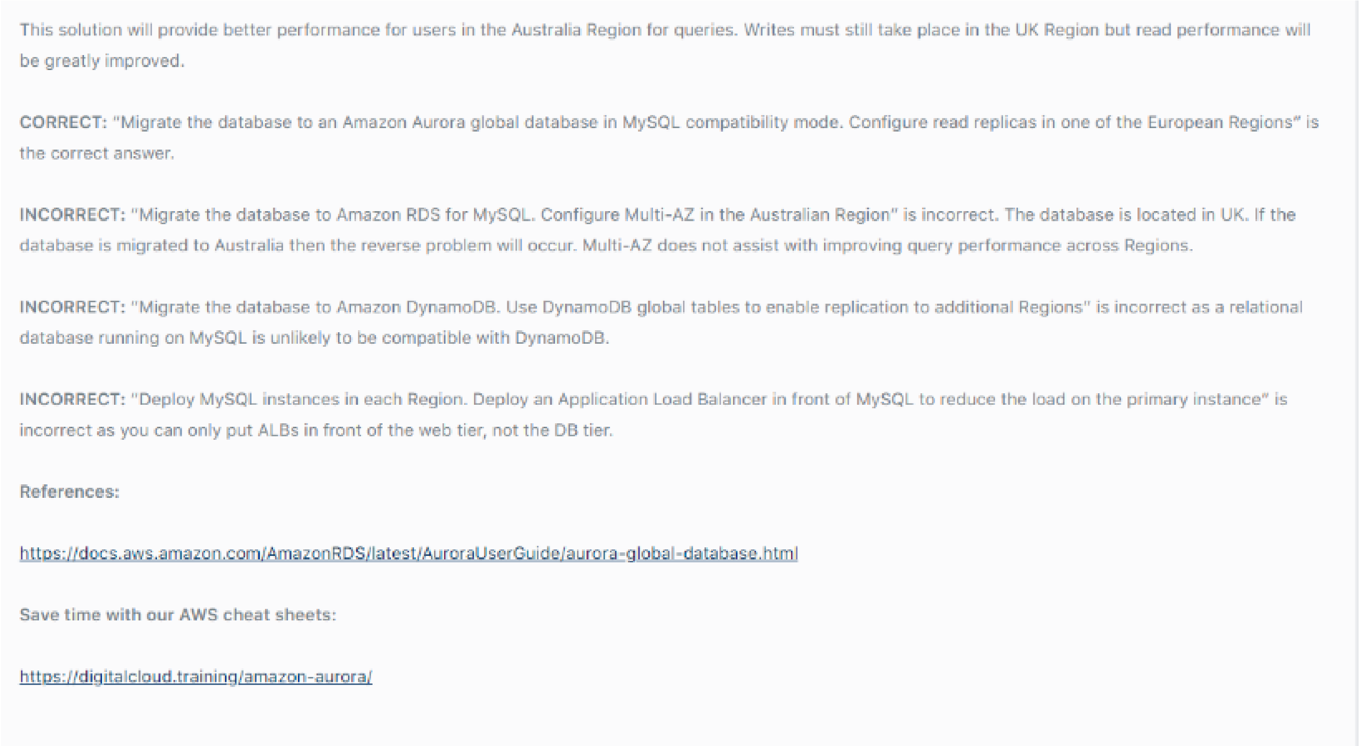
Correct

Explanation:

The issue here is latency with read queries being directed from Australia to UK which is great physical distance. A solution is required for improving read performance in Australia.

An Aurora global database consists of one primary AWS Region where your data is mastered, and up to five read-only, secondary AWS Regions. Aurora replicates data to the secondary AWS Regions with typical latency of under a second. You issue write operations directly to the primary DB instance in the primary AWS Region.

Aurora Global Database:
- Uses physical replication
- One secondary AWS region
- Uses dedicated infrastructure
- No impact on DB performance
- Good for disaster recovery



This solution will provide better performance for users in the Australia Region for queries. Writes must still take place in the UK Region but read performance will be greatly improved.

CORRECT: "Migrate the database to an Amazon Aurora global database in MySQL compatibility mode. Configure read replicas in one of the European Regions" is the correct answer.

INCORRECT: "Migrate the database to Amazon RDS for MySQL. Configure Multi-AZ in the Australian Region" is incorrect. The database is located in UK. If the database is migrated to Australia then the reverse problem will occur. Multi-AZ does not assist with improving query performance across Regions.

INCORRECT: "Migrate the database to Amazon DynamoDB. Use DynamoDB global tables to enable replication to additional Regions" is incorrect as a relational database running on MySQL is unlikely to be compatible with DynamoDB.

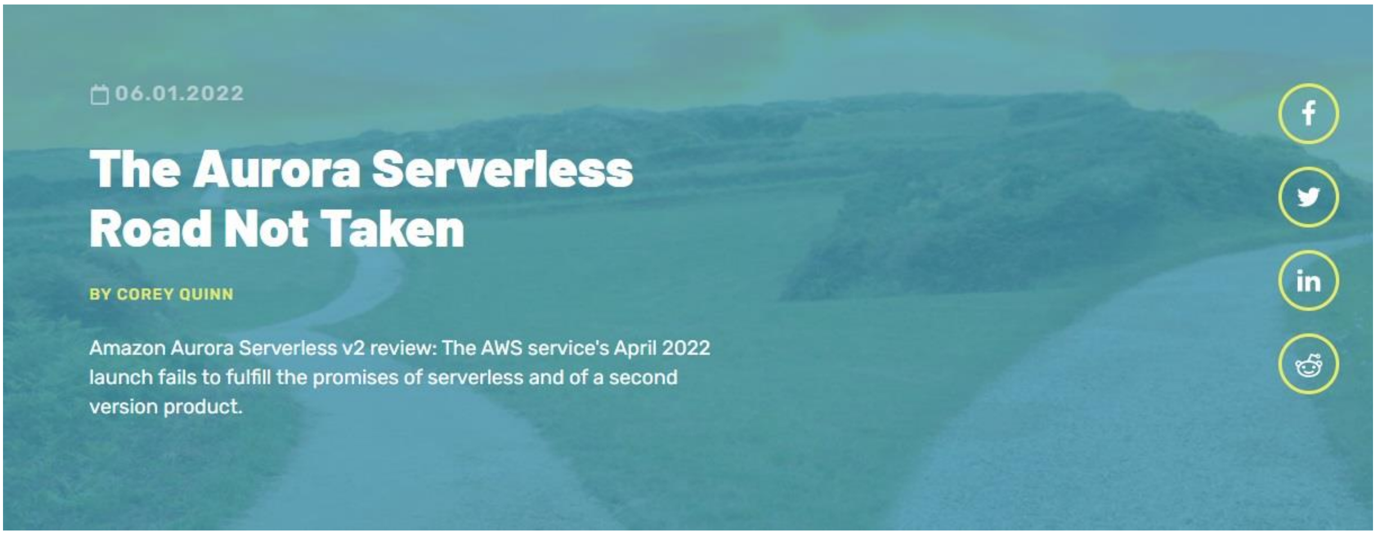
INCORRECT: "Deploy MySQL instances in each Region. Deploy an Application Load Balancer in front of MySQL to reduce the load on the primary instance" is incorrect as you can only put ALBs in front of the web tier, not the DB tier.

References:

https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/aurora-global-database.html

Save time with our AWS cheat sheets:

https://digitalcloud.training/amazon-aurora/

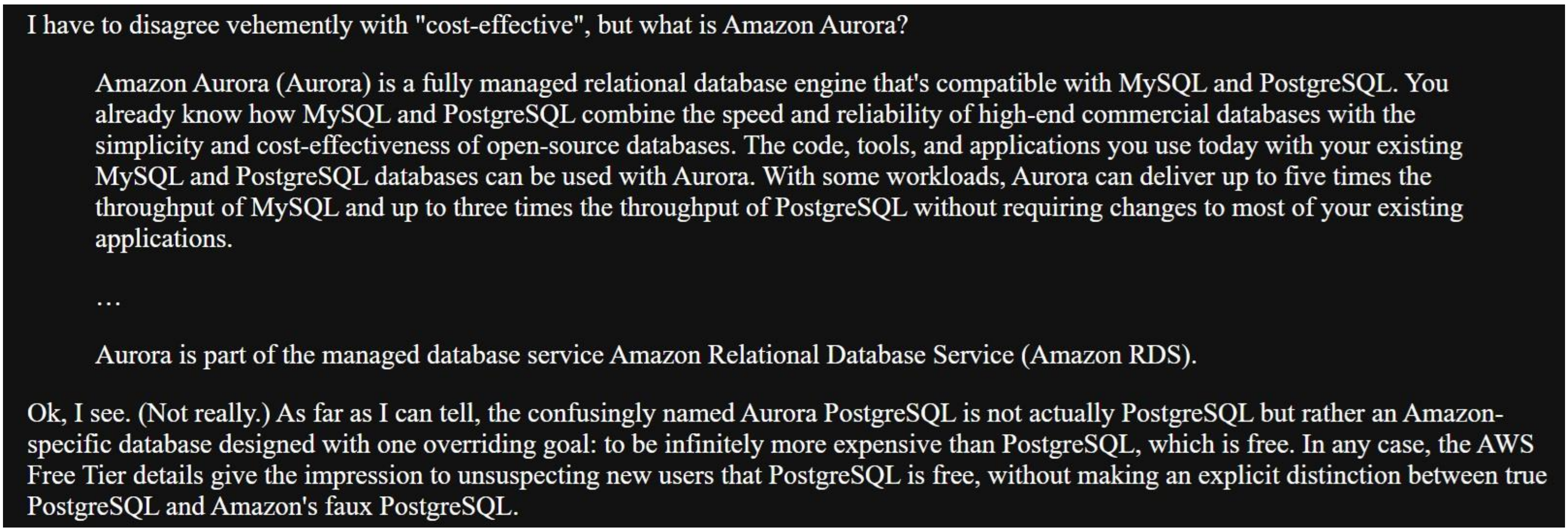


06.01.2022

The Aurora Serverless Road Not Taken

BY COREY QUINN

Amazon Aurora Serverless v2 review: The AWS service's April 2022 launch fails to fulfill the promises of serverless and of a second version product.

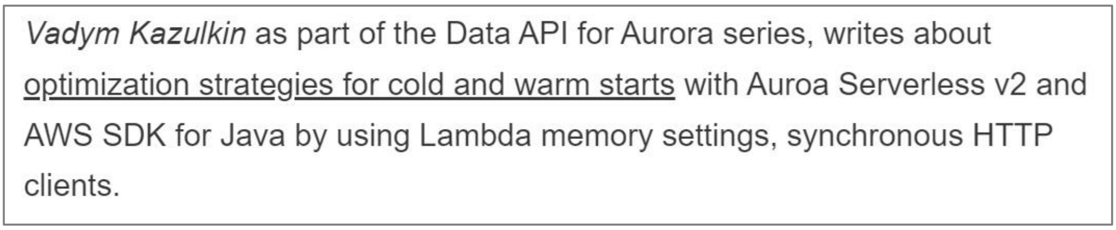


I have to disagree vehemently with "cost-effective", but what is Amazon Aurora?

> Amazon Aurora (Aurora) is a fully managed relational database engine that's compatible with MySQL and PostgreSQL. You already know how MySQL and PostgreSQL combine the speed and reliability of high-end commercial databases with the simplicity and cost-effectiveness of open-source databases. The code, tools, and applications you use today with your existing MySQL and PostgreSQL databases can be used with Aurora. With some workloads, Aurora can deliver up to five times the throughput of MySQL and up to three times the throughput of PostgreSQL without requiring changes to most of your existing applications.
>
> …
>
> Aurora is part of the managed database service Amazon Relational Database Service (Amazon RDS).

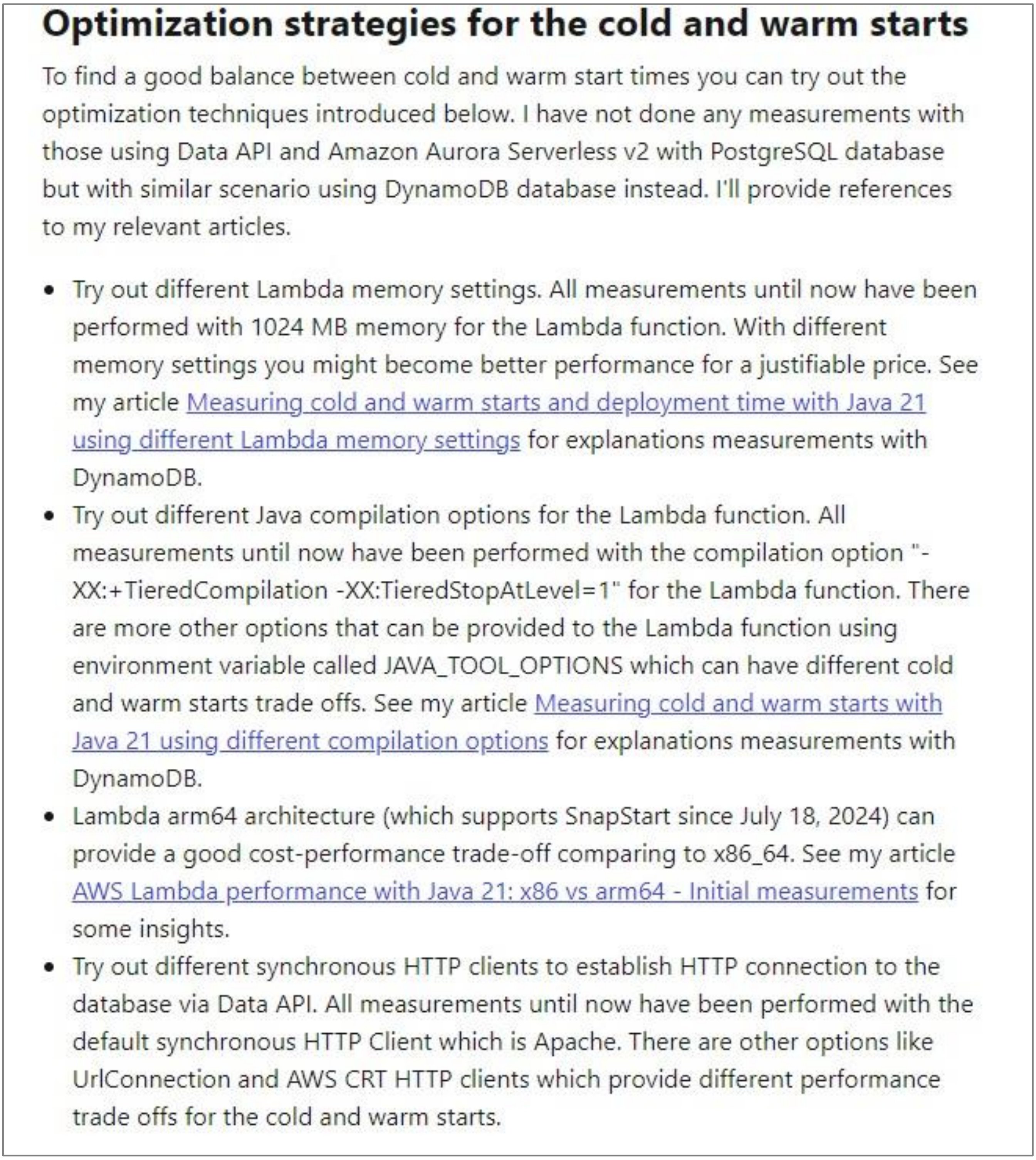
Ok, I see. (Not really.) As far as I can tell, the confusingly named Aurora PostgreSQL is not actually PostgreSQL but rather an Amazon-specific database designed with one overriding goal: to be infinitely more expensive than PostgreSQL, which is free. In any case, the AWS Free Tier details give the impression to unsuspecting new users that PostgreSQL is free, without making an explicit distinction between true PostgreSQL and Amazon's faux PostgreSQL.

> In November 2018, AWS announced the launch of the Data API for Amazon Aurora Serverless. The Data API is an intuitive, secure HTTPS API for running SQL queries against a relational database that enables you to accelerate modern application development. The Data API appeals to customers of all sizes, from startups to enterprises, seeking to minimize the time-consuming network and application configuration tasks needed to securely connect to an Amazon Aurora database. The Data API eliminates the use of drivers and improves application scalability by automatically pooling and sharing database connections (connection pooling) rather than requiring you to manage connections. You can call the Data API via an AWS SDK or the AWS Command Line Interface (AWS CLI).
>
> This feature enabled developers to quickly and securely access Aurora Serverless v1 clusters via a stateless HTTP API. You can use a familiar API interface without needing to know the intricate details of a given database driver. What's more, the Data API handles connection pooling between the Data API and the database. This helps database applications scale by reusing connections without the developer needing to configure or manage a connection pool.
>
> [Abraham 2023]



Vadym Kazulkin as part of the Data API for Aurora series, writes about optimization strategies for cold and warm starts with Auroa Serverless v2 and AWS SDK for Java by using Lambda memory settings, synchronous HTTP clients.

A note in the newsletter of [Noel 2024]



Optimization strategies for the cold and warm starts

To find a good balance between cold and warm start times you can try out the optimization techniques introduced below. I have not done any measurements with those using Data API and Amazon Aurora Serverless v2 with PostgreSQL database but with similar scenario using DynamoDB database instead. I'll provide references to my relevant articles.

- Try out different Lambda memory settings. All measurements until now have been performed with 1024 MB memory for the Lambda function. With different memory settings you might become better performance for a justifiable price. See my article Measuring cold and warm starts and deployment time with Java 21 using different Lambda memory settings for explanations measurements with DynamoDB.
- Try out different Java compilation options for the Lambda function. All measurements until now have been performed with the compilation option "-XX:+TieredCompilation -XX:TieredStopAtLevel=1" for the Lambda function. There are more other options that can be provided to the Lambda function using environment variable called JAVA_TOOL_OPTIONS which can have different cold and warm starts trade offs. See my article Measuring cold and warm starts with Java 21 using different compilation options for explanations measurements with DynamoDB.
- Lambda arm64 architecture (which supports SnapStart since July 18, 2024) can provide a good cost-performance trade-off comparing to x86_64. See my article AWS Lambda performance with Java 21: x86 vs arm64 - Initial measurements for some insights.
- Try out different synchronous HTTP clients to establish HTTP connection to the database via Data API. All measurements until now have been performed with the default synchronous HTTP Client which is Apache. There are other options like UrlConnection and AWS CRT HTTP clients which provide different performance trade offs for the cold and warm starts.

[Kazulkin 2024]

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

Term1

Description of what term means here.

Term2

Description of what term means here.

Term3

Description of what term means here.

Bibliography

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

Image Credits

Sebastiano Ricci

“Aurora and Tithonus”

I. Official

[Surname1]

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

[Surname1]

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

[Verbitski 2018]

Verbitski et al. (2018). **Amazon Aurora: On Avoiding Distributed Consensus for I/Os, Commits, and Membership Changes**. SIGMOD 2018, 10-15 June.

[Abraham 2023]

Abraham, Steve (2023). Introducing the Data API for Amazon Aurora Serverless v2 and Amazon Aurora provisioned clusters. Available at: <https://aws.amazon.com/de/blogs/database/introducing-the-data-api-for-amazon-aurora-serverless-v2-and-amazon-aurora-provisioned-clusters/>

II. Unofficial

[Surname1]

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

[Surname1]

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

https://code.tutsplus.com/tutorials/http-headers-for-dummies--net-8039

[Kazulkin 2024]

Kazulkin, Vadym (2024). Data API for Amazon Aurora Serverless v2 with AWS SDK for Java – Part 8 Optimization strategies for the cold and warm starts. Available at: https://dev.to/aws-builders/data-api-for-amazon-aurora-serverless-v2-with-aws-sdk-for-java-part-8-optimization-strategies-for-the-cold-and-warm-starts-5g88

[Noel 2024]

Noel, Zachariah Noel N (2024). Did Serverless go quiet? *The Serverless Terminal* [Newsletter]. July 15th 2024.

[Quinn 2022]

Quinn, Corey (2022). The Aurora Serverless Road Not Taken. *Last Week in AWS*. Available at: <https://www.lastweekinaws.com/blog/the-aurora-serverless-road-not-taken/>

[Johnson 2024]

Johnson, Jeff (2024). Amazon Web Services Dark Patterns. Available at: <https://lapcatsoftware.com/articles/2024/6/7.html?ck_subscriber_id=1560524742>

III. Critical

[Surname1]

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

[Surname1]

Smith, David (year). Title of Work Here. 1st 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.

IV. General

[Surname1]

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

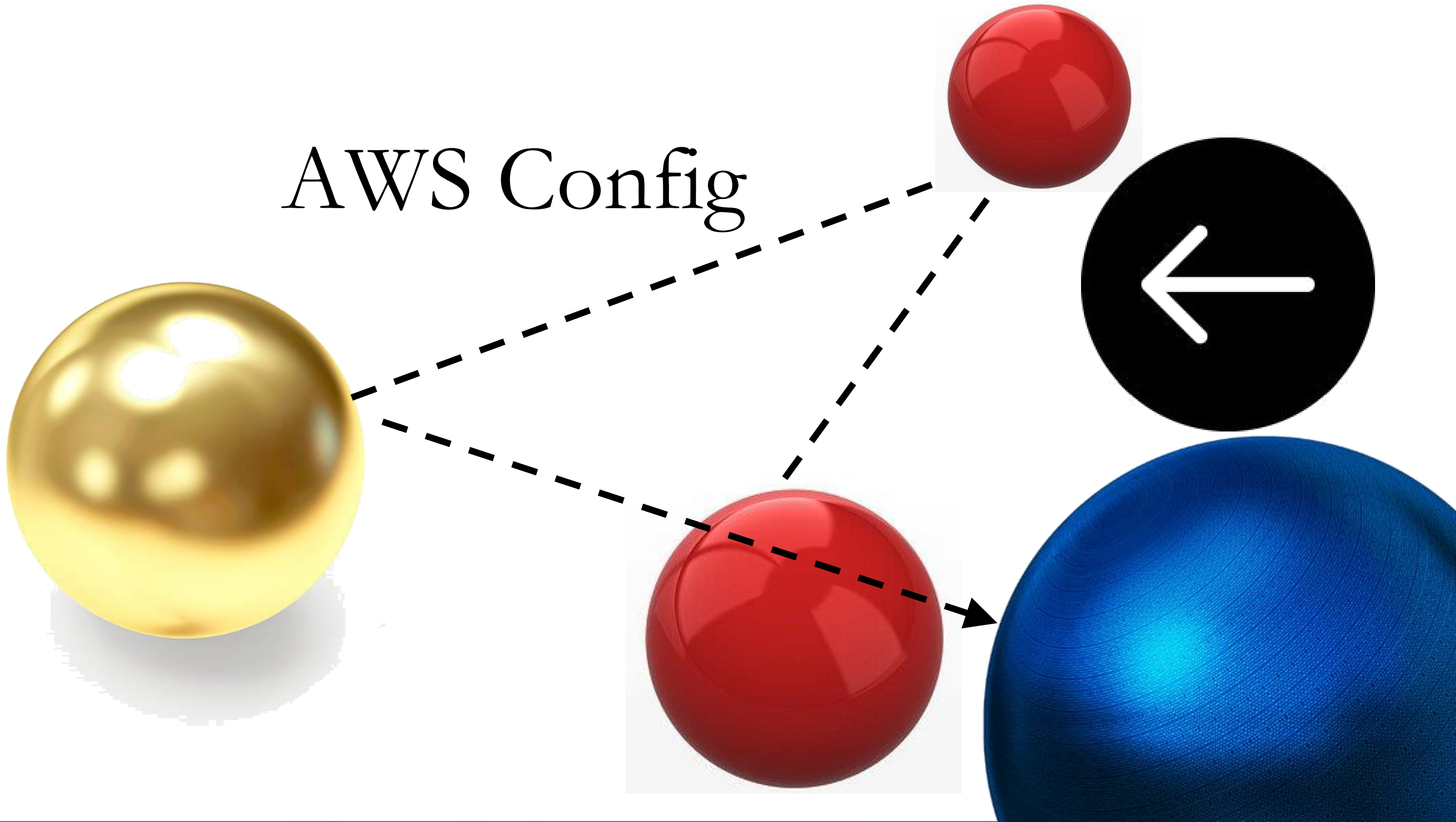
[Surname1]

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

Stefan Podlipnig and Laszlo Böszörmenyi. 2003. A survey of Web cache replacement strategies. ACM Comput. Surv. 35, 4 (December 2003), 374–398. https://doi.org/10.1145/954339.954341

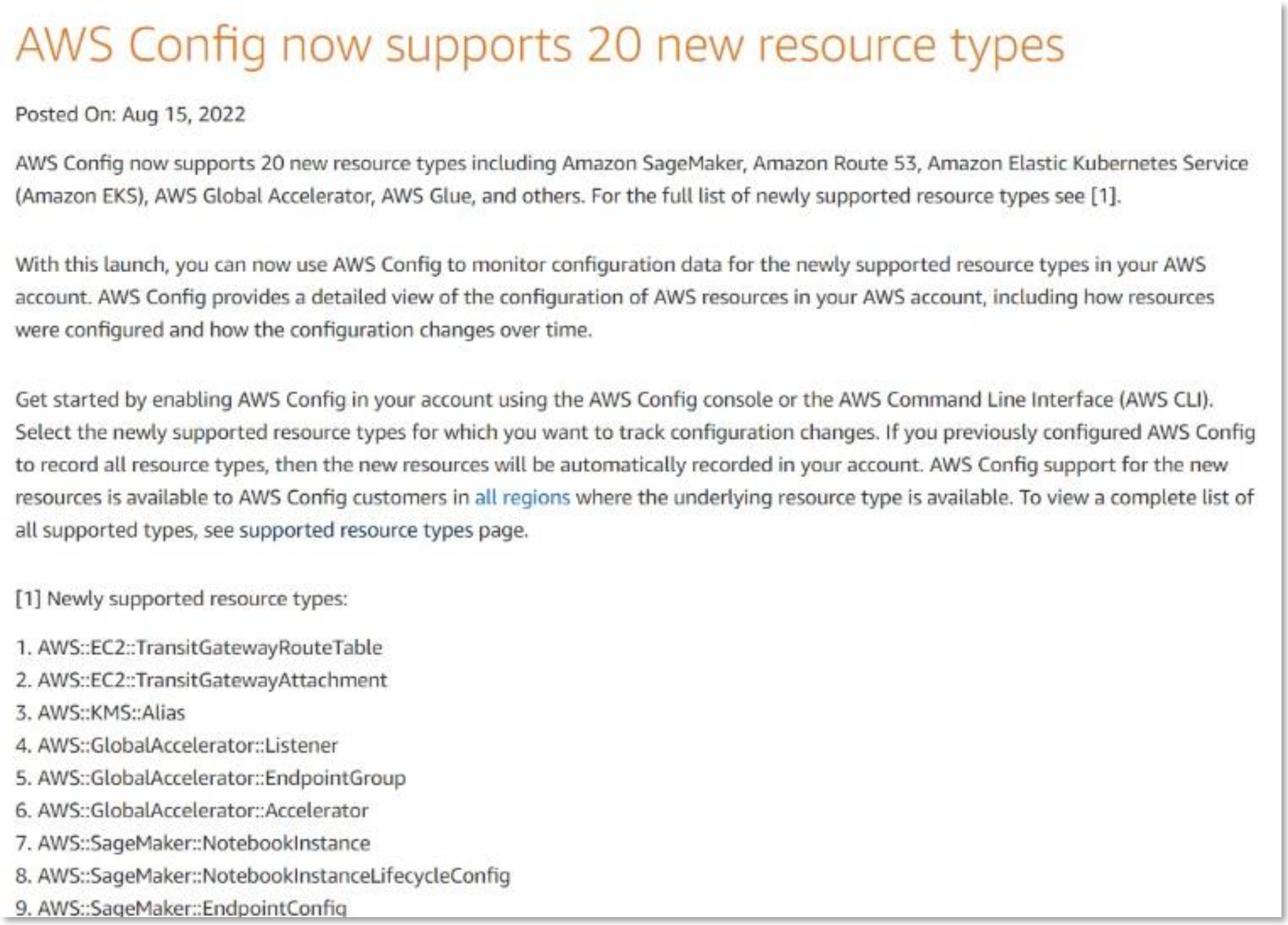
CodeDeploy

DEPLOY



AWS Config

Prashant Prahlad explaining how AWS Config works in 2015 (SEC314-R)



AWS Config now supports 20 new resource types

Posted On: Aug 15, 2022

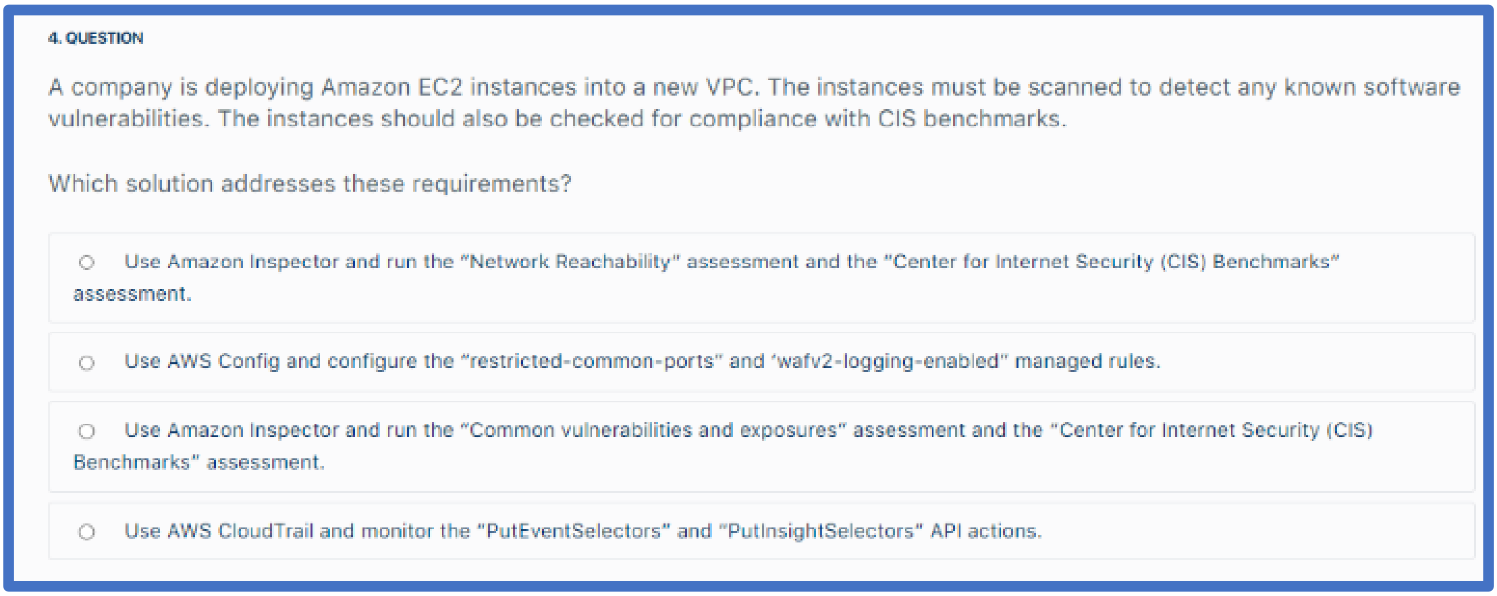
AWS Config now supports 20 new resource types including Amazon SageMaker, Amazon Route 53, Amazon Elastic Kubernetes Service (Amazon EKS), AWS Global Accelerator, AWS Glue, and others. For the full list of newly supported resource types see [1].

With this launch, you can now use AWS Config to monitor configuration data for the newly supported resource types in your AWS account. AWS Config provides a detailed view of the configuration of AWS resources in your AWS account, including how resources were configured and how the configuration changes over time.

Get started by enabling AWS Config in your account using the AWS Config console or the AWS Command Line Interface (AWS CLI). Select the newly supported resource types for which you want to track configuration changes. If you previously configured AWS Config to record all resource types, then the new resources will be automatically recorded in your account. AWS Config support for the new resources is available to AWS Config customers in all regions where the underlying resource type is available. To view a complete list of all supported types, see supported resource types page.

[1] Newly supported resource types:

1. AWS::EC2::TransitGatewayRouteTable
2. AWS::EC2::TransitGatewayAttachment
3. AWS::KMS::Alias
4. AWS::GlobalAccelerator::Listener
5. AWS::GlobalAccelerator::EndpointGroup
6. AWS::GlobalAccelerator::Accelerator
7. AWS::SageMaker::NotebookInstance
8. AWS::SageMaker::NotebookInstanceLifecycleConfig
9. AWS::SageMaker::EndpointConfig



4. QUESTION

A company is deploying Amazon EC2 instances into a new VPC. The instances must be scanned to detect any known software vulnerabilities. The instances should also be checked for compliance with CIS benchmarks.

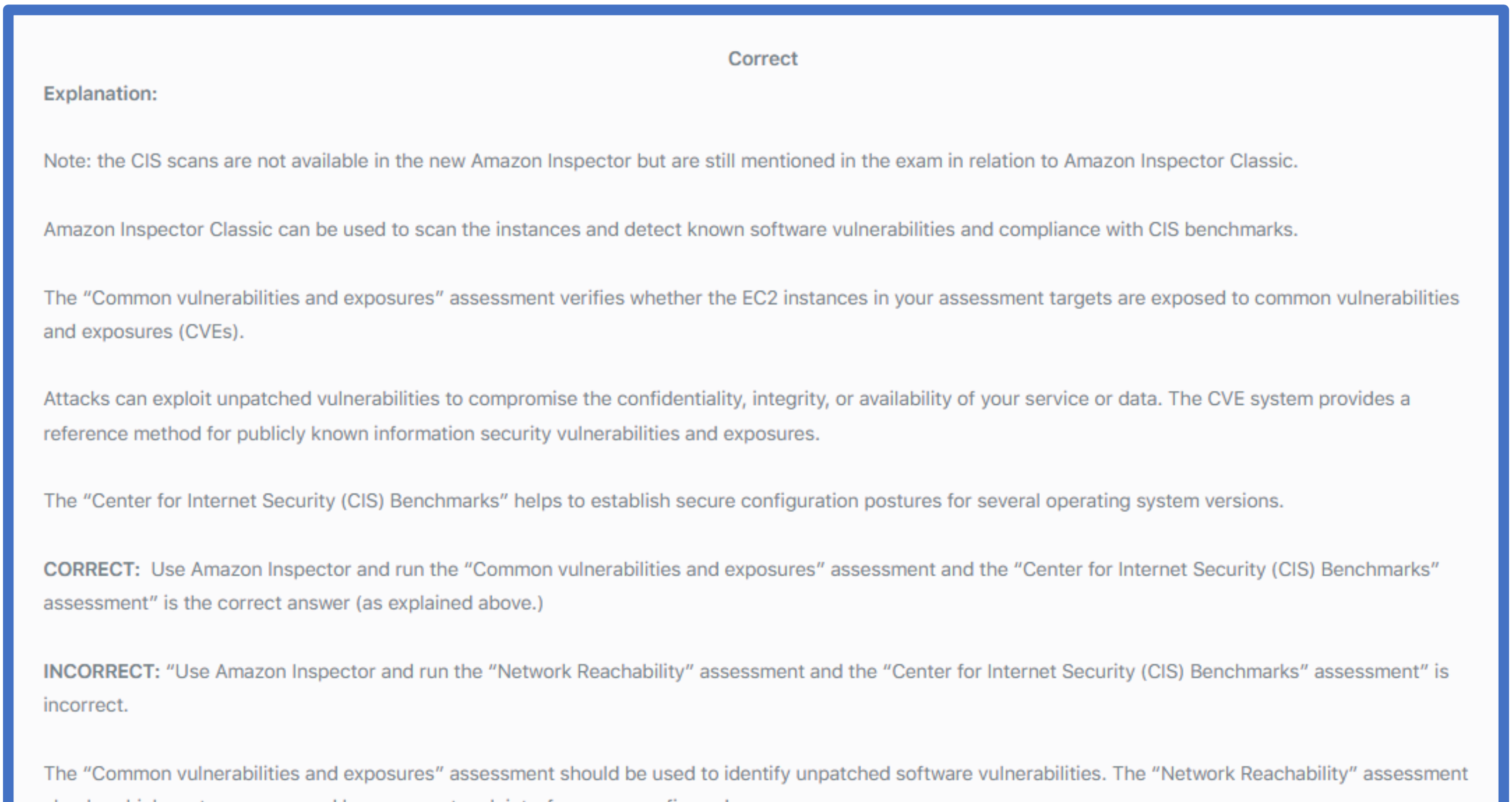
Which solution addresses these requirements?

○ Use Amazon Inspector and run the "Network Reachability" assessment and the "Center for Internet Security (CIS) Benchmarks" assessment.

○ Use AWS Config and configure the "restricted-common-ports" and 'wafv2-logging-enabled" managed rules.

○ Use Amazon Inspector and run the "Common vulnerabilities and exposures" assessment and the "Center for Internet Security (CIS) Benchmarks" assessment.

○ Use AWS CloudTrail and monitor the "PutEventSelectors" and "PutInsightSelectors" API actions.



Correct

Explanation:

Note: the CIS scans are not available in the new Amazon Inspector but are still mentioned in the exam in relation to Amazon Inspector Classic.

Amazon Inspector Classic can be used to scan the instances and detect known software vulnerabilities and compliance with CIS benchmarks.

The “Common vulnerabilities and exposures” assessment verifies whether the EC2 instances in your assessment targets are exposed to common vulnerabilities and exposures (CVEs).

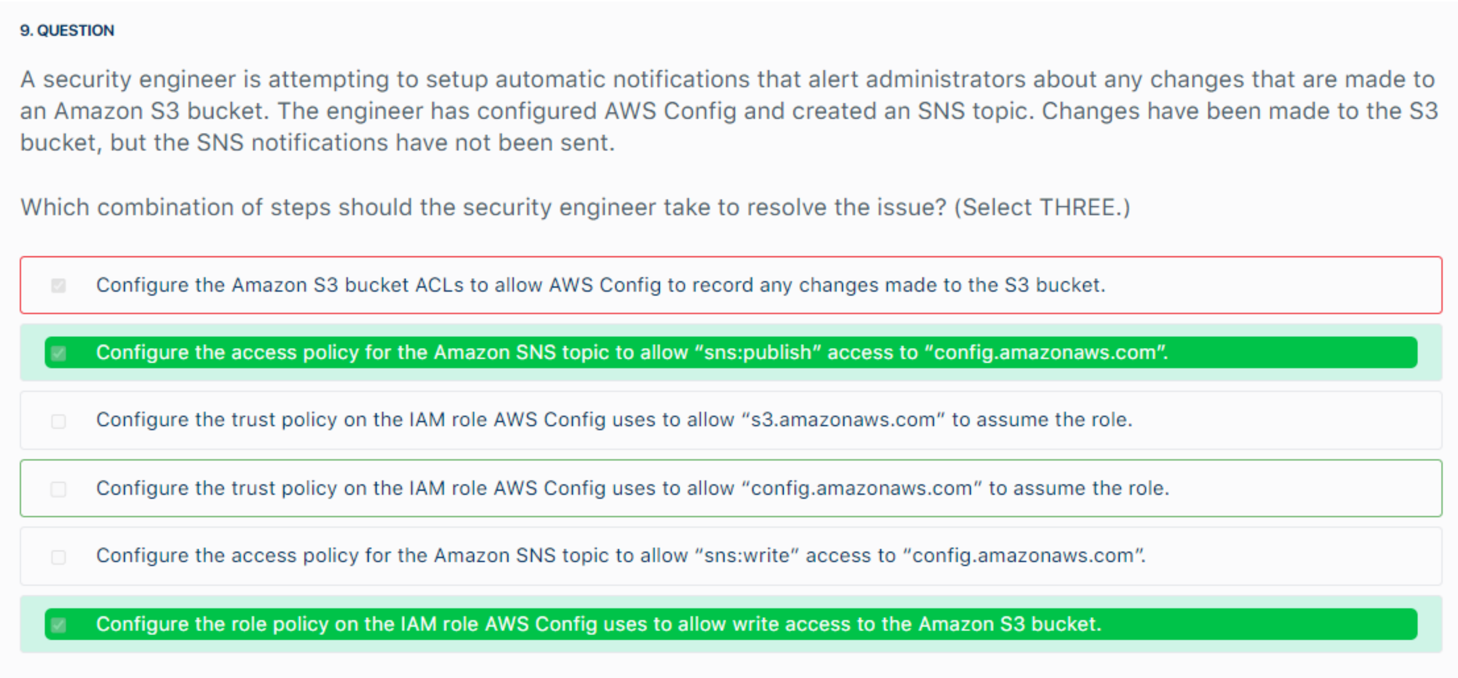
Attacks can exploit unpatched vulnerabilities to compromise the confidentiality, integrity, or availability of your service or data. The CVE system provides a reference method for publicly known information security vulnerabilities and exposures.

The “Center for Internet Security (CIS) Benchmarks” helps to establish secure configuration postures for several operating system versions.

CORRECT: Use Amazon Inspector and run the “Common vulnerabilities and exposures” assessment and the “Center for Internet Security (CIS) Benchmarks” assessment” is the correct answer (as explained above.)

INCORRECT: “Use Amazon Inspector and run the “Network Reachability” assessment and the “Center for Internet Security (CIS) Benchmarks” assessment” is incorrect.

The “Common vulnerabilities and exposures” assessment should be used to identify unpatched software vulnerabilities. The “Network Reachability” assessment

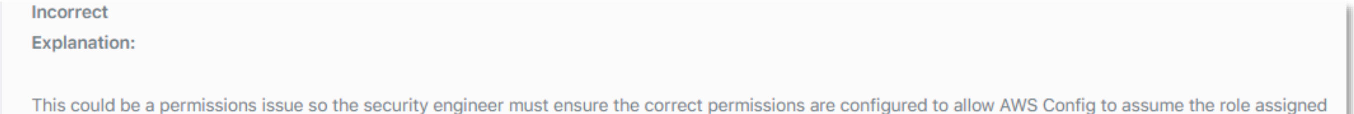


9. QUESTION

A security engineer is attempting to setup automatic notifications that alert administrators about any changes that are made to an Amazon S3 bucket. The engineer has configured AWS Config and created an SNS topic. Changes have been made to the S3 bucket, but the SNS notifications have not been sent.

Which combination of steps should the security engineer take to resolve the issue? (Select THREE.)

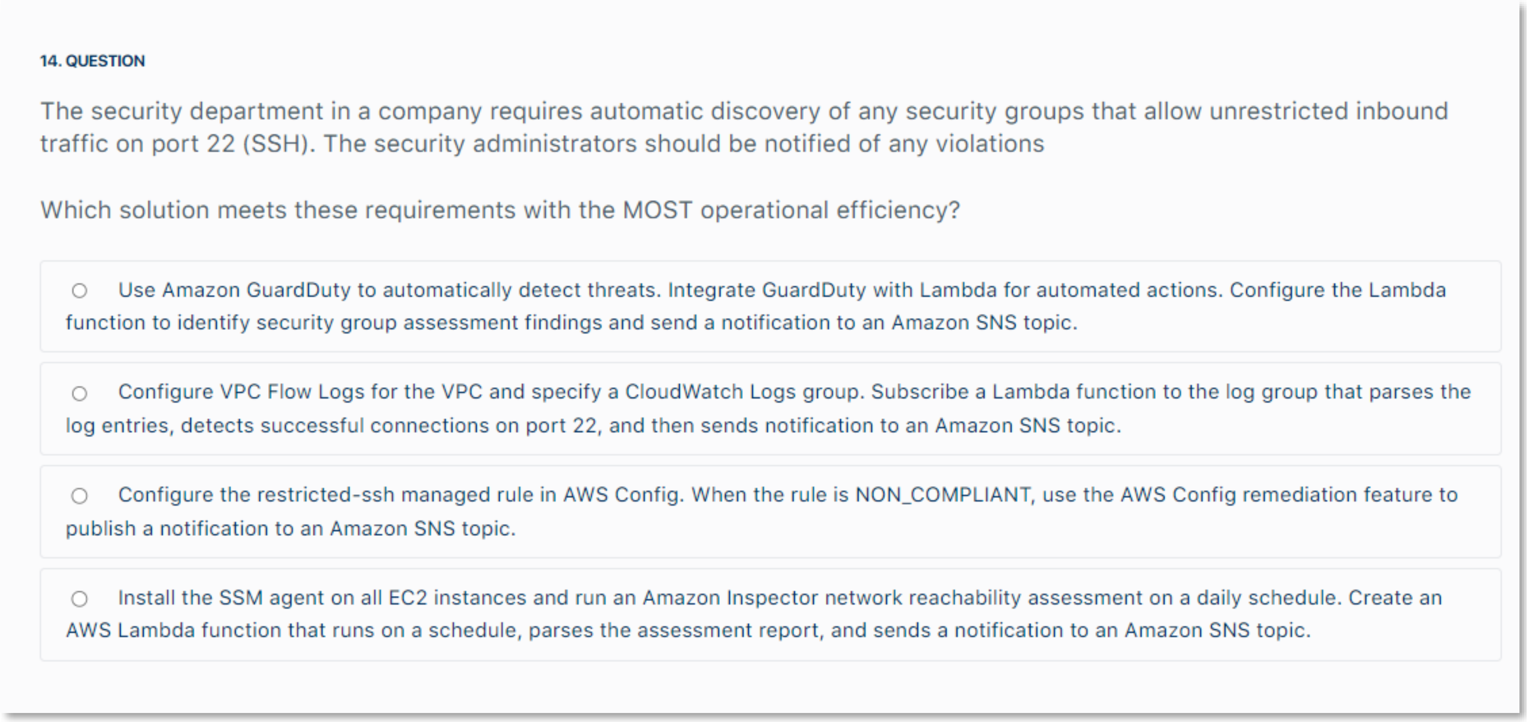
- ☑ Configure the Amazon S3 bucket ACLs to allow AWS Config to record any changes made to the S3 bucket.
- ☑ **Configure the access policy for the Amazon SNS topic to allow "sns:publish" access to "config.amazonaws.com".**
- □ Configure the trust policy on the IAM role AWS Config uses to allow "s3.amazonaws.com" to assume the role.
- □ Configure the trust policy on the IAM role AWS Config uses to allow "config.amazonaws.com" to assume the role.
- □ Configure the access policy for the Amazon SNS topic to allow "sns:write" access to "config.amazonaws.com".
- ☑ **Configure the role policy on the IAM role AWS Config uses to allow write access to the Amazon S3 bucket.**



Incorrect

Explanation:

This could be a permissions issue so the security engineer must ensure the correct permissions are configured to allow AWS Config to assume the role assigned



14. QUESTION

The security department in a company requires automatic discovery of any security groups that allow unrestricted inbound traffic on port 22 (SSH). The security administrators should be notified of any violations

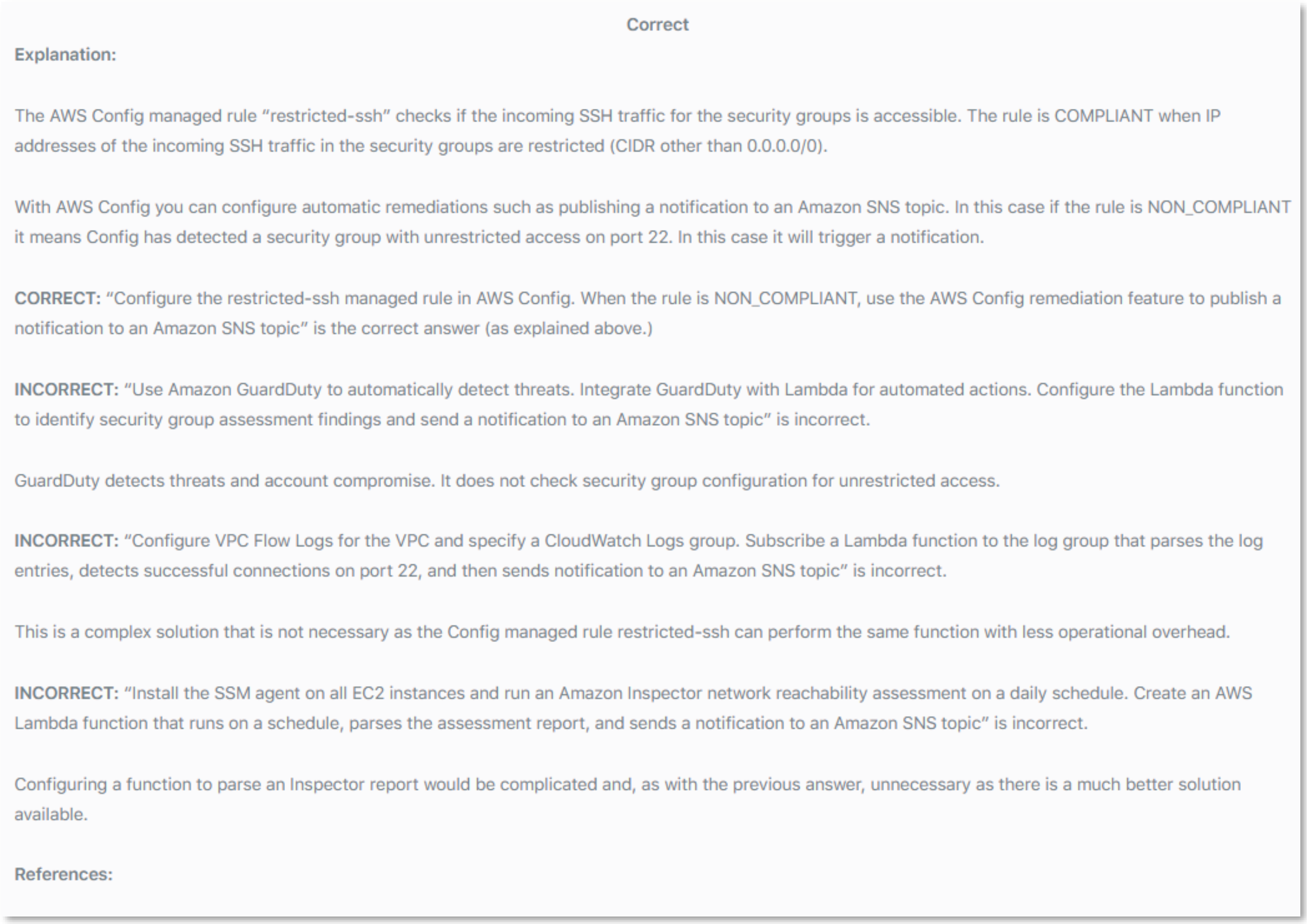
Which solution meets these requirements with the MOST operational efficiency?

○ Use Amazon GuardDuty to automatically detect threats. Integrate GuardDuty with Lambda for automated actions. Configure the Lambda function to identify security group assessment findings and send a notification to an Amazon SNS topic.

○ Configure VPC Flow Logs for the VPC and specify a CloudWatch Logs group. Subscribe a Lambda function to the log group that parses the log entries, detects successful connections on port 22, and then sends notification to an Amazon SNS topic.

○ Configure the restricted-ssh managed rule in AWS Config. When the rule is NON_COMPLIANT, use the AWS Config remediation feature to publish a notification to an Amazon SNS topic.

○ Install the SSM agent on all EC2 instances and run an Amazon Inspector network reachability assessment on a daily schedule. Create an AWS Lambda function that runs on a schedule, parses the assessment report, and sends a notification to an Amazon SNS topic.



Correct

Explanation:

The AWS Config managed rule "restricted-ssh" checks if the incoming SSH traffic for the security groups is accessible. The rule is COMPLIANT when IP addresses of the incoming SSH traffic in the security groups are restricted (CIDR other than 0.0.0.0/0).

With AWS Config you can configure automatic remediations such as publishing a notification to an Amazon SNS topic. In this case if the rule is NON_COMPLIANT it means Config has detected a security group with unrestricted access on port 22. In this case it will trigger a notification.

CORRECT: "Configure the restricted-ssh managed rule in AWS Config. When the rule is NON_COMPLIANT, use the AWS Config remediation feature to publish a notification to an Amazon SNS topic" is the correct answer (as explained above.)

INCORRECT: "Use Amazon GuardDuty to automatically detect threats. Integrate GuardDuty with Lambda for automated actions. Configure the Lambda function to identify security group assessment findings and send a notification to an Amazon SNS topic" is incorrect.

GuardDuty detects threats and account compromise. It does not check security group configuration for unrestricted access.

INCORRECT: "Configure VPC Flow Logs for the VPC and specify a CloudWatch Logs group. Subscribe a Lambda function to the log group that parses the log entries, detects successful connections on port 22, and then sends notification to an Amazon SNS topic" is incorrect.

This is a complex solution that is not necessary as the Config managed rule restricted-ssh can perform the same function with less operational overhead.

INCORRECT: "Install the SSM agent on all EC2 instances and run an Amazon Inspector network reachability assessment on a daily schedule. Create an AWS Lambda function that runs on a schedule, parses the assessment report, and sends a notification to an Amazon SNS topic" is incorrect.

Configuring a function to parse an Inspector report would be complicated and, as with the previous answer, unnecessary as there is a much better solution available.

References:

AWS Cloud Operations & Migrations Blog

Amazon S3 bucket compliance using AWS Config Auto Remediation feature

by Harshitha Putta | on 07 OCT 2019 | in AWS Config, Intermediate (200), Management Tools | Permalink | Share

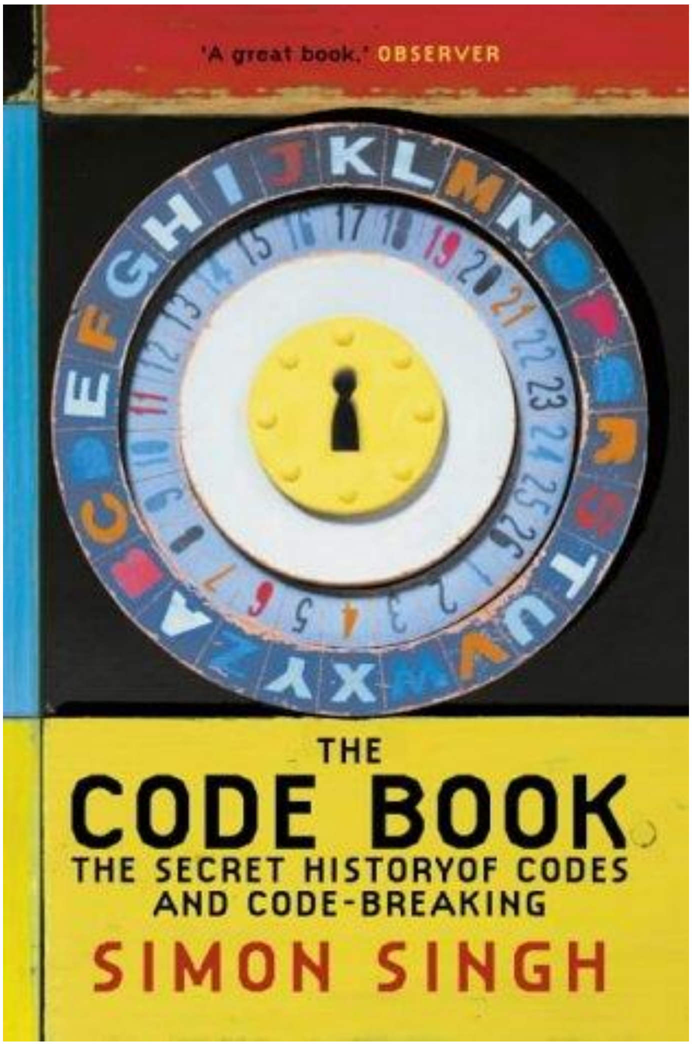
AWS Config keeps track of the configuration of your AWS resources and their relationships to your other resources. It can also evaluate those AWS resources for compliance. This service uses rules that can be configured to evaluate AWS resources against desired configurations.

For example, there are AWS Config rules that check whether or not your Amazon S3 buckets have logging enabled or your IAM users have an MFA device enabled. AWS Config rules use AWS Lambda functions to perform the compliance evaluations, and the Lambda functions return the compliance status of the evaluated resources as compliant or noncompliant. The noncompliant resources are remediated using the remediation action associated to the AWS Config rule. With Auto Remediation feature of AWS Config rules, the remediation action can be executed automatically when a resource is found non-compliant.

https://aws.amazon.com/about-aws/whats-new/2022/08/aws-config-supports-20-new-resource-types/?ck_subscriber_id=1560524742

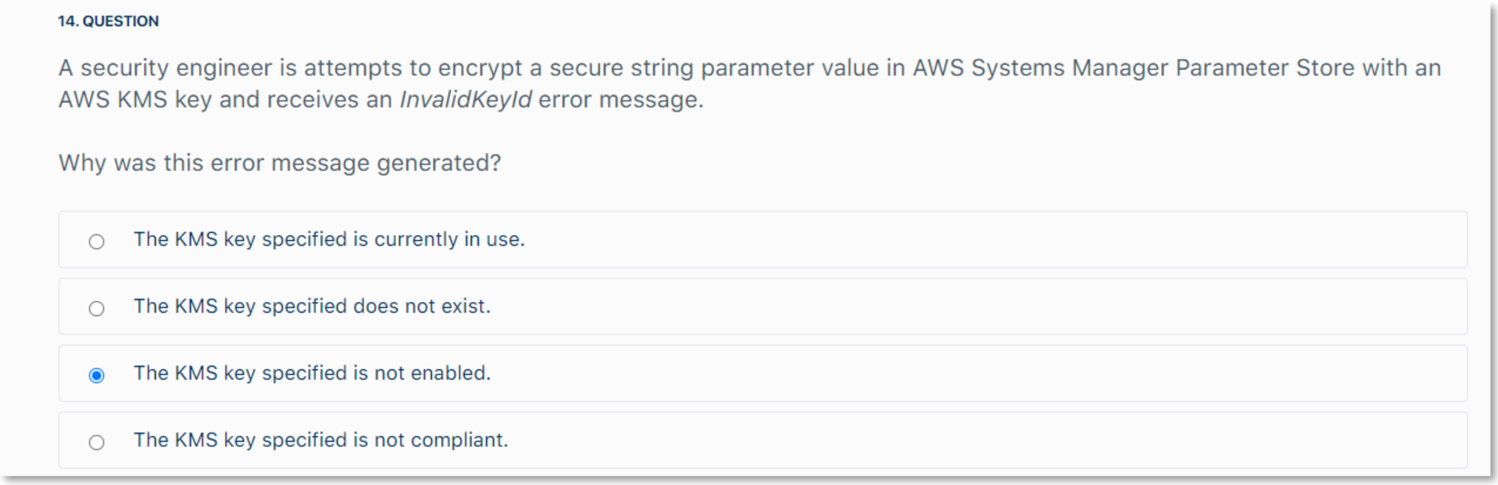
KMS

To understand AWS' Key Management Service (KMS), it is helpful to have a familiarity with cryptography. There is a particular chapter in this book which explains things well. Simon Singh makes these things as simple as they can be. You need to know the sense in which we are talking about keys. Then, the relevance of the distinction between *public* and *private* keys, and *symmetric* and *asymmetric* keys.


'A great book,' OBSERVER
THE
CODE BOOK
THE SECRET HISTORYOF CODES
AND CODE-BREAKING
SIMON SINGH

Peter O’Donnell explaining how KMS works, in 2019

Christoph Paar’s excellent lecture series on cryptography is available on YouTube for free.



14. QUESTION

A security engineer is attempts to encrypt a secure string parameter value in AWS Systems Manager Parameter Store with an AWS KMS key and receives an *InvalidKeyId* error message.

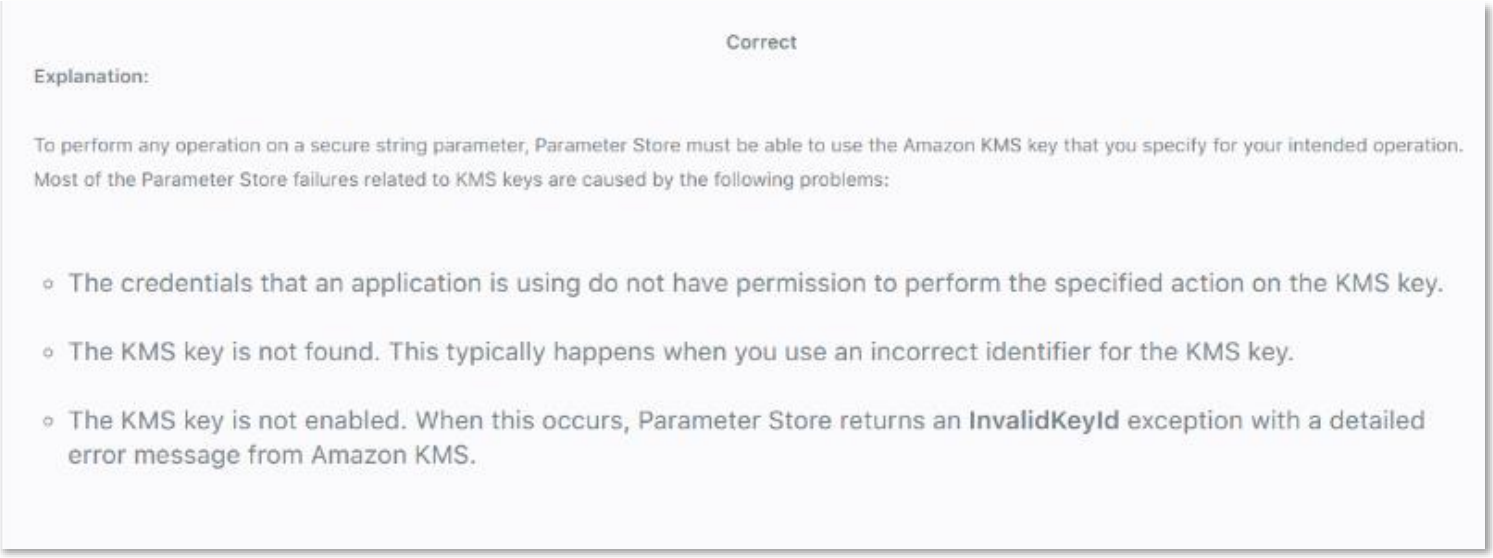
Why was this error message generated?

○ The KMS key specified is currently in use.

○ The KMS key specified does not exist.

● The KMS key specified is not enabled.

○ The KMS key specified is not compliant.

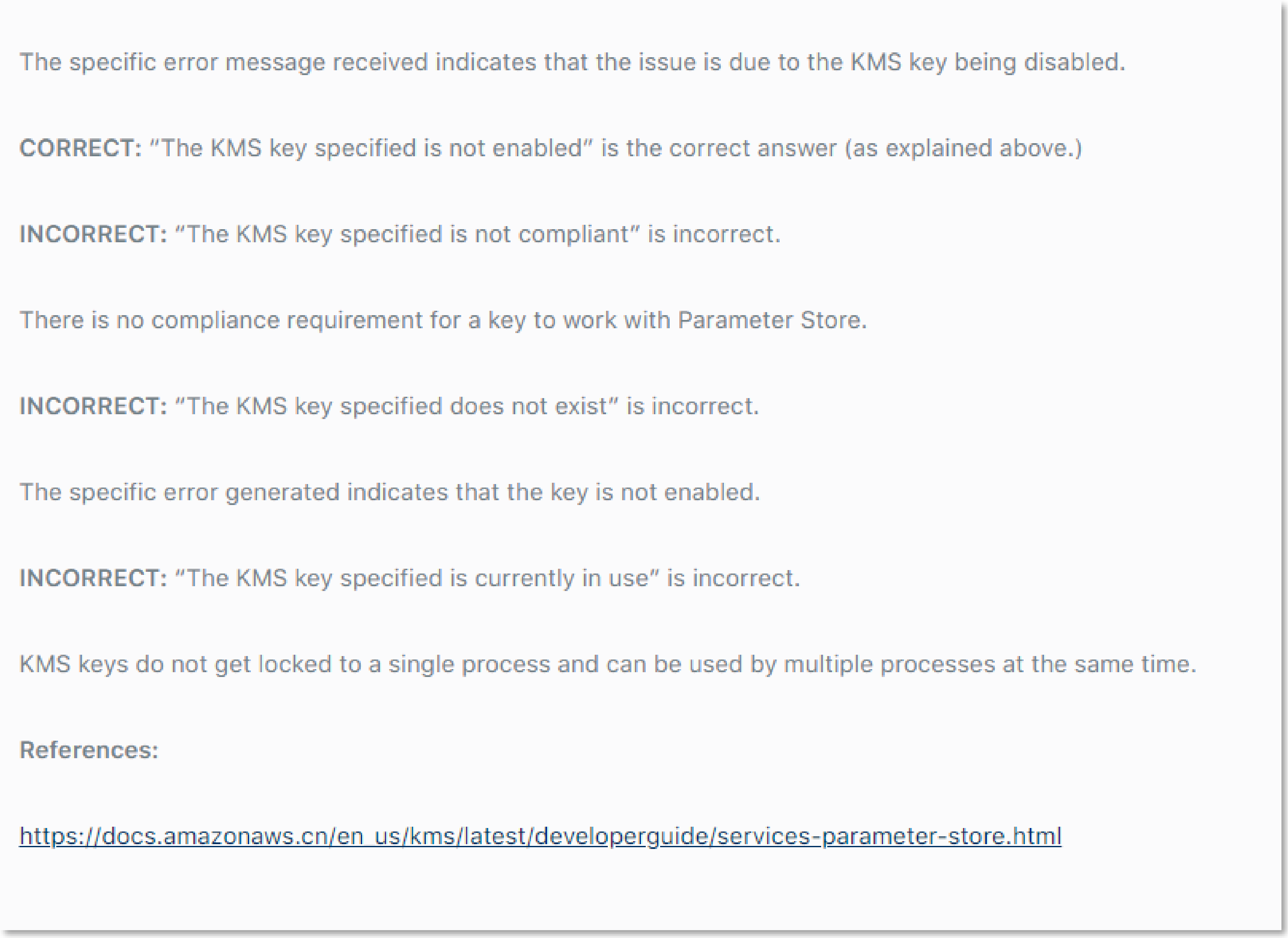


Correct

Explanation:

To perform any operation on a secure string parameter, Parameter Store must be able to use the Amazon KMS key that you specify for your intended operation. Most of the Parameter Store failures related to KMS keys are caused by the following problems:

- The credentials that an application is using do not have permission to perform the specified action on the KMS key.
- The KMS key is not found. This typically happens when you use an incorrect identifier for the KMS key.
- The KMS key is not enabled. When this occurs, Parameter Store returns an **InvalidKeyId** exception with a detailed error message from Amazon KMS.



The specific error message received indicates that the issue is due to the KMS key being disabled.

CORRECT: "The KMS key specified is not enabled" is the correct answer (as explained above.)

INCORRECT: "The KMS key specified is not compliant" is incorrect.

There is no compliance requirement for a key to work with Parameter Store.

INCORRECT: "The KMS key specified does not exist" is incorrect.

The specific error generated indicates that the key is not enabled.

INCORRECT: "The KMS key specified is currently in use" is incorrect.

KMS keys do not get locked to a single process and can be used by multiple processes at the same time.

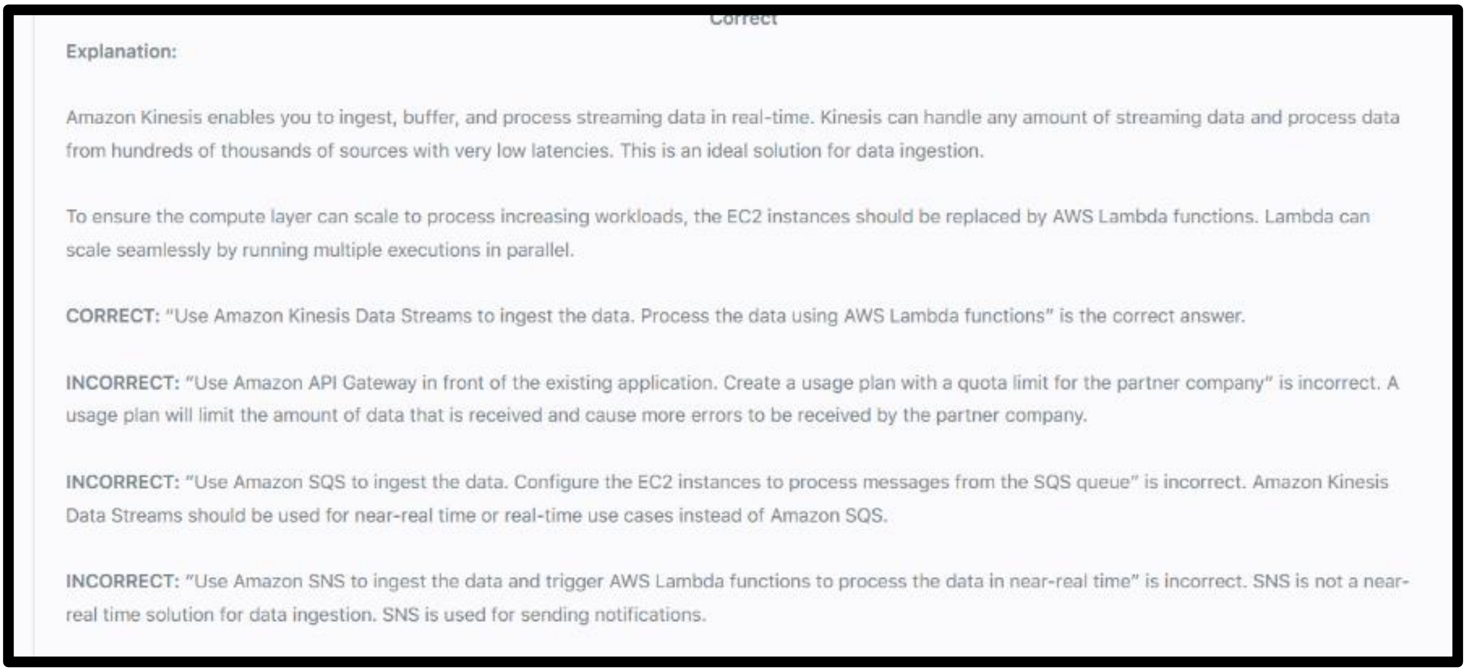
References:

https://docs.amazonaws.cn/en_us/kms/latest/developerguide/services-parameter-store.html

Lambda

Listen, ay—my brother- *don't* allocate (LAMBDA)

Why is it called Lambda?



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.

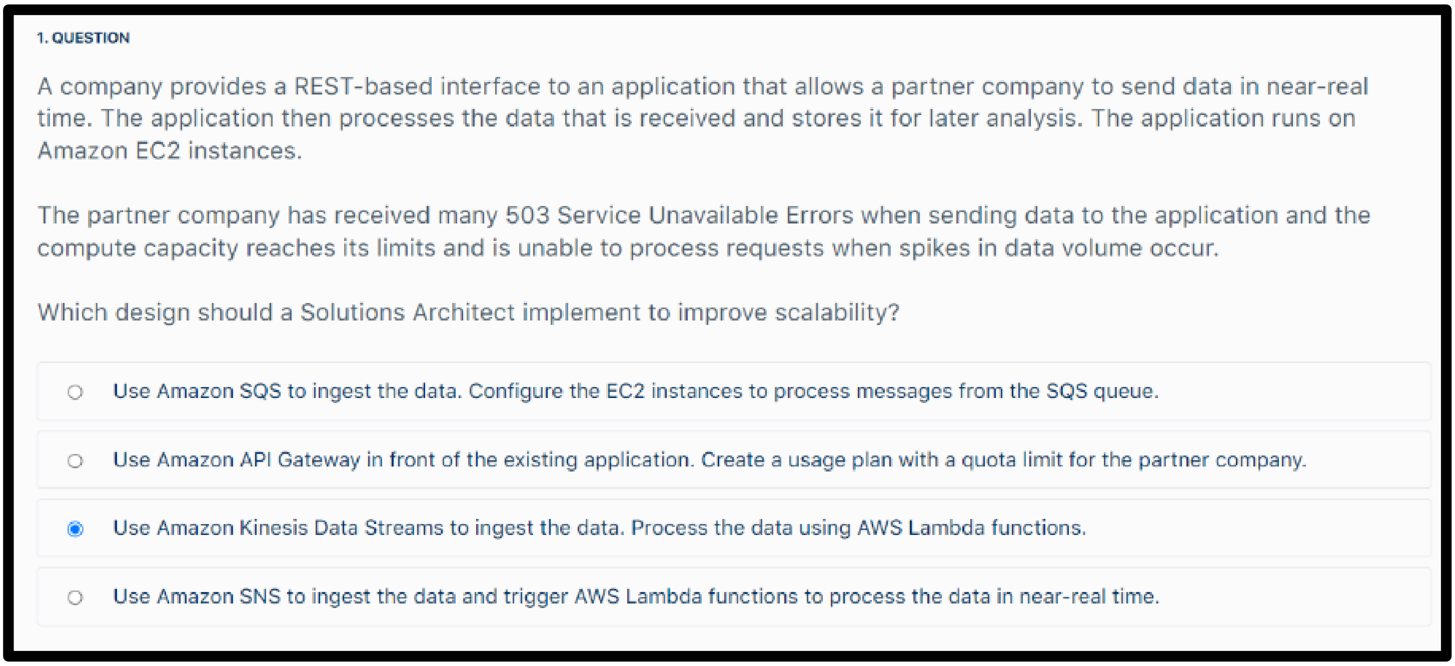
To ensure the compute layer can scale to process increasing workloads, the EC2 instances should be replaced by AWS Lambda functions. Lambda can scale seamlessly by running multiple executions in parallel.

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.



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?

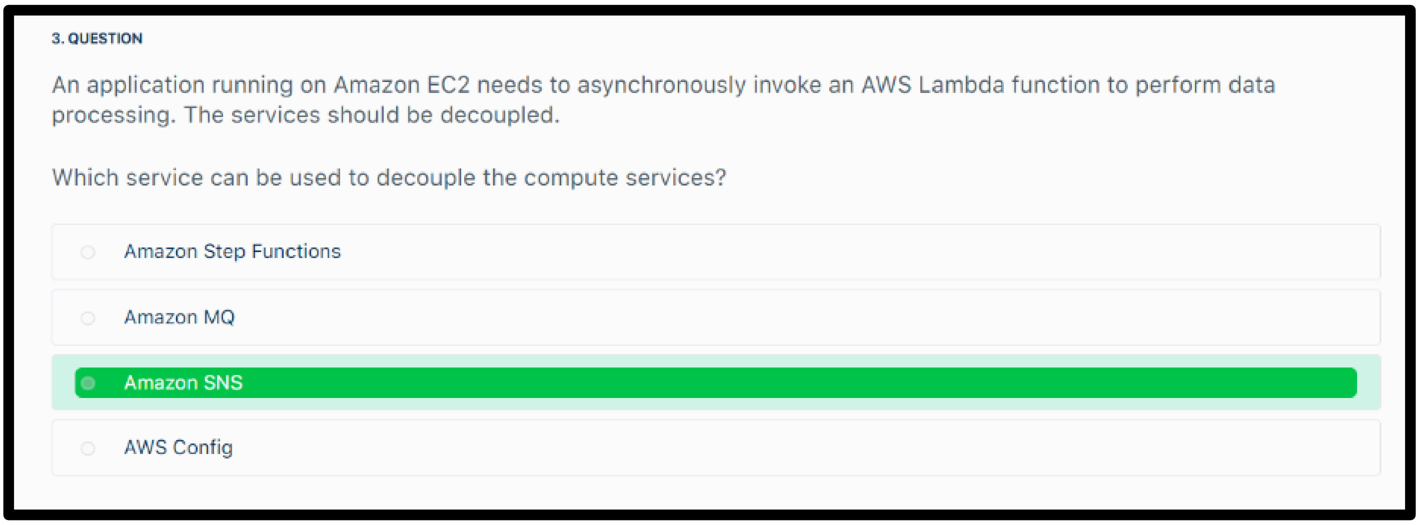
○ Use Amazon SQS to ingest the data. Configure the EC2 instances to process messages from the SQS queue.

○ 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.

The question below highlights how Lambda functions can actually be invoked by the SNS. You naturally think of SNS as quite passive: it merely *notifies* you of things. Therefore, this question from Neal Davis reminds us that SNS can actually bring about action:



3. QUESTION

An application running on Amazon EC2 needs to asynchronously invoke an AWS Lambda function to perform data processing. The services should be decoupled.

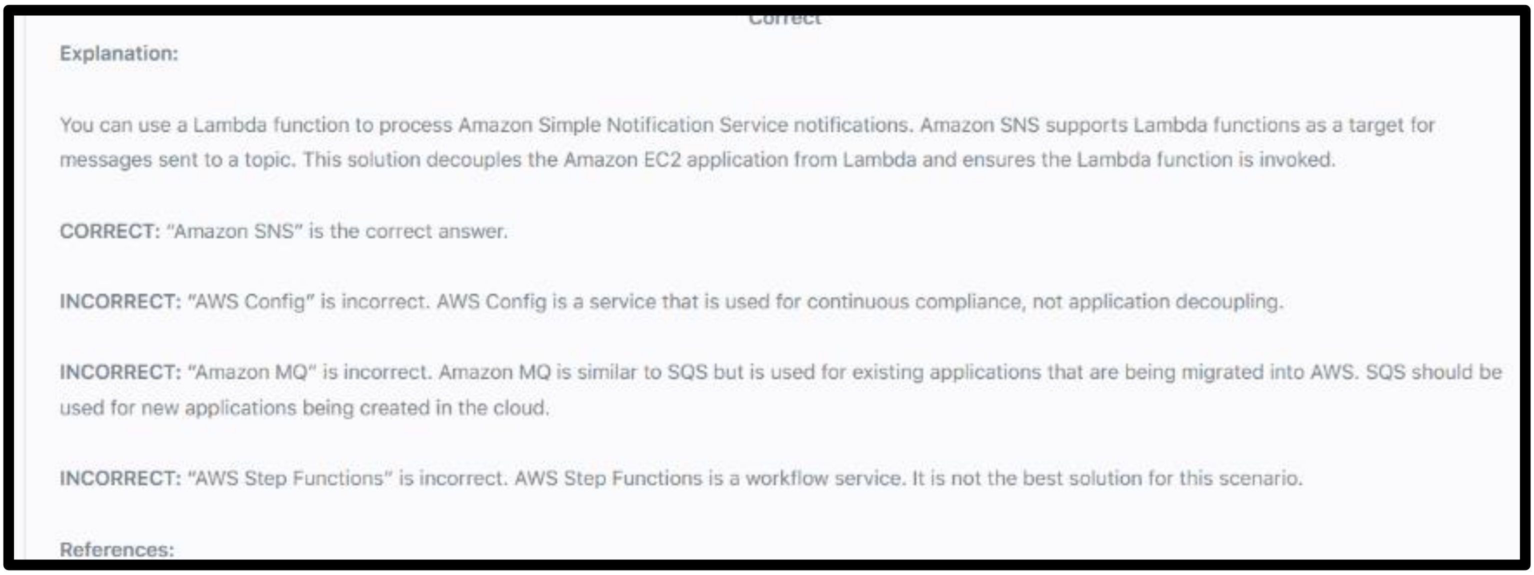
Which service can be used to decouple the compute services?

○ Amazon Step Functions

○ Amazon MQ

● Amazon SNS

○ AWS Config



Correct

Explanation:

You can use a Lambda function to process Amazon Simple Notification Service notifications. Amazon SNS supports Lambda functions as a target for messages sent to a topic. This solution decouples the Amazon EC2 application from Lambda and ensures the Lambda function is invoked.

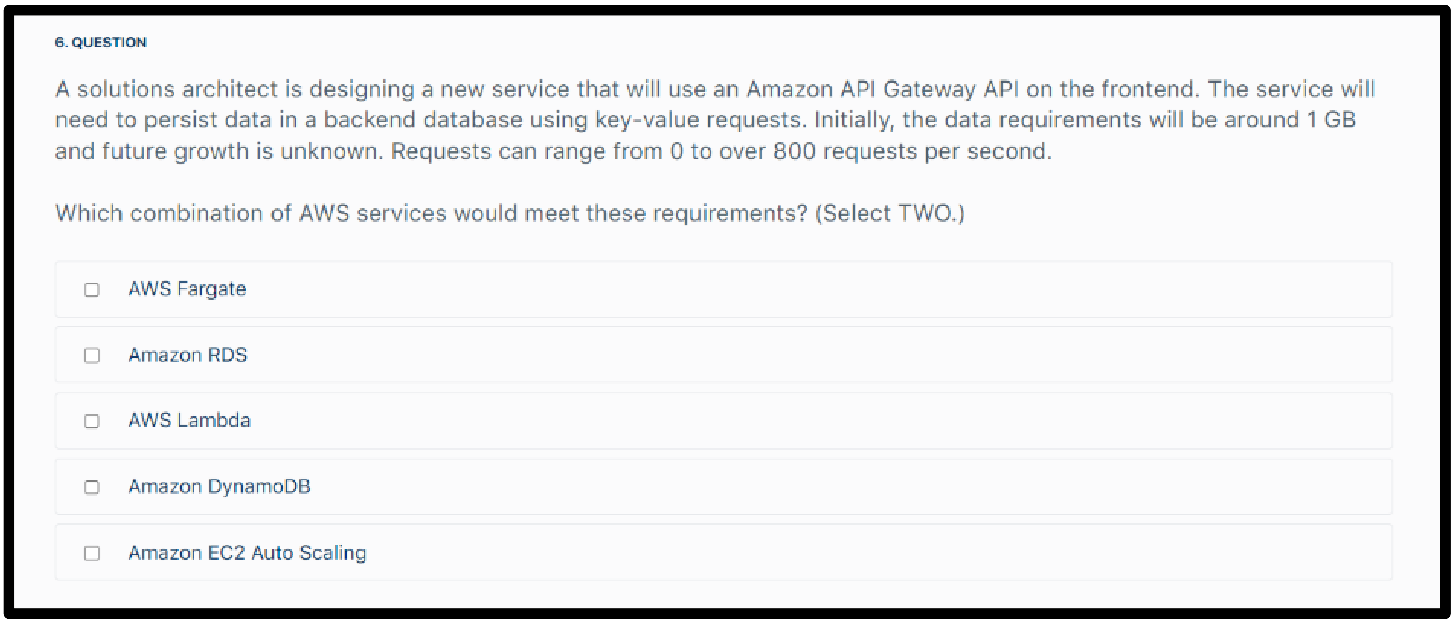
CORRECT: "Amazon SNS" is the correct answer.

INCORRECT: "AWS Config" is incorrect. AWS Config is a service that is used for continuous compliance, not application decoupling.

INCORRECT: "Amazon MQ" is incorrect. Amazon MQ is similar to SQS but is used for existing applications that are being migrated into AWS. SQS should be used for new applications being created in the cloud.

INCORRECT: "AWS Step Functions" is incorrect. AWS Step Functions is a workflow service. It is not the best solution for this scenario.

References:



6. QUESTION

A solutions architect is designing a new service that will use an Amazon API Gateway API on the frontend. The service will need to persist data in a backend database using key-value requests. Initially, the data requirements will be around 1 GB and future growth is unknown. Requests can range from 0 to over 800 requests per second.

Which combination of AWS services would meet these requirements? (Select TWO.)

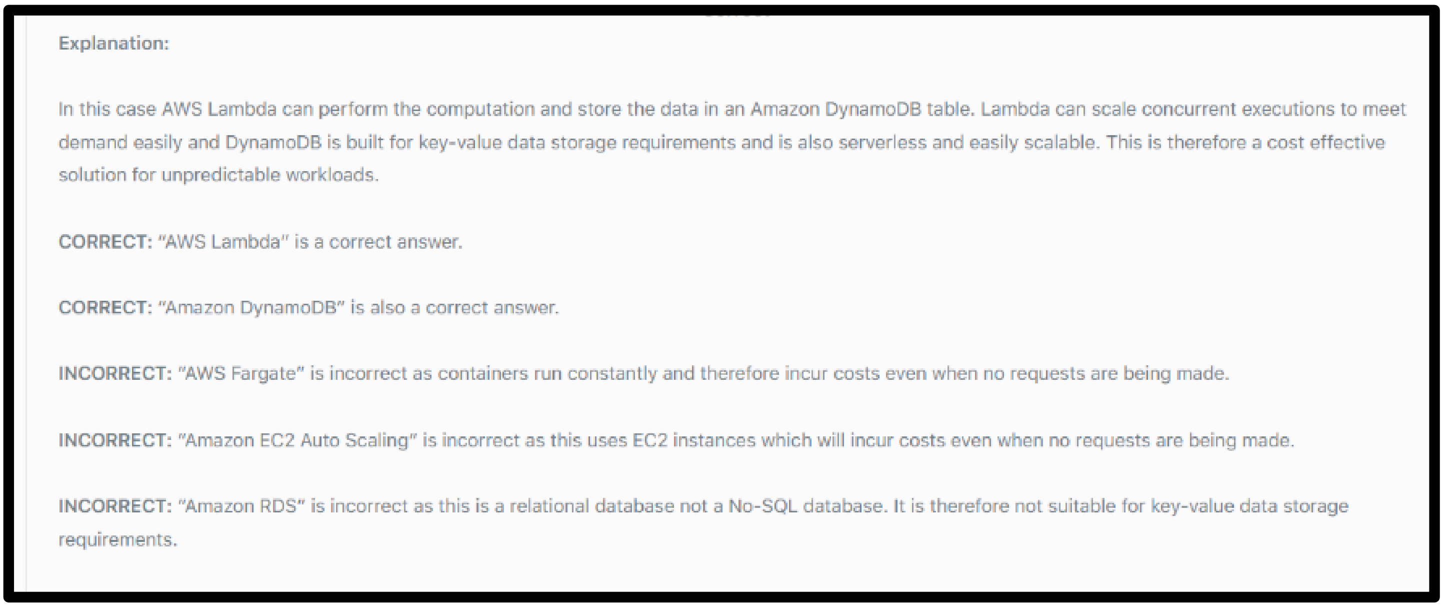
□ AWS Fargate

□ Amazon RDS

□ AWS Lambda

□ Amazon DynamoDB

□ Amazon EC2 Auto Scaling



Explanation:

In this case AWS Lambda can perform the computation and store the data in an Amazon DynamoDB table. Lambda can scale concurrent executions to meet demand easily and DynamoDB is built for key-value data storage requirements and is also serverless and easily scalable. This is therefore a cost effective solution for unpredictable workloads.

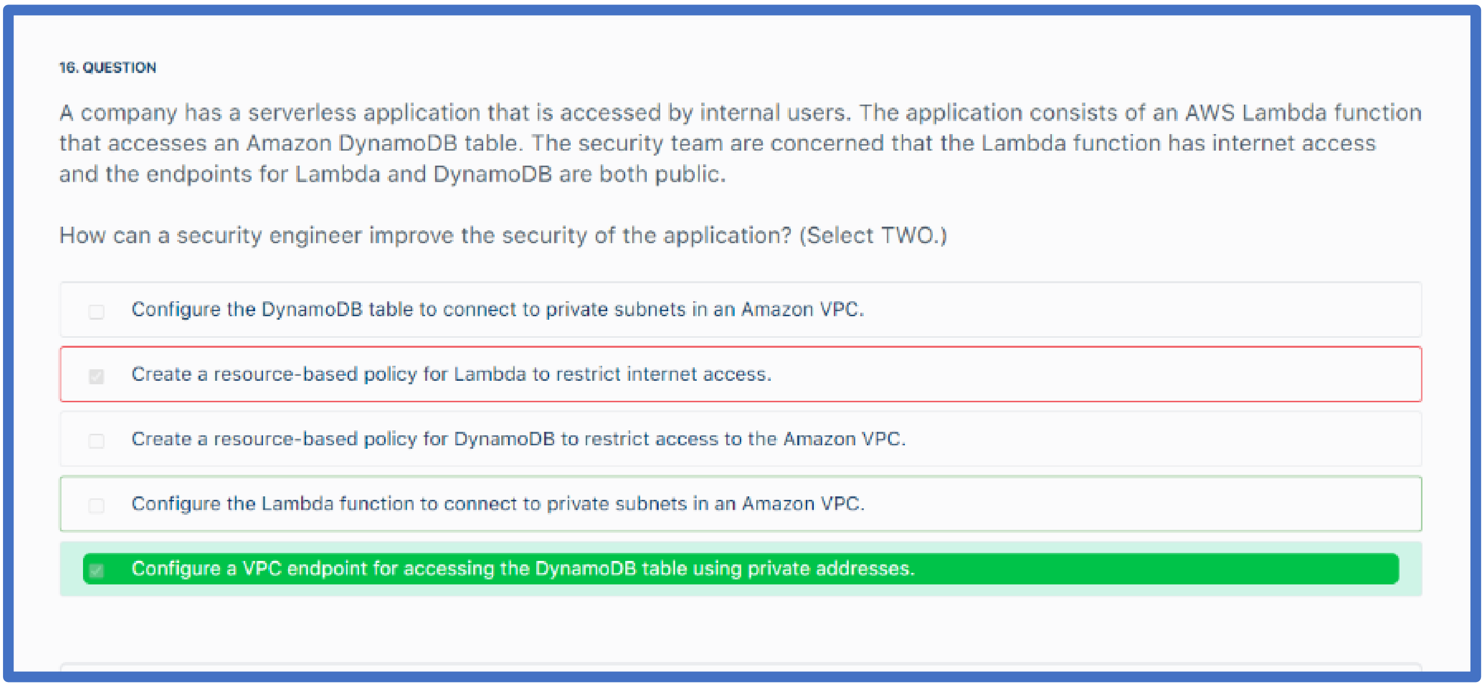
CORRECT: "AWS Lambda" is a correct answer.

CORRECT: "Amazon DynamoDB" is also a correct answer.

INCORRECT: "AWS Fargate" is incorrect as containers run constantly and therefore incur costs even when no requests are being made.

INCORRECT: "Amazon EC2 Auto Scaling" is incorrect as this uses EC2 instances which will incur costs even when no requests are being made.

INCORRECT: "Amazon RDS" is incorrect as this is a relational database not a No-SQL database. It is therefore not suitable for key-value data storage requirements.

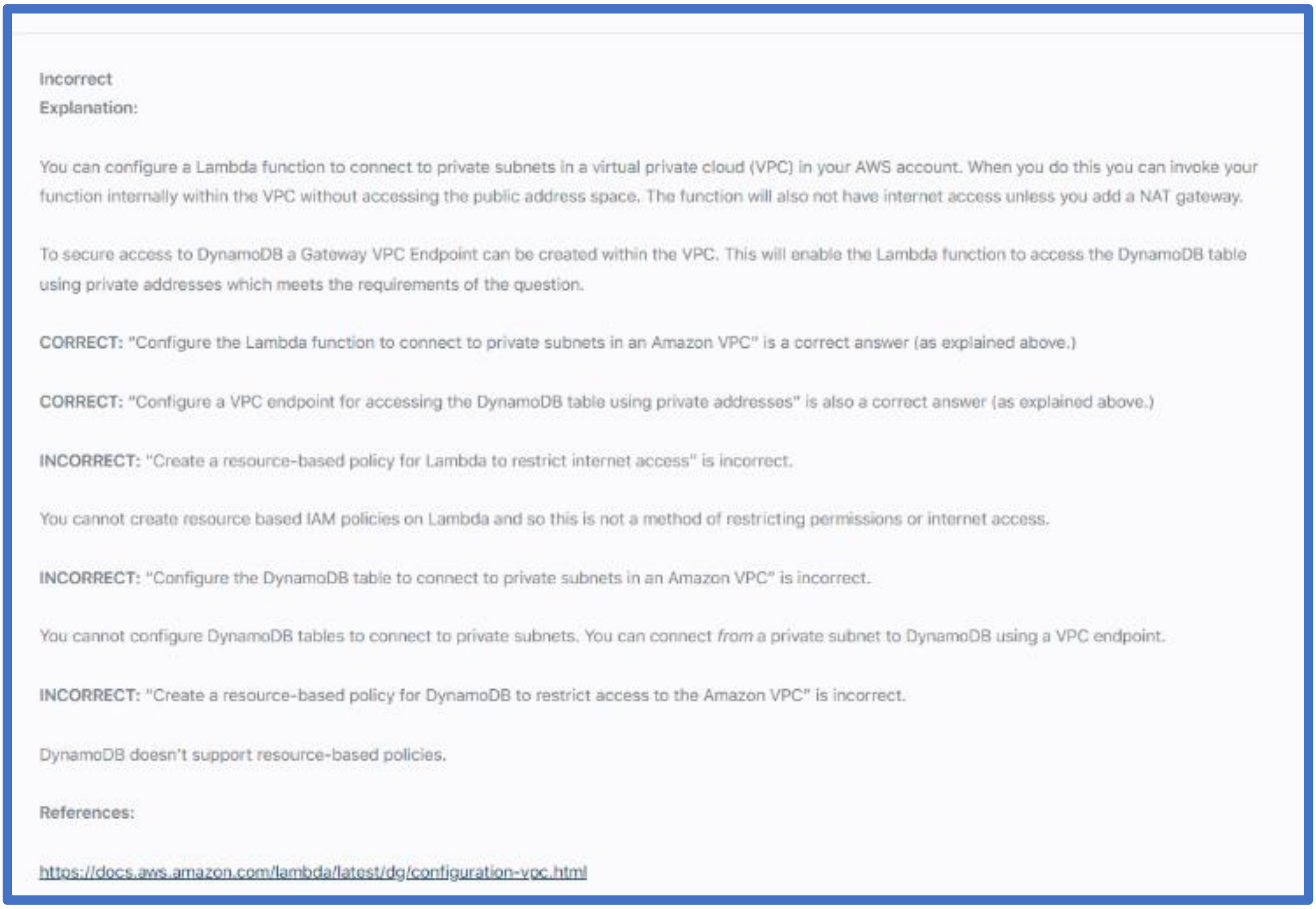


16. QUESTION

A company has a serverless application that is accessed by internal users. The application consists of an AWS Lambda function that accesses an Amazon DynamoDB table. The security team are concerned that the Lambda function has internet access and the endpoints for Lambda and DynamoDB are both public.

How can a security engineer improve the security of the application? (Select TWO.)

- □ Configure the DynamoDB table to connect to private subnets in an Amazon VPC.
- ☑ Create a resource-based policy for Lambda to restrict internet access.
- □ Create a resource-based policy for DynamoDB to restrict access to the Amazon VPC.
- □ Configure the Lambda function to connect to private subnets in an Amazon VPC.
- ☑ Configure a VPC endpoint for accessing the DynamoDB table using private addresses.



Incorrect

Explanation:

You can configure a Lambda function to connect to private subnets in a virtual private cloud (VPC) in your AWS account. When you do this you can invoke your function internally within the VPC without accessing the public address space. The function will also not have internet access unless you add a NAT gateway.

To secure access to DynamoDB a Gateway VPC Endpoint can be created within the VPC. This will enable the Lambda function to access the DynamoDB table using private addresses which meets the requirements of the question.

CORRECT: "Configure the Lambda function to connect to private subnets in an Amazon VPC" is a correct answer (as explained above.)

CORRECT: "Configure a VPC endpoint for accessing the DynamoDB table using private addresses" is also a correct answer (as explained above.)

INCORRECT: "Create a resource-based policy for Lambda to restrict internet access" is incorrect.

You cannot create resource based IAM policies on Lambda and so this is not a method of restricting permissions or internet access.

INCORRECT: "Configure the DynamoDB table to connect to private subnets in an Amazon VPC" is incorrect.

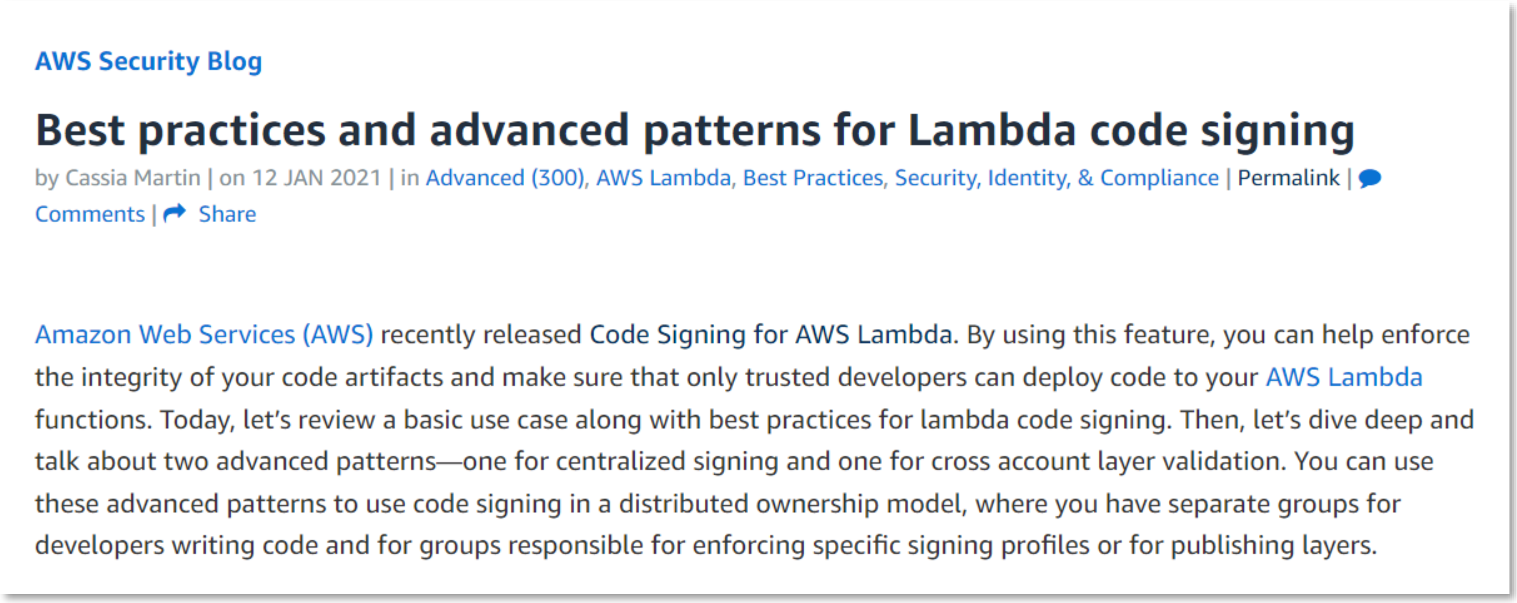
You cannot configure DynamoDB tables to connect to private subnets. You can connect *from* a private subnet to DynamoDB using a VPC endpoint.

INCORRECT: "Create a resource-based policy for DynamoDB to restrict access to the Amazon VPC" is incorrect.

DynamoDB doesn't support resource-based policies.

References:

https://docs.aws.amazon.com/lambda/latest/dg/configuration-vpc.html

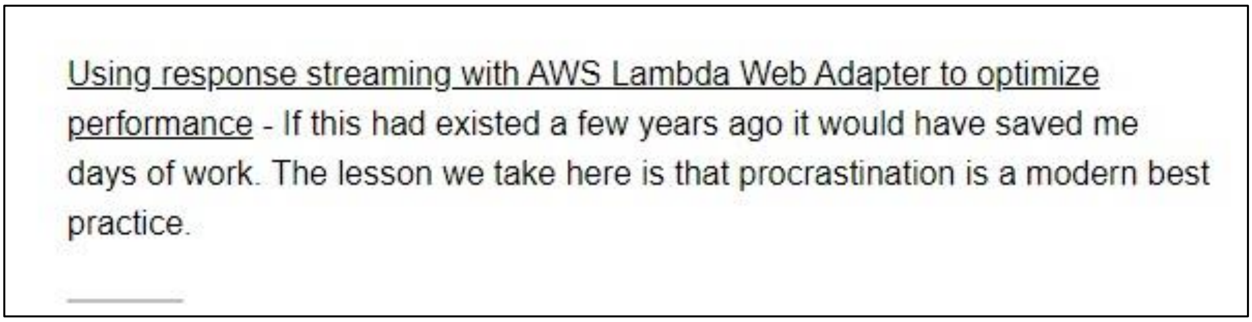


AWS Security Blog

Best practices and advanced patterns for Lambda code signing

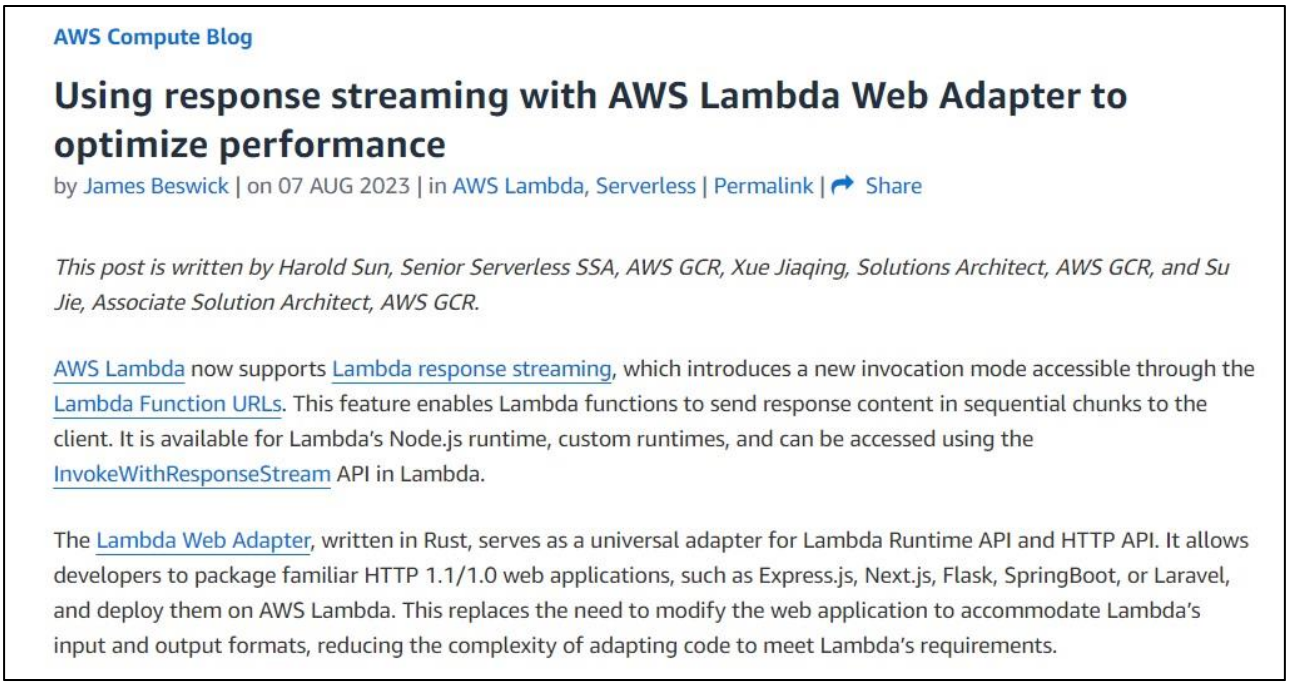
by Cassia Martin | on 12 JAN 2021 | in Advanced (300), AWS Lambda, Best Practices, Security, Identity, & Compliance | Permalink | Comments | Share

Amazon Web Services (AWS) recently released Code Signing for AWS Lambda. By using this feature, you can help enforce the integrity of your code artifacts and make sure that only trusted developers can deploy code to your AWS Lambda functions. Today, let's review a basic use case along with best practices for lambda code signing. Then, let's dive deep and talk about two advanced patterns—one for centralized signing and one for cross account layer validation. You can use these advanced patterns to use code signing in a distributed ownership model, where you have separate groups for developers writing code and for groups responsible for enforcing specific signing profiles or for publishing layers.



Using response streaming with AWS Lambda Web Adapter to optimize performance - If this had existed a few years ago it would have saved me days of work. The lesson we take here is that procrastination is a modern best practice.

Corey Quinn writing in an email on 14th August 2023



AWS Compute Blog

Using response streaming with AWS Lambda Web Adapter to optimize performance

by James Beswick | on 07 AUG 2023 | in AWS Lambda, Serverless | Permalink | ➦ Share

This post is written by Harold Sun, Senior Serverless SSA, AWS GCR, Xue Jiaqing, Solutions Architect, AWS GCR, and Su Jie, Associate Solution Architect, AWS GCR.

AWS Lambda now supports Lambda response streaming, which introduces a new invocation mode accessible through the Lambda Function URLs. This feature enables Lambda functions to send response content in sequential chunks to the client. It is available for Lambda's Node.js runtime, custom runtimes, and can be accessed using the InvokeWithResponseStream API in Lambda.

The Lambda Web Adapter, written in Rust, serves as a universal adapter for Lambda Runtime API and HTTP API. It allows developers to package familiar HTTP 1.1/1.0 web applications, such as Express.js, Next.js, Flask, SpringBoot, or Laravel, and deploy them on AWS Lambda. This replaces the need to modify the web application to accommodate Lambda's input and output formats, reducing the complexity of adapting code to meet Lambda's requirements.

In the above article we are told that Web Adapter is an extension to AWS Lambda. Now, what is the Web Adapter? Well, it is an adapter for the Lambda runtime API and HTTP API. "It allows developers to package familiar HTTP 1.1/1.0 web applications, such as Express.js, Next.js, Flask, Springboot, or Laravel and deploy them on AWS Lambda."

You might still be confused about what, exactly, the web adapter is. So, let's continue. We are told that the Lambda Web Adapter:

> Replaces the need to modify the web application to accommodate Lambda's input and output formats

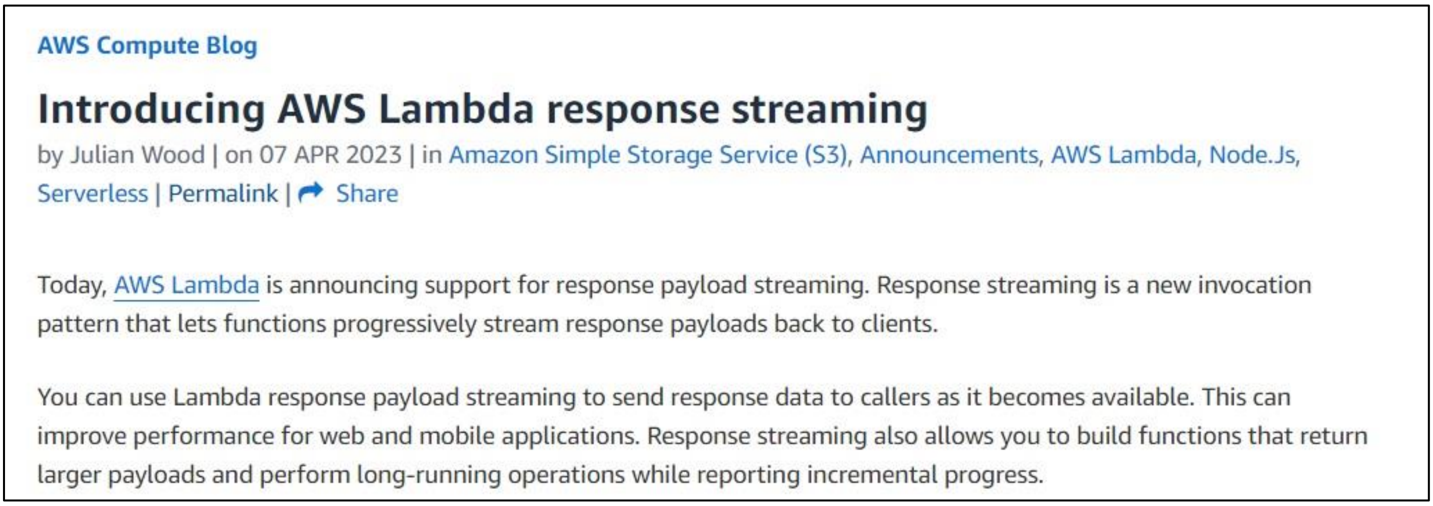
So, it really is a mediator. It allows web applications which would usually need to be modified, if they are to integrate with Lambda, to be used easily with Lambda. This is what the Lambda Web Adapter achieves. Let's look at what is does in more concrete terms now:

> To use Lambda Web Adapter with docker images, package your web app (http api) in a Dockerfile, and add one line to copy Lambda Web Adapter binary to /opt/extensions inside your container.
>
> By default, Lambda Web Adapter assumes the web app is listening on port 8080. If not, you can specify the port via configuration.

If you read the extract above, it is clear that in order to adapt your application to AWS Lambda, you may package your web app into a Dockerfile. Then, you add a binary to your container. Specifically, you add the binary to /opt/extensions.

A web app needs to communicate with Lambda. It does this over the network, using ports. The question is therefore raised Which port is the web app listening on? Well, by default Lambda Web Adapter assumes the web app is listening on port 8080. It is possible to specify a different port, however.

What on earth is AWS Lambda response streaming?



AWS Compute Blog

Introducing AWS Lambda response streaming

by Julian Wood | on 07 APR 2023 | in Amazon Simple Storage Service (S3), Announcements, AWS Lambda, Node.Js, Serverless | Permalink | Share

Today, AWS Lambda is announcing support for response payload streaming. Response streaming is a new invocation pattern that lets functions progressively stream response payloads back to clients.

You can use Lambda response payload streaming to send response data to callers as it becomes available. This can improve performance for web and mobile applications. Response streaming also allows you to build functions that return larger payloads and perform long-running operations while reporting incremental progress.

Above we can see an AWS blogpost by Julian Wood. It was published in early 2023. It announces response streaming. The question we need to answer is this: What is response streaming?

Wood states:

> Today, AWS Lambda is announcing support for response payload streaming.
>
> Response streaming is a new invocation pattern that lets functions progressively stream response payloads back to clients.

Here we find the answer to our questions. The 'response' is the message that is sent following a *request*. To *stream* a response is to progressively send it, as opposed to sending it all in one go. So, "response streaming" refers to the phenomenon of progressively communicating a response to a client.

Now, why might we want to do this? After all, surely I would want my response all in one go. I cannot possibly imaging how a response which comes in dribs and drabs could be an *advantage*. Wood suggests that by *streaming* a response, we can overcome certain limitations:

> You can use Lambda response payload streaming to send response data to callers as it becomes available. This can improve performance for web and mobile applications.
>
> Response streaming also allows you to build functions that return larger payloads and perform long-running operations while reporting incremental progress.

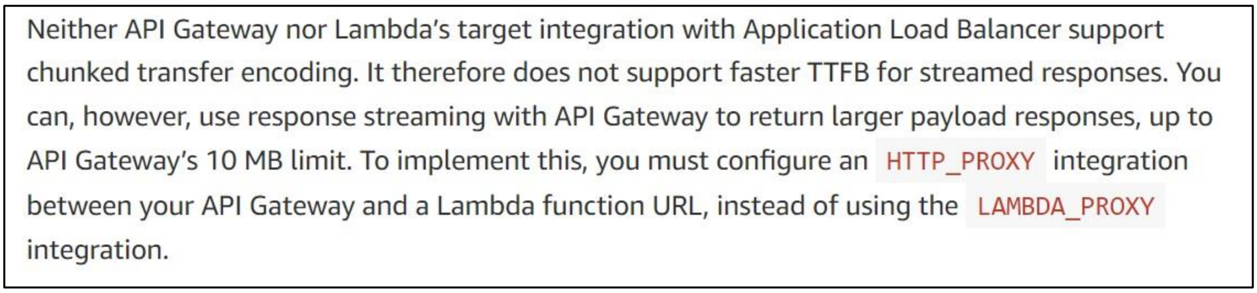
[Wood 2023]

In general, we want to minimise how long it takes to get the first byte loaded on the client's machine. The metric which assesses this concern is called the *Time to First Byte*.

We can improve the Time to First Byte, and therefore the visitor's experience, if we use response streaming:

> In traditional request-response models, the response needs to be fully generated and buffered before it is returned to the client.
>
> This can delay the time to first byte (TTFB) performance while the client waits for the response to be generated. Web applications are especially sensitive to TTFB and page load performance.
>
> Response streaming lets you send partial responses back to the client as they become ready, improving TTFB latency to within milliseconds. For web applications, this can improve visitor experience and search engine rankings.

Consider the paragraph below, which is quite confusing:



Neither API Gateway nor Lambda's target integration with Application Load Balancer support chunked transfer encoding. It therefore does not support faster TTFB for streamed responses. You can, however, use response streaming with API Gateway to return larger payload responses, up to API Gateway's 10 MB limit. To implement this, you must configure an `HTTP_PROXY` integration between your API Gateway and a Lambda function URL, instead of using the `LAMBDA_PROXY` integration.

What does it mean by Lambda's "target integration"?
What is the significance of a chunked transfer encoding?

AWS Lambda now detects and stops recursive loops in Lambda functions

Posted On: Jul 13, 2023

AWS Lambda can now detect and stop recursive loops in Lambda functions. Customers build event-driven applications using Lambda functions to process events from sources like Amazon SQS and Amazon SNS. However, in certain scenarios, due to resource misconfiguration or code defect, a processed event may be sent back to the same service or resource that invoked the Lambda function. This can cause an unintended recursive loop, and result in unintended usage and costs for customers. With this launch, Lambda will stop recursive invocations between Amazon SQS, AWS Lambda, and Amazon SNS after 16 recursive calls.

When a function sends an event to Amazon SQS or Amazon SNS using a supported AWS SDK version or higher, Lambda tracks the number of times a function has been invoked based on that event. If a function is invoked by the same triggering event more than 16 times, Lambda will stop the next invocation and sends the event to a Dead-Letter Queue or on-failure destination, if configured. Customers will also receive an AWS Health Dashboard notification with troubleshooting steps.

This feature is turned on by default, and is available in the following AWS Regions: Asia Pacific (Hong Kong, Jakarta, Osaka, Mumbai, Seoul, Singapore, Sydney, Tokyo), Africa (Cape Town), Canada (Central), Europe (Frankfurt, Ireland, London, Milan, Paris, Stockholm), South America (Sao Paulo), US East (Ohio, N.Virginia), US West (Oregon, N.California). To turn off the feature for your AWS Account, please contact AWS Support. For further information, please refer to our documentation or the launch blog post.

AWS Lambda Monitoring — A Full Guide

Maximize Your Serverless Success with the Complete AWS Lambda Monitoring Guide

Manoj Fernando · Follow

Published in AWS in Plain English · 11 min read · Jun 19

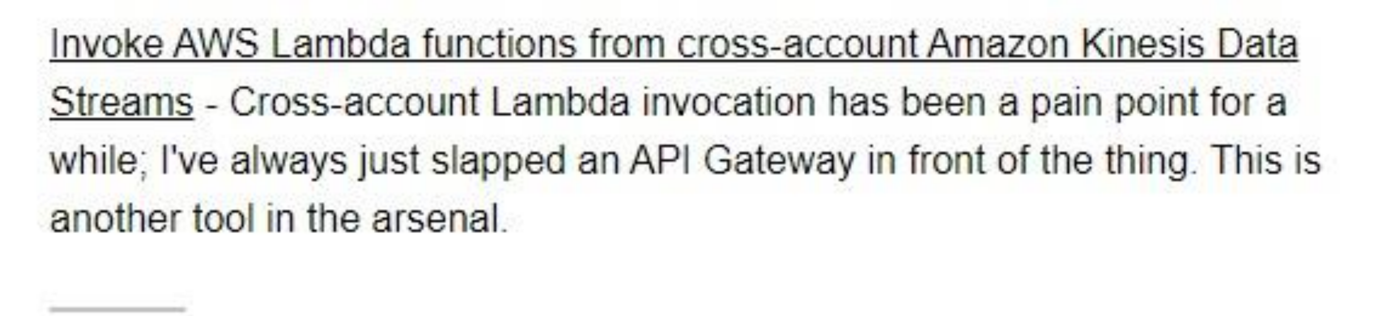
AWS Lambda supports starting from timestamp for Kafka event sources

Posted On: Jun 21, 2023

AWS Lambda now supports starting from a specific timestamp when using Amazon Managed Streaming for Apache Kafka (MSK) or Self-Managed Kafka as an event source. Previously, Kafka event source mappings could only have starting positions of trim horizon or latest. Now with starting from a timestamp, you can start processing messages at a precise point in time. This is useful for situations like Disaster Recovery, where you need a new consumer to quickly start processing where you previously left off.

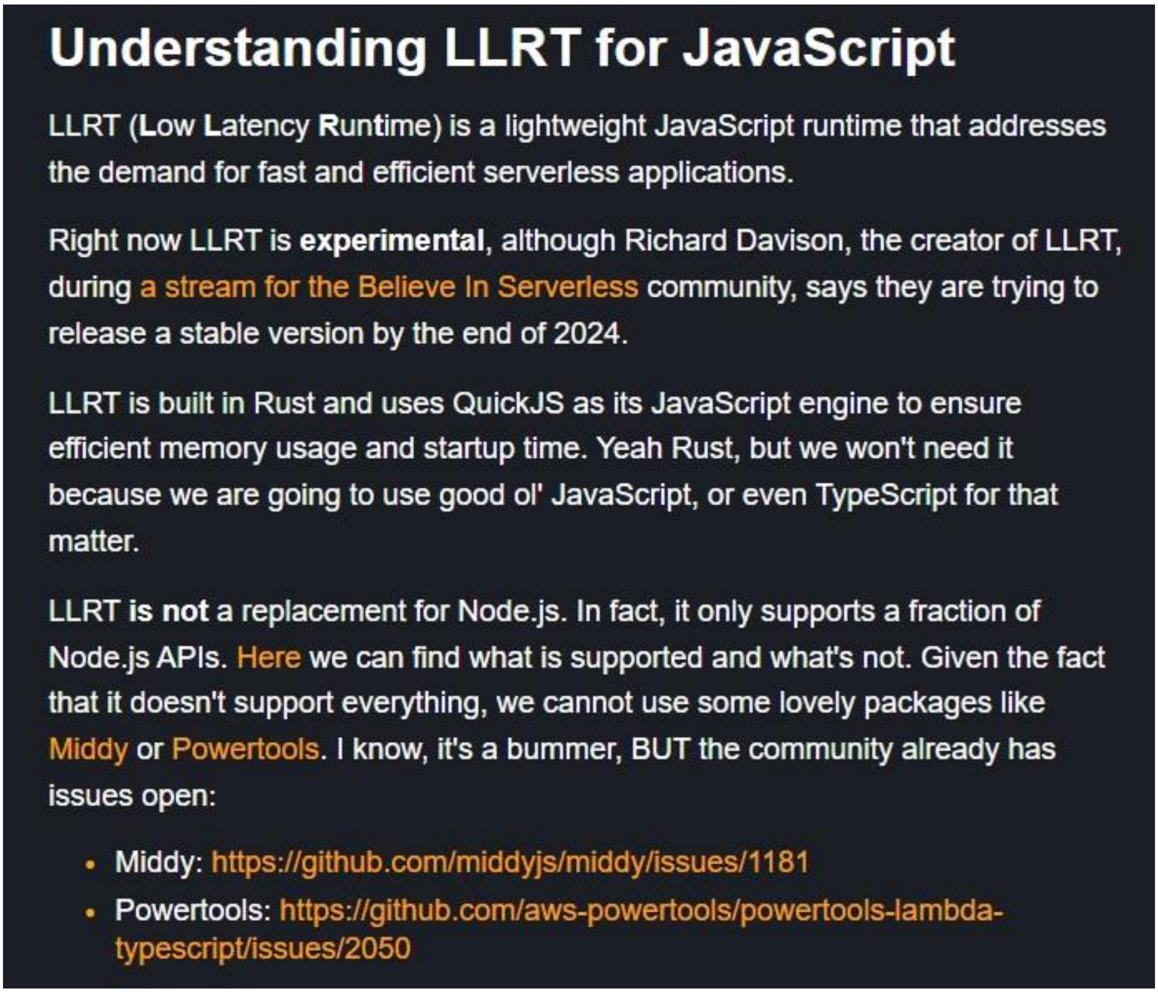
When a Kafka event source mapping is configured to start from a specific timestamp, the event source mapping will start processing messages in a topic at or the first message after the specified timestamp. To use this feature, create a new Kafka event source mapping, set the StartingPosition to AT_TIMESTAMP, and set StartingPositionTimestamp to the desired starting position. The StartingPositionTimestamp needs to be formatted in Unix time seconds. Please note that in Kafka, the starting position is only used for a new consumer group or when an existing consumer group points to an offset that is invalid (expires). New Kafka event source mappings will generate a new consumer group ID if not otherwise configured with a specific consumer group ID.

This feature incurs no additional charge. You pay for the Lambda invocations triggered by the event source mapping connected to Kafka. To learn more, see the Lambda Developer Guide for Amazon MSK and Apache Kafka.



Invoke AWS Lambda functions from cross-account Amazon Kinesis Data Streams - Cross-account Lambda invocation has been a pain point for a while; I've always just slapped an API Gateway in front of the thing. This is another tool in the arsenal.

Corey Quinn writing on 25th March 2024



Understanding LLRT for JavaScript

LLRT (**L**ow **L**atency **R**un**t**ime) is a lightweight JavaScript runtime that addresses the demand for fast and efficient serverless applications.

Right now LLRT is **experimental**, although Richard Davison, the creator of LLRT, during a stream for the Believe In Serverless community, says they are trying to release a stable version by the end of 2024.

LLRT is built in Rust and uses QuickJS as its JavaScript engine to ensure efficient memory usage and startup time. Yeah Rust, but we won't need it because we are going to use good ol' JavaScript, or even TypeScript for that matter.

LLRT **is not** a replacement for Node.js. In fact, it only supports a fraction of Node.js APIs. Here we can find what is supported and what's not. Given the fact that it doesn't support everything, we cannot use some lovely packages like Middy or Powertools. I know, it's a bummer, BUT the community already has issues open:

- Middy: https://github.com/middyjs/middy/issues/1181
- Powertools: https://github.com/aws-powertools/powertools-lambda-typescript/issues/2050

[Depascale 2024]

Bibliography

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

I. Official

[Surname1]

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

[Wood 2020]

Wood, Julian (2020). Choosing Events, Queues, Topics and Streams. April 2020. Available at:
<https://www.youtube.com/watch?v=d9Jb1WKCLd8&ab_channel=AWSOnlineTechTalks>.

[Brooker 2020]

Brooker, Adrian Costin and Marc Brooker (2020). How AWS's Firecracker virtual machines work. Available at:
<https://www.amazon.science/blog/how-awss-firecracker-virtual-machines-work?utm_source=substack&utm_medium=email>

[Verbitski 2018]

Verbitski et al. (2018). **Amazon Aurora: On Avoiding Distributed Consensus for I/Os, Commits, and Membership Changes**. SIGMOD 2018, 10-15 June.

[Sun 2023]

Sun, Harold (2023). Using Response Streaming with AWS Lambda Web Adapter to optimize performance. *AWS Compute Blog*. Aug 7th 2023. Available at: <https://aws.amazon.com/blogs/compute/using-response-streaming-with-aws-lambda-web-adapter-to-optimize-performance/?ck_subscriber_id=1560524742>

[Wood 2023]

Wood, Julian (2023). Introducing AWS Lambda response streaming. *AWS Compute Blog*. Apr 7th 2023. Available at:
<https://aws.amazon.com/blogs/compute/introducing-aws-lambda-response-streaming/>

[AWS 2023]

AWS Lambda now detects and stops recursive loops in Lambda functions. [Announcement]. July 13th 2023. Available at: https://aws.amazon.com/about-aws/whats-new/2023/07/aws-lambda-detects-recursive-loops-lambda-functions/?ck_subscriber_id=1560524742

[Mesrobian 2018]

Mesrobian, Holly and Marc Brooker (2018). A Serverless Journey: Under the Hood of AWS Lambda. *Reinvent 2018* [Conference]. Available at: <https://www.youtube.com/watch?v=QdzV04T_kec&ab_channel=AmazonWebServices>

II. Unofficial

[Fernando 2023]

Fernando, Manoj (2023). AWS Lambda Monitoring – A Full Guide. Available at: <https://aws.plainenglish.io/aws-lambda-monitoring-a-full-guide-3cc68c6052fd>

[Lakomy 2022]

Guide to AWS Lambda Function URLs. Available at: <https://cloudash.dev/blog/guide-to-lambda-function-urls?ck_subscriber_id=1560524742 >

[Surname1]

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

[Surname1]

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

https://code.tutsplus.com/tutorials/http-headers-for-dummies--net-8039

Using Golang for your AWS Lambda Functions. Available at: https://dev.to/aws-builders/using-golang-for-your-aws-lambda-functions-3ko1?utm_source=substack&utm_medium=email

[Depascale 2024]

Depascale, Matteo (2024). Building Lightning-Fast AWS Lambda Functions with LLRT and Terraform. Available at: <https://cloudnature.net/blog/aws-lambda-llrt-terraform?utm_source=substack&utm_medium=email>

III. Critical

[Surname1]

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

[Surname1]

Smith, David (year). Title of Work Here. 1st 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.

IV. General

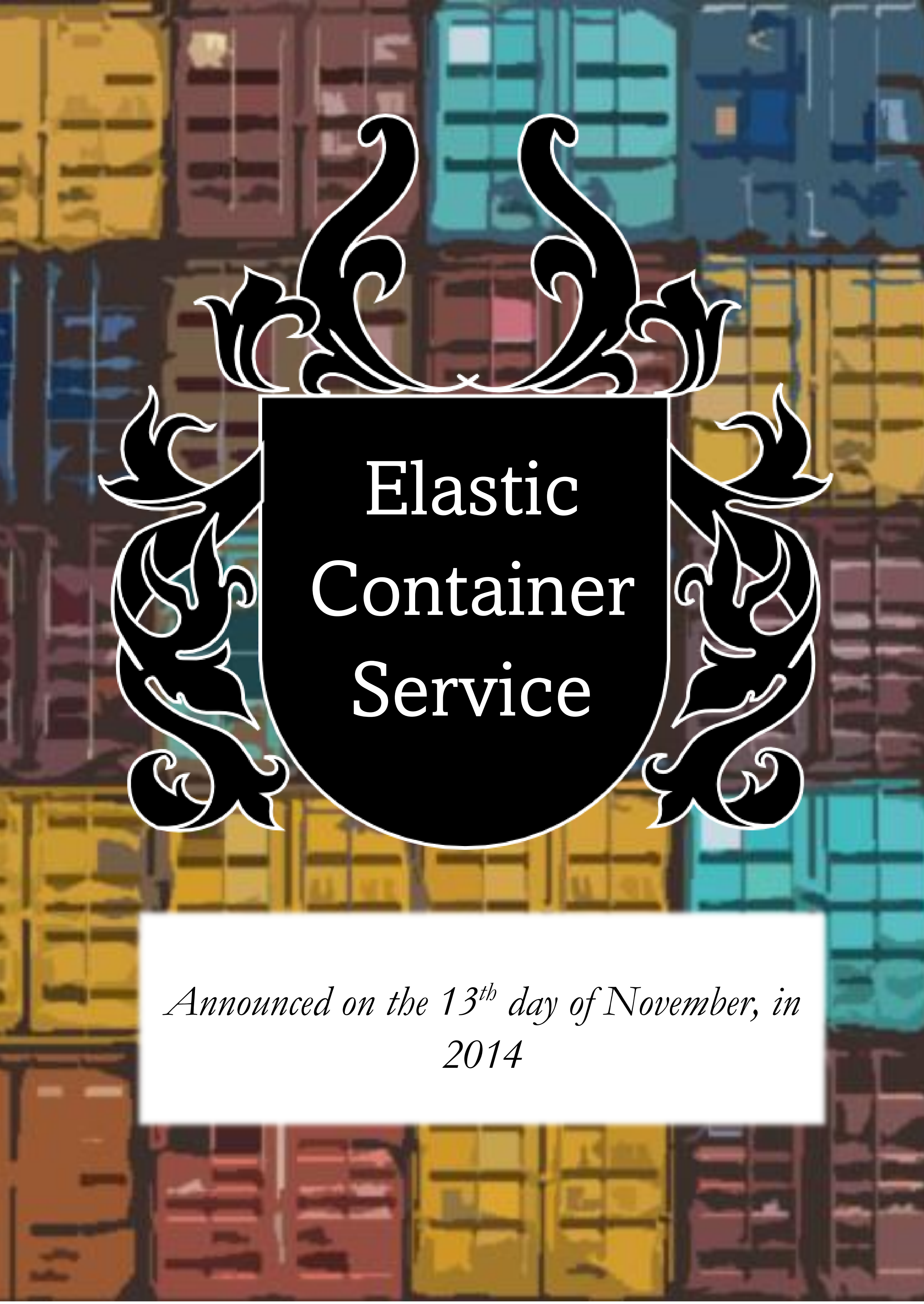
[Surname1]

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

[Surname1]

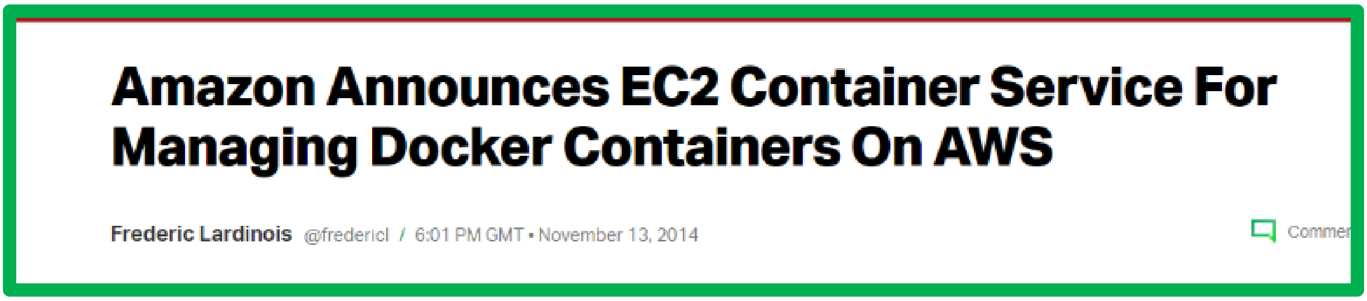
Smith, David (year). Title of Work Here. 1st Jan 2022. City: Publisher. Available at:
<URL here>.

Elastic Container Service



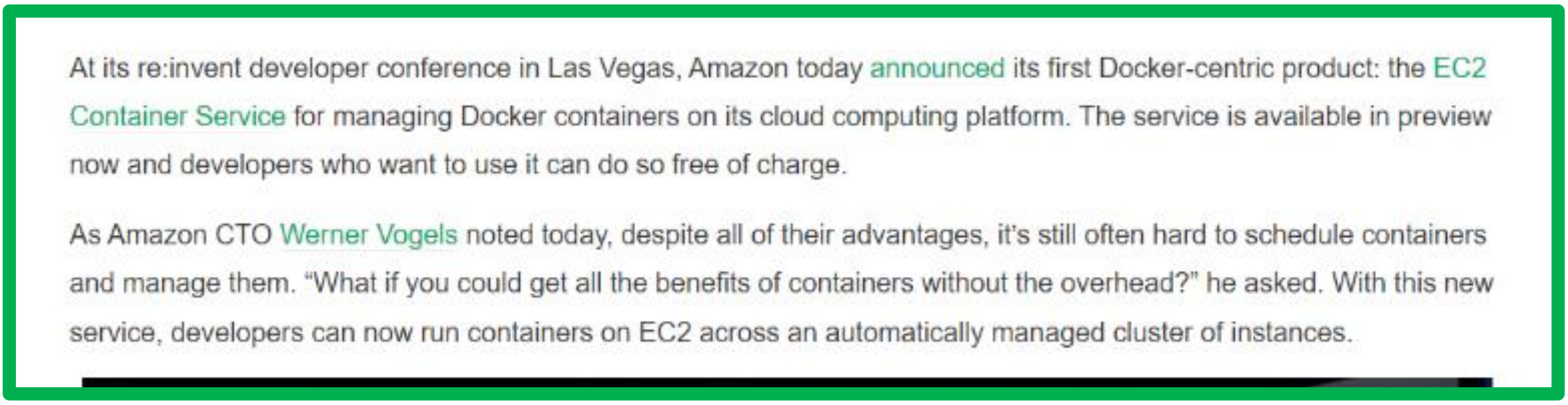
Elastic Container Service

Announced on the 13th day of November, in 2014



Amazon Announces EC2 Container Service For Managing Docker Containers On AWS

Frederic Lardinois @fredericl / 6:01 PM GMT • November 13, 2014



At its re:invent developer conference in Las Vegas, Amazon today announced its first Docker-centric product: the EC2 Container Service for managing Docker containers on its cloud computing platform. The service is available in preview now and developers who want to use it can do so free of charge.

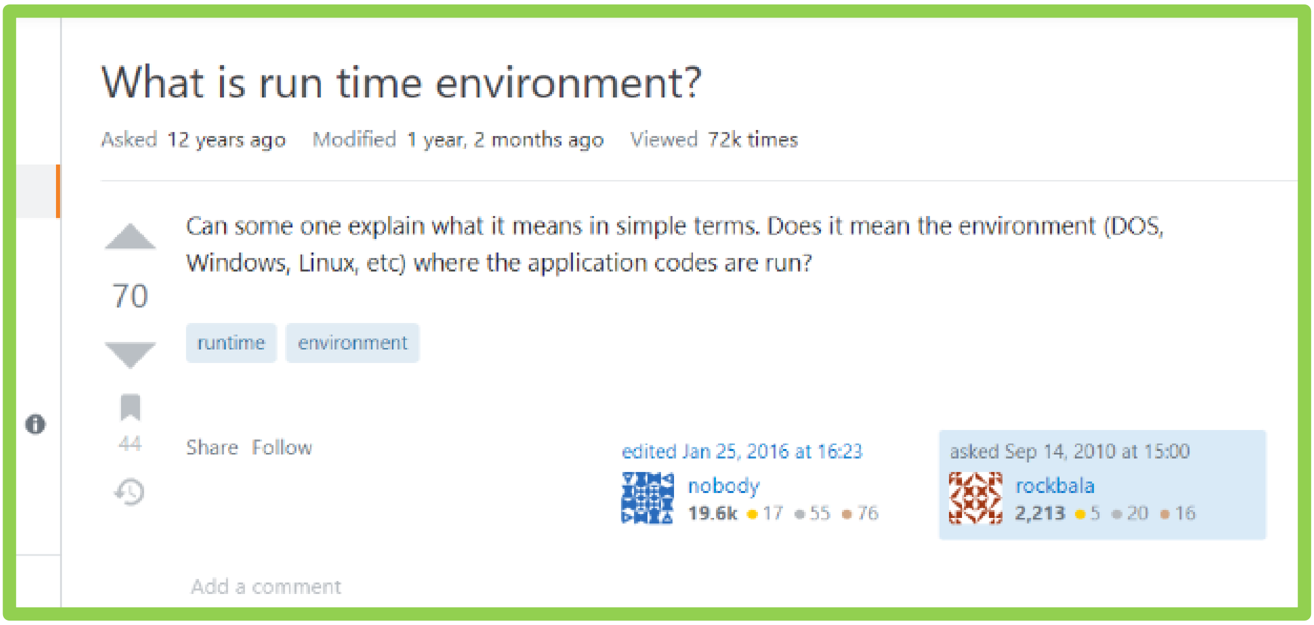
As Amazon CTO Werner Vogels noted today, despite all of their advantages, it's still often hard to schedule containers and manage them. "What if you could get all the benefits of containers without the overhead?" he asked. With this new service, developers can now run containers on EC2 across an automatically managed cluster of instances.

Paul Duffy explaining how the "EC2 container service" works, on stage at Re:Invent in 2014

TEN QUESTIONS

An important briefing which will prime you to understand Amazon's ECS

1. **What is a container?**
2. **What is Docker?**
3. **What is Kubernetes?**
4. **What is a scheduler?**
5. **What is a runtime environment?**



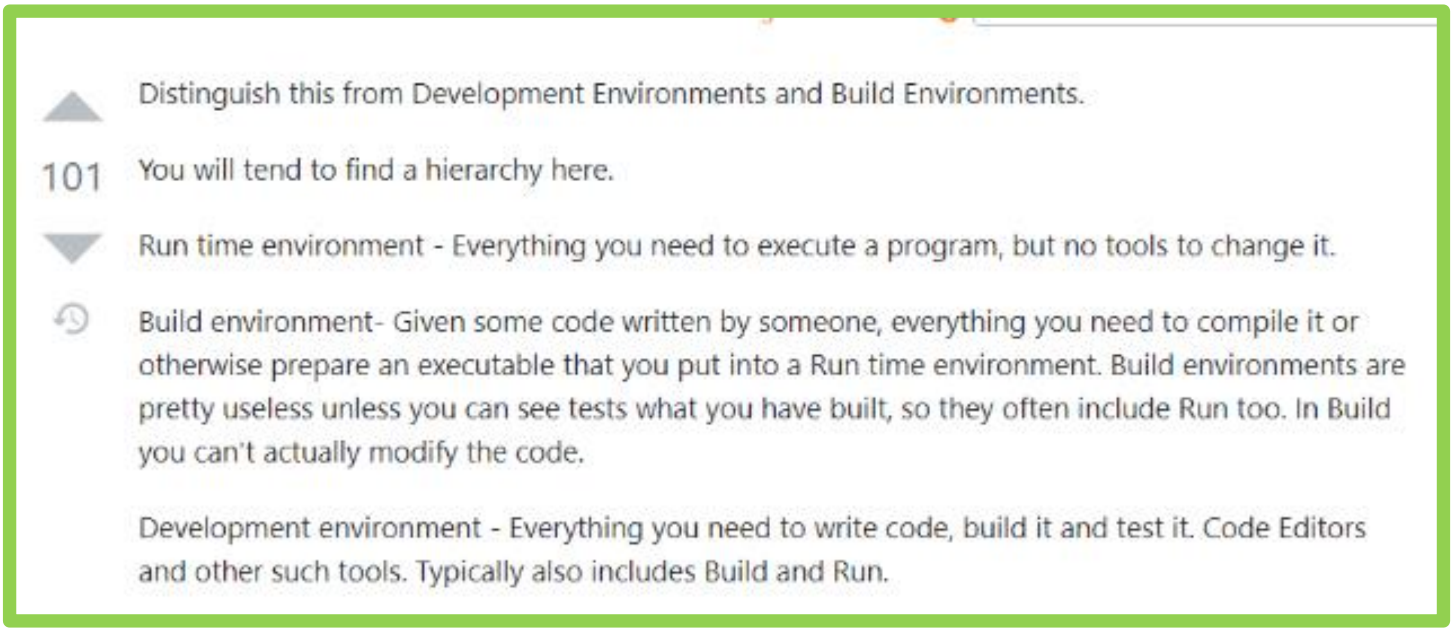
What is run time environment?

Asked 12 years ago Modified 1 year, 2 months ago Viewed 72k times

Can some one explain what it means in simple terms. Does it mean the environment (DOS, Windows, Linux, etc) where the application codes are run?

edited Jan 25, 2016 at 16:23
nobody
19.6k ● 17 ● 55 ● 76

asked Sep 14, 2010 at 15:00
rockbala
2,213 ● 5 ● 20 ● 16



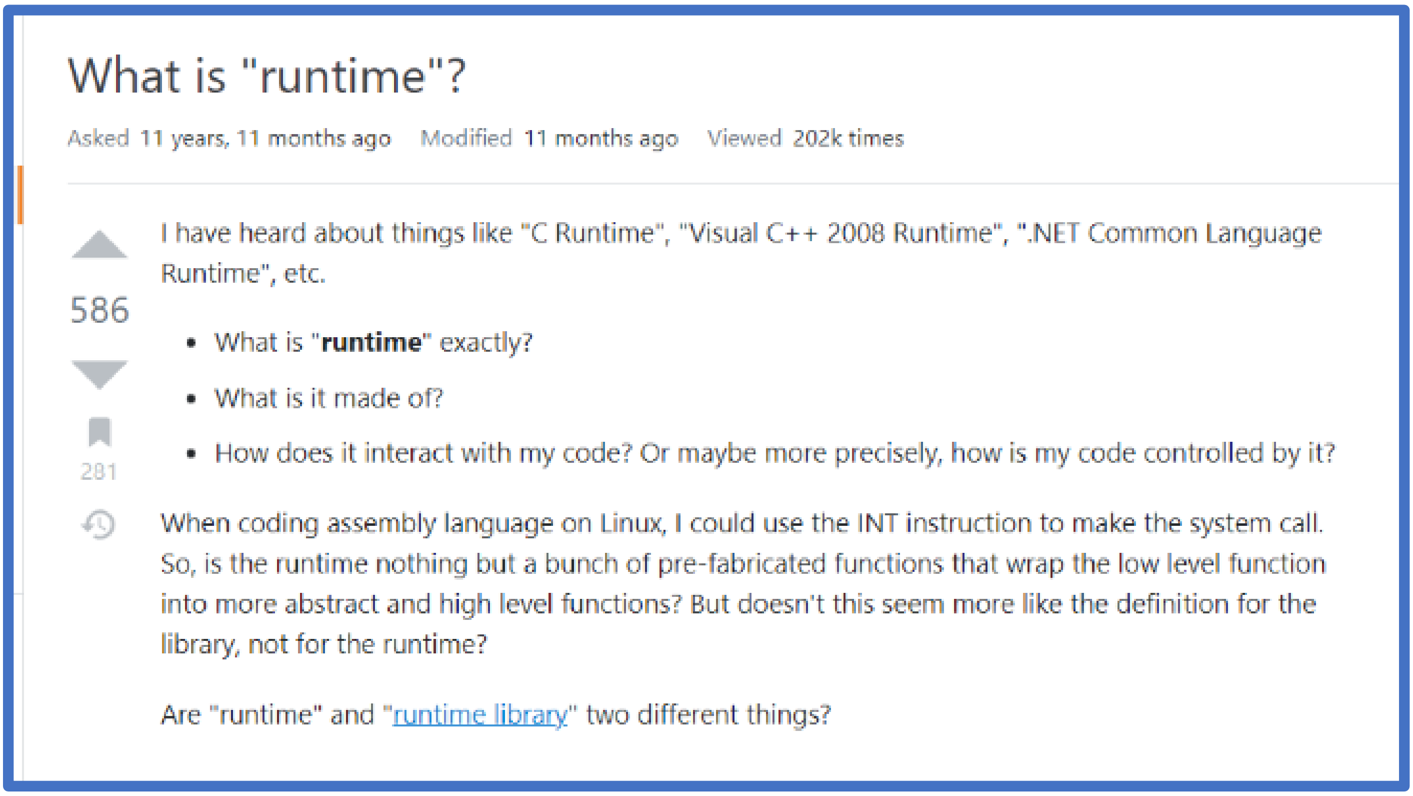
Distinguish this from Development Environments and Build Environments.

You will tend to find a hierarchy here.

Run time environment - Everything you need to execute a program, but no tools to change it.

Build environment- Given some code written by someone, everything you need to compile it or otherwise prepare an executable that you put into a Run time environment. Build environments are pretty useless unless you can see tests what you have built, so they often include Run too. In Build you can't actually modify the code.

Development environment - Everything you need to write code, build it and test it. Code Editors and other such tools. Typically also includes Build and Run.



What is "runtime"?

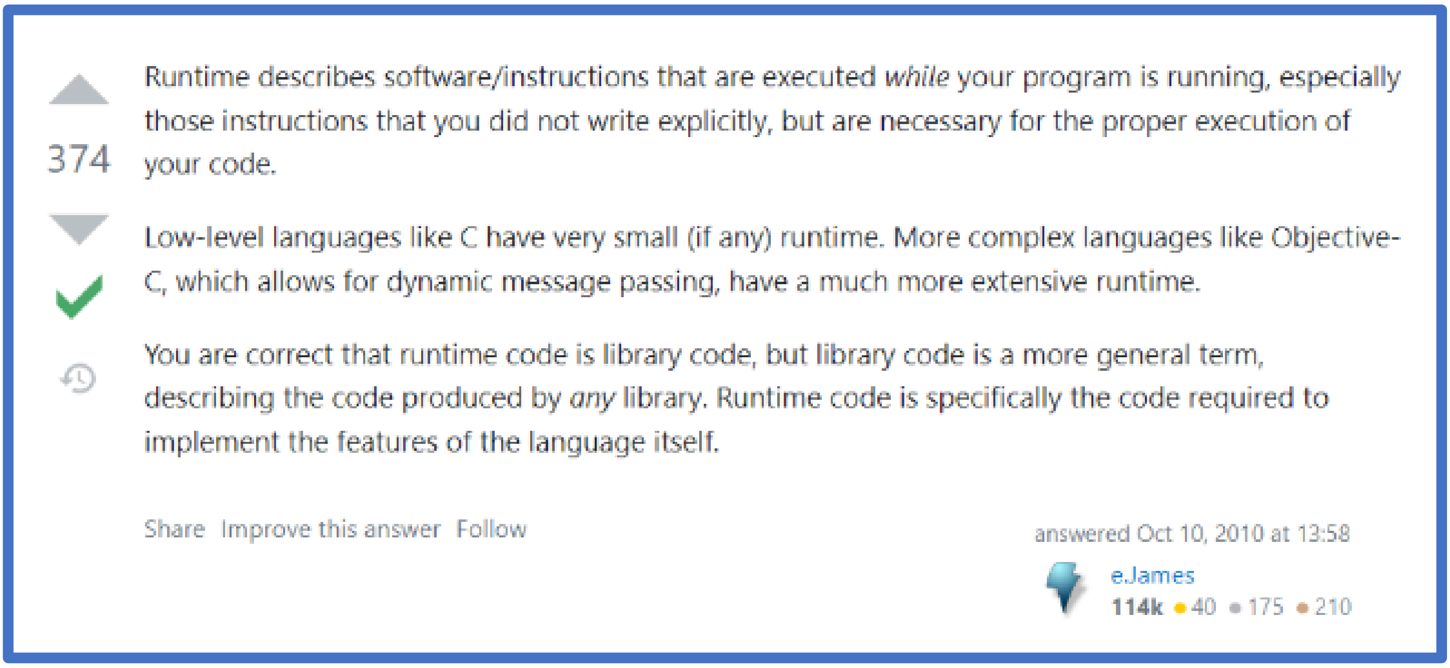
Asked 11 years, 11 months ago Modified 11 months ago Viewed 202k times

I have heard about things like "C Runtime", "Visual C++ 2008 Runtime", ".NET Common Language Runtime", etc.

- What is "**runtime**" exactly?
- What is it made of?
- How does it interact with my code? Or maybe more precisely, how is my code controlled by it?

When coding assembly language on Linux, I could use the INT instruction to make the system call. So, is the runtime nothing but a bunch of pre-fabricated functions that wrap the low level function into more abstract and high level functions? But doesn't this seem more like the definition for the library, not for the runtime?

Are "runtime" and "runtime library" two different things?



Runtime describes software/instructions that are executed *while* your program is running, especially those instructions that you did not write explicitly, but are necessary for the proper execution of your code.

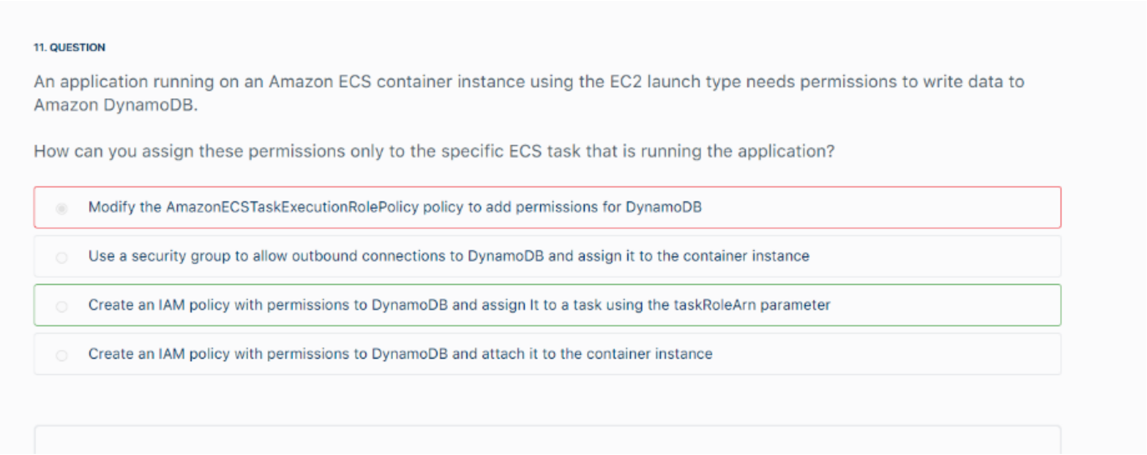
Low-level languages like C have very small (if any) runtime. More complex languages like Objective-C, which allows for dynamic message passing, have a much more extensive runtime.

You are correct that runtime code is library code, but library code is a more general term, describing the code produced by *any* library. Runtime code is specifically the code required to implement the features of the language itself.

answered Oct 10, 2010 at 13:58

e.James
114k ● 40 ● 175 ● 210

TASKMASTER



11. QUESTION

An application running on an Amazon ECS container instance using the EC2 launch type needs permissions to write data to Amazon DynamoDB.

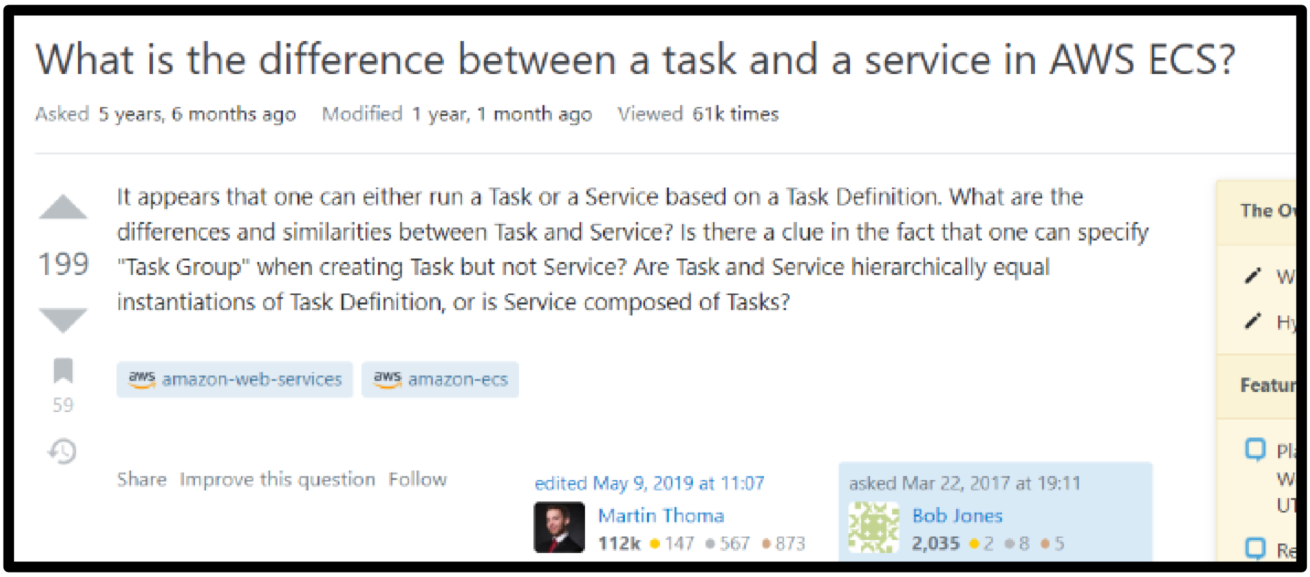
How can you assign these permissions only to the specific ECS task that is running the application?

○ Modify the AmazonECSTaskExecutionRolePolicy policy to add permissions for DynamoDB

○ Use a security group to allow outbound connections to DynamoDB and assign it to the container instance

○ Create an IAM policy with permissions to DynamoDB and assign It to a task using the taskRoleArn parameter

○ Create an IAM policy with permissions to DynamoDB and attach it to the container instance

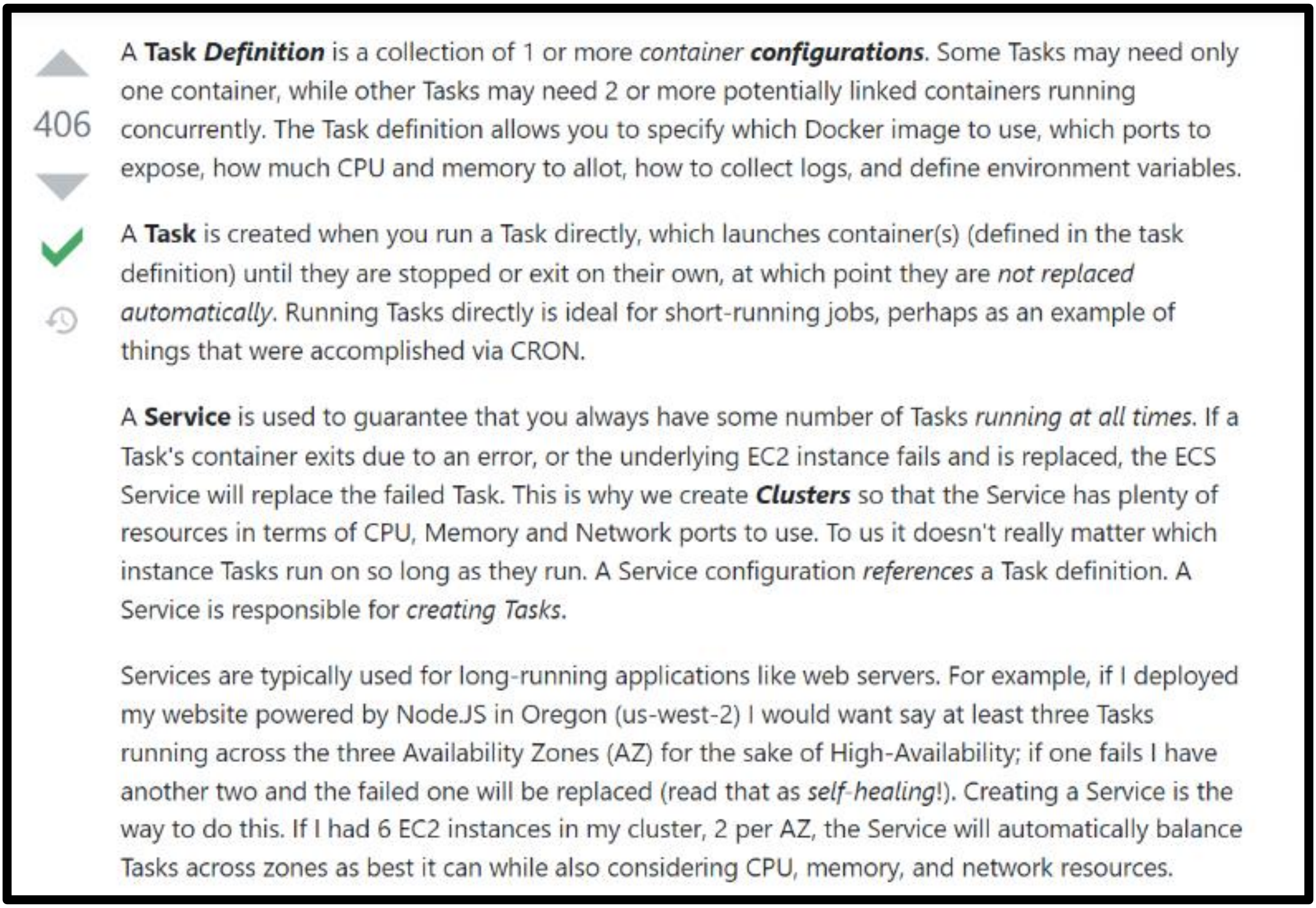


What is the difference between a task and a service in AWS ECS?

Asked 5 years, 6 months ago Modified 1 year, 1 month ago Viewed 61k times

199

It appears that one can either run a Task or a Service based on a Task Definition. What are the differences and similarities between Task and Service? Is there a clue in the fact that one can specify "Task Group" when creating Task but not Service? Are Task and Service hierarchically equal instantiations of Task Definition, or is Service composed of Tasks?

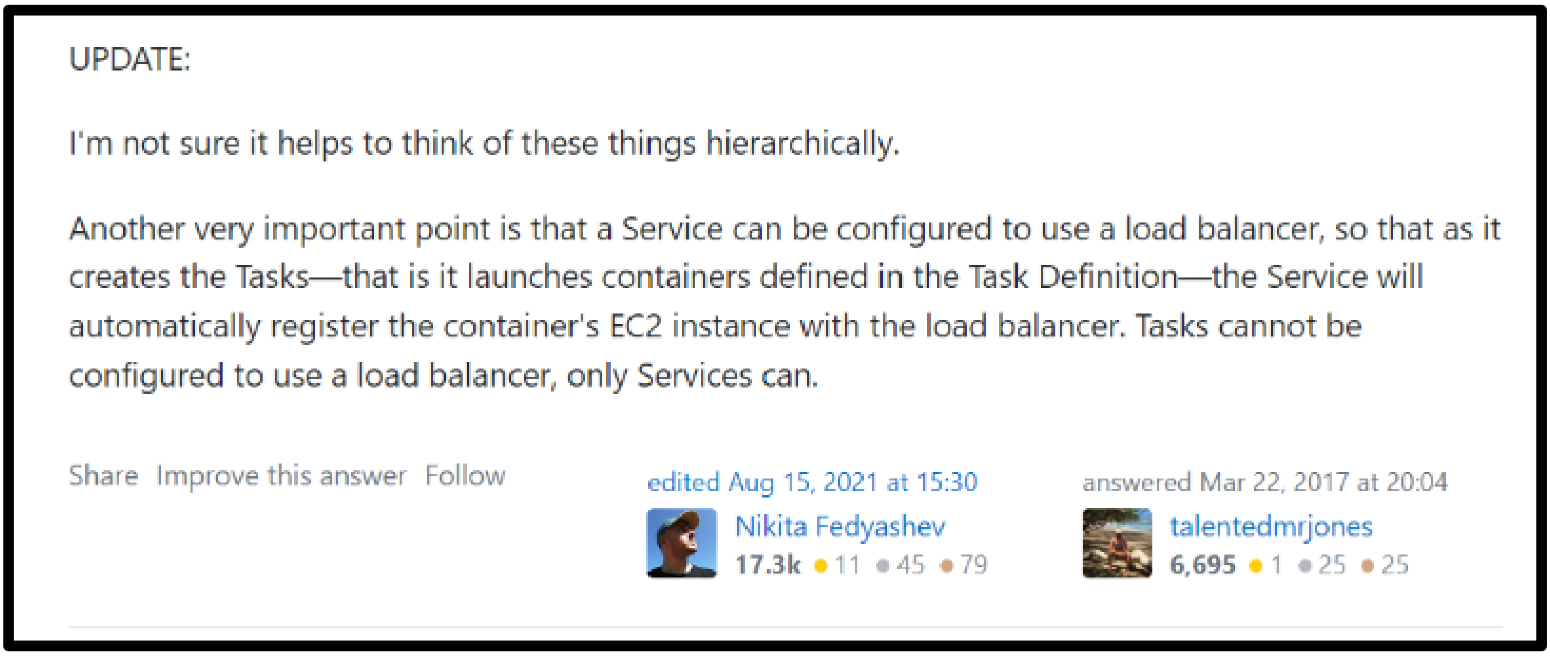


A **Task *Definition*** is a collection of 1 or more *container* ***configurations***. Some Tasks may need only one container, while other Tasks may need 2 or more potentially linked containers running concurrently. The Task definition allows you to specify which Docker image to use, which ports to expose, how much CPU and memory to allot, how to collect logs, and define environment variables.

A **Task** is created when you run a Task directly, which launches container(s) (defined in the task definition) until they are stopped or exit on their own, at which point they are *not replaced automatically*. Running Tasks directly is ideal for short-running jobs, perhaps as an example of things that were accomplished via CRON.

A **Service** is used to guarantee that you always have some number of Tasks *running at all times*. If a Task's container exits due to an error, or the underlying EC2 instance fails and is replaced, the ECS Service will replace the failed Task. This is why we create ***Clusters*** so that the Service has plenty of resources in terms of CPU, Memory and Network ports to use. To us it doesn't really matter which instance Tasks run on so long as they run. A Service configuration *references* a Task definition. A Service is responsible for *creating Tasks*.

Services are typically used for long-running applications like web servers. For example, if I deployed my website powered by NodeJS in Oregon (us-west-2) I would want say at least three Tasks running across the three Availability Zones (AZ) for the sake of High-Availability; if one fails I have another two and the failed one will be replaced (read that as *self-healing*!). Creating a Service is the way to do this. If I had 6 EC2 instances in my cluster, 2 per AZ, the Service will automatically balance Tasks across zones as best it can while also considering CPU, memory, and network resources.



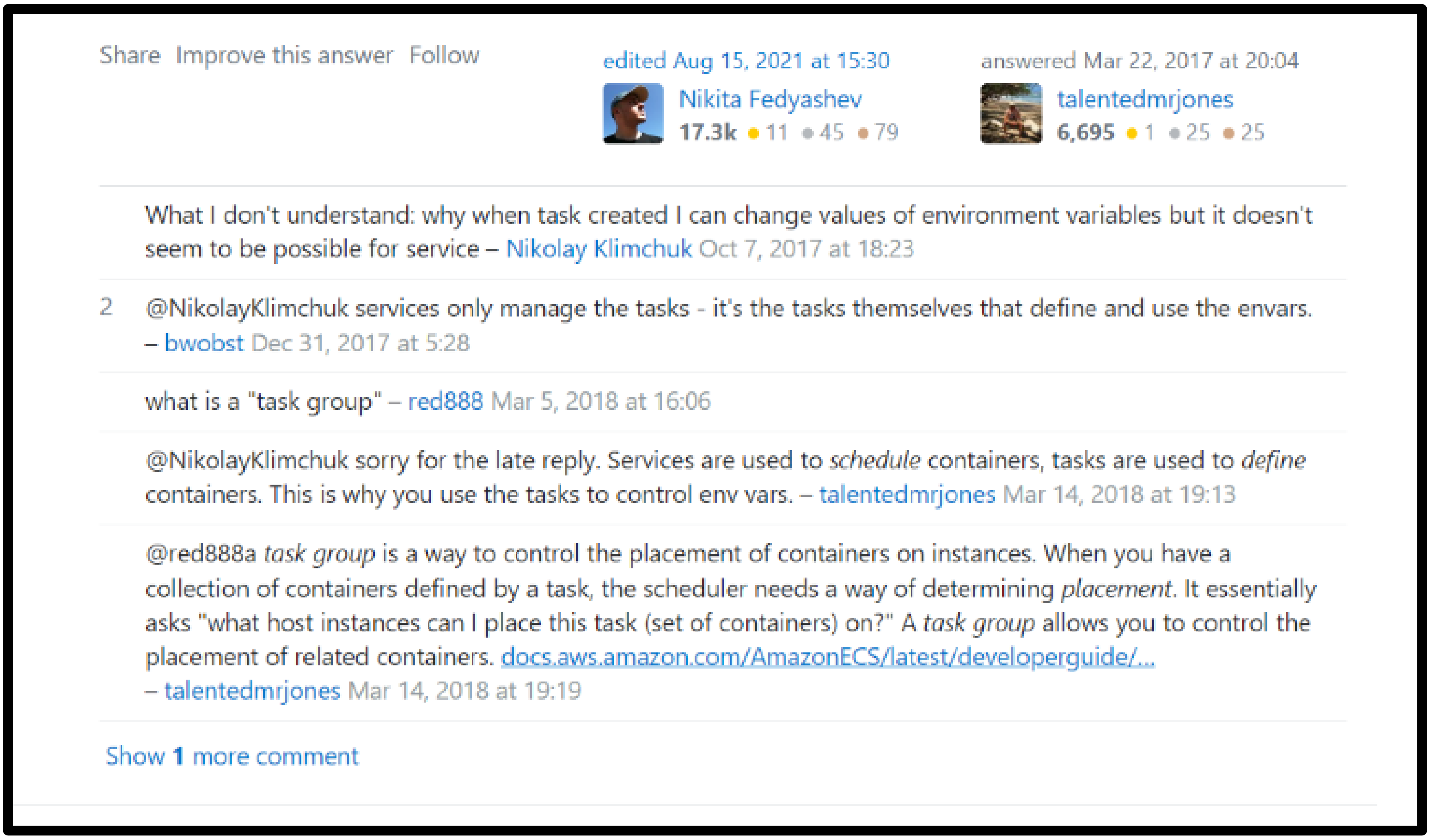
UPDATE:

I'm not sure it helps to think of these things hierarchically.

Another very important point is that a Service can be configured to use a load balancer, so that as it creates the Tasks—that is it launches containers defined in the Task Definition—the Service will automatically register the container's EC2 instance with the load balancer. Tasks cannot be configured to use a load balancer, only Services can.

edited Aug 15, 2021 at 15:30
Nikita Fedyashev
17.3k • 11 • 45 • 79

answered Mar 22, 2017 at 20:04
talentedmrjones
6,695 • 1 • 25 • 25



Share Improve this answer Follow

edited Aug 15, 2021 at 15:30
Nikita Fedyashev
17.3k ● 11 ● 45 ● 79

answered Mar 22, 2017 at 20:04
talentedmrjones
6,695 ● 1 ● 25 ● 25

What I don't understand: why when task created I can change values of environment variables but it doesn't seem to be possible for service – Nikolay Klimchuk Oct 7, 2017 at 18:23

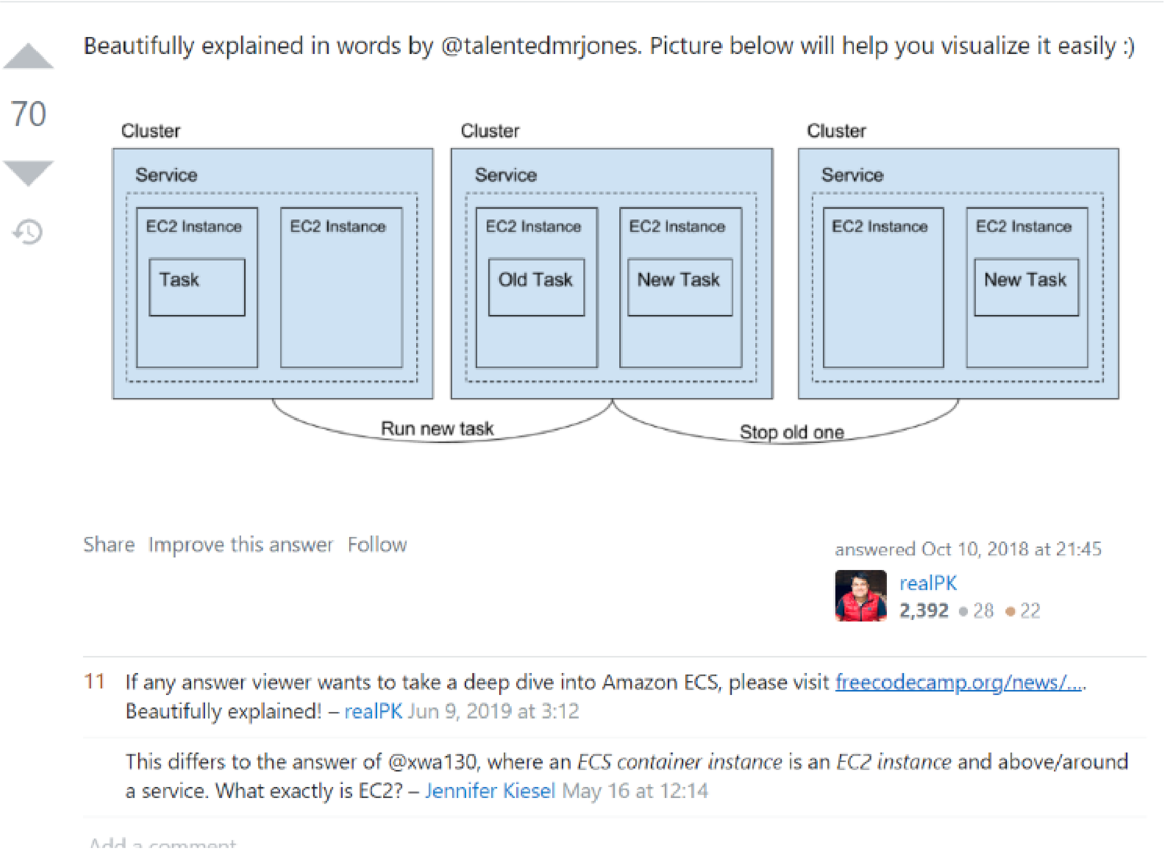
2 @NikolayKlimchuk services only manage the tasks - it's the tasks themselves that define and use the envars. – bwobst Dec 31, 2017 at 5:28

what is a "task group" – red888 Mar 5, 2018 at 16:06

@NikolayKlimchuk sorry for the late reply. Services are used to *schedule* containers, tasks are used to *define* containers. This is why you use the tasks to control env vars. – talentedmrjones Mar 14, 2018 at 19:13

@red888a *task group* is a way to control the placement of containers on instances. When you have a collection of containers defined by a task, the scheduler needs a way of determining *placement*. It essentially asks "what host instances can I place this task (set of containers) on?" A *task group* allows you to control the placement of related containers. docs.aws.amazon.com/AmazonECS/latest/developerguide/... – talentedmrjones Mar 14, 2018 at 19:19

Show **1** more comment



70

Beautifully explained in words by @talentedmrjones. Picture below will help you visualize it easily :)

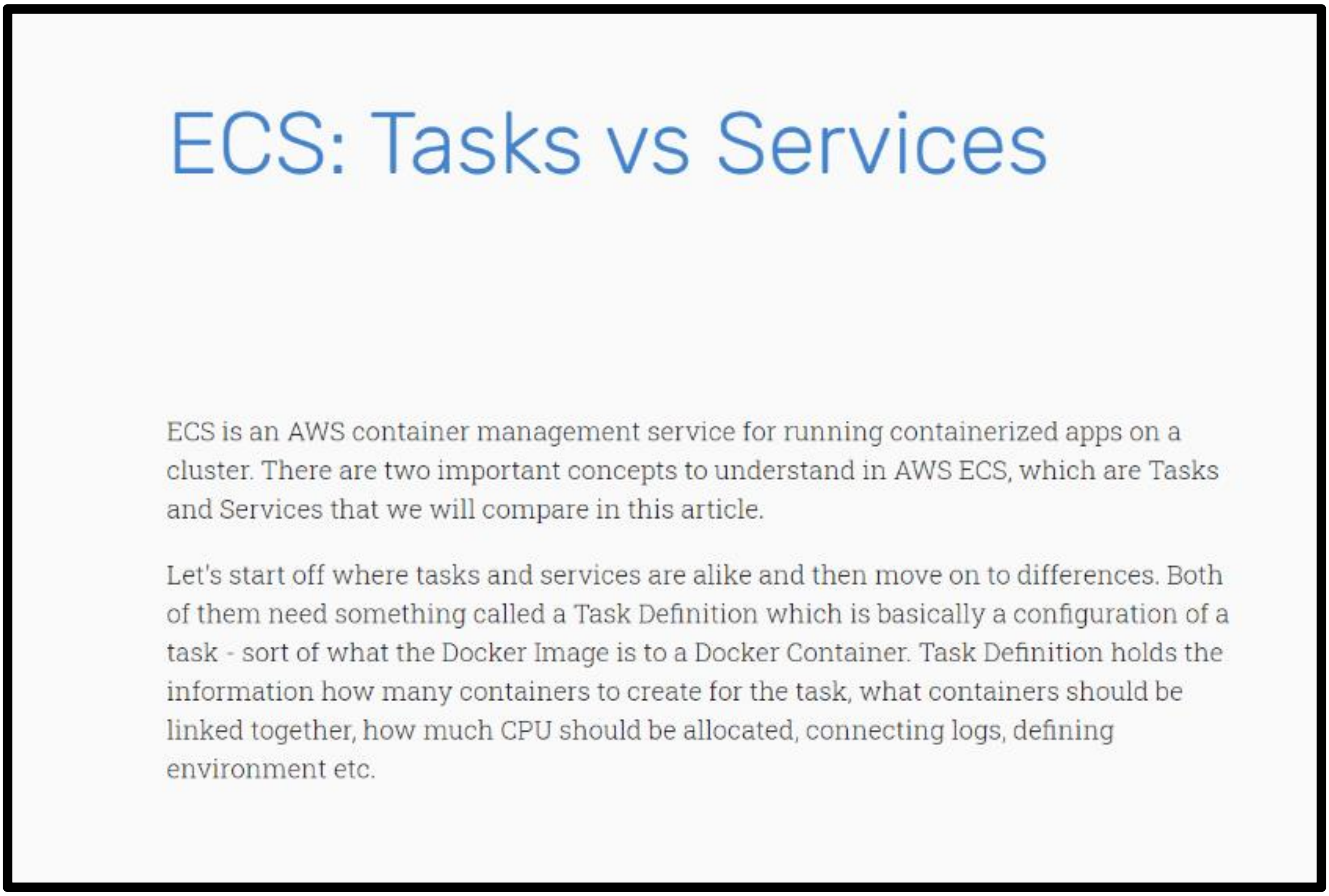
Share Improve this answer Follow

answered Oct 10, 2018 at 21:45
realPK
2,392 ● 28 ● 22

11 If any answer viewer wants to take a deep dive into Amazon ECS, please visit freecodecamp.org/news/.... Beautifully explained! – realPK Jun 9, 2019 at 3:12

This differs to the answer of @xwa130, where an *ECS container instance* is an *EC2 instance* and above/around a service. What exactly is EC2? – Jennifer Kiesel May 16 at 12:14

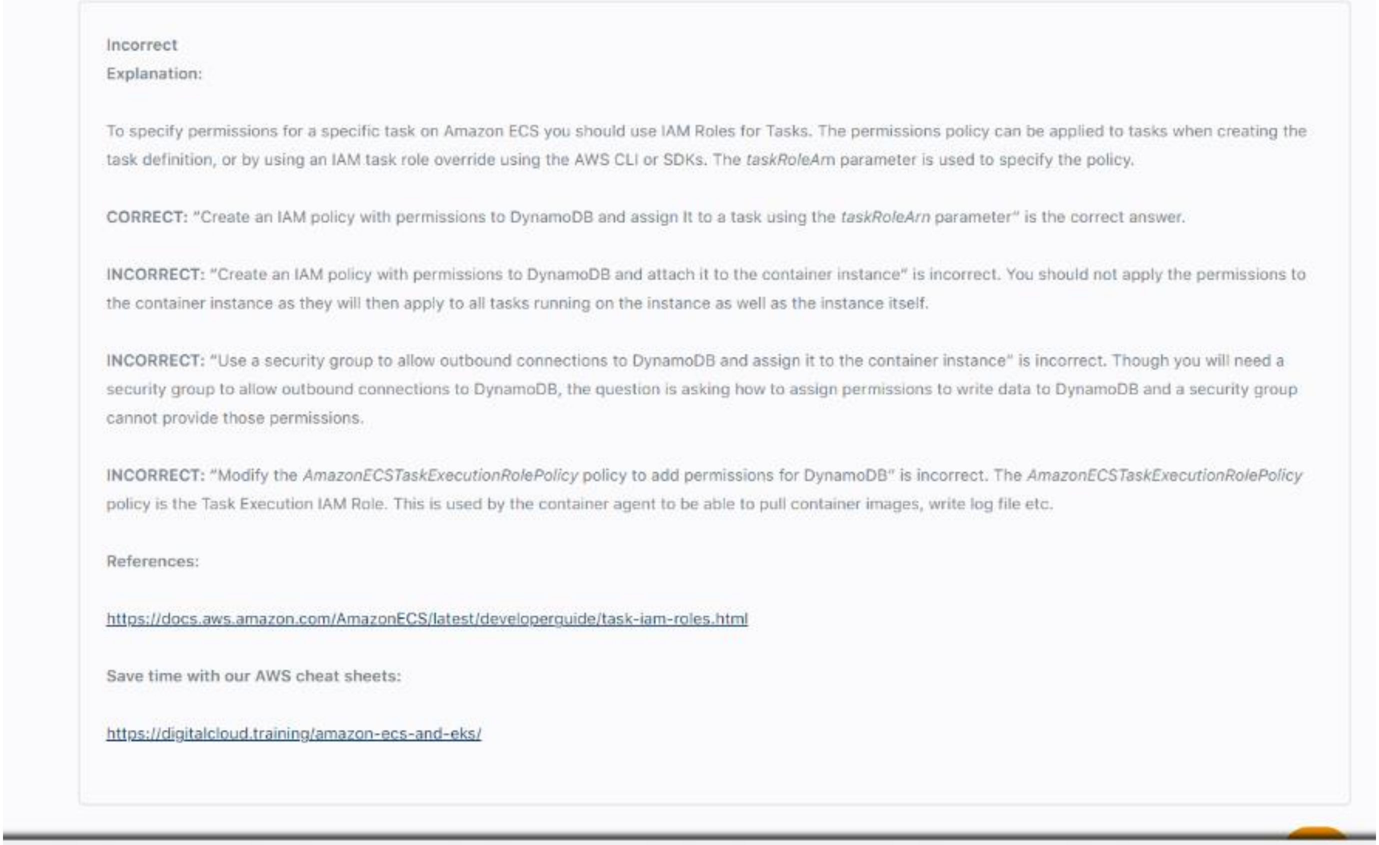
Add a comment



ECS: Tasks vs Services

ECS is an AWS container management service for running containerized apps on a cluster. There are two important concepts to understand in AWS ECS, which are Tasks and Services that we will compare in this article.

Let's start off where tasks and services are alike and then move on to differences. Both of them need something called a Task Definition which is basically a configuration of a task - sort of what the Docker Image is to a Docker Container. Task Definition holds the information how many containers to create for the task, what containers should be linked together, how much CPU should be allocated, connecting logs, defining environment etc.



Incorrect

Explanation:

To specify permissions for a specific task on Amazon ECS you should use IAM Roles for Tasks. The permissions policy can be applied to tasks when creating the task definition, or by using an IAM task role override using the AWS CLI or SDKs. The *taskRoleArn* parameter is used to specify the policy.

CORRECT: "Create an IAM policy with permissions to DynamoDB and assign It to a task using the *taskRoleArn* parameter" is the correct answer.

INCORRECT: "Create an IAM policy with permissions to DynamoDB and attach it to the container instance" is incorrect. You should not apply the permissions to the container instance as they will then apply to all tasks running on the instance as well as the instance itself.

INCORRECT: "Use a security group to allow outbound connections to DynamoDB and assign it to the container instance" is incorrect. Though you will need a security group to allow outbound connections to DynamoDB, the question is asking how to assign permissions to write data to DynamoDB and a security group cannot provide those permissions.

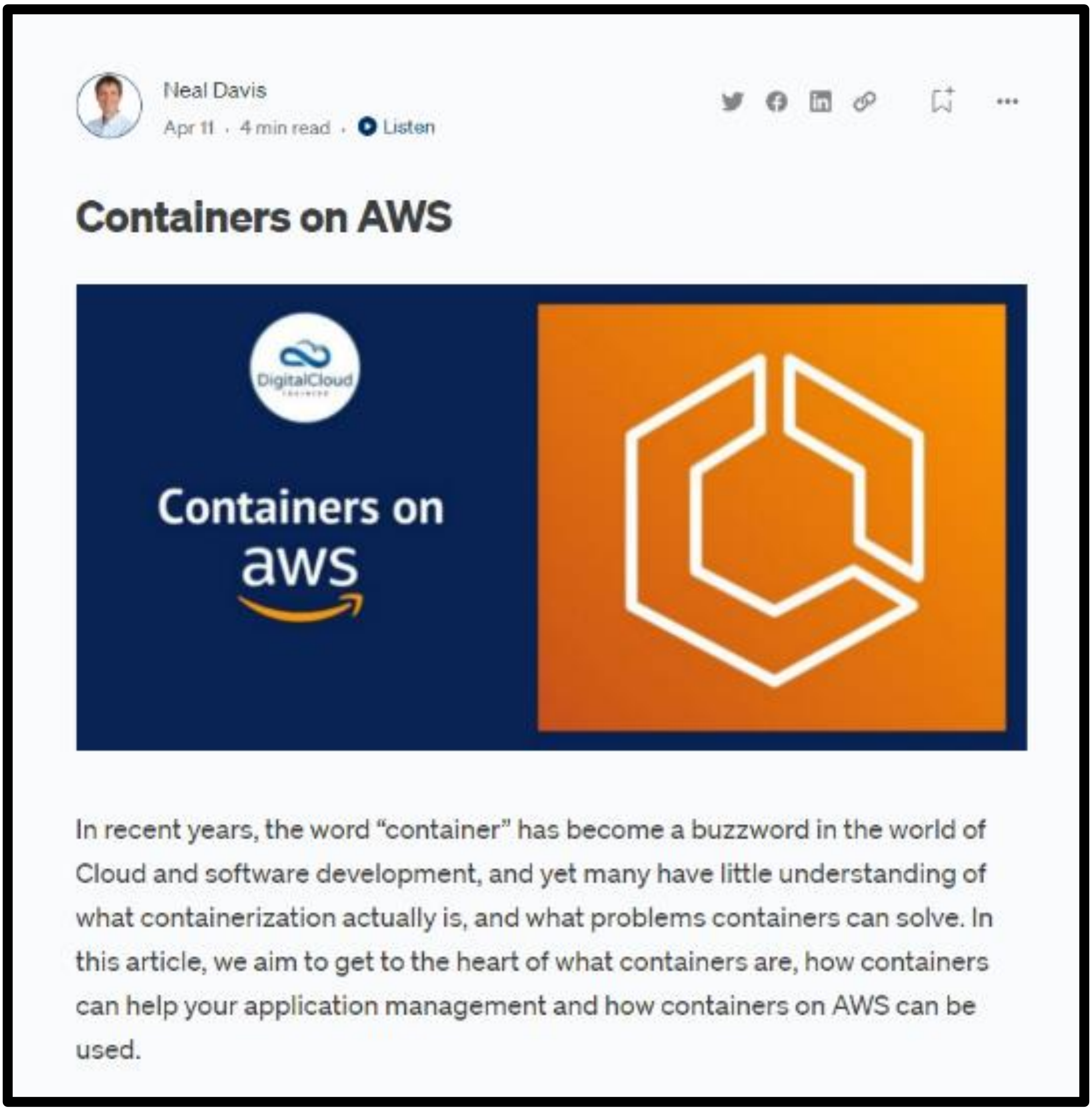
INCORRECT: "Modify the *AmazonECSTaskExecutionRolePolicy* policy to add permissions for DynamoDB" is incorrect. The *AmazonECSTaskExecutionRolePolicy* policy is the Task Execution IAM Role. This is used by the container agent to be able to pull container images, write log file etc.

References:

https://docs.aws.amazon.com/AmazonECS/latest/developerguide/task-iam-roles.html

Save time with our AWS cheat sheets:

https://digitalcloud.training/amazon-ecs-and-eks/

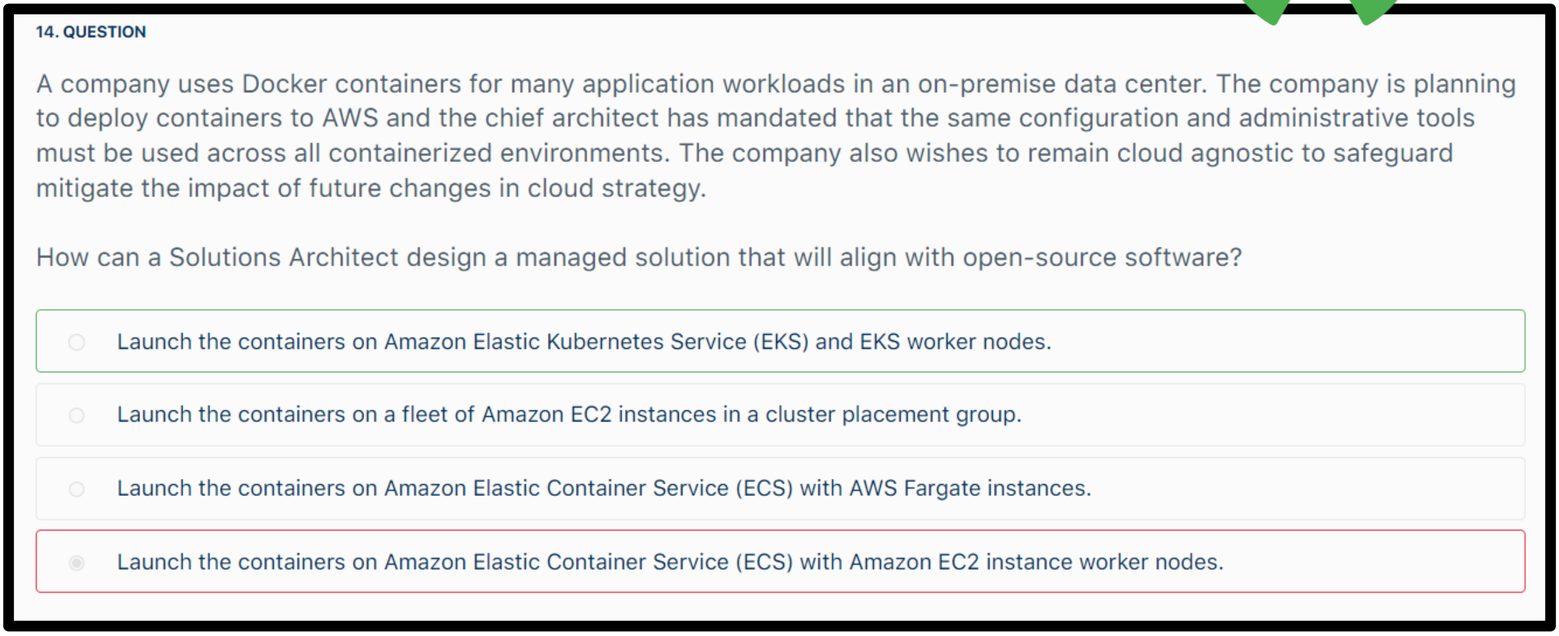


Neal Davis

Apr 11 · 4 min read · Listen

Containers on AWS

In recent years, the word "container" has become a buzzword in the world of Cloud and software development, and yet many have little understanding of what containerization actually is, and what problems containers can solve. In this article, we aim to get to the heart of what containers are, how containers can help your application management and how containers on AWS can be used.



14. QUESTION

A company uses Docker containers for many application workloads in an on-premise data center. The company is planning to deploy containers to AWS and the chief architect has mandated that the same configuration and administrative tools must be used across all containerized environments. The company also wishes to remain cloud agnostic to safeguard mitigate the impact of future changes in cloud strategy.

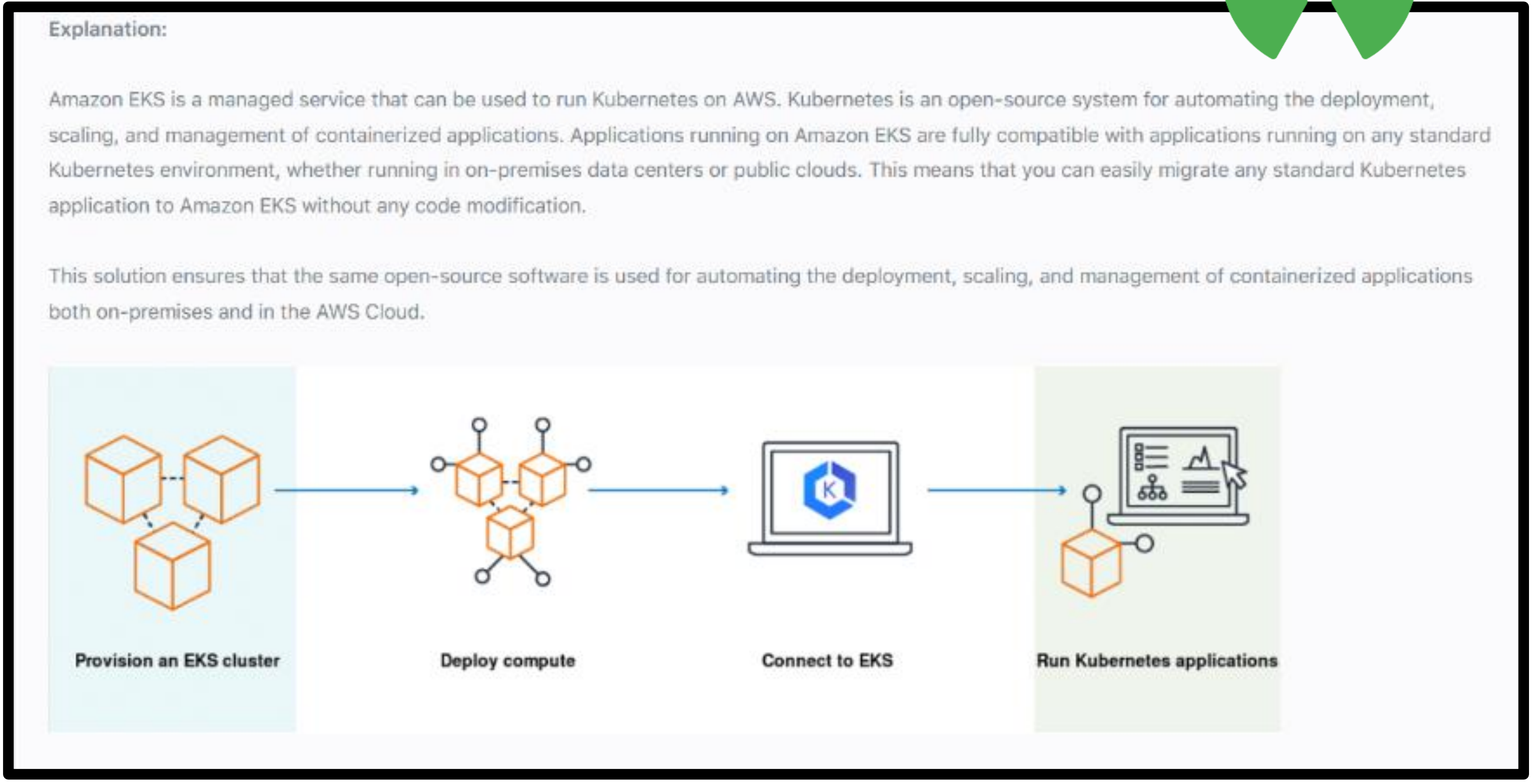
How can a Solutions Architect design a managed solution that will align with open-source software?

○ Launch the containers on Amazon Elastic Kubernetes Service (EKS) and EKS worker nodes.

○ Launch the containers on a fleet of Amazon EC2 instances in a cluster placement group.

○ Launch the containers on Amazon Elastic Container Service (ECS) with AWS Fargate instances.

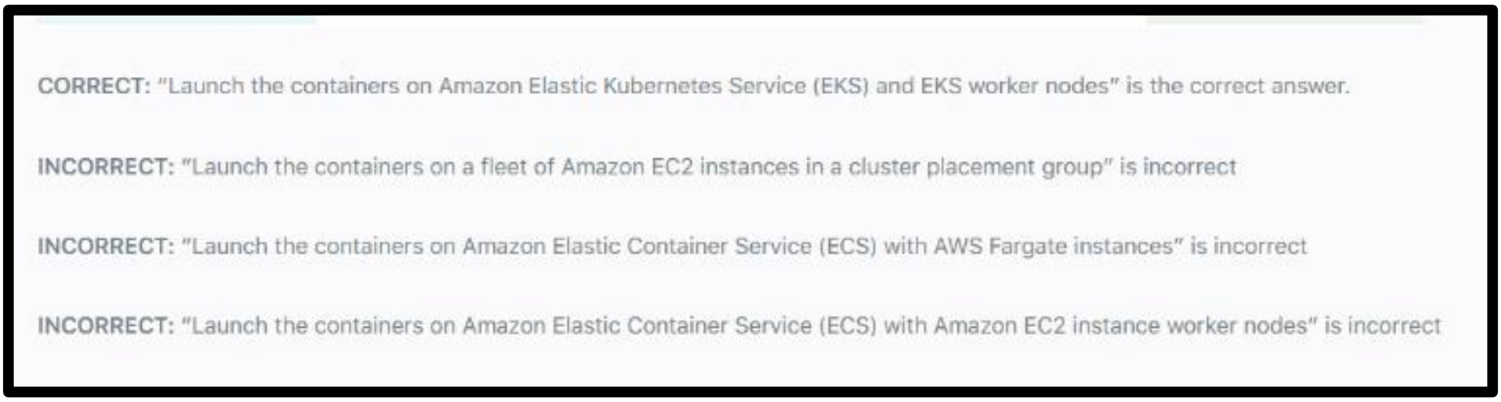
● Launch the containers on Amazon Elastic Container Service (ECS) with Amazon EC2 instance worker nodes.



Explanation:

Amazon EKS is a managed service that can be used to run Kubernetes on AWS. Kubernetes is an open-source system for automating the deployment, scaling, and management of containerized applications. Applications running on Amazon EKS are fully compatible with applications running on any standard Kubernetes environment, whether running in on-premises data centers or public clouds. This means that you can easily migrate any standard Kubernetes application to Amazon EKS without any code modification.

This solution ensures that the same open-source software is used for automating the deployment, scaling, and management of containerized applications both on-premises and in the AWS Cloud.

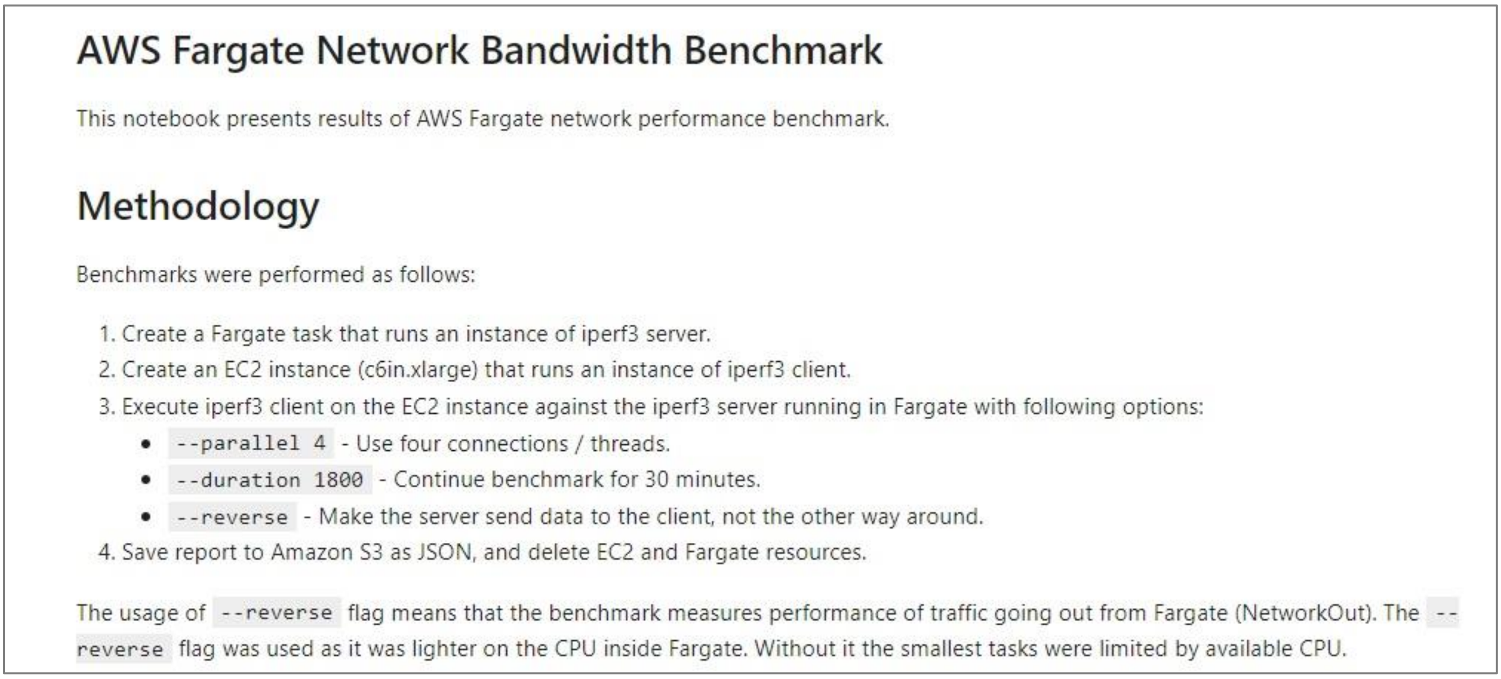


CORRECT: "Launch the containers on Amazon Elastic Kubernetes Service (EKS) and EKS worker nodes" is the correct answer.

INCORRECT: "Launch the containers on a fleet of Amazon EC2 instances in a cluster placement group" is incorrect

INCORRECT: "Launch the containers on Amazon Elastic Container Service (ECS) with AWS Fargate instances" is incorrect

INCORRECT: "Launch the containers on Amazon Elastic Container Service (ECS) with Amazon EC2 instance worker nodes" is incorrect



AWS Fargate Network Bandwidth Benchmark

This notebook presents results of AWS Fargate network performance benchmark.

Methodology

Benchmarks were performed as follows:

1. Create a Fargate task that runs an instance of iperf3 server.
2. Create an EC2 instance (c6in.xlarge) that runs an instance of iperf3 client.
3. Execute iperf3 client on the EC2 instance against the iperf3 server running in Fargate with following options:
    - `--parallel 4` - Use four connections / threads.
    - `--duration 1800` - Continue benchmark for 30 minutes.
    - `--reverse` - Make the server send data to the client, not the other way around.
4. Save report to Amazon S3 as JSON, and delete EC2 and Fargate resources.

The usage of `--reverse` flag means that the benchmark measures performance of traffic going out from Fargate (NetworkOut). The `--reverse` flag was used as it was lighter on the CPU inside Fargate. Without it the smallest tasks were limited by available CPU.

[sjakthol 2024]

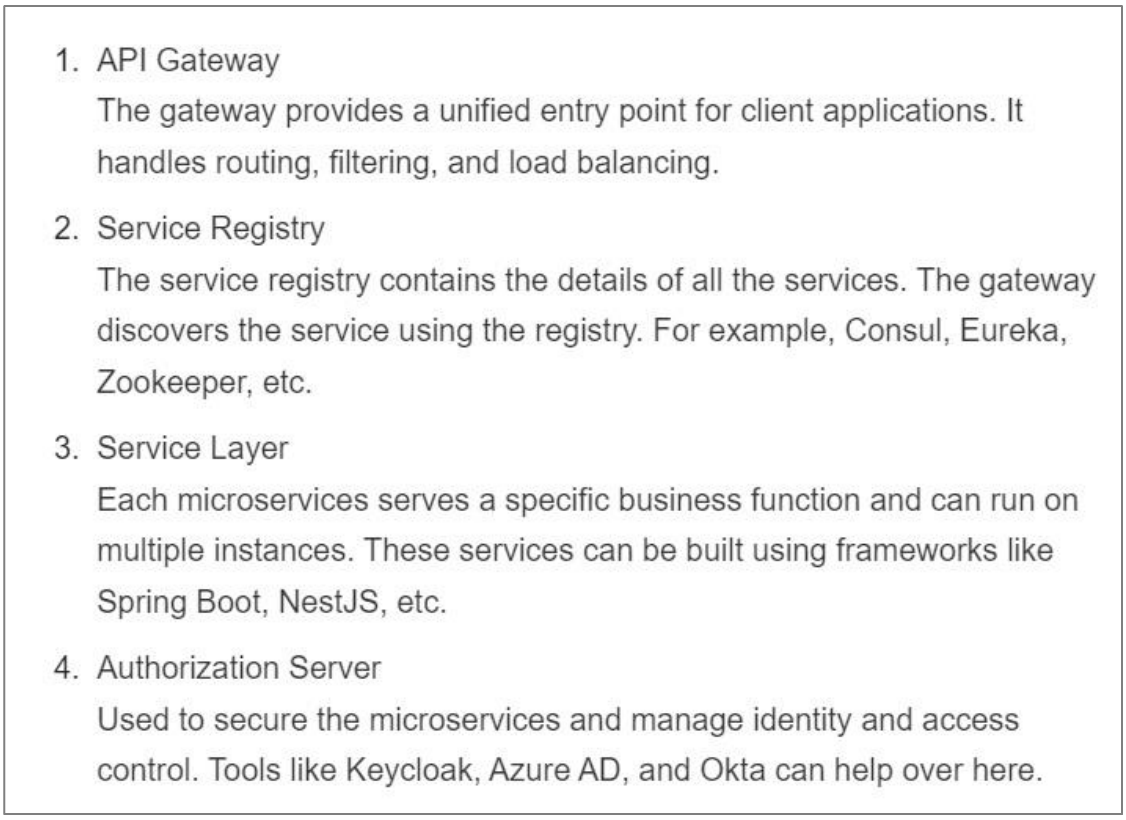
EP121: 9 Essential Components of a Production Microservice Application

BYTEBYTEGO
JUL 20

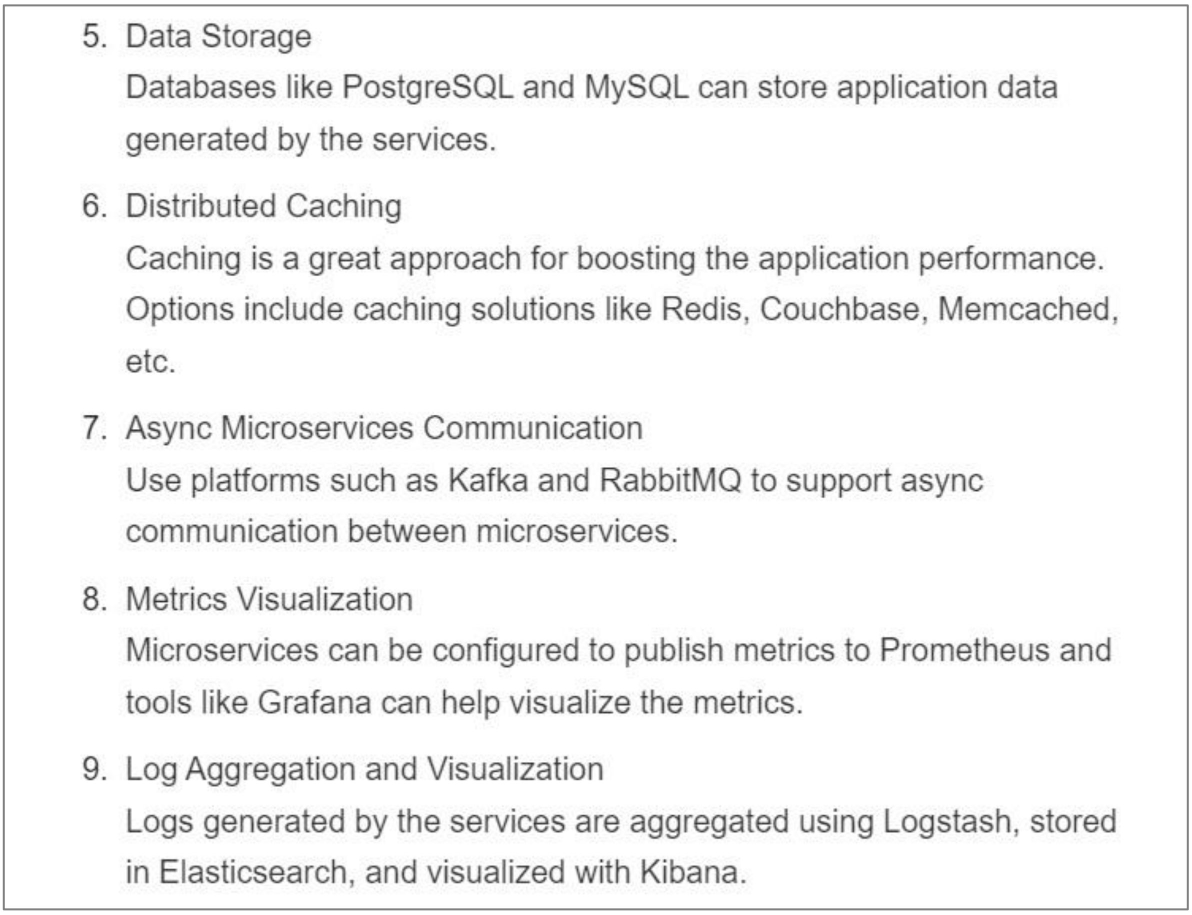
READ IN APP ↗

This week's system design interview:

- Linux Crash Course - Understanding File Permissions (Youtube video)
- 9 Essential Components of a Production Microservice Application
- Iterative, Agile, Waterfall, Spiral Model, RAD Model... What are the differences?
- Design Patterns Cheat Sheet - Part 1 and Part 2
- SPONSOR US



1. API Gateway
   The gateway provides a unified entry point for client applications. It handles routing, filtering, and load balancing.
2. Service Registry
   The service registry contains the details of all the services. The gateway discovers the service using the registry. For example, Consul, Eureka, Zookeeper, etc.
3. Service Layer
   Each microservices serves a specific business function and can run on multiple instances. These services can be built using frameworks like Spring Boot, NestJS, etc.
4. Authorization Server
   Used to secure the microservices and manage identity and access control. Tools like Keycloak, Azure AD, and Okta can help over here.



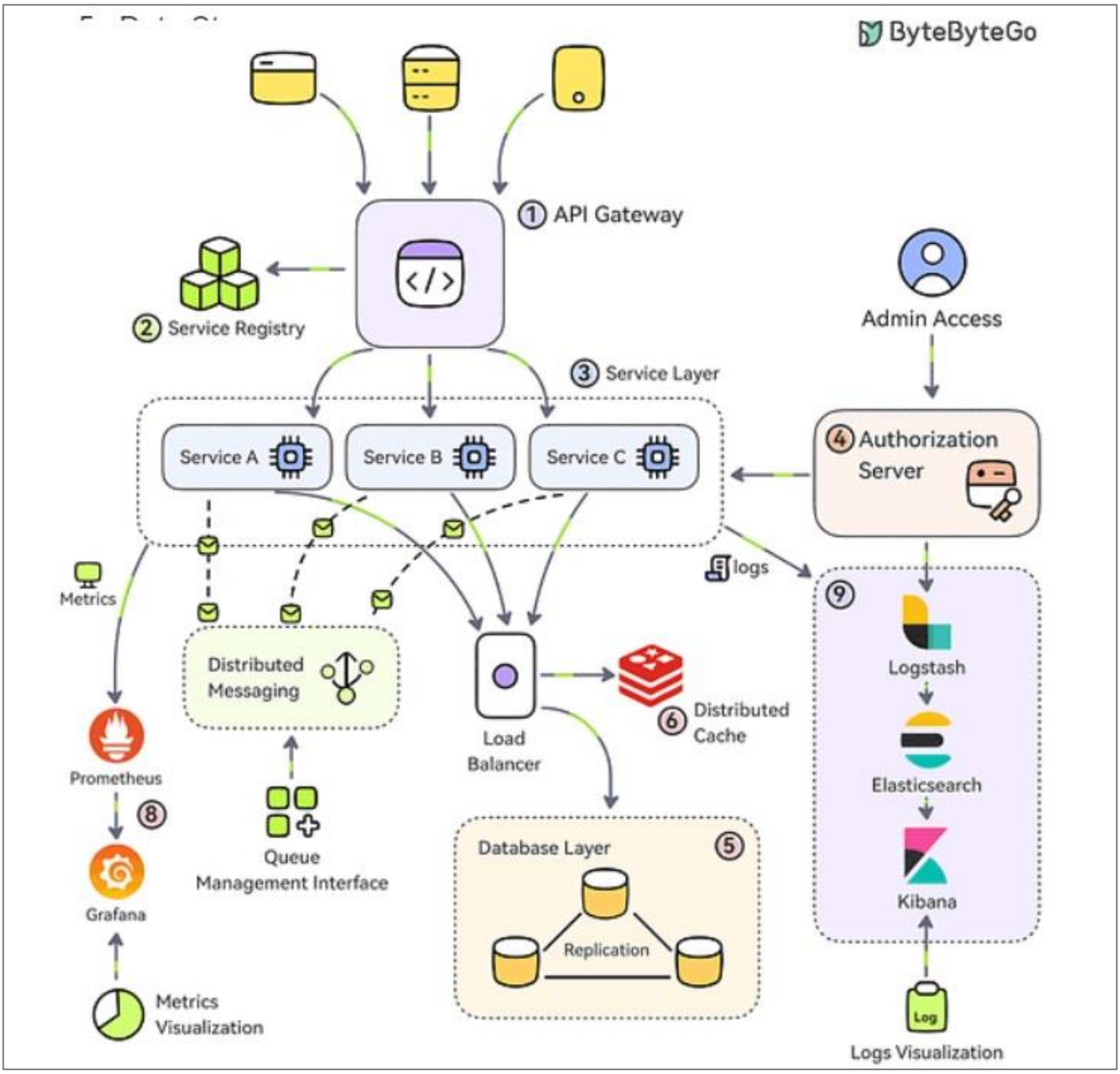
5. Data Storage
   Databases like PostgreSQL and MySQL can store application data generated by the services.

6. Distributed Caching
   Caching is a great approach for boosting the application performance. Options include caching solutions like Redis, Couchbase, Memcached, etc.

7. Async Microservices Communication
   Use platforms such as Kafka and RabbitMQ to support async communication between microservices.

8. Metrics Visualization
   Microservices can be configured to publish metrics to Prometheus and tools like Grafana can help visualize the metrics.

9. Log Aggregation and Visualization
   Logs generated by the services are aggregated using Logstash, stored in Elasticsearch, and visualized with Kibana.



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

Cluster

Important term within Amazon ECS. The documentation writes:

> An Amazon ECS *cluster* is a logical grouping of tasks or services.
>
> You can use clusters to isolate your applications. This way, they don't use the same underlying infrastructure. When your tasks are run on Fargate, your cluster resources are also managed by Fargate.

Task

This is an important term within Amazon ECS. The documentation writes:

> A *task* is the **instantiation of a task definition** within a cluster.
>
> After you create a task definition for your application within Amazon ECS, you can specify the number of tasks to run on your cluster. You can run a standalone task, or you can run a task as part of a service.

Task Definition

"Task definition" is an important expression within Amazon's ECS. The documentation writes:

> A *task definition* is a text file that describes one or more containers that form your application.
>
> It's in JSON format. You can use it to describe up to a maximum of ten containers. The task definition functions as a blueprint for your application.

Service

This is an important term, but AWS's explanation of it is poot. The documentation writes:

> You can use an Amazon ECS *service* to run and maintain your desired number of tasks simultaneously in an Amazon ECS cluster.
>
> How it works is that, if any of your tasks fail or stop for any reason, the Amazon ECS service scheduler launches another instance based on your task definition. It does this to replace it and thereby maintain your desired number of tasks in the service.

Scheduler

We are told:

> Amazon Elastic Container Service (Amazon ECS) is a shared state, optimistic concurrency system that provides flexible scheduling capabilities for your tasks and containers.
>
> The Amazon ECS schedulers use the same cluster state information as the Amazon ECS API to make appropriate placement decisions.

Custom scheduler

Service Scheduler

Container

The documentation writes:

> To deploy applications on Amazon ECS, your application components must be configured to run in *containers*. A container is a standardized unit of software development that holds everything that your software application requires to run.

> This includes relevant code, runtime, system tools, and system libraries. Containers are created from a read-only template that's called an *image*.

Image

Important term within Amazon’s ECS. We’re told:

> Containers are created from a read-only template that's called an *image*.

Fargate

The Elastic Container Service (ECS) was announced by AWS in 2014. Three years later, AWS announced a service named after a shopping street in Sheffield, England (we do not know why). Its name was *Fargate*.

Fargate provides a new *way* of using the Elastic Container Service. You can now use it without concerning yourself with the provisioning of containers.

The documentation writes:

> AWS Fargate is a technology that you can use with Amazon ECS to run containers without having to manage servers or clusters of Amazon EC2 instances.
>
> With AWS Fargate, you no longer have to provision, configure, or scale clusters of virtual machines to run containers. This removes the need to choose server types, decide when to scale your clusters, or optimize cluster packing.

Launch type

Bibliography

I. Official

[Surname1]

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

[Surname1]

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

II. Unofficial

[sjakthol 2024]

AWS Fargate Network Bandwidth Benchmark. Available at: <https://github.com/sjakthol/aws-network-benchmark/blob/main/analysis/2024/results.ipynb?ck_subscriber_id=1560524742>

[Davis 2022]

"Containers on AWS". *Medium*. Available at: https://neal-davis.medium.com/containers-on-aws-7160928c4339

[Ferrè 2021]

Ferrè, Massimo re (2023). Configuring a timeout for Amazon ECS tasks. Available at: <https://it20.info/2023/03/configuring-a-timeout-for-amazon-ecs-tasks/?ck_subscriber_id=1560524742>

[Fraser 2018]

Fraser, Dominic (year). A beginner's guide to Amazon's Elastic Container Service. 20st May 2018. FreeCodeCamp. Available at:

<https://www.freecodecamp.org/news/amazon-ecs-terms-and-architecture-807d8c4960fd/>.

[Nguyen 2017]

Nguyen, Teng (2017). "Gentle Introduction to how ECS works with example tutorial". 10th Sept 2017. Medium. Available at: https://medium.com/boltops/gentle-introduction-to-how-aws-ecs-works-with-example-tutorial-cea3d27ce63d

[Verch 2024]

Verch, Shaun (2024). Our unusual journey to ECS on EC2. Oso. Available at: <https://www.osohq.com/post/ecs-on-ec2>

III. Critical

[Surname1]

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

[Surname1]

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

IV. General

[Surname1]

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

[Surname1]

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

[Tamang 2024]

Tamang, Kisan (2024). How Containers work? Deep diver into Containerization. *The Cloud Handbook*. May 14th 2024.

[ByteByteGo 2024]

9 Essential Components of Production Microservice Application. ByteByteGo [Email newsletter]. July 20th 2024.

Amazon WorkMail

amazon WorkMail

Amazon WorkMail now supports Audit Logging - I keep forgetting that Amazon has their own hosted email service. Given that it only now in 2024 got audit logging, I'm guessing everyone else forgot too.

Email from Corey Quinn on 25th March 2024

Bibliography

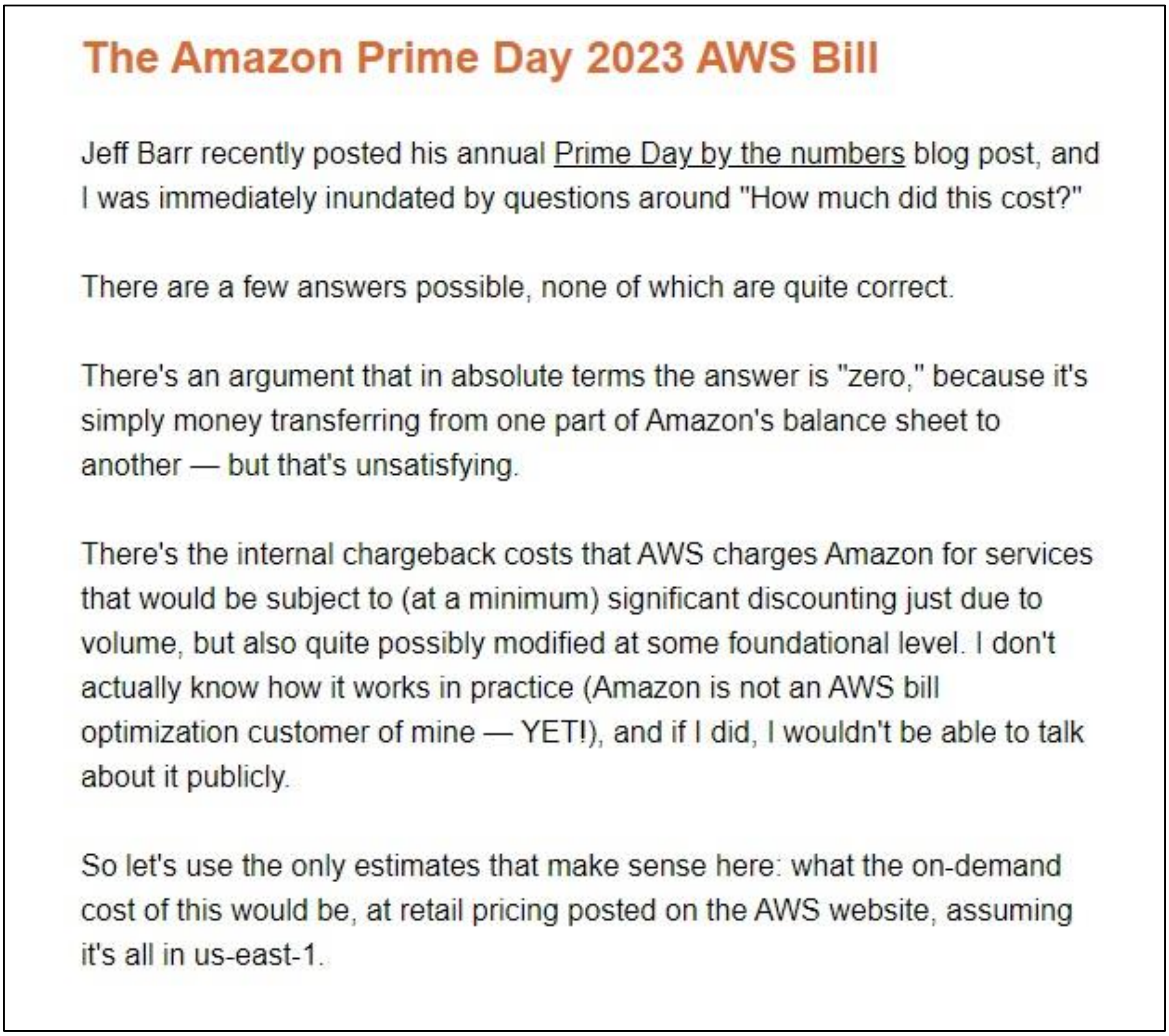
I. Official

II. Unofficial

III. Critical

IV. General

AWS Budgets



The Amazon Prime Day 2023 AWS Bill

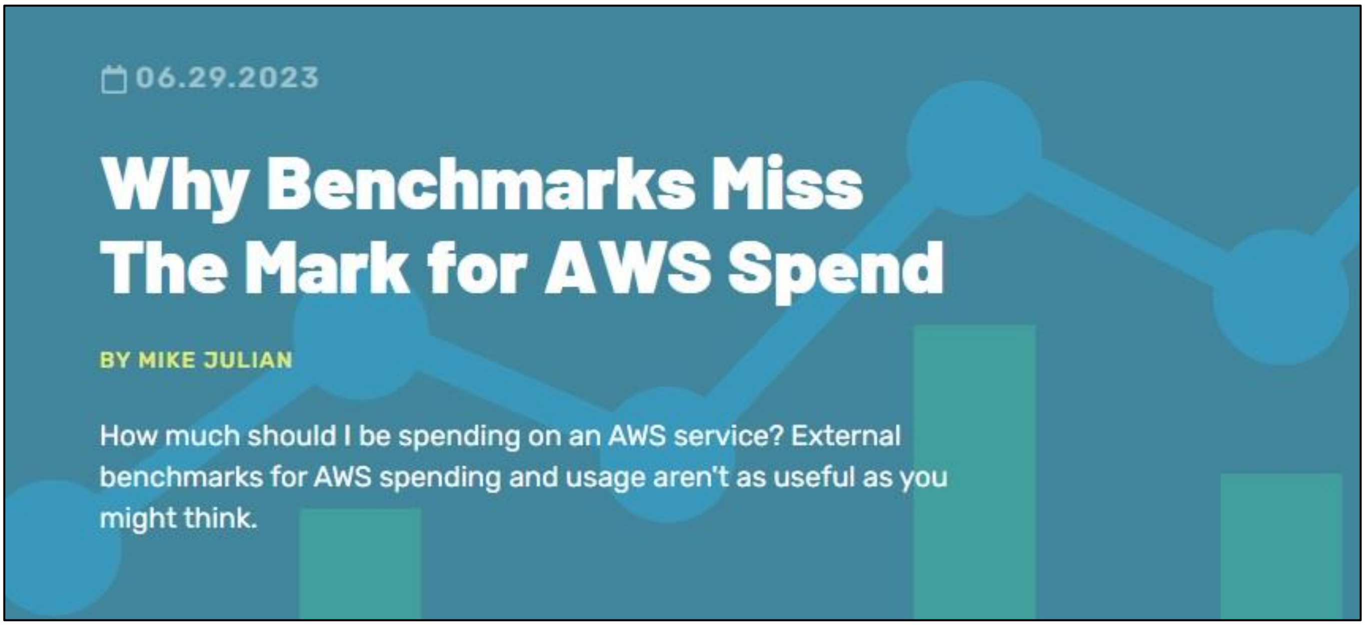
Jeff Barr recently posted his annual Prime Day by the numbers blog post, and I was immediately inundated by questions around "How much did this cost?"

There are a few answers possible, none of which are quite correct.

There's an argument that in absolute terms the answer is "zero," because it's simply money transferring from one part of Amazon's balance sheet to another — but that's unsatisfying.

There's the internal chargeback costs that AWS charges Amazon for services that would be subject to (at a minimum) significant discounting just due to volume, but also quite possibly modified at some foundational level. I don't actually know how it works in practice (Amazon is not an AWS bill optimization customer of mine — YET!), and if I did, I wouldn't be able to talk about it publicly.

So let's use the only estimates that make sense here: what the on-demand cost of this would be, at retail pricing posted on the AWS website, assuming it's all in us-east-1.

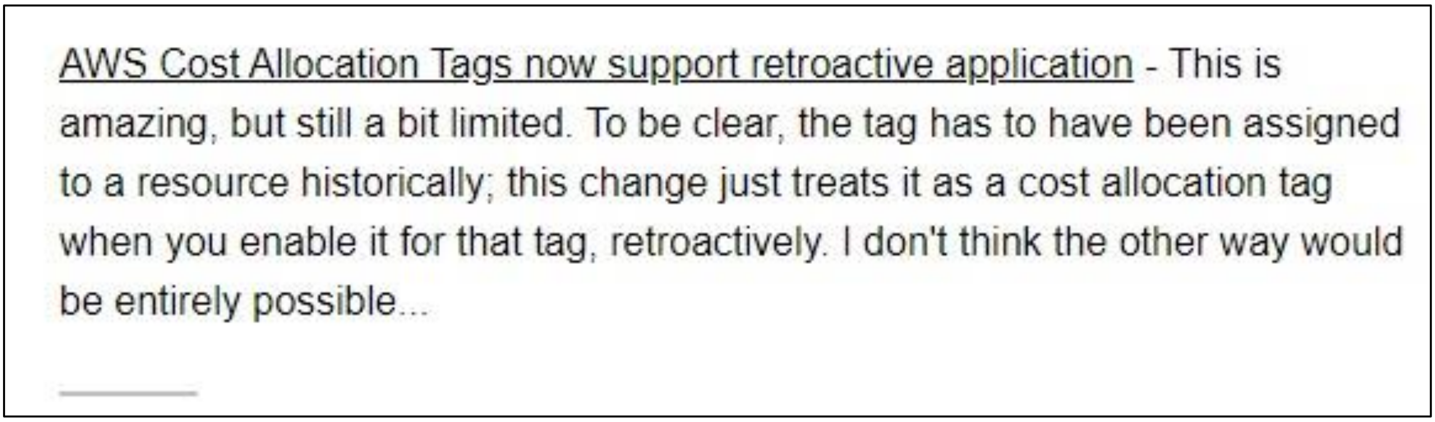


06.29.2023

Why Benchmarks Miss The Mark for AWS Spend

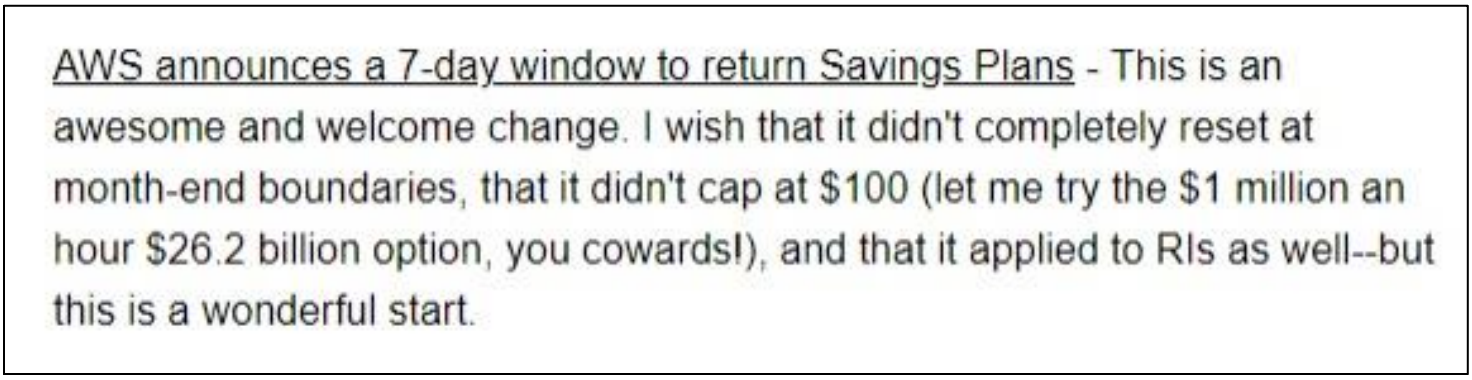
BY MIKE JULIAN

How much should I be spending on an AWS service? External benchmarks for AWS spending and usage aren't as useful as you might think.



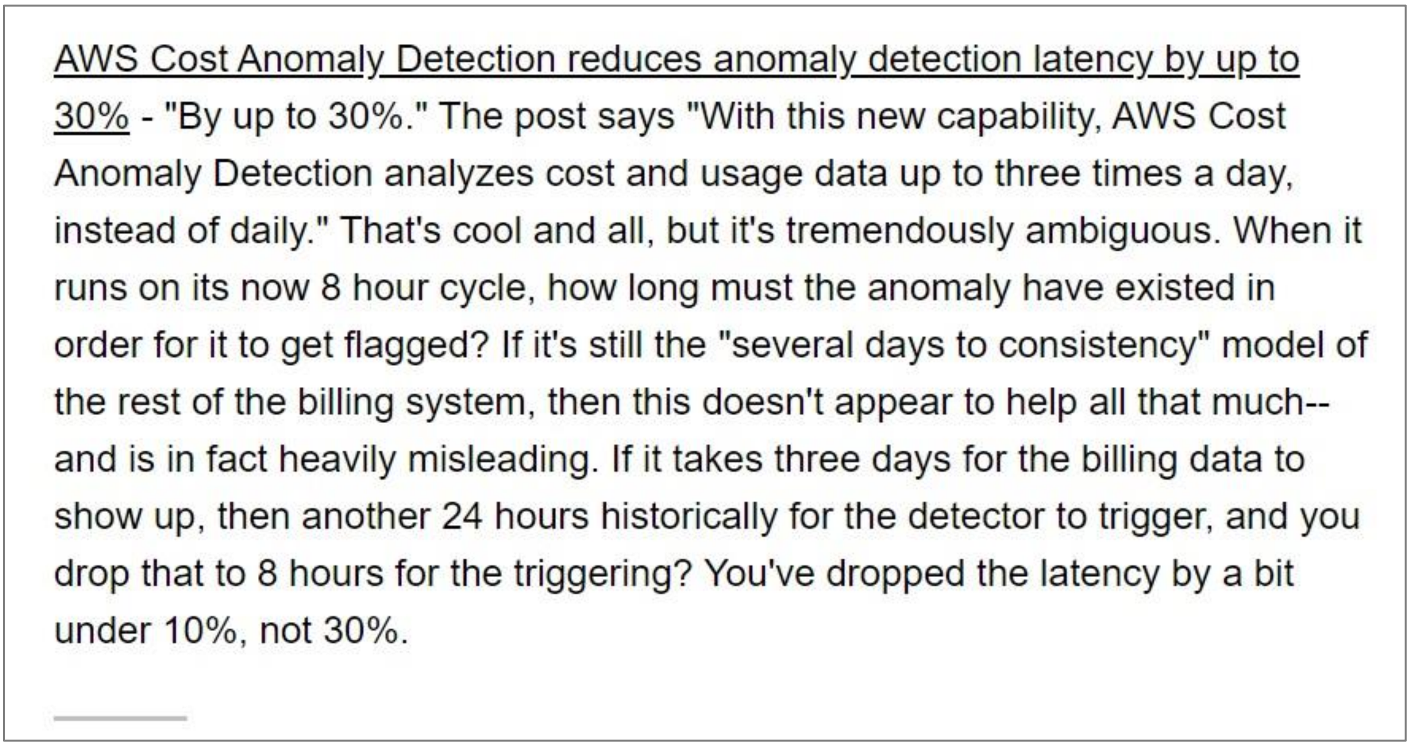
AWS Cost Allocation Tags now support retroactive application - This is amazing, but still a bit limited. To be clear, the tag has to have been assigned to a resource historically; this change just treats it as a cost allocation tag when you enable it for that tag, retroactively. I don't think the other way would be entirely possible...

Email from Corey Quinn on 1st April 2024



AWS announces a 7-day window to return Savings Plans - This is an awesome and welcome change. I wish that it didn't completely reset at month-end boundaries, that it didn't cap at $100 (let me try the $1 million an hour $26.2 billion option, you cowards!), and that it applied to RIs as well--but this is a wonderful start.

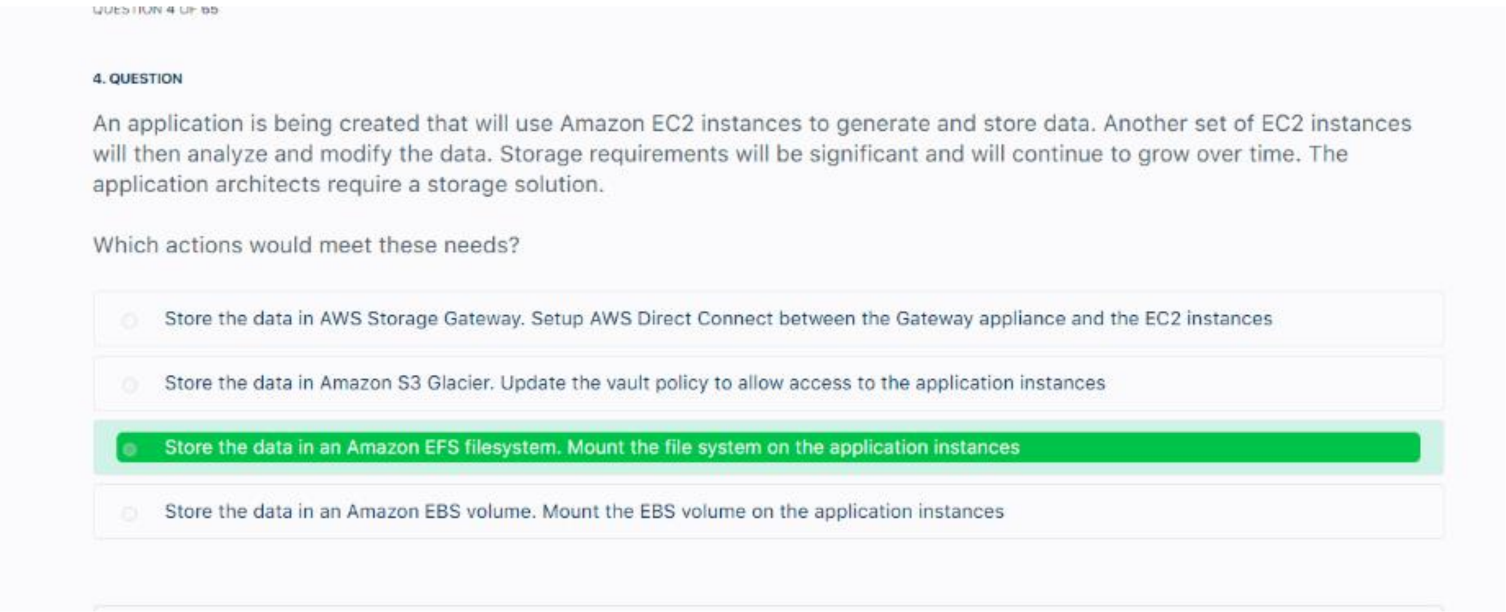
Email from Corey Quinn on 14st April 2024



AWS Cost Anomaly Detection reduces anomaly detection latency by up to 30% - "By up to 30%." The post says "With this new capability, AWS Cost Anomaly Detection analyzes cost and usage data up to three times a day, instead of daily." That's cool and all, but it's tremendously ambiguous. When it runs on its now 8 hour cycle, how long must the anomaly have existed in order for it to get flagged? If it's still the "several days to consistency" model of the rest of the billing system, then this doesn't appear to help all that much--and is in fact heavily misleading. If it takes three days for the billing data to show up, then another 24 hours historically for the detector to trigger, and you drop that to 8 hours for the triggering? You've dropped the latency by a bit under 10%, not 30%.

Corey Quinn in an email on May 13th 2024

Elastic File System (EFS)



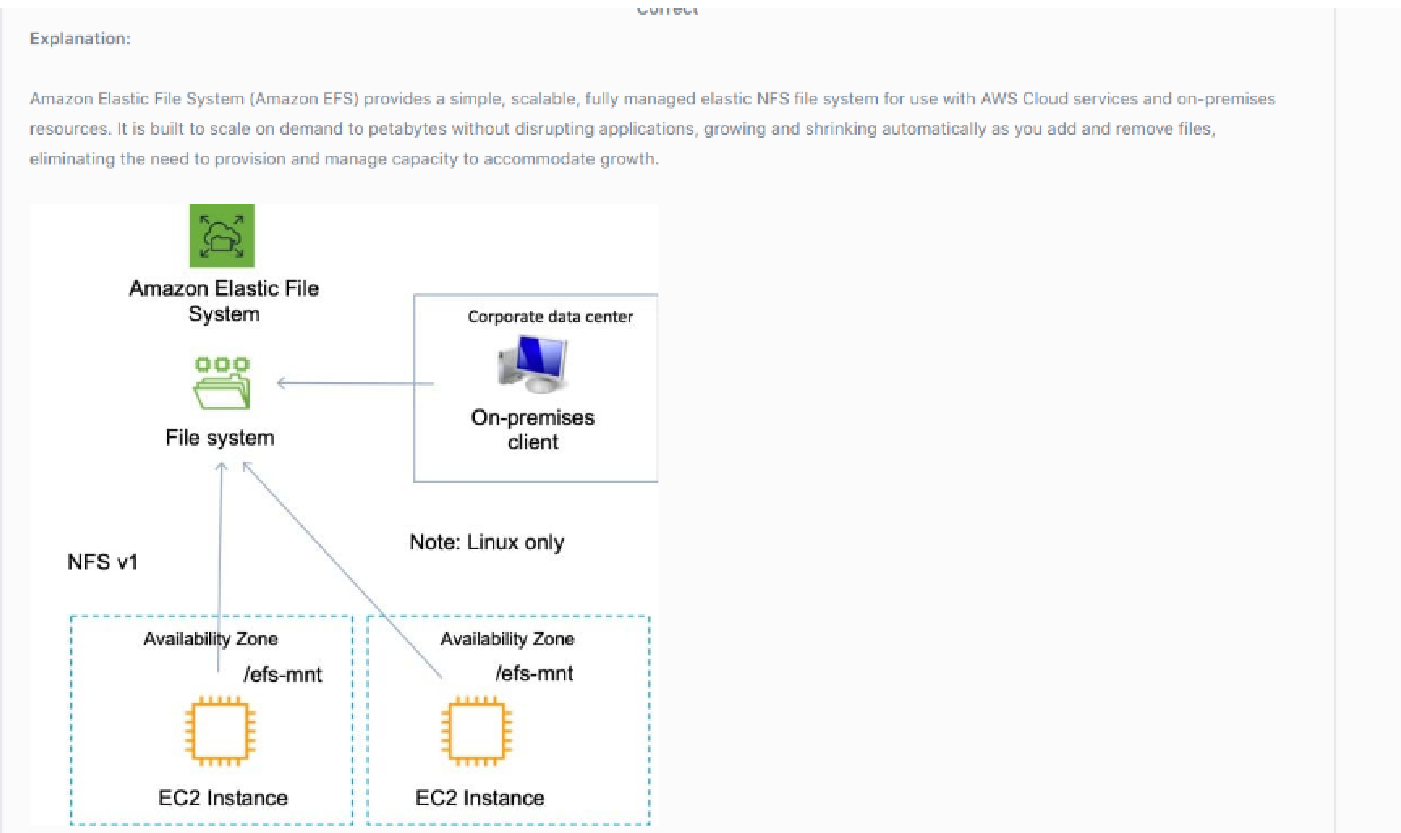
QUESTION 4 OF 65

4. QUESTION

An application is being created that will use Amazon EC2 instances to generate and store data. Another set of EC2 instances will then analyze and modify the data. Storage requirements will be significant and will continue to grow over time. The application architects require a storage solution.

Which actions would meet these needs?

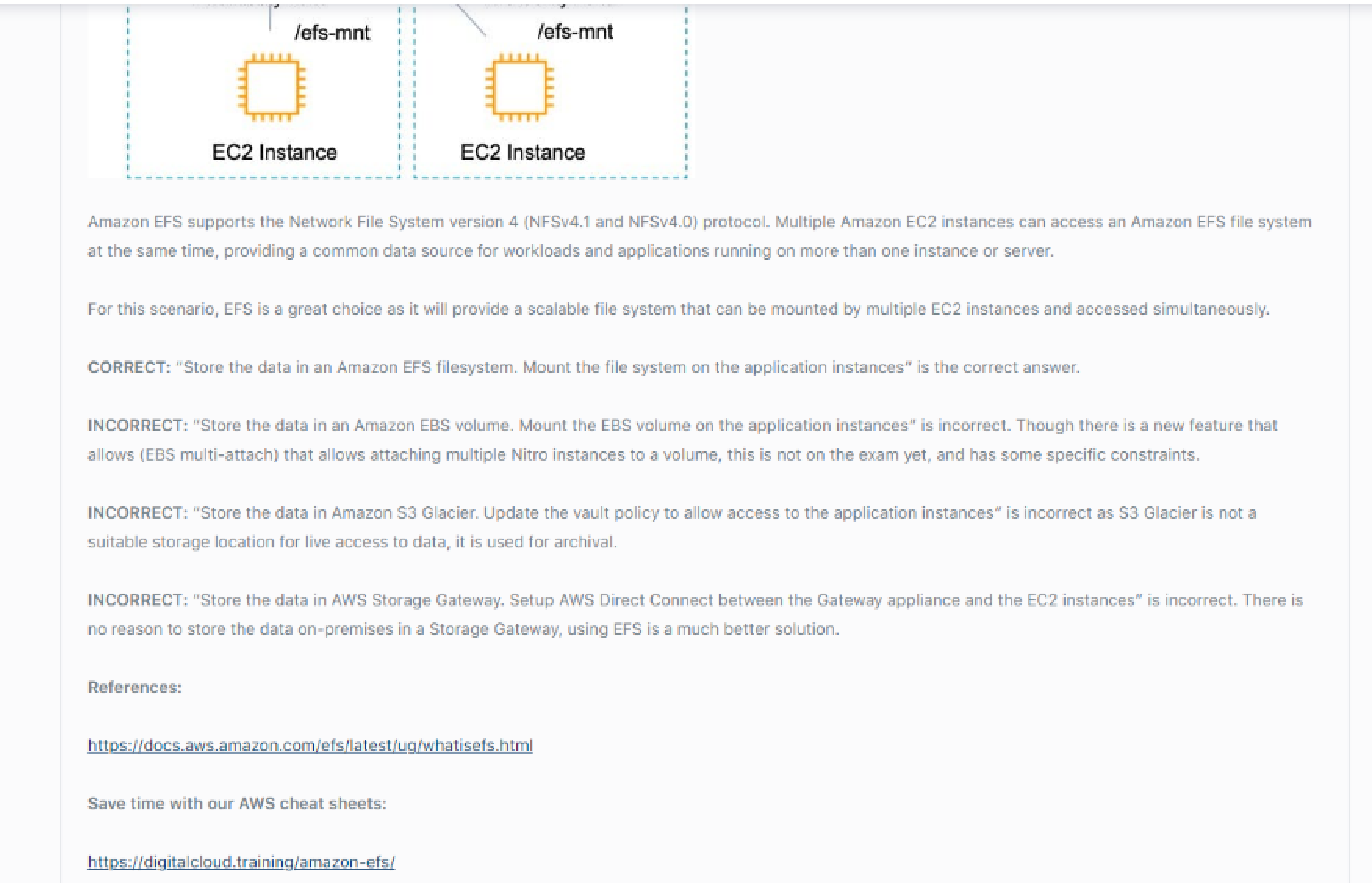
- ○ Store the data in AWS Storage Gateway. Setup AWS Direct Connect between the Gateway appliance and the EC2 instances
- ○ Store the data in Amazon S3 Glacier. Update the vault policy to allow access to the application instances
- ● Store the data in an Amazon EFS filesystem. Mount the file system on the application instances
- ○ Store the data in an Amazon EBS volume. Mount the EBS volume on the application instances



Correct

Explanation:

Amazon Elastic File System (Amazon EFS) provides a simple, scalable, fully managed elastic NFS file system for use with AWS Cloud services and on-premises resources. It is built to scale on demand to petabytes without disrupting applications, growing and shrinking automatically as you add and remove files, eliminating the need to provision and manage capacity to accommodate growth.



Amazon EFS supports the Network File System version 4 (NFSv4.1 and NFSv4.0) protocol. Multiple Amazon EC2 instances can access an Amazon EFS file system at the same time, providing a common data source for workloads and applications running on more than one instance or server.

For this scenario, EFS is a great choice as it will provide a scalable file system that can be mounted by multiple EC2 instances and accessed simultaneously.

CORRECT: "Store the data in an Amazon EFS filesystem. Mount the file system on the application instances" is the correct answer.

INCORRECT: "Store the data in an Amazon EBS volume. Mount the EBS volume on the application instances" is incorrect. Though there is a new feature that allows (EBS multi-attach) that allows attaching multiple Nitro instances to a volume, this is not on the exam yet, and has some specific constraints.

INCORRECT: "Store the data in Amazon S3 Glacier. Update the vault policy to allow access to the application instances" is incorrect as S3 Glacier is not a suitable storage location for live access to data, it is used for archival.

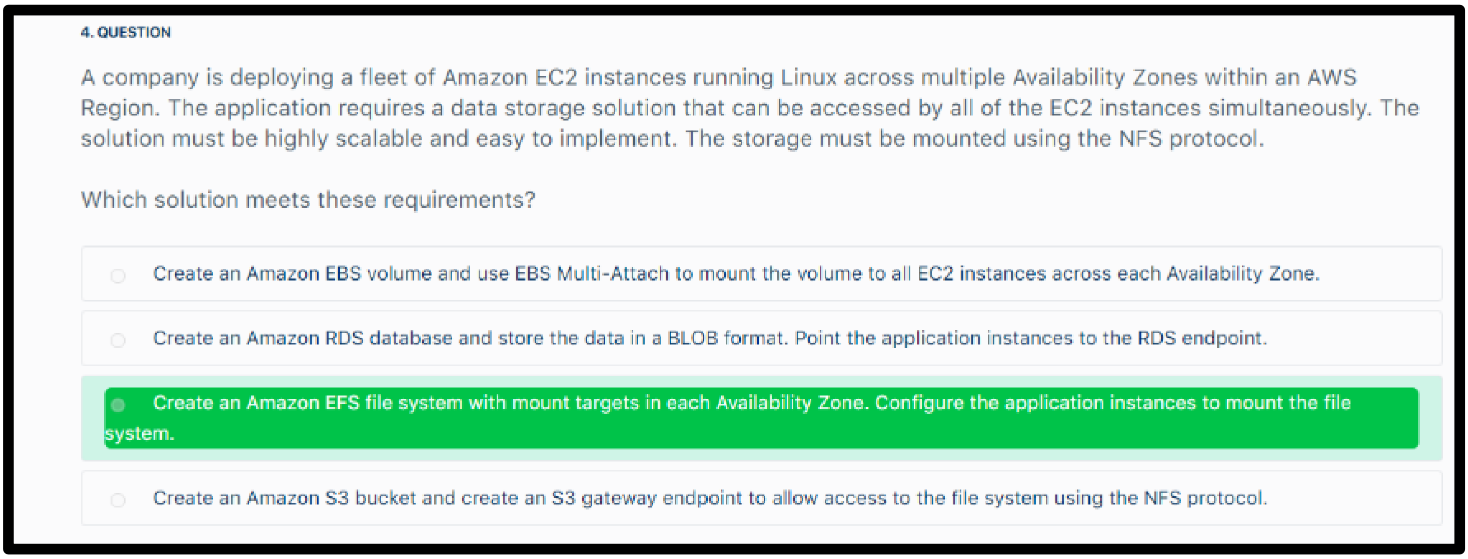
INCORRECT: "Store the data in AWS Storage Gateway. Setup AWS Direct Connect between the Gateway appliance and the EC2 instances" is incorrect. There is no reason to store the data on-premises in a Storage Gateway, using EFS is a much better solution.

References:

https://docs.aws.amazon.com/efs/latest/ug/whatisefs.html

Save time with our AWS cheat sheets:

https://digitalcloud.training/amazon-efs/



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.

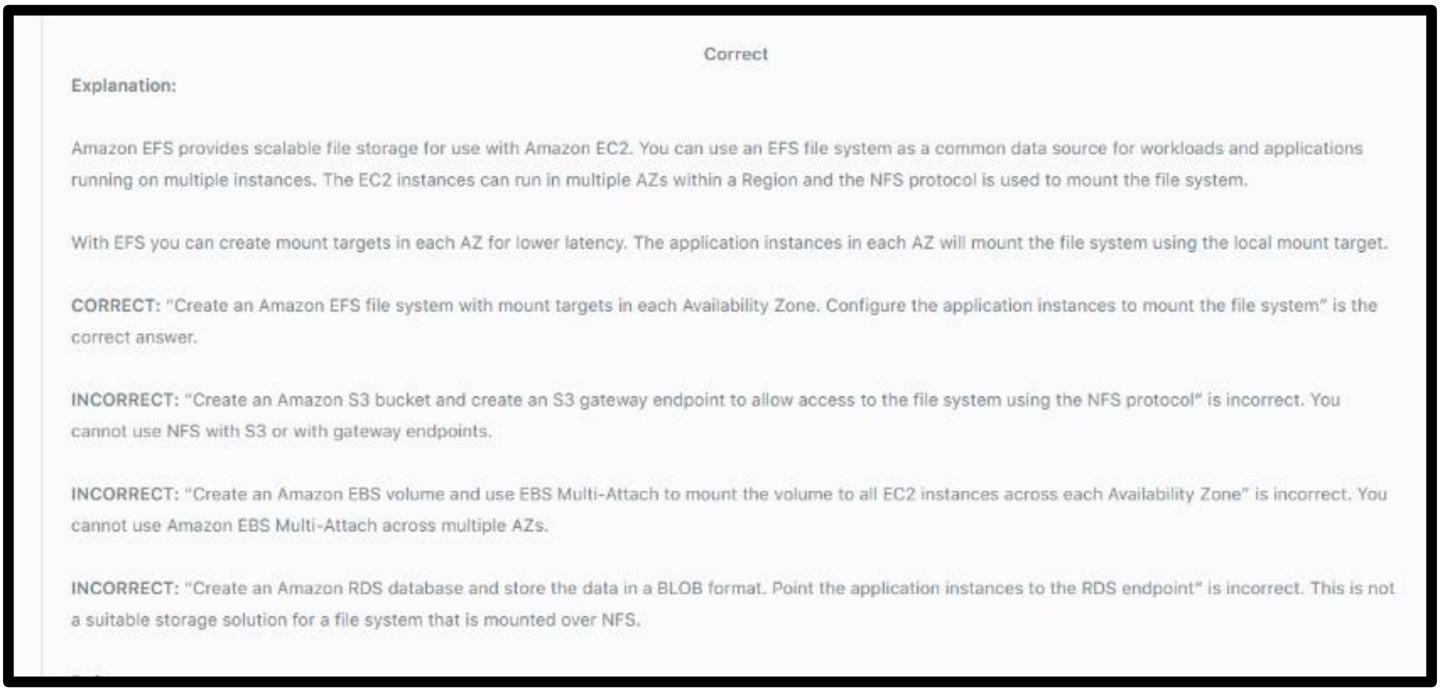
Which solution meets these requirements?

○ 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.

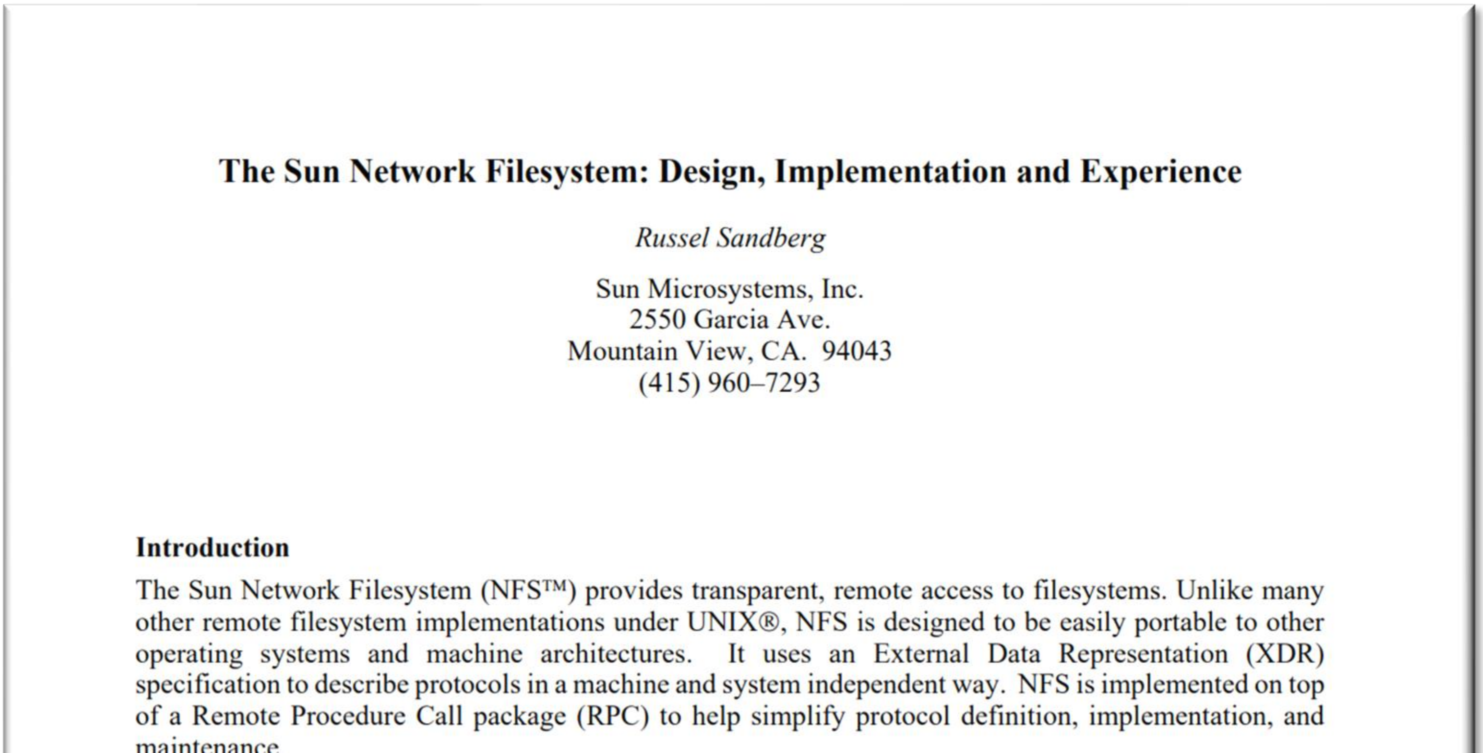
CORRECT: "Create an Amazon EFS file system with mount targets in each Availability Zone. Configure the application instances to mount the file system" is the correct answer.

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.

What on earth is NFS?



The Sun Network Filesystem: Design, Implementation and Experience

Russel Sandberg

Sun Microsystems, Inc.
2550 Garcia Ave.
Mountain View, CA. 94043
(415) 960–7293

Introduction

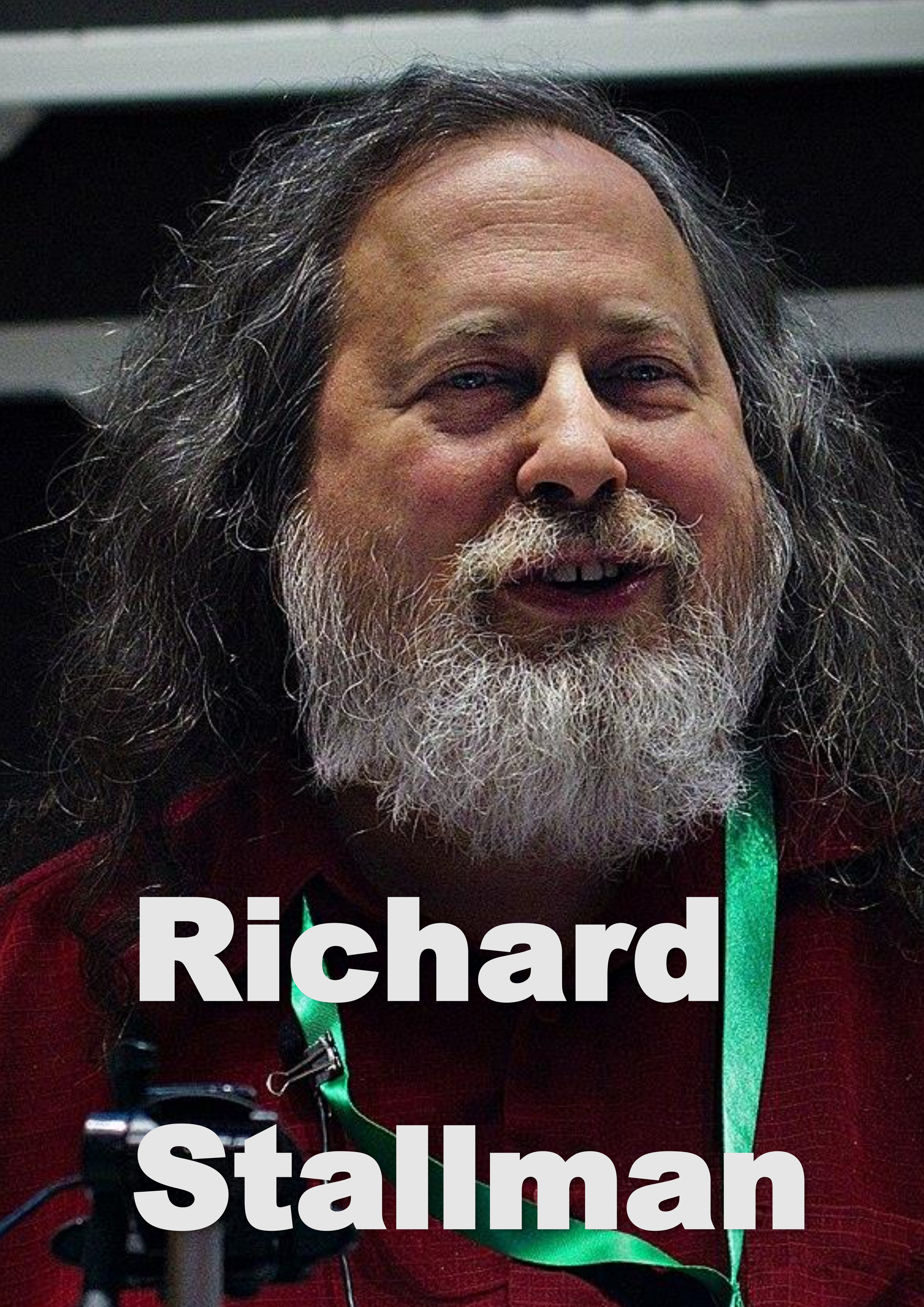
The Sun Network Filesystem (NFS™) provides transparent, remote access to filesystems. Unlike many other remote filesystem implementations under UNIX®, NFS is designed to be easily portable to other operating systems and machine architectures. It uses an External Data Representation (XDR) specification to describe protocols in a machine and system independent way. NFS is implemented on top of a Remote Procedure Call package (RPC) to help simplify protocol definition, implementation, and maintenance.

NFS

Bill Joy

What on earth is POSIX?

Richard Stallman came up with the name.



Richard Stallman

NTFS

New Technology

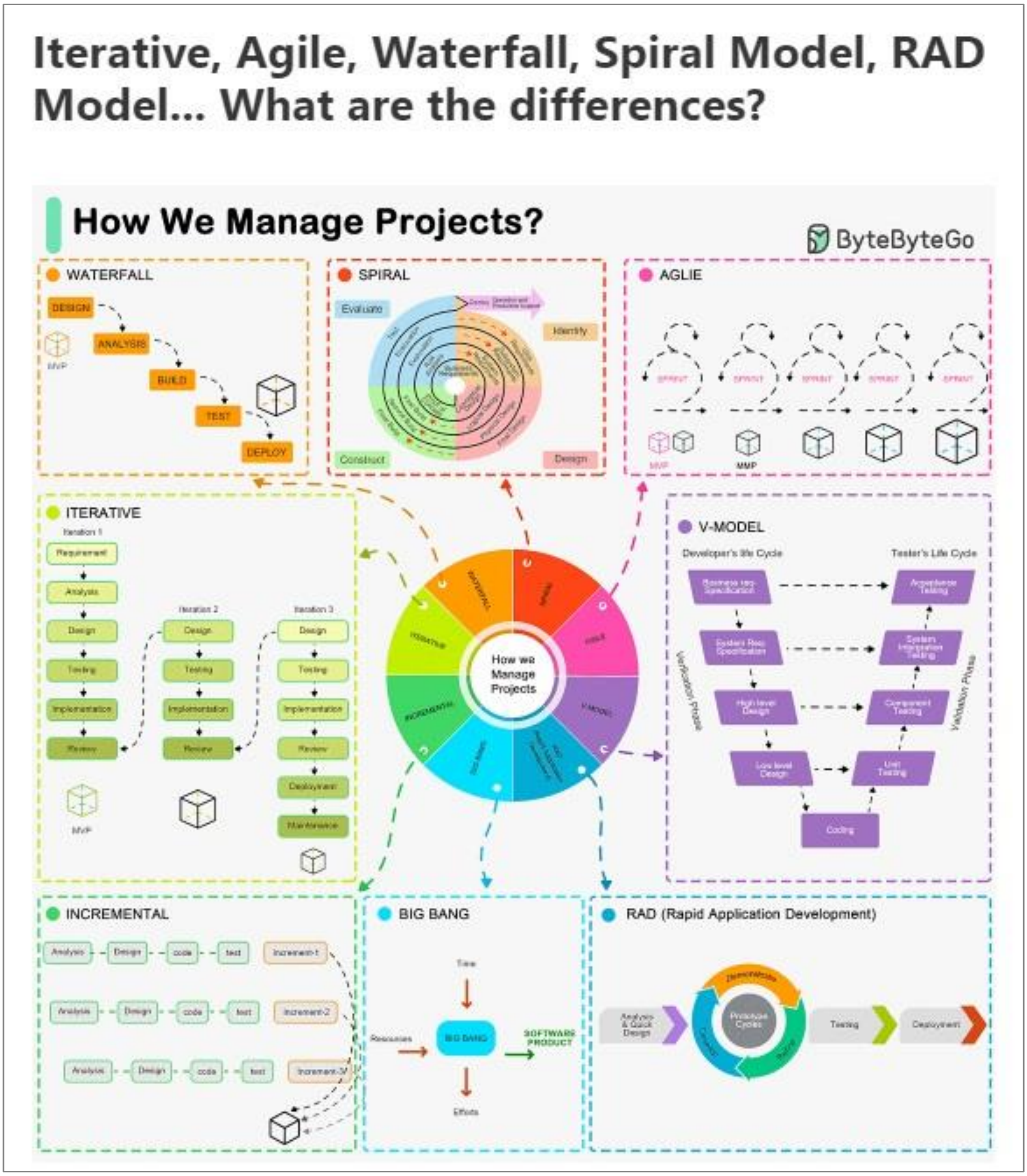
File System

What is NTFS?

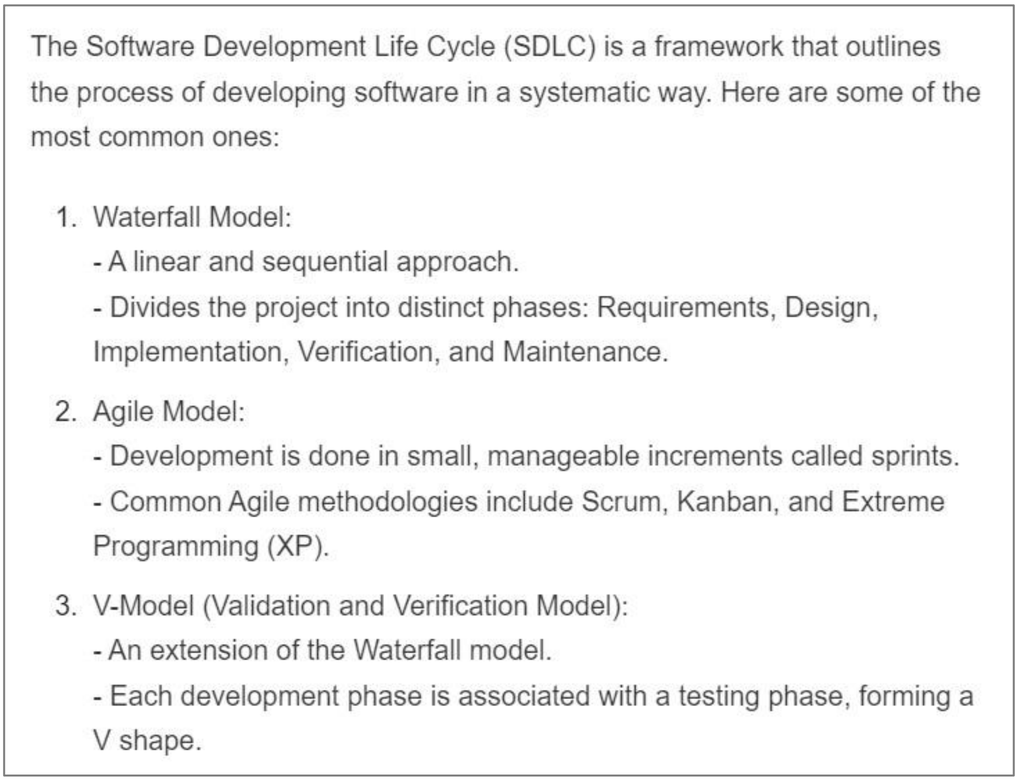
Bibliography

I. Official

CodePipeline

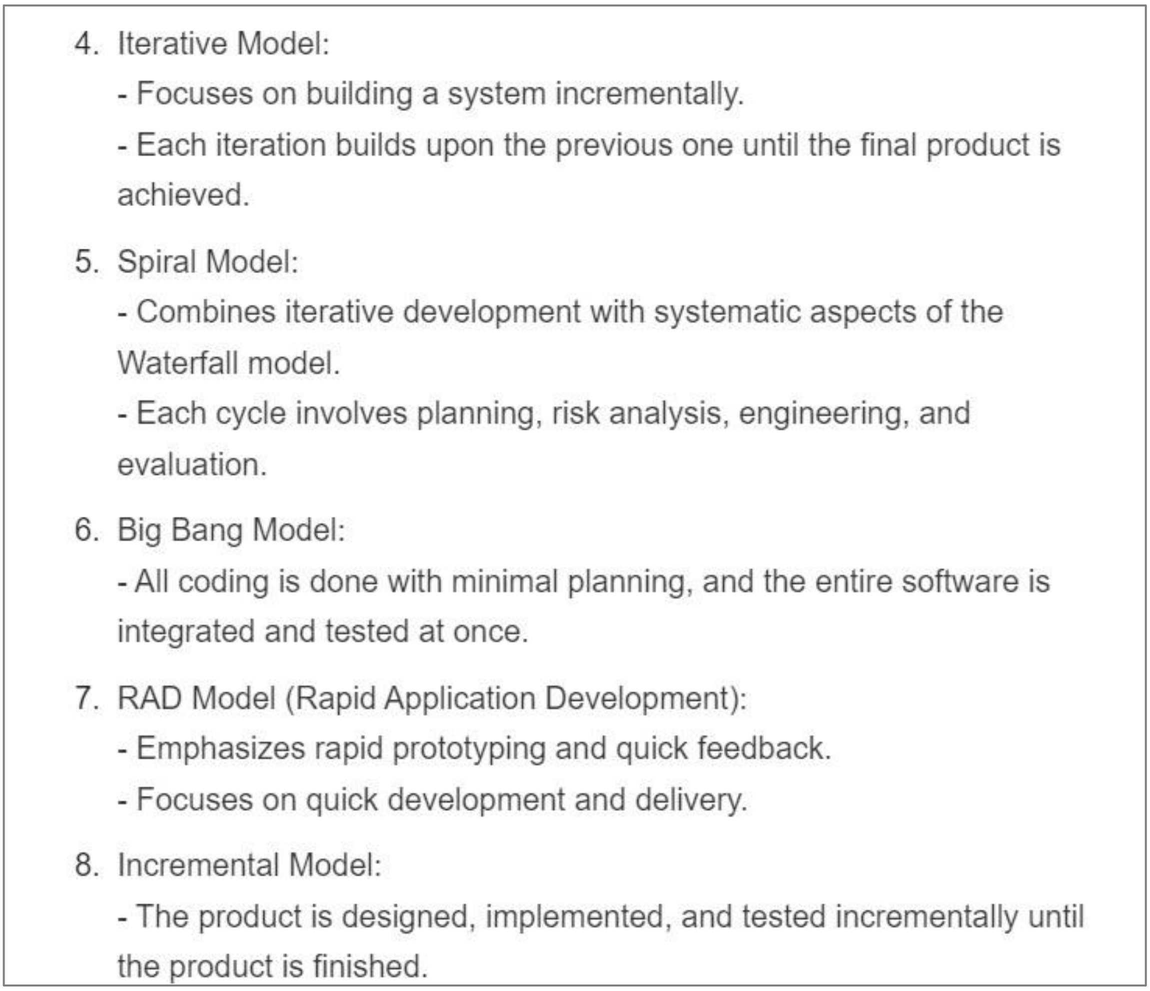


Iterative, Agile, Waterfall, Spiral Model, RAD Model... What are the differences?

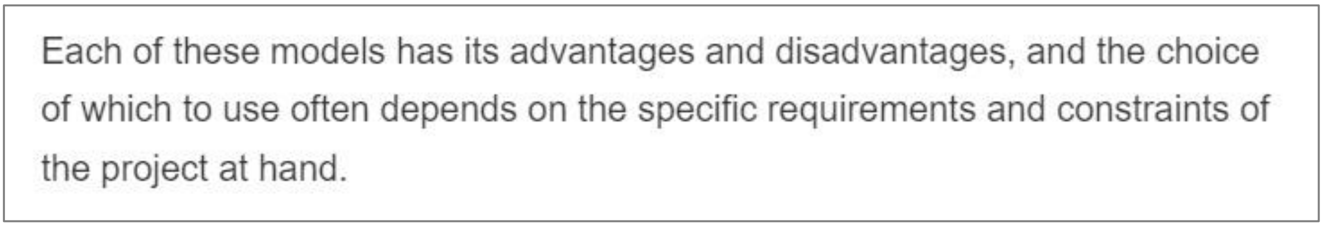


The Software Development Life Cycle (SDLC) is a framework that outlines the process of developing software in a systematic way. Here are some of the most common ones:

1. Waterfall Model:
   - A linear and sequential approach.
   - Divides the project into distinct phases: Requirements, Design, Implementation, Verification, and Maintenance.
2. Agile Model:
   - Development is done in small, manageable increments called sprints.
   - Common Agile methodologies include Scrum, Kanban, and Extreme Programming (XP).
3. V-Model (Validation and Verification Model):
   - An extension of the Waterfall model.
   - Each development phase is associated with a testing phase, forming a V shape.



4. Iterative Model:
   - Focuses on building a system incrementally.
   - Each iteration builds upon the previous one until the final product is achieved.
5. Spiral Model:
   - Combines iterative development with systematic aspects of the Waterfall model.
   - Each cycle involves planning, risk analysis, engineering, and evaluation.
6. Big Bang Model:
   - All coding is done with minimal planning, and the entire software is integrated and tested at once.
7. RAD Model (Rapid Application Development):
   - Emphasizes rapid prototyping and quick feedback.
   - Focuses on quick development and delivery.
8. Incremental Model:
   - The product is designed, implemented, and tested incrementally until the product is finished.

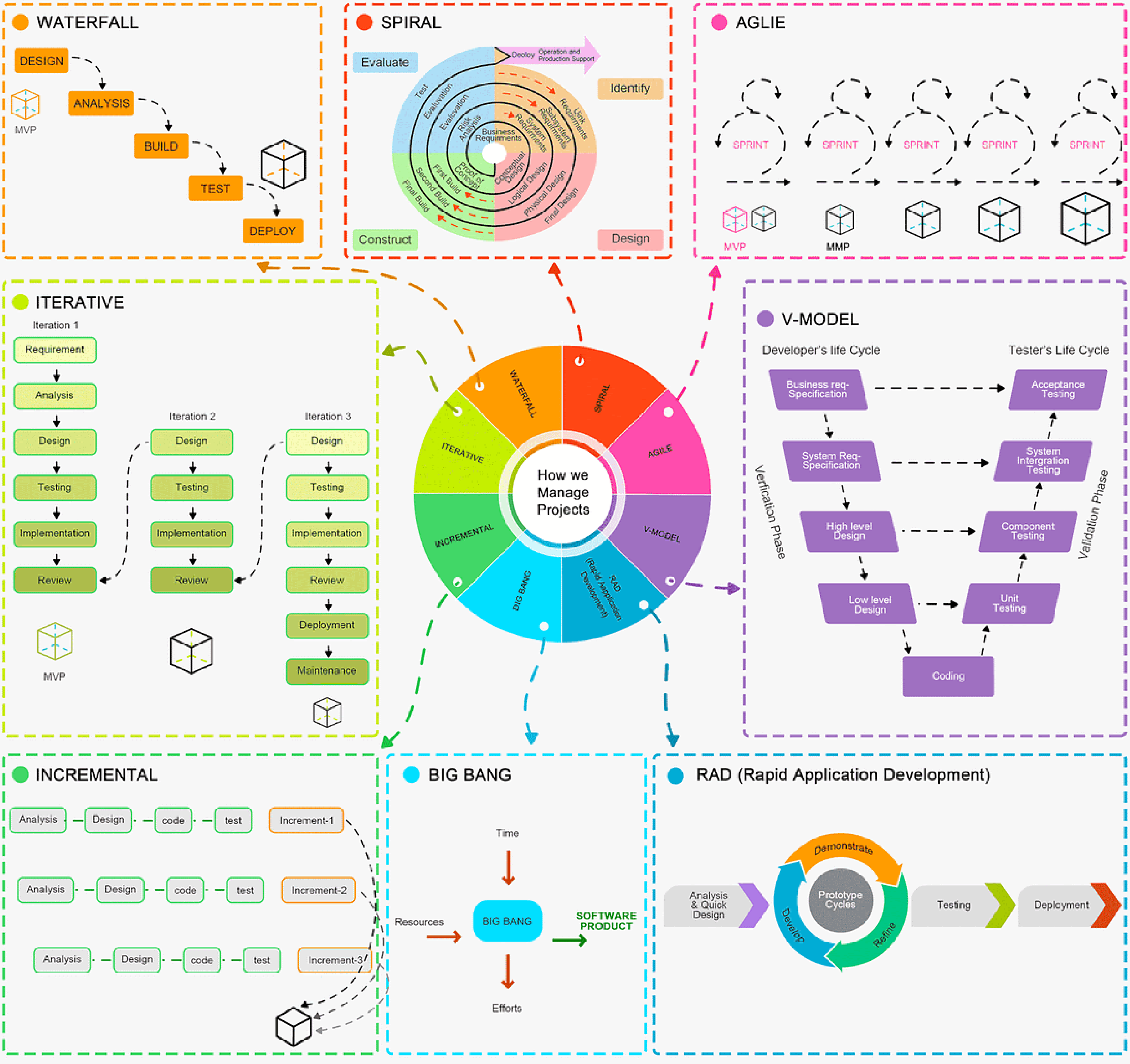


Each of these models has its advantages and disadvantages, and the choice of which to use often depends on the specific requirements and constraints of the project at hand.

[ByteByteGo 2024]

How We Manage Projects?

ByteByteGo



How we Manage Projects

WATERFALL · SPIRAL · AGILE · V-MODEL · RAD (Rapid Application Development) · BIG BANG · INCREMENTAL · ITERATIVE

WATERFALL

DESIGN → ANALYSIS → BUILD → TEST → DEPLOY

MVP

SPIRAL

Evaluate · Identify · Design · Construct

Deploy · Operation and Production Support

Test · Evaluation · Evaluation · Risk Analysis · Business Requirements · System Requirements · Subsystem Requirements · Unit Requirements · Conceptual Design · Logical Design · Physical Design · Final Design · Proof of Concept · First Build · Second Build · Final Build

AGLIE

SPRINT · SPRINT · SPRINT · SPRINT · SPRINT

MVP · MMP

ITERATIVE

Iteration 1: Requirement → Analysis → Design → Testing → Implementation → Review

MVP

Iteration 2: Design → Testing → Implementation → Review

Iteration 3: Design → Testing → Implementation → Review → Deployment → Maintenance

V-MODEL

Developer's life Cycle · Tester's Life Cycle

Verfication Phase · Validation Phase

Business req-Specification → Acceptance Testing

System Req-Specification → System Intergration Testing

High level Design → Component Testing

Low level Design → Unit Testing

Coding

INCREMENTAL

Analysis – Design – code – test – Increment-1

Analysis – Design – code – test – Increment-2

Analysis – Design – code – test – Increment-3

BIG BANG

Time

Resources → BIG BANG → SOFTWARE PRODUCT

Efforts

RAD (Rapid Application Development)

Analysis & Quick Design → Prototype Cycles (Demonstrate · Refine · Develop) → Testing → Deployment

Bibliography

I. Official

II. Unofficial

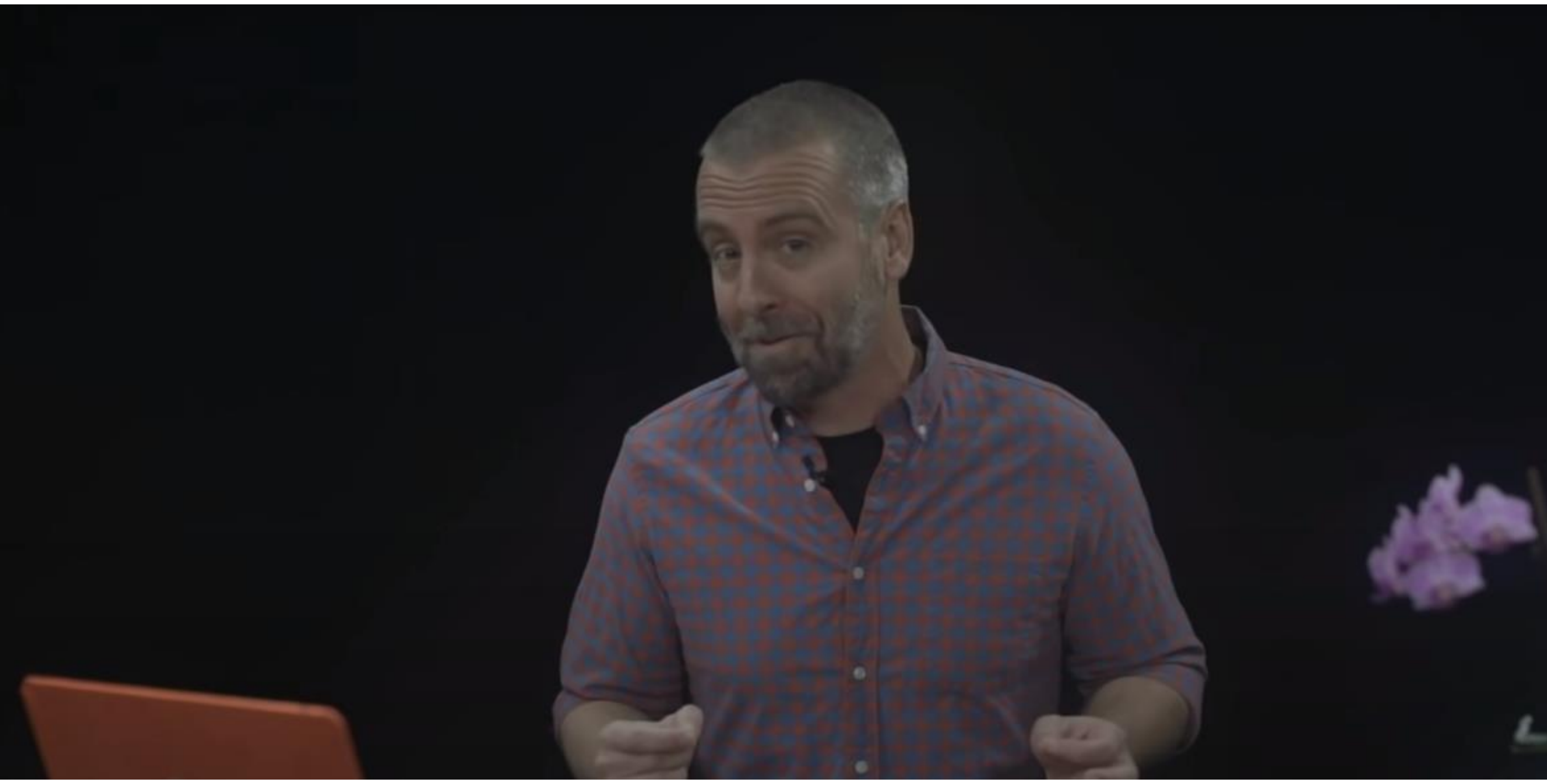
III. Critical

IV. General

[ByteByteGo 2024]

Nine essential components of a microservices architecture. ByteByteGo [Email newsletter]. July 20th 2024.

API Gateway

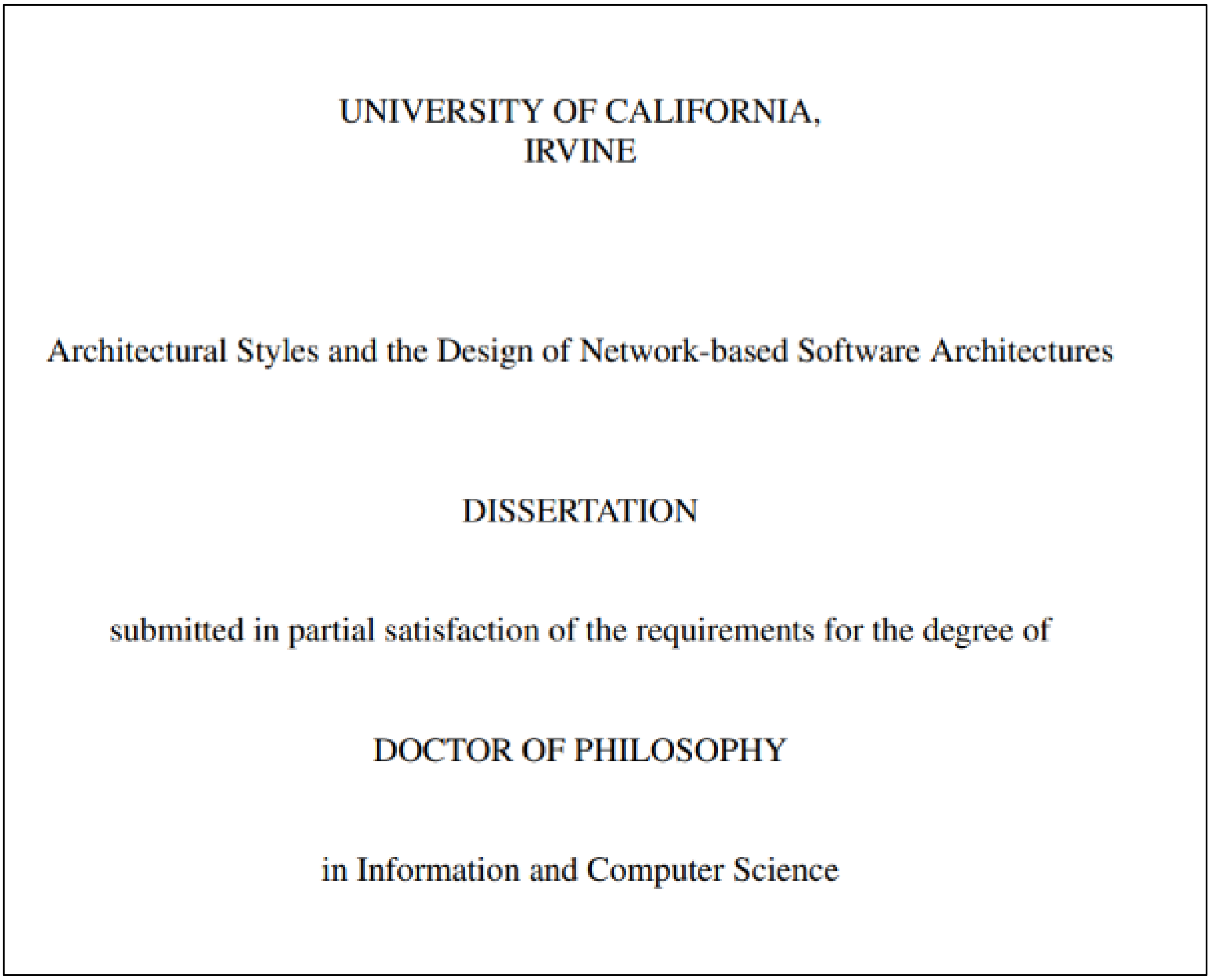


Craig Dennis explaining how an API works on his free course, on YouTube

Someone from IBM explaining what an API is

What on earth is Representational State Transfer (REST)?

Roy Fielding published his PhD dissertation in 2000. In this work, he proposed something called the RESTful architecture. You could just read the first chapter. He mentions a Monty Python sketch at one point.



UNIVERSITY OF CALIFORNIA,
IRVINE

Architectural Styles and the Design of Network-based Software Architectures

DISSERTATION

submitted in partial satisfaction of the requirements for the degree of

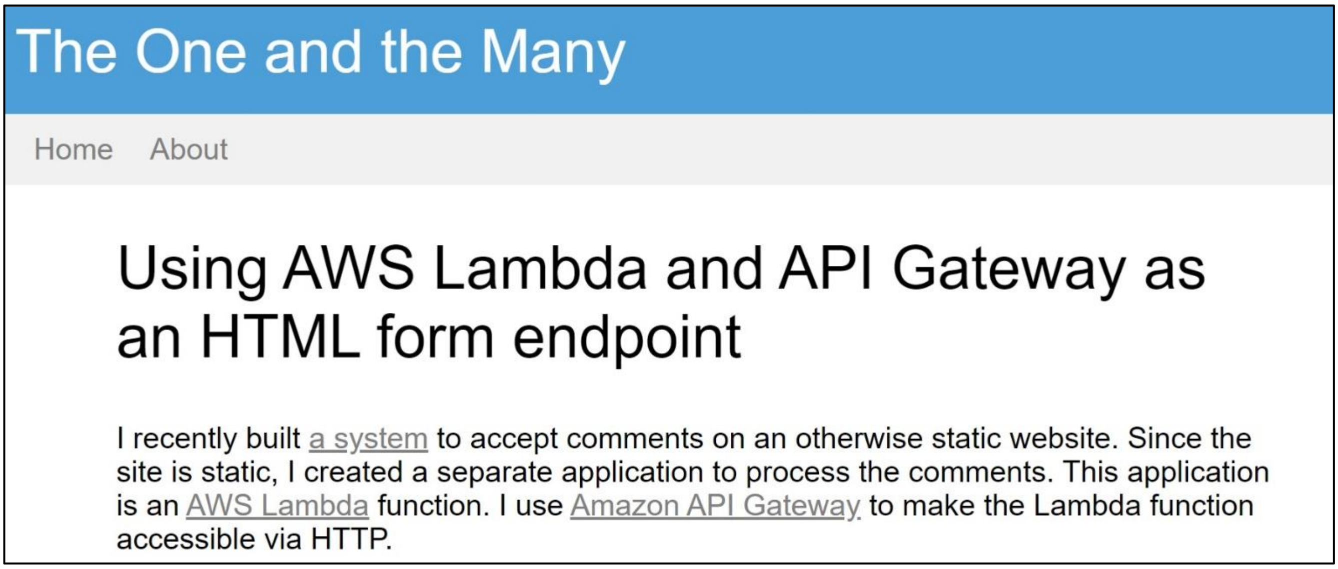
DOCTOR OF PHILOSOPHY

in Information and Computer Science

Caleb Curry explains what the expression “REST API” might mean

Chris Munns explaining API Gateway (23rd January in either 2018 or 2019)

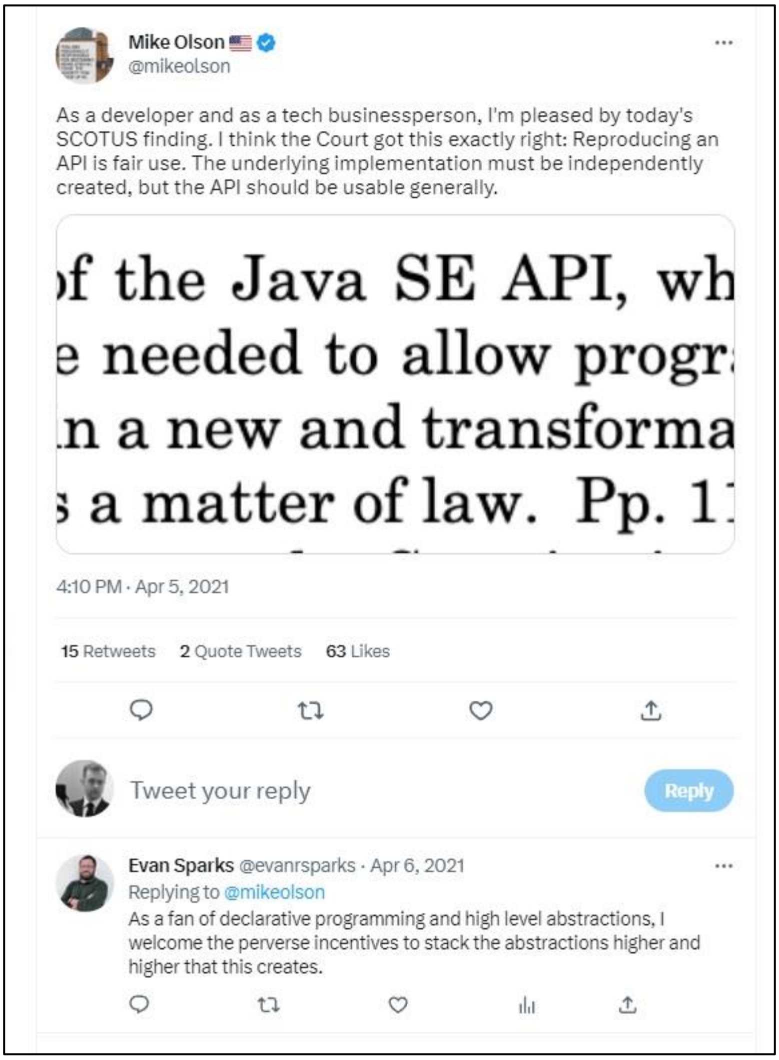
This is a great blogpost going through the details of using API Gateway:



The One and the Many

Using AWS Lambda and API Gateway as an HTML form endpoint

I recently built a system to accept comments on an otherwise static website. Since the site is static, I created a separate application to process the comments. This application is an AWS Lambda function. I use Amazon API Gateway to make the Lambda function accessible via HTTP.



Mike Olson 🇺🇸 @mikeolson

As a developer and as a tech businessperson, I'm pleased by today's SCOTUS finding. I think the Court got this exactly right: Reproducing an API is fair use. The underlying implementation must be independently created, but the API should be usable generally.

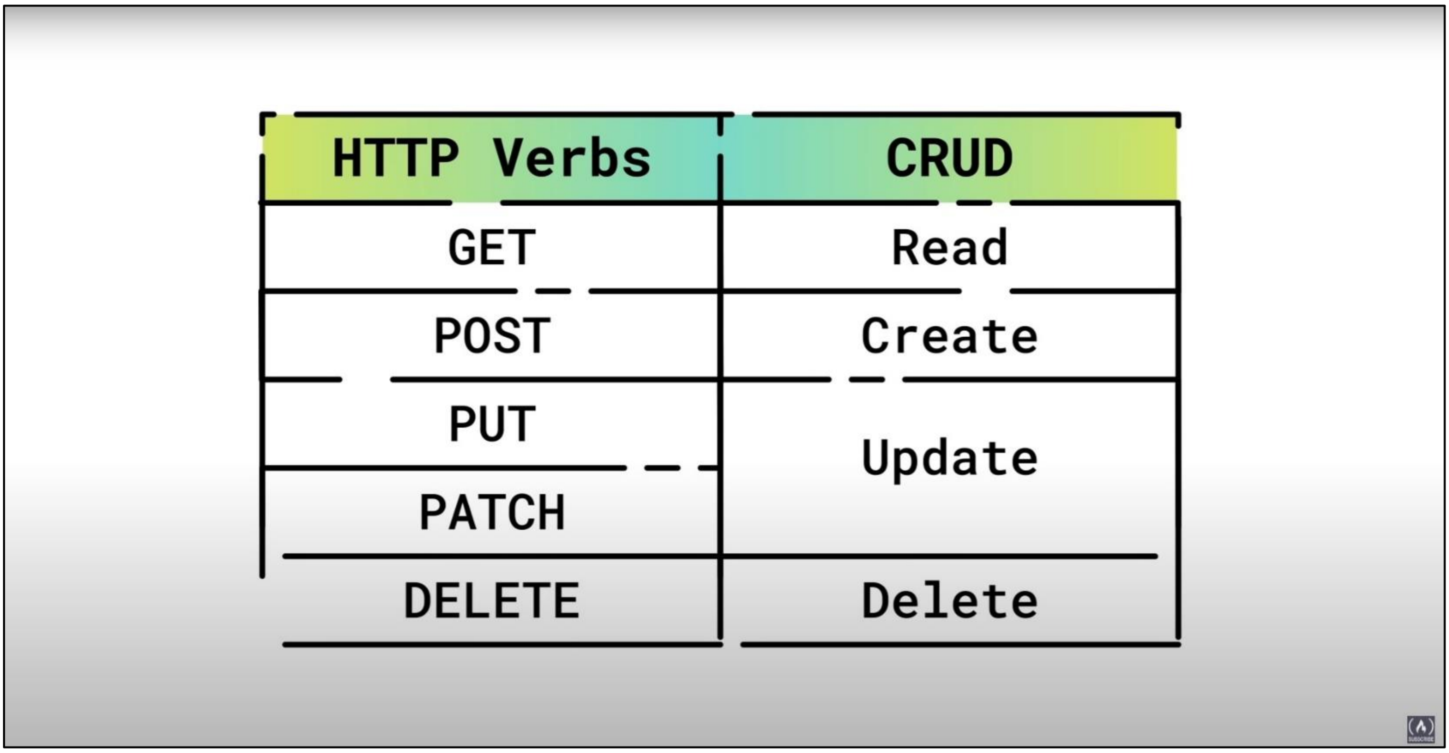
of the Java SE API, wh
e needed to allow progr
n a new and transforma
s a matter of law. Pp. 11

4:10 PM · Apr 5, 2021

15 Retweets 2 Quote Tweets 63 Likes

Evan Sparks @evanrsparks · Apr 6, 2021
Replying to @mikeolson
As a fan of declarative programming and high level abstractions, I welcome the perverse incentives to stack the abstractions higher and higher that this creates.

Craig Dennis' Presentation

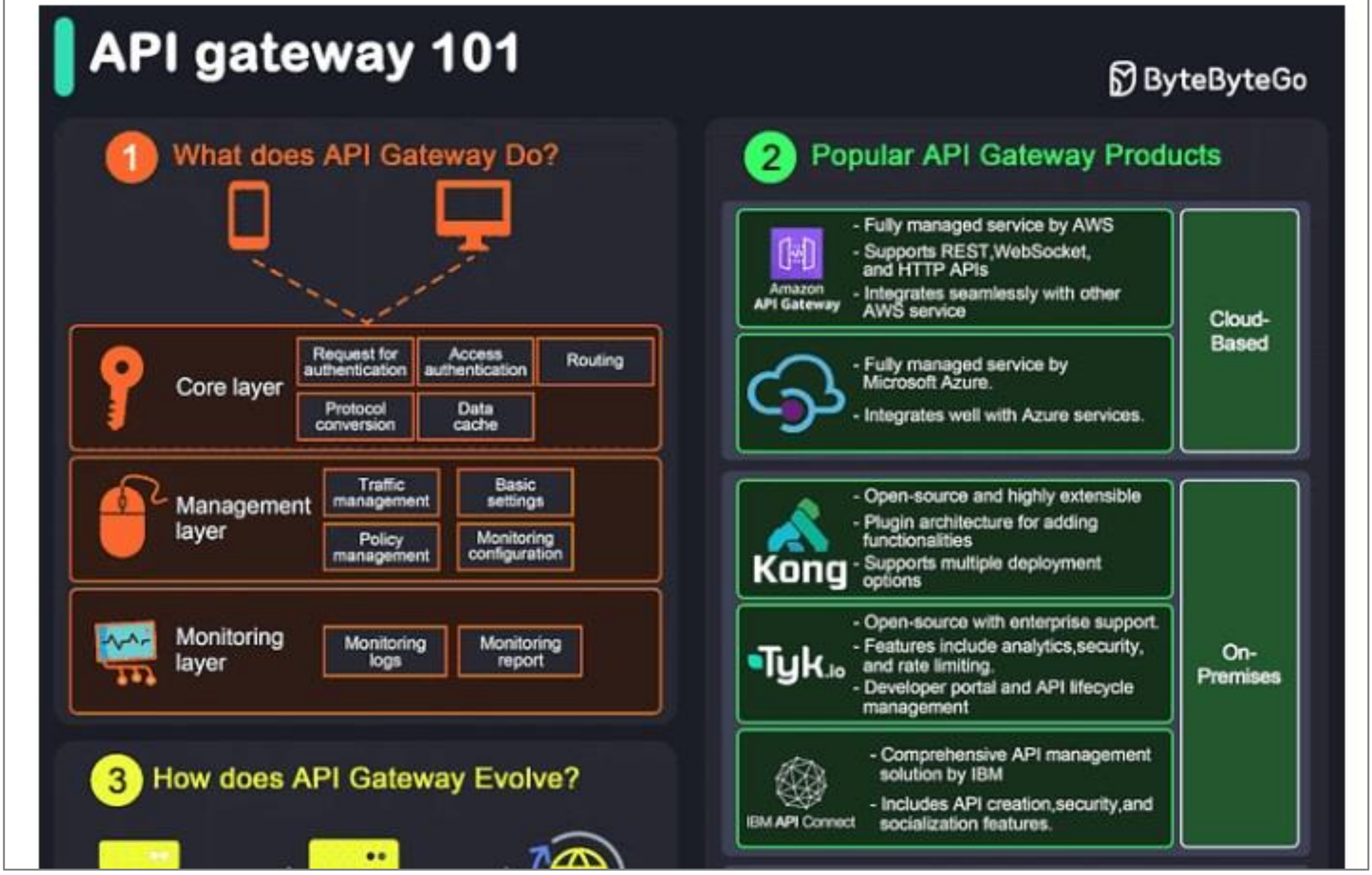


HTTP Verbs	CRUD
GET	Read
POST	Create
PUT	Update
PATCH	
DELETE	Delete

I have found Video 6 to be extremely helpful.

API Gateway 101

An API gateway is a server that acts as an API front-end, receiving API requests, enforcing throttling and security policies, passing requests to the back-end service, and then returning the appropriate result to the client.



API gateway 101

ByteByteGo

1 What does API Gateway Do?

Core layer: Request for authentication, Access authentication, Routing, Protocol conversion, Data cache

Management layer: Traffic management, Basic settings, Policy management, Monitoring configuration

Monitoring layer: Monitoring logs, Monitoring report

2 Popular API Gateway Products

Cloud-Based

Amazon API Gateway
- Fully managed service by AWS
- Supports REST,WebSocket, and HTTP APIs
- Integrates seamlessly with other AWS service

- Fully managed service by Microsoft Azure.
- Integrates well with Azure services.

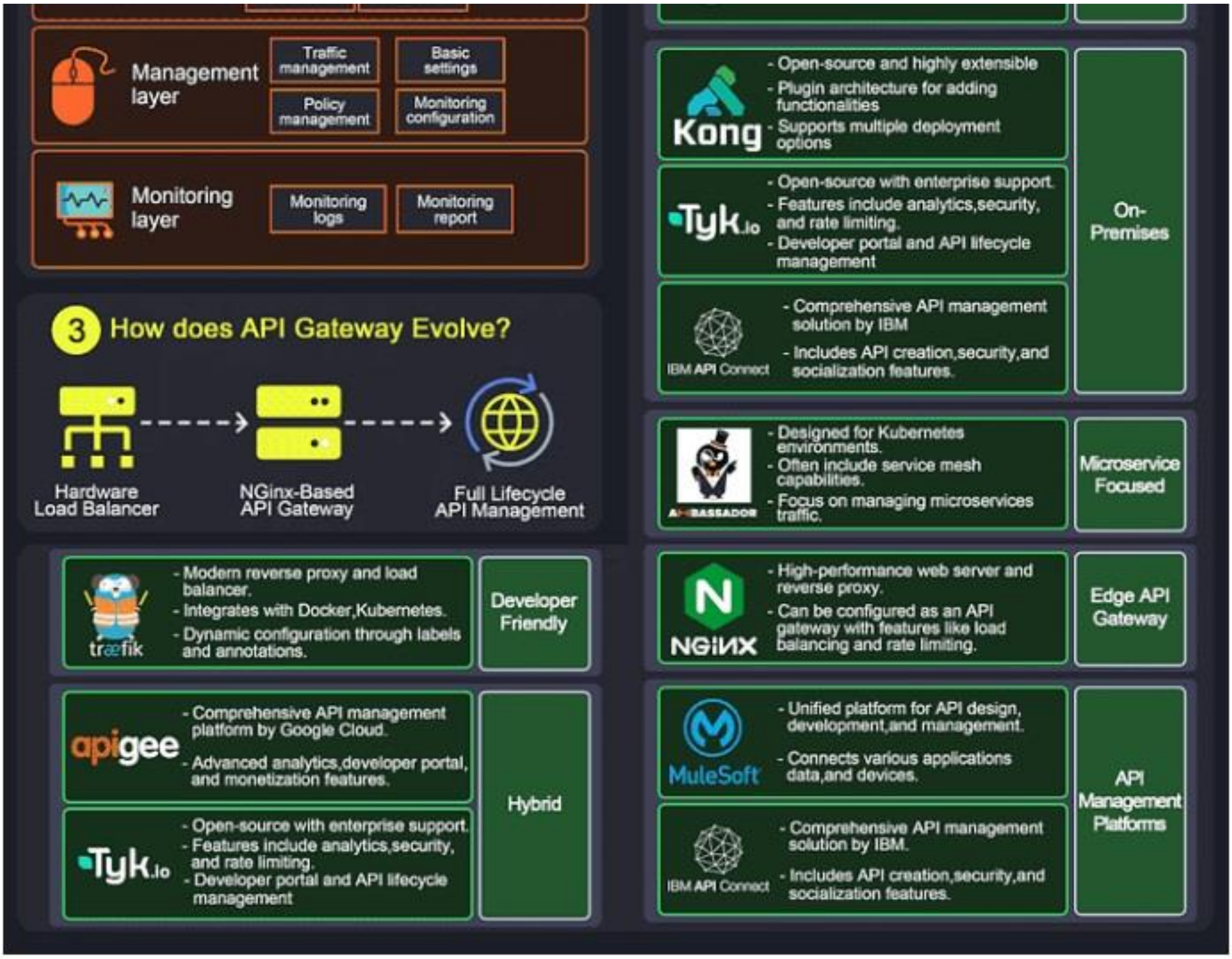
On-Premises

Kong
- Open-source and highly extensible
- Plugin architecture for adding functionalities
- Supports multiple deployment options

Tyk.io
- Open-source with enterprise support.
- Features include analytics,security, and rate limiting.
- Developer portal and API lifecycle management

IBM API Connect
- Comprehensive API management solution by IBM
- Includes API creation,security,and socialization features.

3 How does API Gateway Evolve?



Management layer

Traffic management

Basic settings

Policy management

Monitoring configuration

Monitoring layer

Monitoring logs

Monitoring report

3 How does API Gateway Evolve?

Hardware Load Balancer → NGinx-Based API Gateway → Full Lifecycle API Management

Developer Friendly

traefik
- Modern reverse proxy and load balancer.
- Integrates with Docker,Kubernetes.
- Dynamic configuration through labels and annotations.

Hybrid

apigee
- Comprehensive API management platform by Google Cloud.
- Advanced analytics,developer portal, and monetization features.

Tyk.io
- Open-source with enterprise support.
- Features include analytics,security, and rate limiting.
- Developer portal and API lifecycle management

On-Premises

Kong
- Open-source and highly extensible
- Plugin architecture for adding functionalities
- Supports multiple deployment options

Tyk.io
- Open-source with enterprise support.
- Features include analytics,security, and rate limiting.
- Developer portal and API lifecycle management

IBM API Connect
- Comprehensive API management solution by IBM
- Includes API creation,security,and socialization features.

Microservice Focused

AMBASSADOR
- Designed for Kubernetes environments.
- Often include service mesh capabilities.
- Focus on managing microservices traffic.

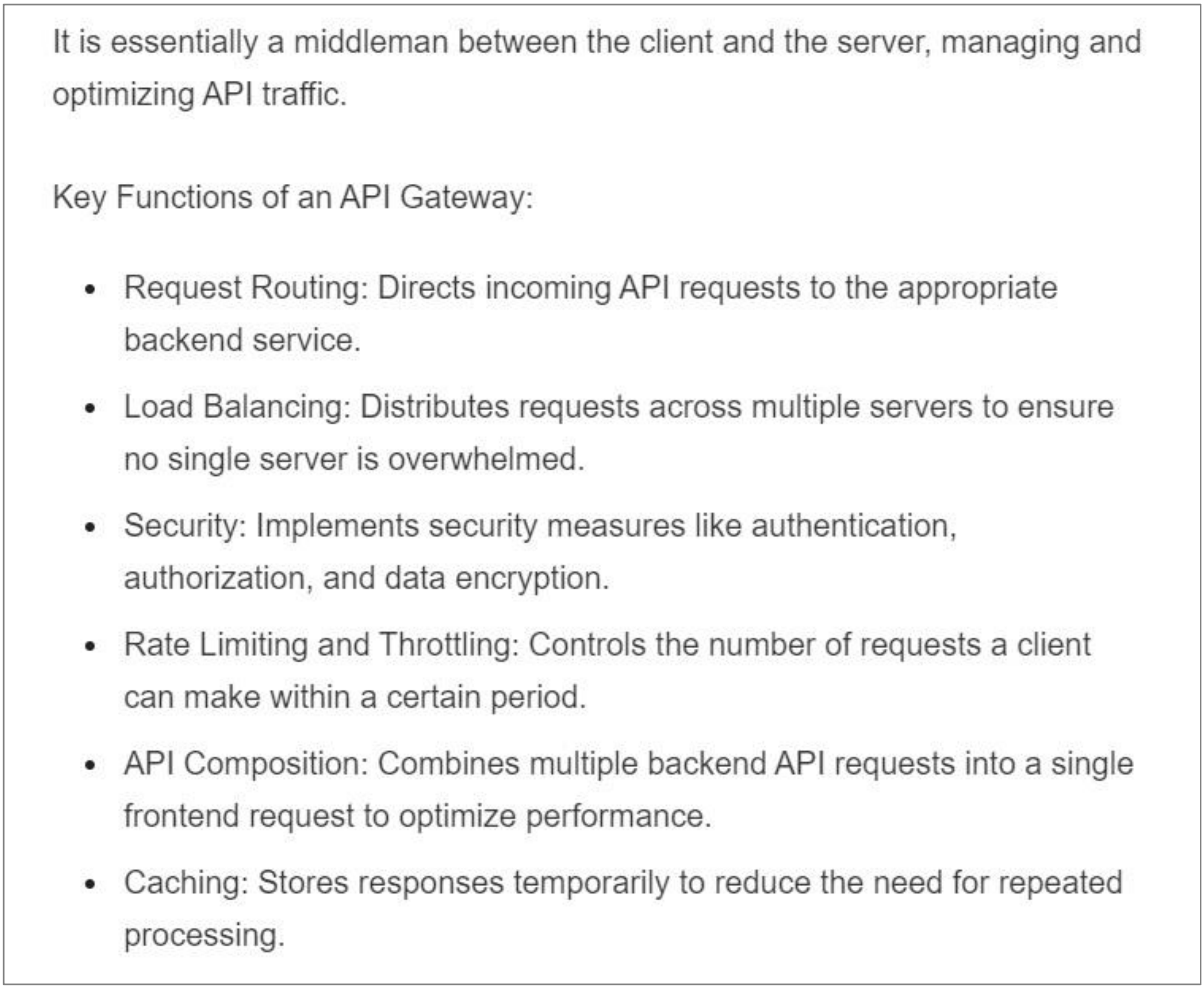
Edge API Gateway

NGINX
- High-performance web server and reverse proxy.
- Can be configured as an API gateway with features like load balancing and rate limiting.

API Management Platforms

MuleSoft
- Unified platform for API design, development,and management.
- Connects various applications data,and devices.

IBM API Connect
- Comprehensive API management solution by IBM.
- Includes API creation,security,and socialization features.



It is essentially a middleman between the client and the server, managing and optimizing API traffic.

Key Functions of an API Gateway:

- Request Routing: Directs incoming API requests to the appropriate backend service.
- Load Balancing: Distributes requests across multiple servers to ensure no single server is overwhelmed.
- Security: Implements security measures like authentication, authorization, and data encryption.
- Rate Limiting and Throttling: Controls the number of requests a client can make within a certain period.
- API Composition: Combines multiple backend API requests into a single frontend request to optimize performance.
- Caching: Stores responses temporarily to reduce the need for repeated processing.

[BBG 2024]

Bibliography

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

I. Official

II. Unofficial

[Summercat 2017]

https://blog.summercat.com/using-aws-lambda-and-api-gateway-as-html-form-endpoint.html

[Brodhagen 2016]

Brodhagen, Kenn (2016). How to create a Request object for AWS API Gateway & Lambda. Available at:

https://www.youtube.com/watch?v=2Z-Utw_xl4c&ab_channel=KennBrodhagen

[BBG 2024]

API Gateway 101. *Byte Byte Go* [Online Newsletter]. July 27th 2024.

[Taylor 2024]

Taylor, Thomas (2024). API Key Authentication with API Gateway using AWS SDK. Available at: <https://dev.to/aws-builders/api-key-authentication-with-api-gateway-using-aws-cdk-5cjd?utm_source=substack&utm_medium=email>

III. Critical

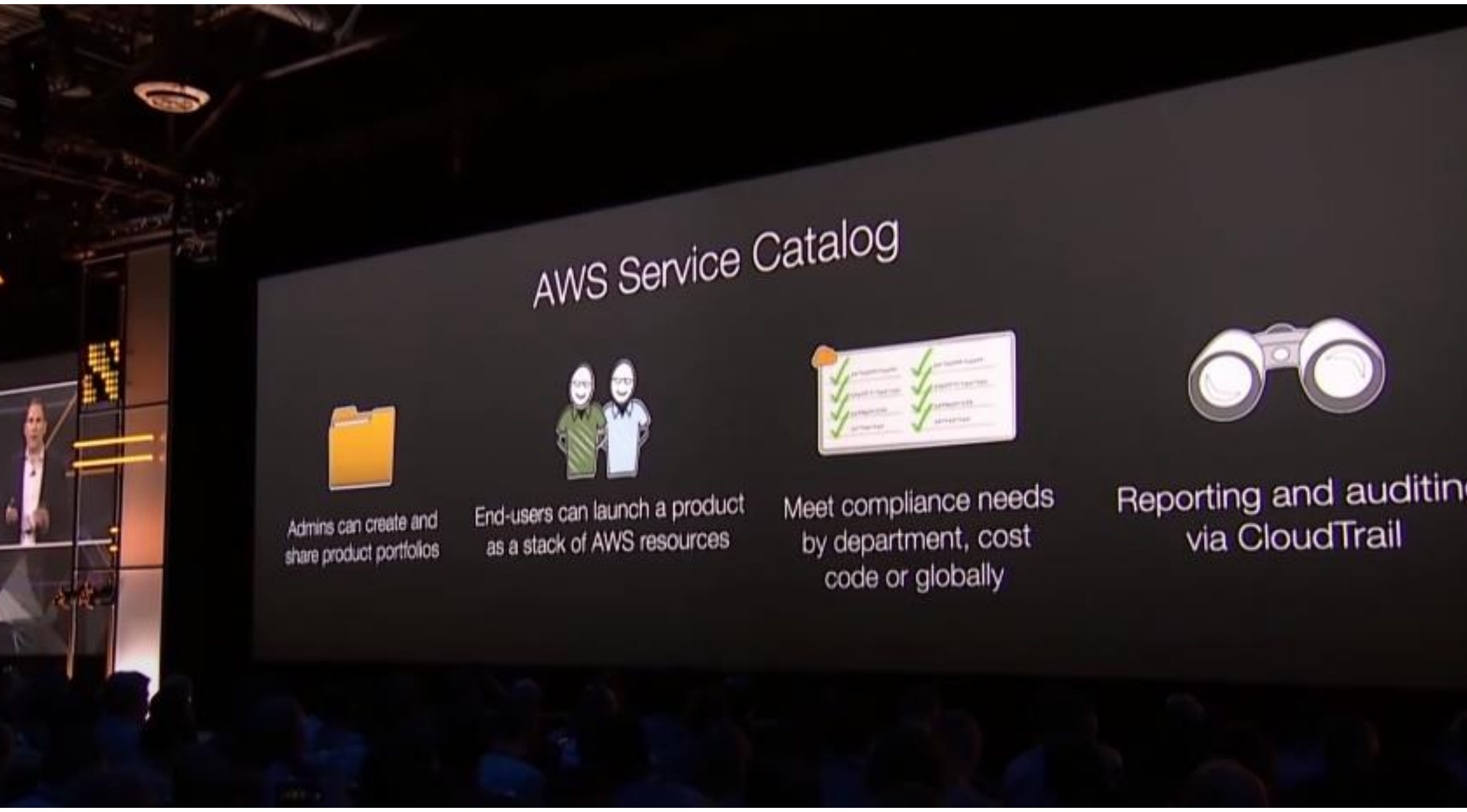
Device Farm

Service Catalog

Introducing the AWS Service Catalog

A service for discovering and provisioning applications in the cloud

Available early 2015



AWS Service Catalog

Admins can create and share product portfolios

End-users can launch a product as a stack of AWS resources

Meet compliance needs by department, cost code or globally

Reporting and auditin via CloudTrail

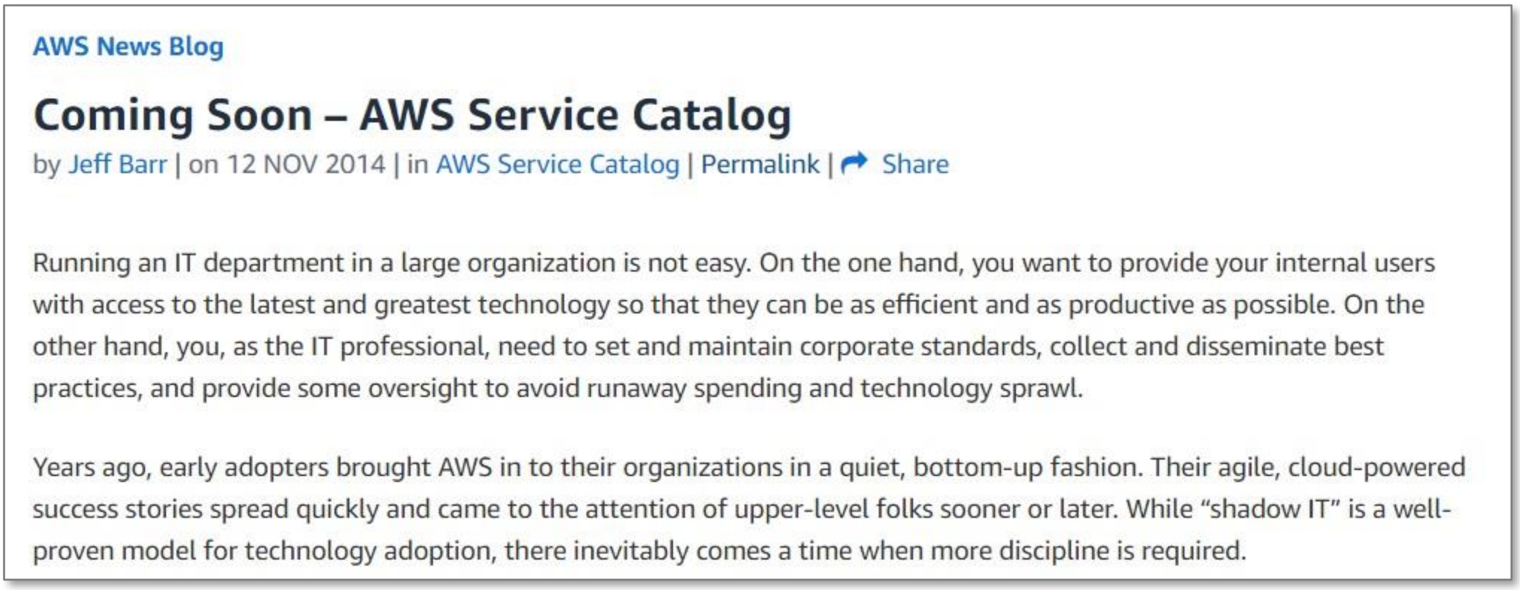
“And so it’s my pleasure to announce today—coming in early 2015—the AWS Service Catalog.

This will allow enterprise administrators to create portfolios of products, set up in the configuration that they want deployed.

They can make them easily discoverable for employees of the company, on a portal that they host.

Meet the compliance needs that they have in difference businesses.

Then, again, they have that strong visibility, as all the activity is tracked in AWS CloudTrail.”



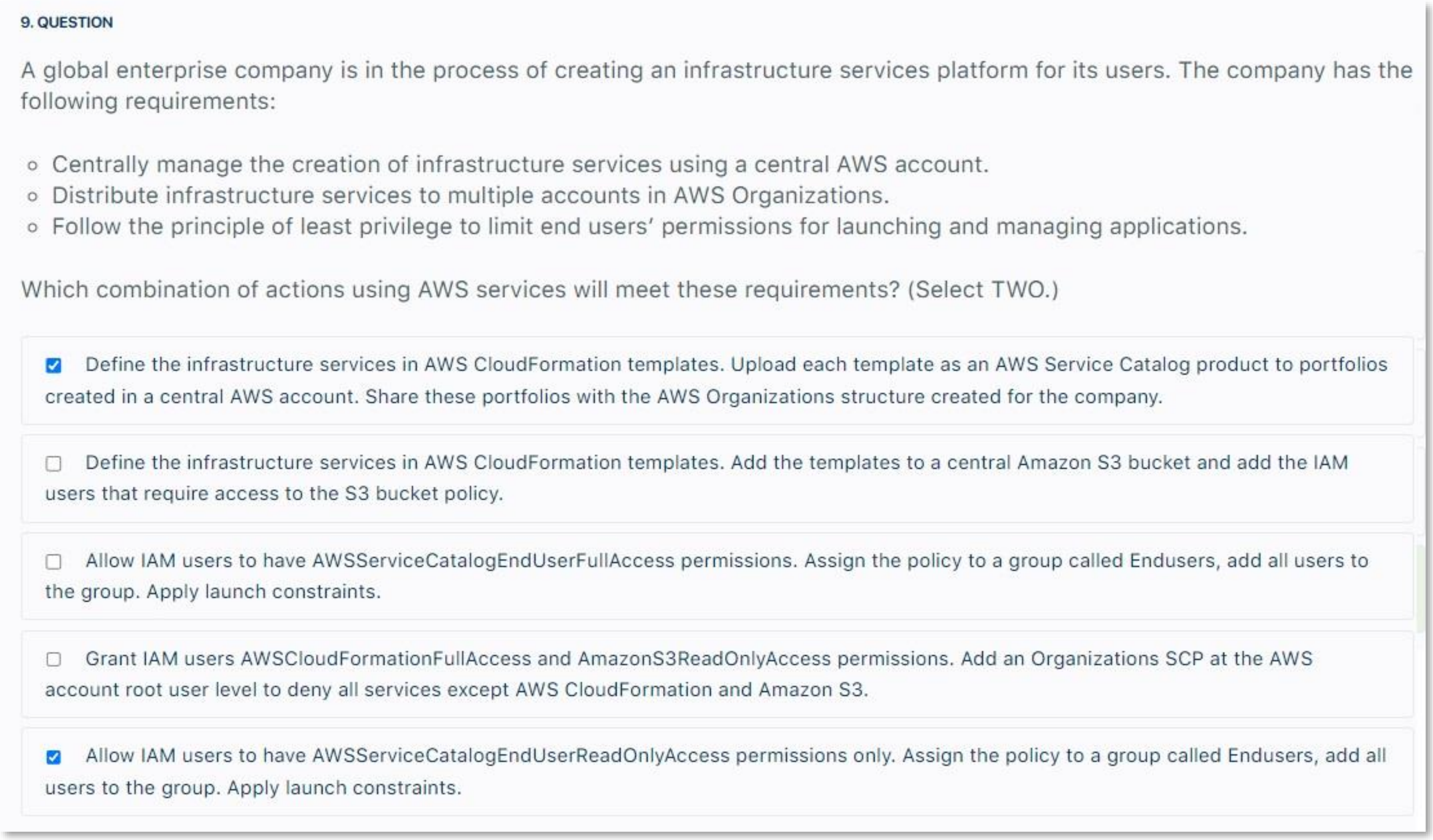
AWS News Blog

Coming Soon – AWS Service Catalog

by Jeff Barr | on 12 NOV 2014 | in AWS Service Catalog | Permalink | Share

Running an IT department in a large organization is not easy. On the one hand, you want to provide your internal users with access to the latest and greatest technology so that they can be as efficient and as productive as possible. On the other hand, you, as the IT professional, need to set and maintain corporate standards, collect and disseminate best practices, and provide some oversight to avoid runaway spending and technology sprawl.

Years ago, early adopters brought AWS in to their organizations in a quiet, bottom-up fashion. Their agile, cloud-powered success stories spread quickly and came to the attention of upper-level folks sooner or later. While "shadow IT" is a well-proven model for technology adoption, there inevitably comes a time when more discipline is required.



9. QUESTION

A global enterprise company is in the process of creating an infrastructure services platform for its users. The company has the following requirements:

- Centrally manage the creation of infrastructure services using a central AWS account.
- Distribute infrastructure services to multiple accounts in AWS Organizations.
- Follow the principle of least privilege to limit end users' permissions for launching and managing applications.

Which combination of actions using AWS services will meet these requirements? (Select TWO.)

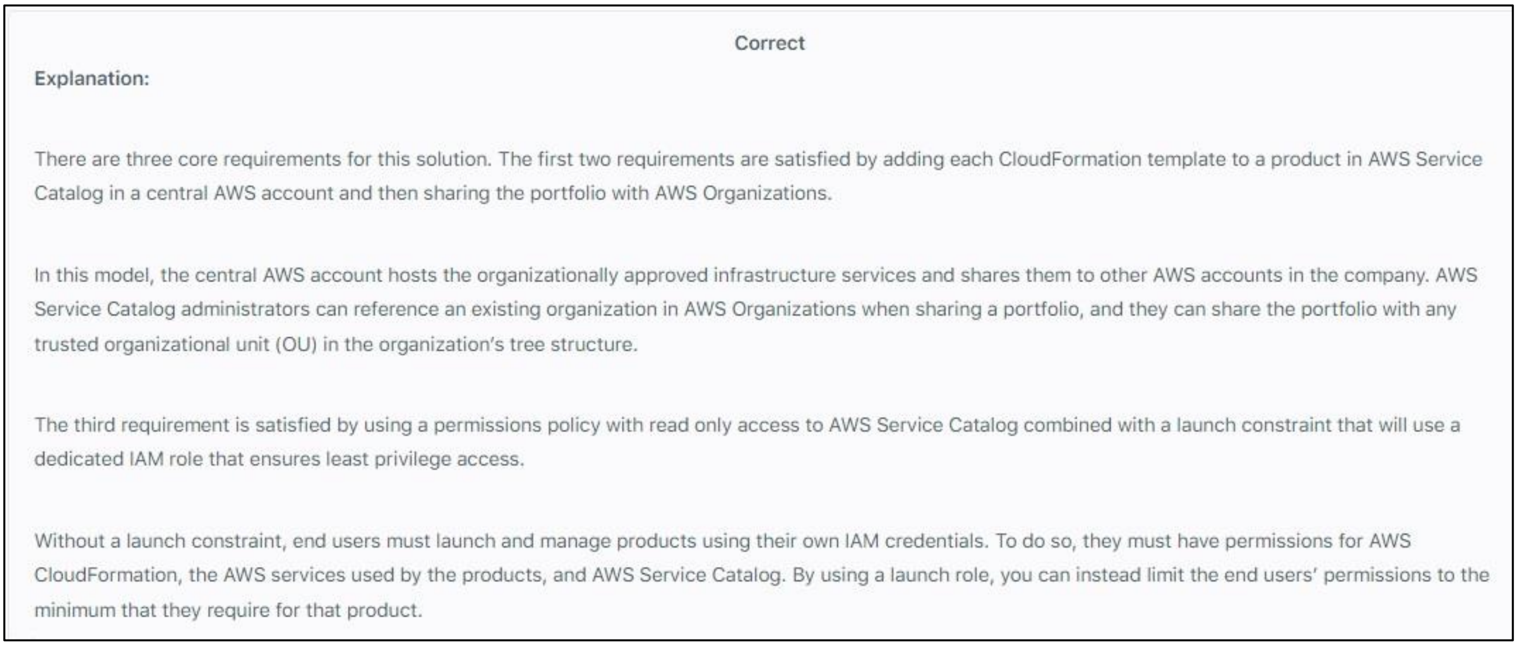
☑ Define the infrastructure services in AWS CloudFormation templates. Upload each template as an AWS Service Catalog product to portfolios created in a central AWS account. Share these portfolios with the AWS Organizations structure created for the company.

□ Define the infrastructure services in AWS CloudFormation templates. Add the templates to a central Amazon S3 bucket and add the IAM users that require access to the S3 bucket policy.

□ Allow IAM users to have AWSServiceCatalogEndUserFullAccess permissions. Assign the policy to a group called Endusers, add all users to the group. Apply launch constraints.

□ Grant IAM users AWSCloudFormationFullAccess and AmazonS3ReadOnlyAccess permissions. Add an Organizations SCP at the AWS account root user level to deny all services except AWS CloudFormation and Amazon S3.

☑ Allow IAM users to have AWSServiceCatalogEndUserReadOnlyAccess permissions only. Assign the policy to a group called Endusers, add all users to the group. Apply launch constraints.



Correct

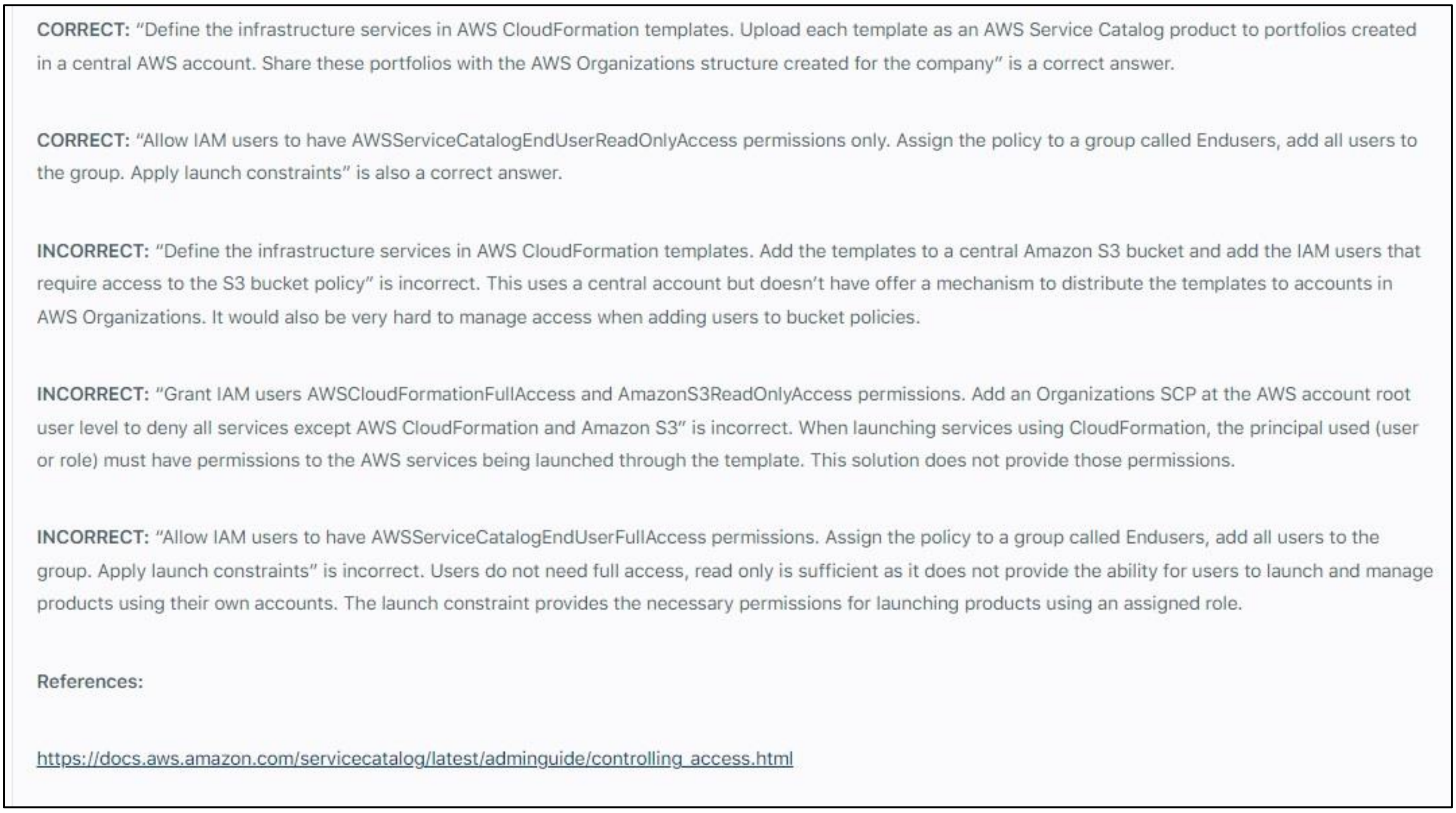
Explanation:

There are three core requirements for this solution. The first two requirements are satisfied by adding each CloudFormation template to a product in AWS Service Catalog in a central AWS account and then sharing the portfolio with AWS Organizations.

In this model, the central AWS account hosts the organizationally approved infrastructure services and shares them to other AWS accounts in the company. AWS Service Catalog administrators can reference an existing organization in AWS Organizations when sharing a portfolio, and they can share the portfolio with any trusted organizational unit (OU) in the organization's tree structure.

The third requirement is satisfied by using a permissions policy with read only access to AWS Service Catalog combined with a launch constraint that will use a dedicated IAM role that ensures least privilege access.

Without a launch constraint, end users must launch and manage products using their own IAM credentials. To do so, they must have permissions for AWS CloudFormation, the AWS services used by the products, and AWS Service Catalog. By using a launch role, you can instead limit the end users' permissions to the minimum that they require for that product.



CORRECT: "Define the infrastructure services in AWS CloudFormation templates. Upload each template as an AWS Service Catalog product to portfolios created in a central AWS account. Share these portfolios with the AWS Organizations structure created for the company" is a correct answer.

CORRECT: "Allow IAM users to have AWSServiceCatalogEndUserReadOnlyAccess permissions only. Assign the policy to a group called Endusers, add all users to the group. Apply launch constraints" is also a correct answer.

INCORRECT: "Define the infrastructure services in AWS CloudFormation templates. Add the templates to a central Amazon S3 bucket and add the IAM users that require access to the S3 bucket policy" is incorrect. This uses a central account but doesn't have offer a mechanism to distribute the templates to accounts in AWS Organizations. It would also be very hard to manage access when adding users to bucket policies.

INCORRECT: "Grant IAM users AWSCloudFormationFullAccess and AmazonS3ReadOnlyAccess permissions. Add an Organizations SCP at the AWS account root user level to deny all services except AWS CloudFormation and Amazon S3" is incorrect. When launching services using CloudFormation, the principal used (user or role) must have permissions to the AWS services being launched through the template. This solution does not provide those permissions.

INCORRECT: "Allow IAM users to have AWSServiceCatalogEndUserFullAccess permissions. Assign the policy to a group called Endusers, add all users to the group. Apply launch constraints" is incorrect. Users do not need full access, read only is sufficient as it does not provide the ability for users to launch and manage products using their own accounts. The launch constraint provides the necessary permissions for launching products using an assigned role.

References:

https://docs.aws.amazon.com/servicecatalog/latest/adminguide/controlling_access.html

Bibliography

I. Official

[Lal 2016]

Lal, Abishek (2014). Introducing AWS Service Catalog. *ReInvent 2014* [Conference]. Available at: <https://www.youtube.com/watch?v=5mn3NdJWL-Y&ab_channel=AmazonWebServices>

II. Unofficial

III. Critical

IV. General

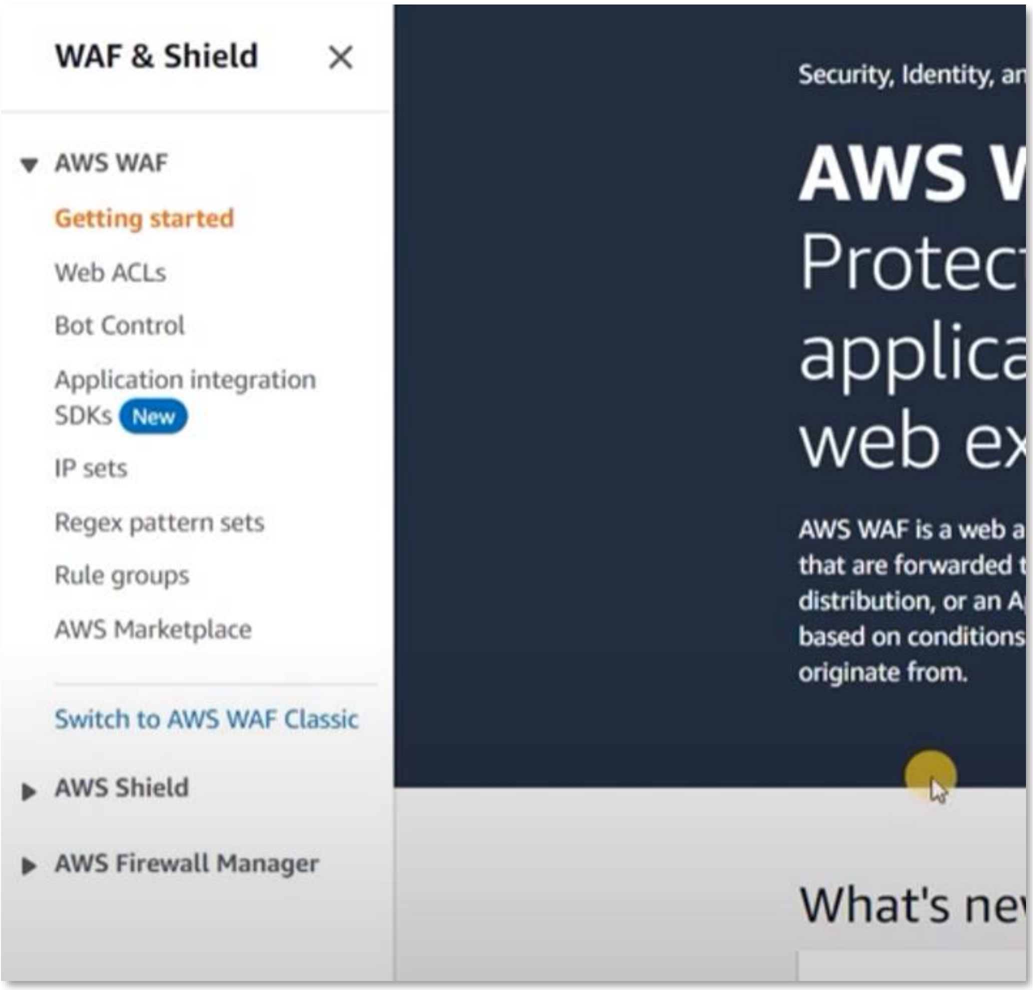
Web Application Firewall (WAF)

Nate Dye explaining how AWS WAF works, in October 2015. WAF stands for Web Application Firewall. Image from Re:Invent 2015.

Mike Pound explaining how an SQL injection attack works

What on earth is “Regex”?

In the left-hand side of the console, you might see this term.



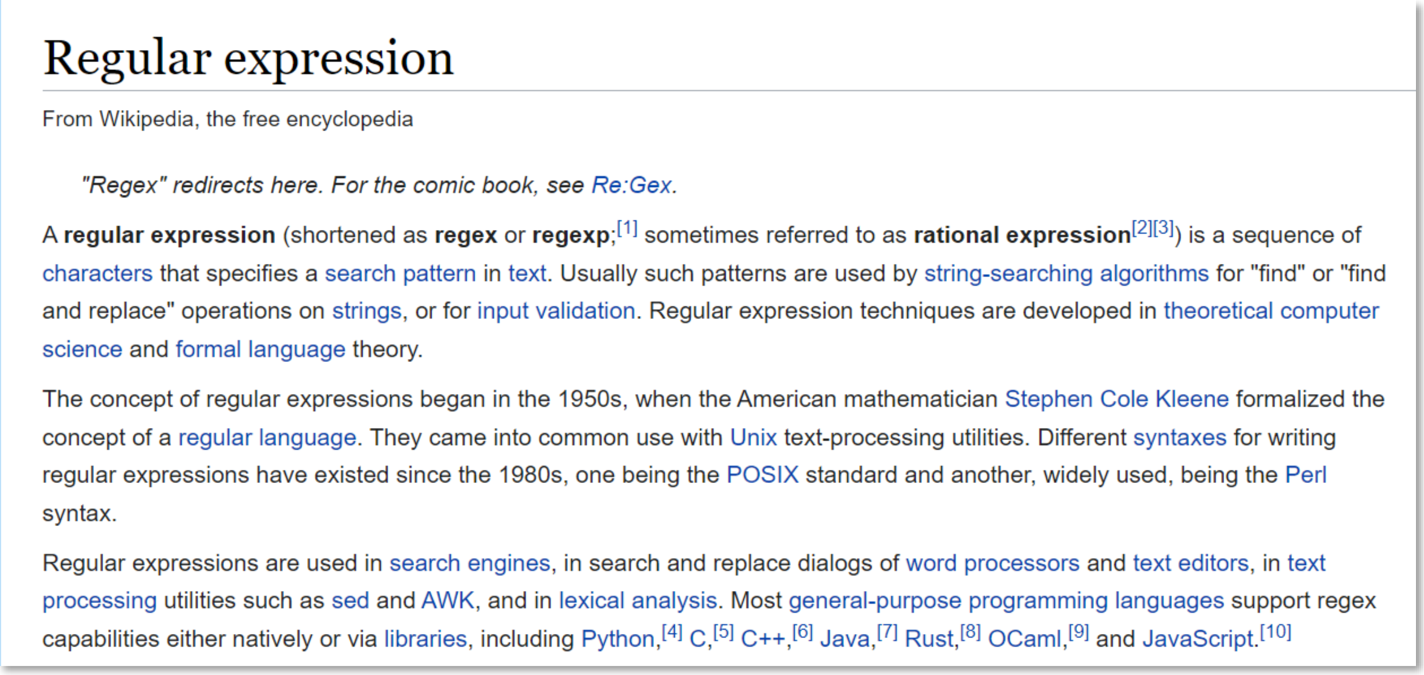
Security, Identity, an

AWS W
Protec
applica
web ex

AWS WAF is a web a
that are forwarded t
distribution, or an A
based on conditions
originate from.

What's nev

It turns out that “regex” is short for “regular expression”. Regular expressions are used to search for characters, in a way that accounts for variations in the characters.



Regular expression

From Wikipedia, the free encyclopedia

"Regex" redirects here. For the comic book, see Re:Gex.

A **regular expression** (shortened as **regex** or **regexp**;[1] sometimes referred to as **rational expression**[2][3]) is a sequence of characters that specifies a search pattern in text. Usually such patterns are used by string-searching algorithms for "find" or "find and replace" operations on strings, or for input validation. Regular expression techniques are developed in theoretical computer science and formal language theory.

The concept of regular expressions began in the 1950s, when the American mathematician Stephen Cole Kleene formalized the concept of a regular language. They came into common use with Unix text-processing utilities. Different syntaxes for writing regular expressions have existed since the 1980s, one being the POSIX standard and another, widely used, being the Perl syntax.

Regular expressions are used in search engines, in search and replace dialogs of word processors and text editors, in text processing utilities such as sed and AWK, and in lexical analysis. Most general-purpose programming languages support regex capabilities either natively or via libraries, including Python,[4] C,[5] C++,[6] Java,[7] Rust,[8] OCaml,[9] and JavaScript.[10]

Introduction to Regex

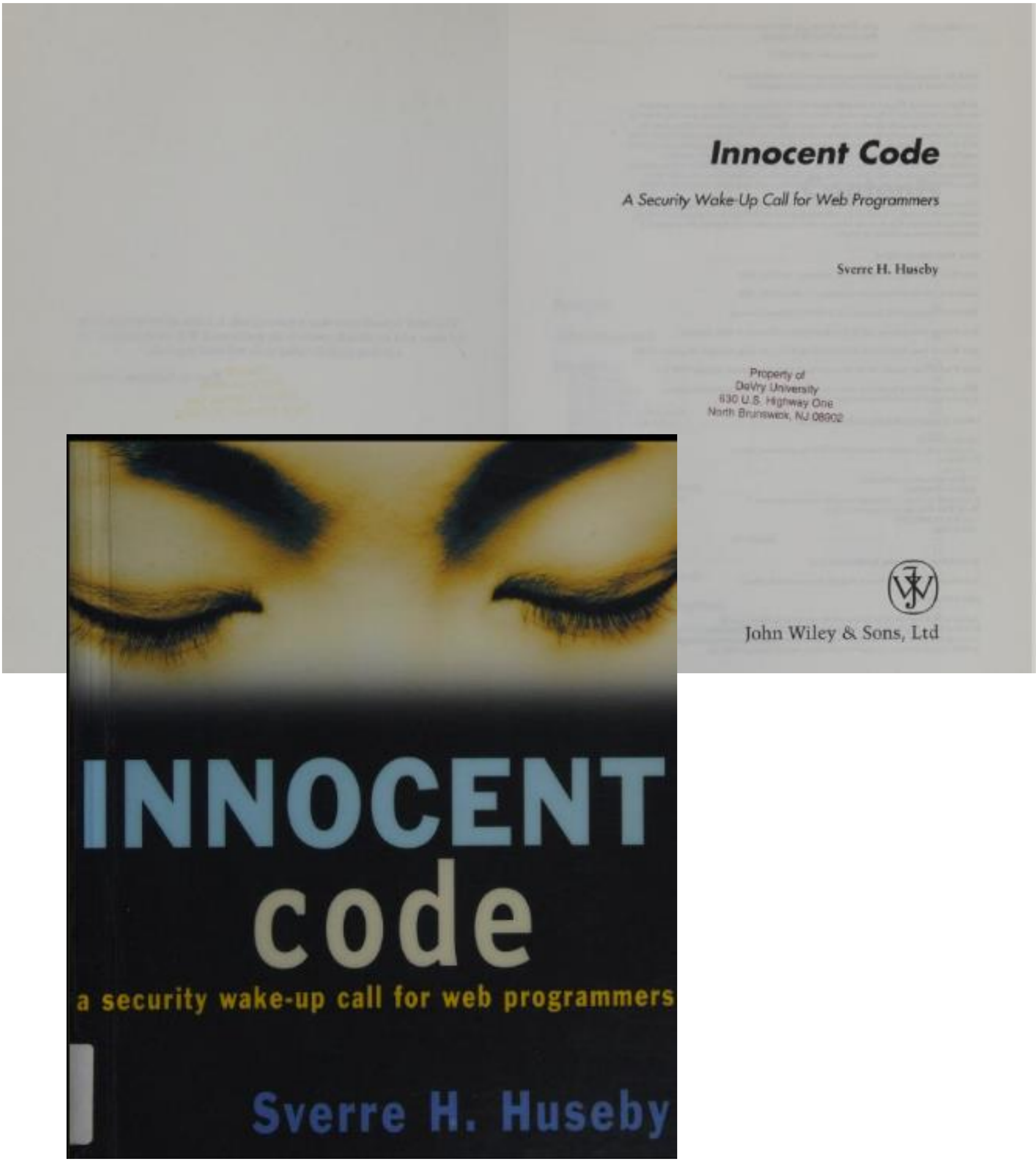
https://betterprogramming.pub/introduction-to-regex-8c18abdd4f70

A Brief Introduction to Regular Expressions

https://tldp.org/LDP/abs/html/x17129.html

WEB application firewall

Sir Tim Berners-Lee had a project called WorldWideWeb



Innocent Code

A Security Wake-Up Call for Web Programmers

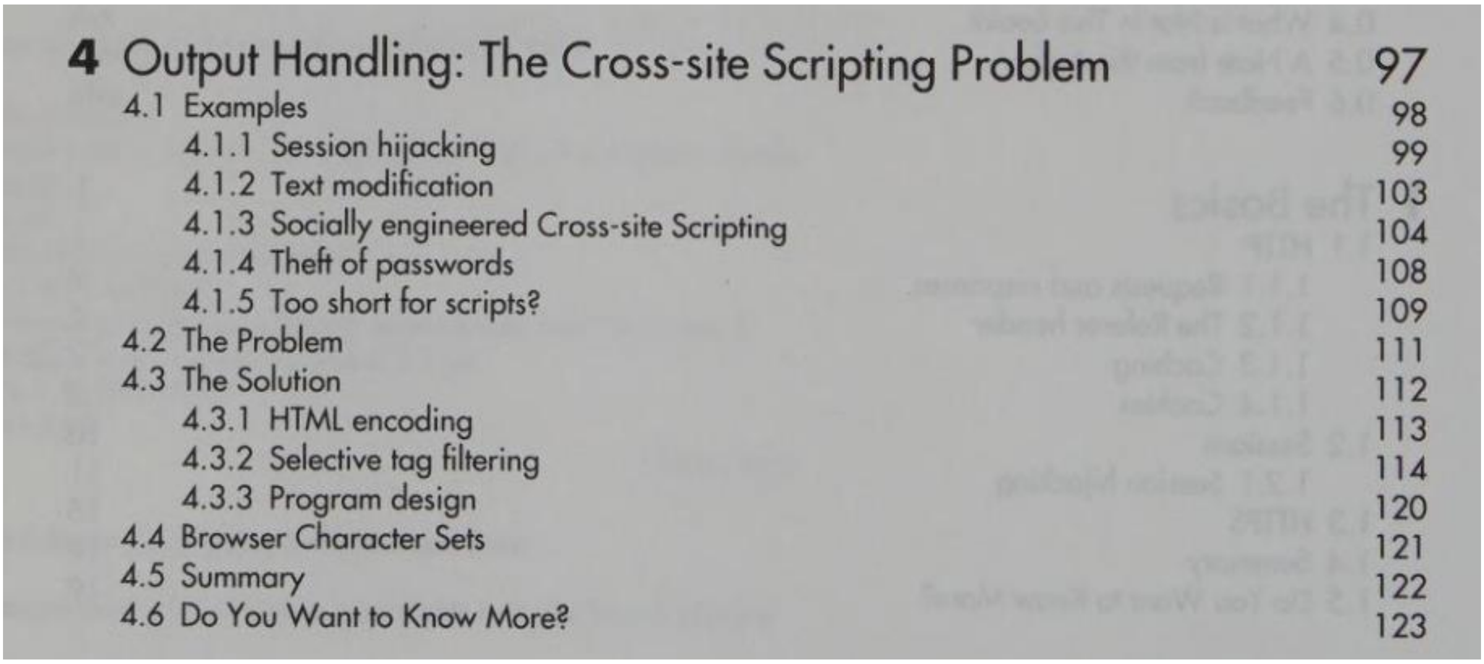
Sverre H. Huseby

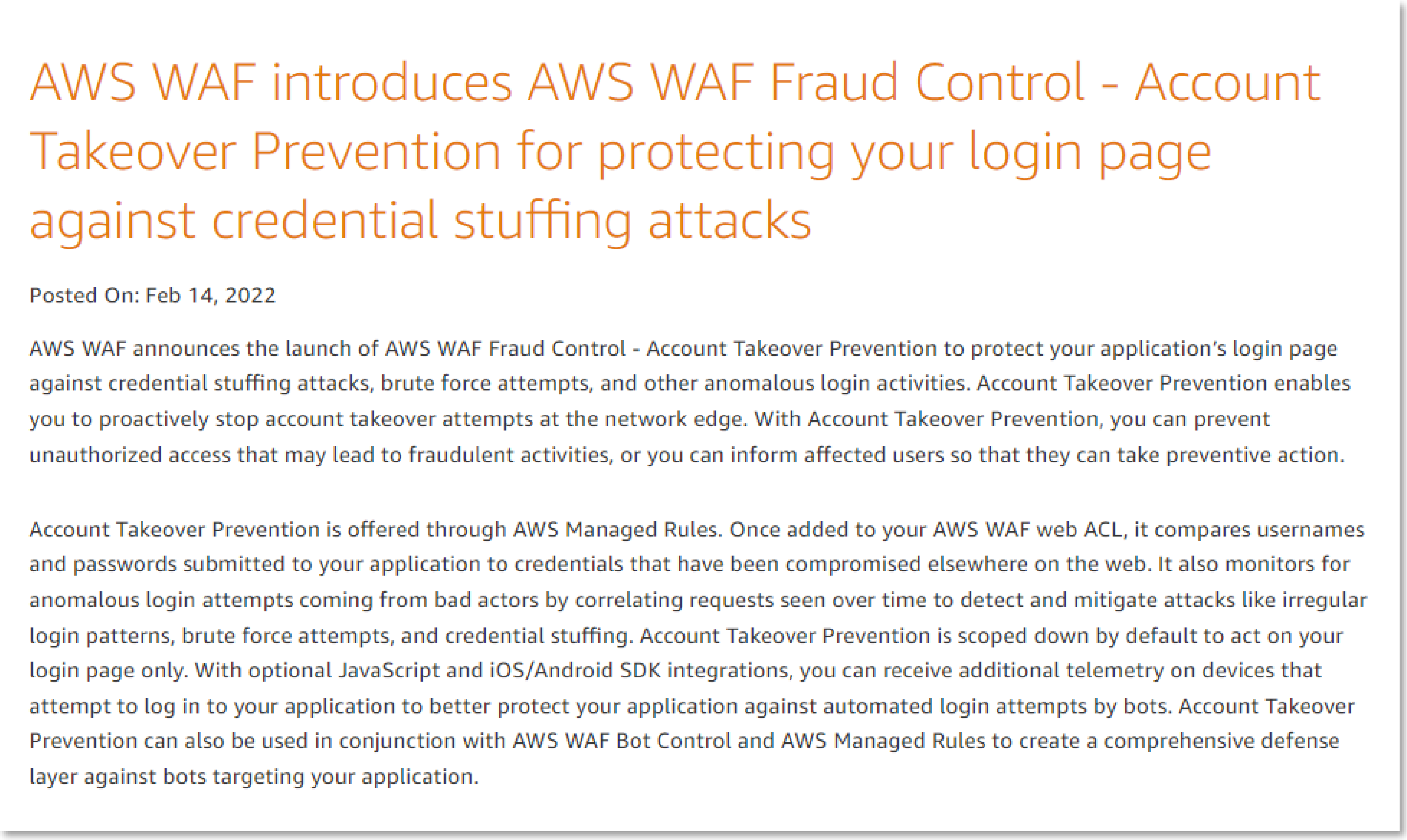
John Wiley & Sons, Ltd

INNOCENT
code

a security wake-up call for web programmers

Sverre H. Huseby





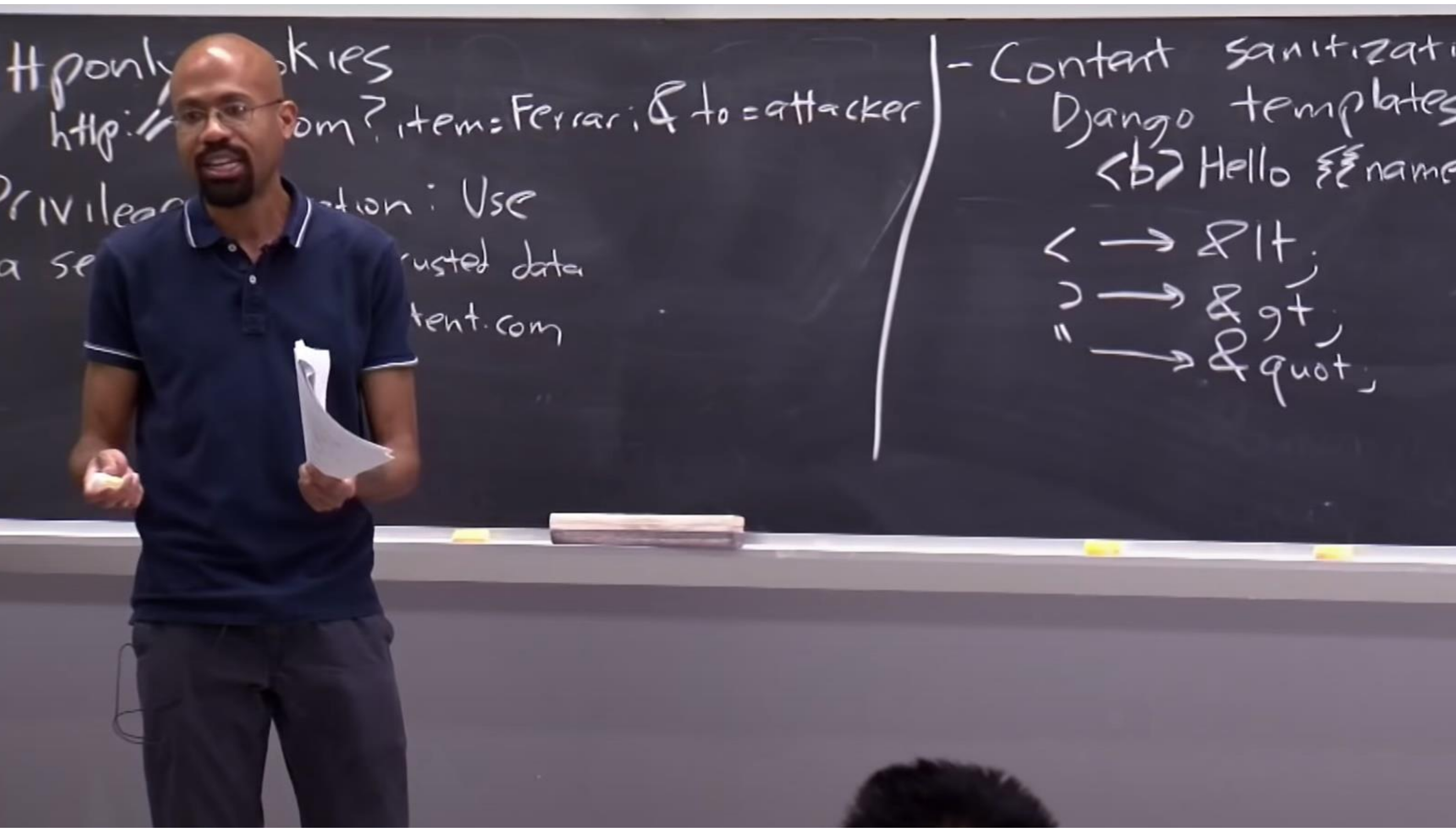
AWS WAF introduces AWS WAF Fraud Control - Account Takeover Prevention for protecting your login page against credential stuffing attacks

Posted On: Feb 14, 2022

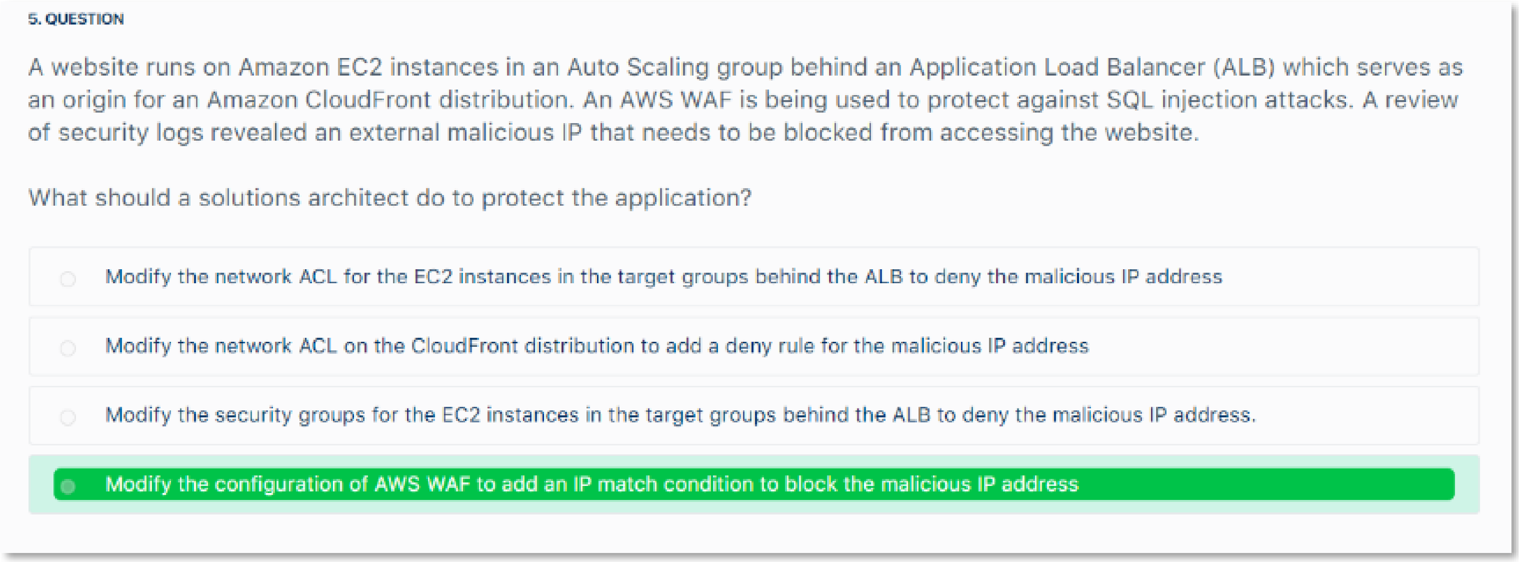
AWS WAF announces the launch of AWS WAF Fraud Control - Account Takeover Prevention to protect your application's login page against credential stuffing attacks, brute force attempts, and other anomalous login activities. Account Takeover Prevention enables you to proactively stop account takeover attempts at the network edge. With Account Takeover Prevention, you can prevent unauthorized access that may lead to fraudulent activities, or you can inform affected users so that they can take preventive action.

Account Takeover Prevention is offered through AWS Managed Rules. Once added to your AWS WAF web ACL, it compares usernames and passwords submitted to your application to credentials that have been compromised elsewhere on the web. It also monitors for anomalous login attempts coming from bad actors by correlating requests seen over time to detect and mitigate attacks like irregular login patterns, brute force attempts, and credential stuffing. Account Takeover Prevention is scoped down by default to act on your login page only. With optional JavaScript and iOS/Android SDK integrations, you can receive additional telemetry on devices that attempt to log in to your application to better protect your application against automated login attempts by bots. Account Takeover Prevention can also be used in conjunction with AWS WAF Bot Control and AWS Managed Rules to create a comprehensive defense layer against bots targeting your application.

Taking it further



James Mickens, an American computer scientist, delivering a lecture on web security at MIT (Massachusetts Institute of Technology)



5. QUESTION

A website runs on Amazon EC2 instances in an Auto Scaling group behind an Application Load Balancer (ALB) which serves as an origin for an Amazon CloudFront distribution. An AWS WAF is being used to protect against SQL injection attacks. A review of security logs revealed an external malicious IP that needs to be blocked from accessing the website.

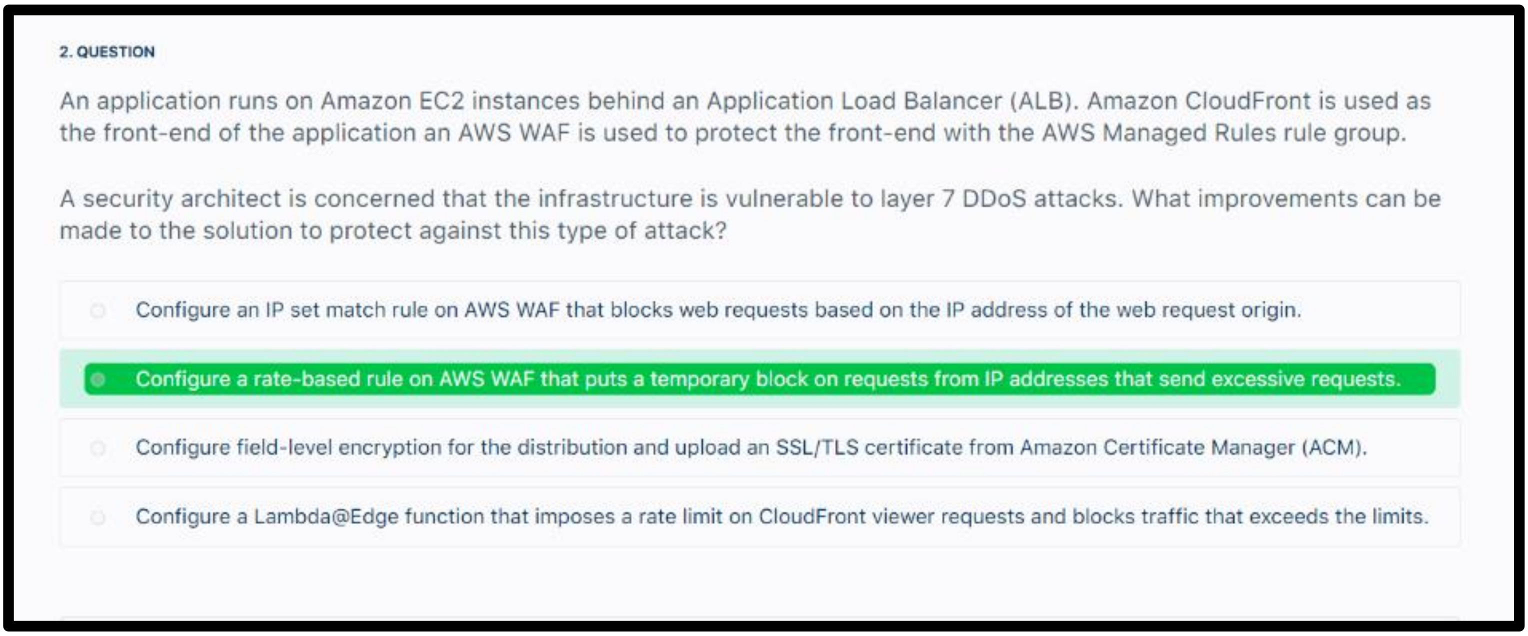
What should a solutions architect do to protect the application?

○ Modify the network ACL for the EC2 instances in the target groups behind the ALB to deny the malicious IP address

○ Modify the network ACL on the CloudFront distribution to add a deny rule for the malicious IP address

○ Modify the security groups for the EC2 instances in the target groups behind the ALB to deny the malicious IP address.

● Modify the configuration of AWS WAF to add an IP match condition to block the malicious IP address



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 an 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?

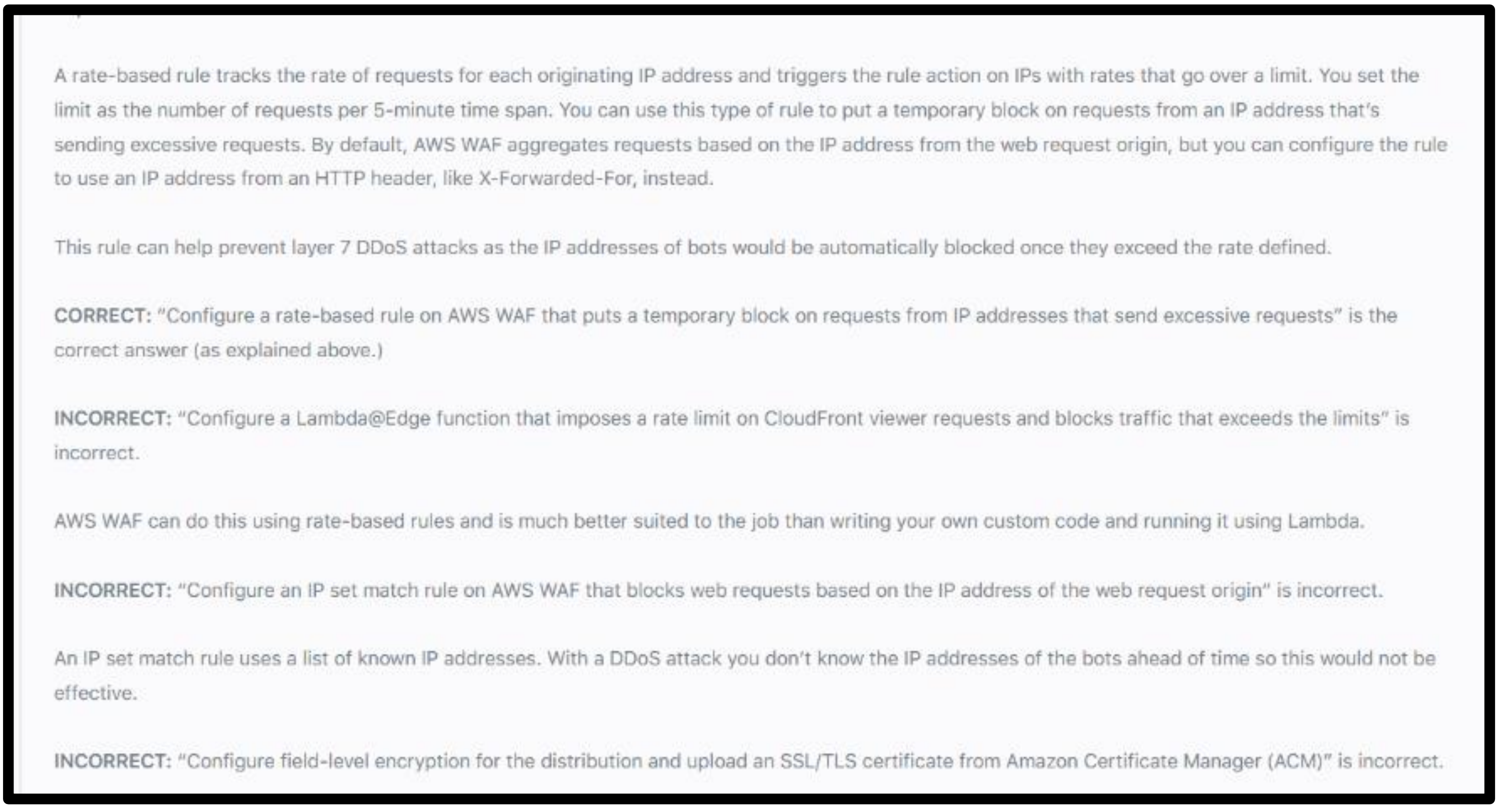
○ Configure an IP set match rule on AWS WAF that blocks web requests based on the IP address of the web request origin.

● 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.

A question from Neal Davis, for people preparing for the Security Speciality exam



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.)

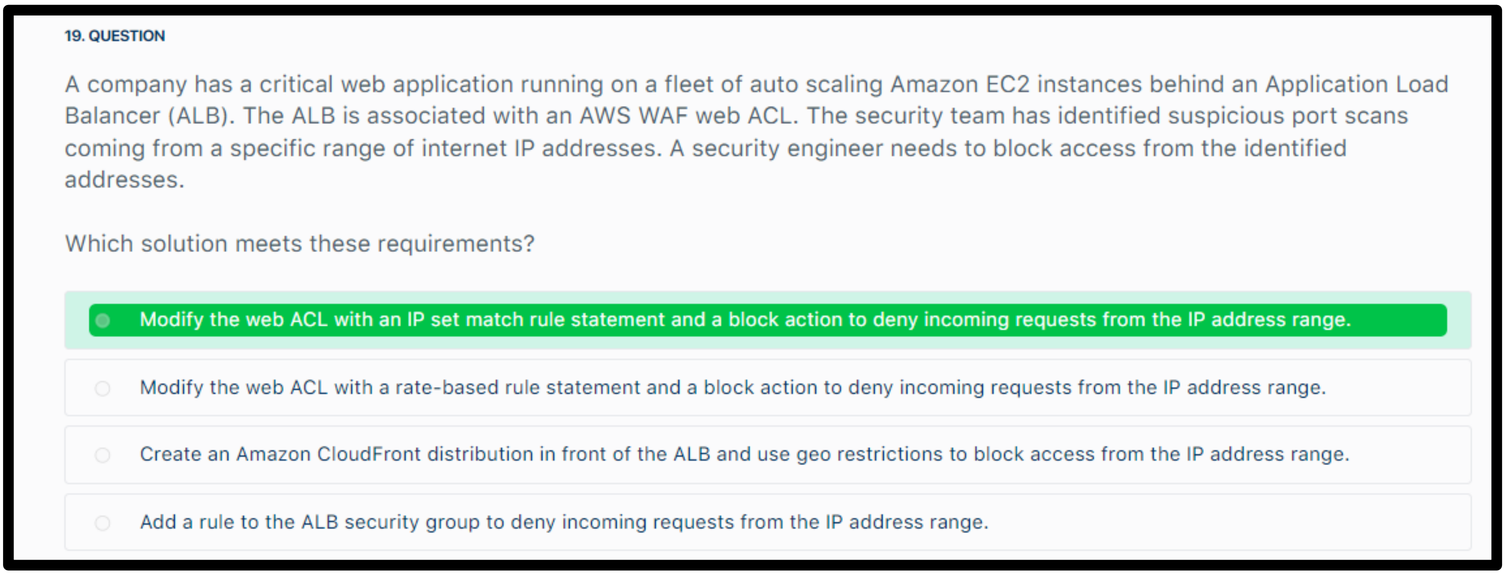
INCORRECT: "Configure a Lambda@Edge function that imposes a rate limit on CloudFront viewer requests and blocks traffic that exceeds the limits" is incorrect.

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.



19. QUESTION

A company has a critical web application running on a fleet of auto scaling Amazon EC2 instances behind an Application Load Balancer (ALB). The ALB is associated with an AWS WAF web ACL. The security team has identified suspicious port scans coming from a specific range of internet IP addresses. A security engineer needs to block access from the identified addresses.

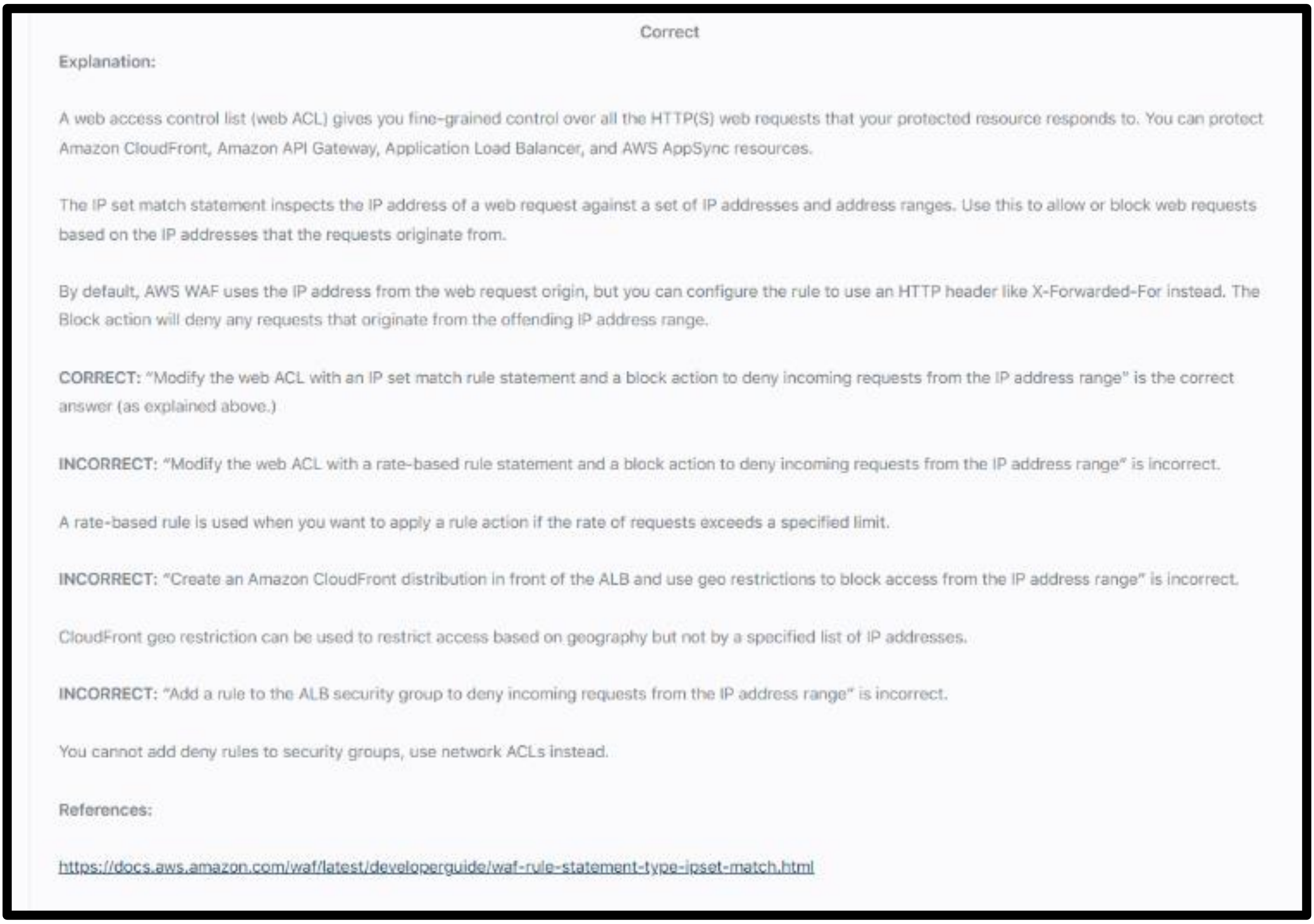
Which solution meets these requirements?

● Modify the web ACL with an IP set match rule statement and a block action to deny incoming requests from the IP address range.

○ Modify the web ACL with a rate-based rule statement and a block action to deny incoming requests from the IP address range.

○ Create an Amazon CloudFront distribution in front of the ALB and use geo restrictions to block access from the IP address range.

○ Add a rule to the ALB security group to deny incoming requests from the IP address range.



Correct

Explanation:

A web access control list (web ACL) gives you fine-grained control over all the HTTP(S) web requests that your protected resource responds to. You can protect Amazon CloudFront, Amazon API Gateway, Application Load Balancer, and AWS AppSync resources.

The IP set match statement inspects the IP address of a web request against a set of IP addresses and address ranges. Use this to allow or block web requests based on the IP addresses that the requests originate from.

By default, AWS WAF uses the IP address from the web request origin, but you can configure the rule to use an HTTP header like X-Forwarded-For instead. The Block action will deny any requests that originate from the offending IP address range.

CORRECT: "Modify the web ACL with an IP set match rule statement and a block action to deny incoming requests from the IP address range" is the correct answer (as explained above.)

INCORRECT: "Modify the web ACL with a rate-based rule statement and a block action to deny incoming requests from the IP address range" is incorrect.

A rate-based rule is used when you want to apply a rule action if the rate of requests exceeds a specified limit.

INCORRECT: "Create an Amazon CloudFront distribution in front of the ALB and use geo restrictions to block access from the IP address range" is incorrect.

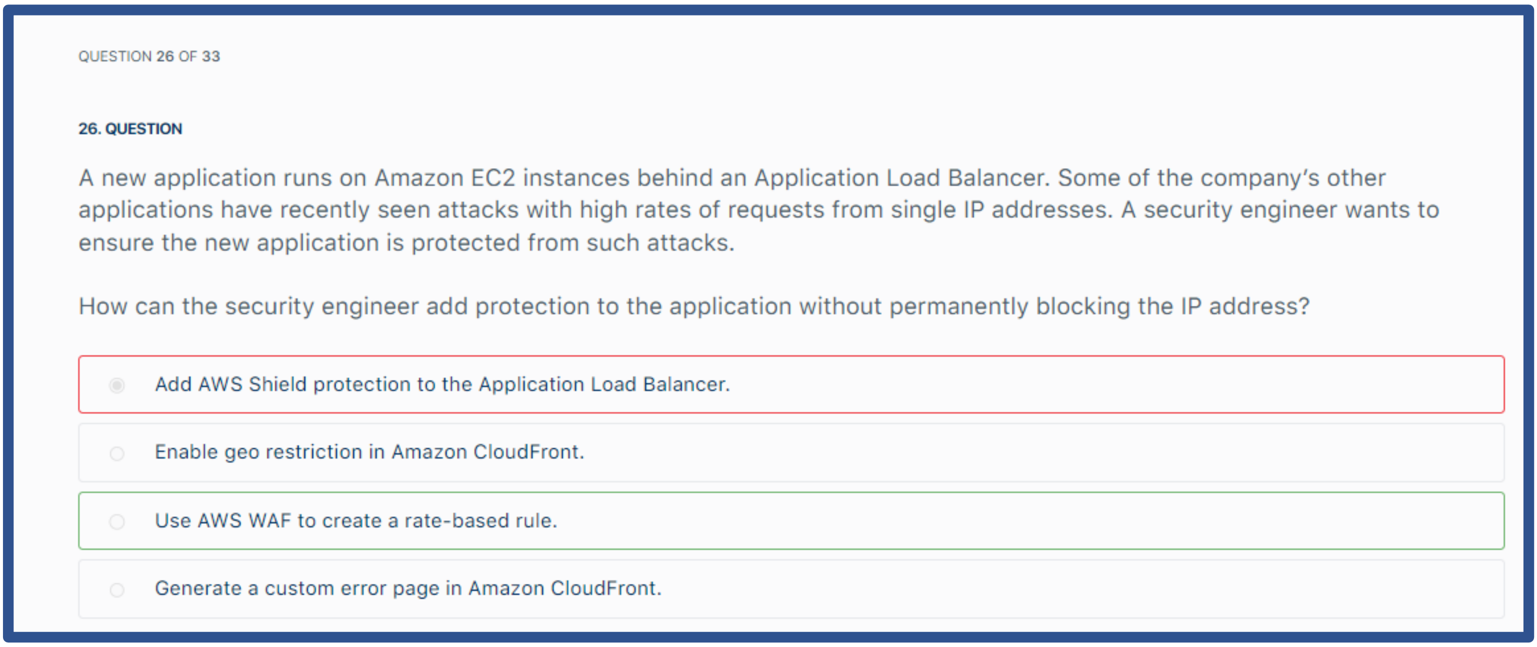
CloudFront geo restriction can be used to restrict access based on geography but not by a specified list of IP addresses.

INCORRECT: "Add a rule to the ALB security group to deny incoming requests from the IP address range" is incorrect.

You cannot add deny rules to security groups, use network ACLs instead.

References:

https://docs.aws.amazon.com/waf/latest/developerguide/waf-rule-statement-type-ipset-match.html



QUESTION 26 OF 33

26. QUESTION

A new application runs on Amazon EC2 instances behind an Application Load Balancer. Some of the company's other applications have recently seen attacks with high rates of requests from single IP addresses. A security engineer wants to ensure the new application is protected from such attacks.

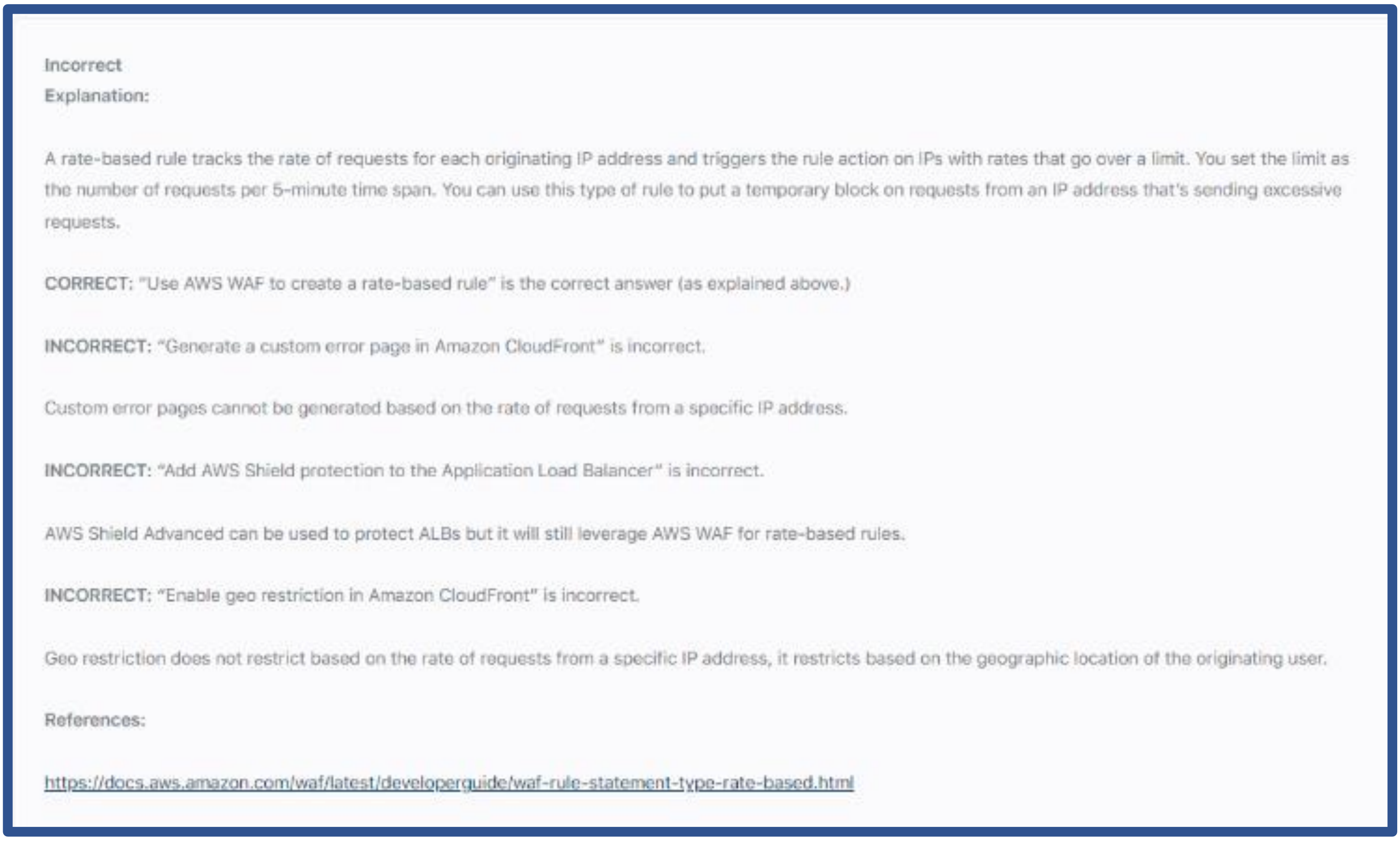
How can the security engineer add protection to the application without permanently blocking the IP address?

● Add AWS Shield protection to the Application Load Balancer.

○ Enable geo restriction in Amazon CloudFront.

○ Use AWS WAF to create a rate-based rule.

○ Generate a custom error page in Amazon CloudFront.



Incorrect
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.

CORRECT: "Use AWS WAF to create a rate-based rule" is the correct answer (as explained above.)

INCORRECT: "Generate a custom error page in Amazon CloudFront" is incorrect.

Custom error pages cannot be generated based on the rate of requests from a specific IP address.

INCORRECT: "Add AWS Shield protection to the Application Load Balancer" is incorrect.

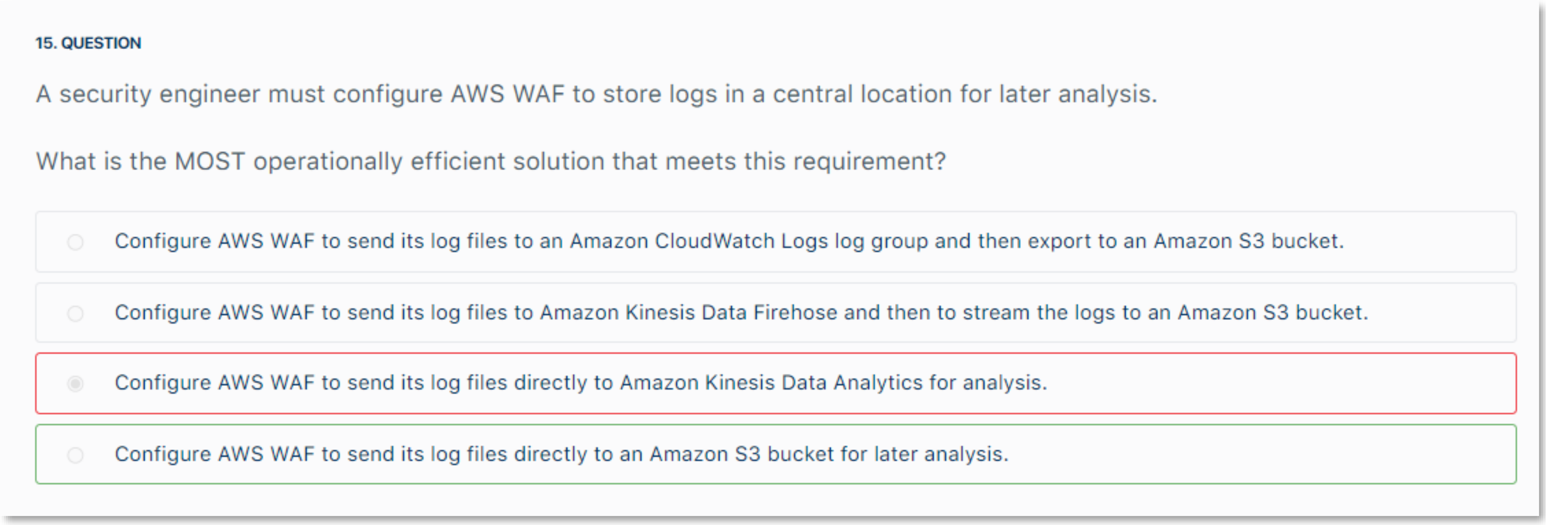
AWS Shield Advanced can be used to protect ALBs but it will still leverage AWS WAF for rate-based rules.

INCORRECT: "Enable geo restriction in Amazon CloudFront" is incorrect.

Geo restriction does not restrict based on the rate of requests from a specific IP address, it restricts based on the geographic location of the originating user.

References:

https://docs.aws.amazon.com/waf/latest/developerguide/waf-rule-statement-type-rate-based.html



15. QUESTION

A security engineer must configure AWS WAF to store logs in a central location for later analysis.

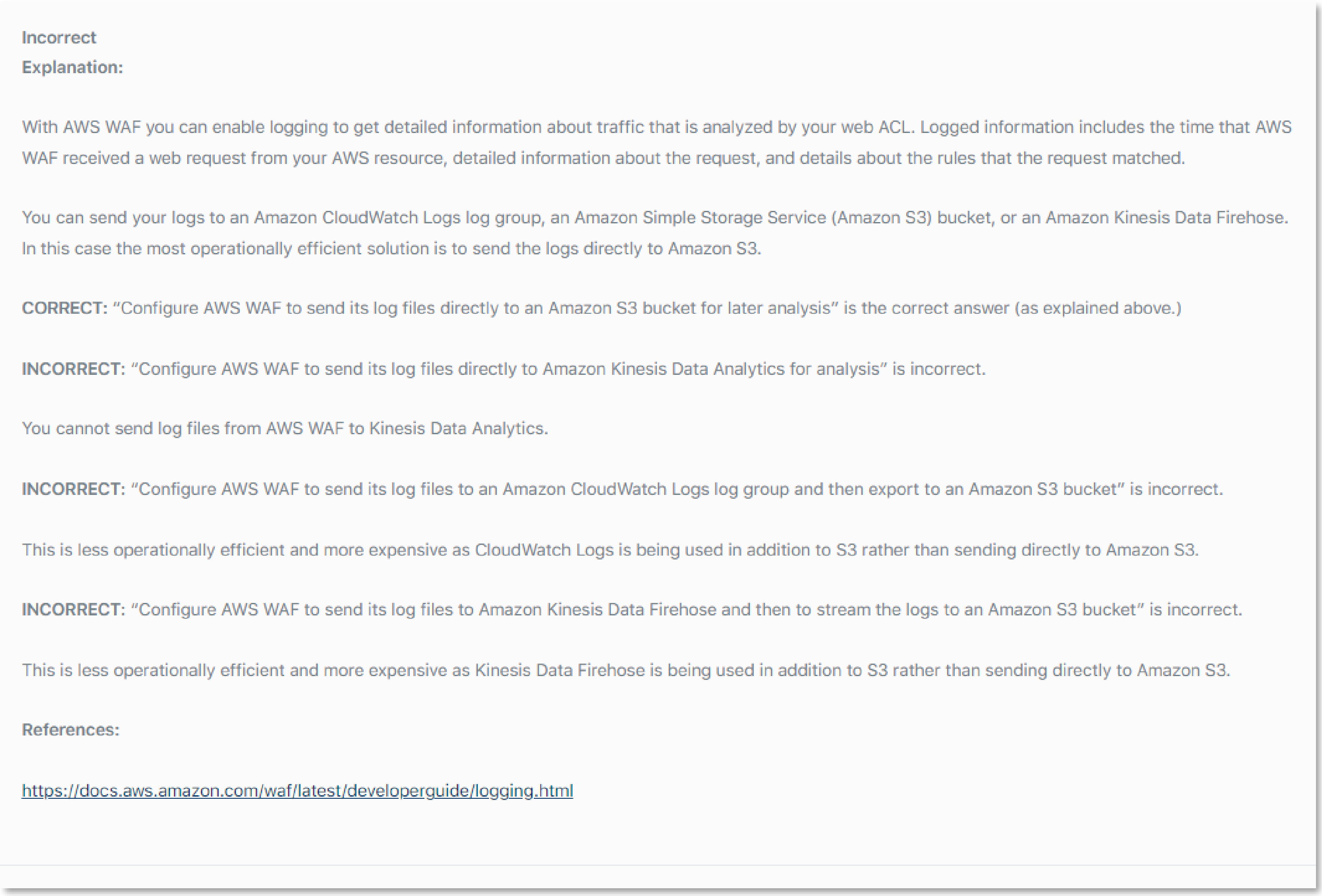
What is the MOST operationally efficient solution that meets this requirement?

○ Configure AWS WAF to send its log files to an Amazon CloudWatch Logs log group and then export to an Amazon S3 bucket.

○ Configure AWS WAF to send its log files to Amazon Kinesis Data Firehose and then to stream the logs to an Amazon S3 bucket.

● Configure AWS WAF to send its log files directly to Amazon Kinesis Data Analytics for analysis.

○ Configure AWS WAF to send its log files directly to an Amazon S3 bucket for later analysis.



Incorrect

Explanation:

With AWS WAF you can enable logging to get detailed information about traffic that is analyzed by your web ACL. Logged information includes the time that AWS WAF received a web request from your AWS resource, detailed information about the request, and details about the rules that the request matched.

You can send your logs to an Amazon CloudWatch Logs log group, an Amazon Simple Storage Service (Amazon S3) bucket, or an Amazon Kinesis Data Firehose. In this case the most operationally efficient solution is to send the logs directly to Amazon S3.

CORRECT: “Configure AWS WAF to send its log files directly to an Amazon S3 bucket for later analysis” is the correct answer (as explained above.)

INCORRECT: “Configure AWS WAF to send its log files directly to Amazon Kinesis Data Analytics for analysis” is incorrect.

You cannot send log files from AWS WAF to Kinesis Data Analytics.

INCORRECT: “Configure AWS WAF to send its log files to an Amazon CloudWatch Logs log group and then export to an Amazon S3 bucket” is incorrect.

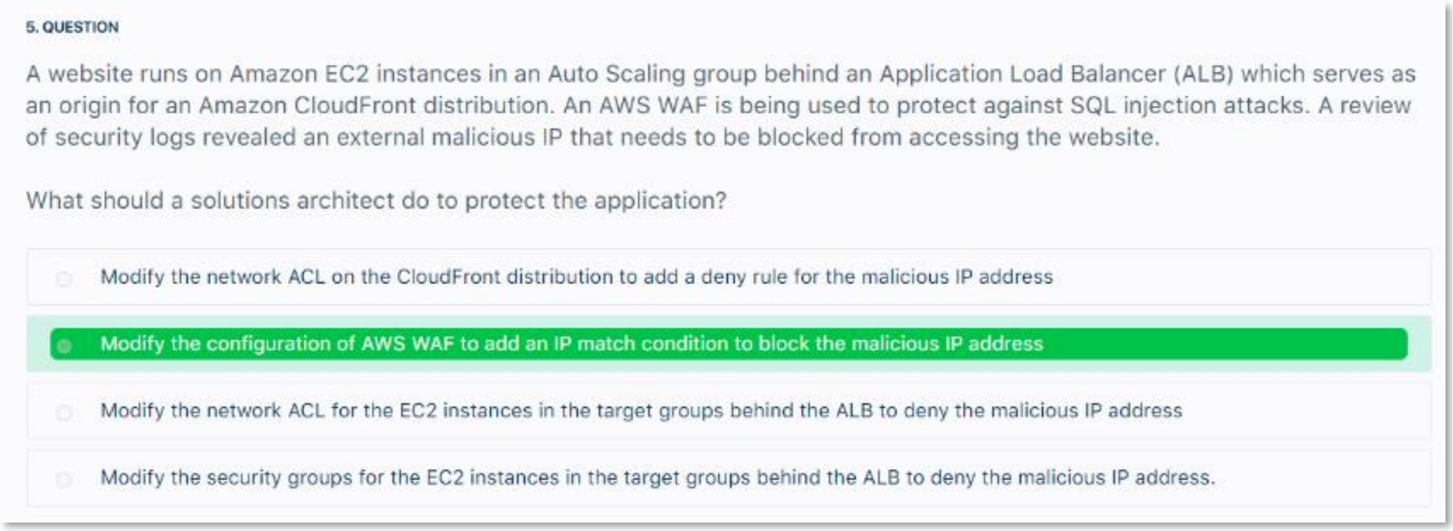
This is less operationally efficient and more expensive as CloudWatch Logs is being used in addition to S3 rather than sending directly to Amazon S3.

INCORRECT: “Configure AWS WAF to send its log files to Amazon Kinesis Data Firehose and then to stream the logs to an Amazon S3 bucket” is incorrect.

This is less operationally efficient and more expensive as Kinesis Data Firehose is being used in addition to S3 rather than sending directly to Amazon S3.

References:

https://docs.aws.amazon.com/waf/latest/developerguide/logging.html



5. QUESTION

A website runs on Amazon EC2 instances in an Auto Scaling group behind an Application Load Balancer (ALB) which serves as an origin for an Amazon CloudFront distribution. An AWS WAF is being used to protect against SQL injection attacks. A review of security logs revealed an external malicious IP that needs to be blocked from accessing the website.

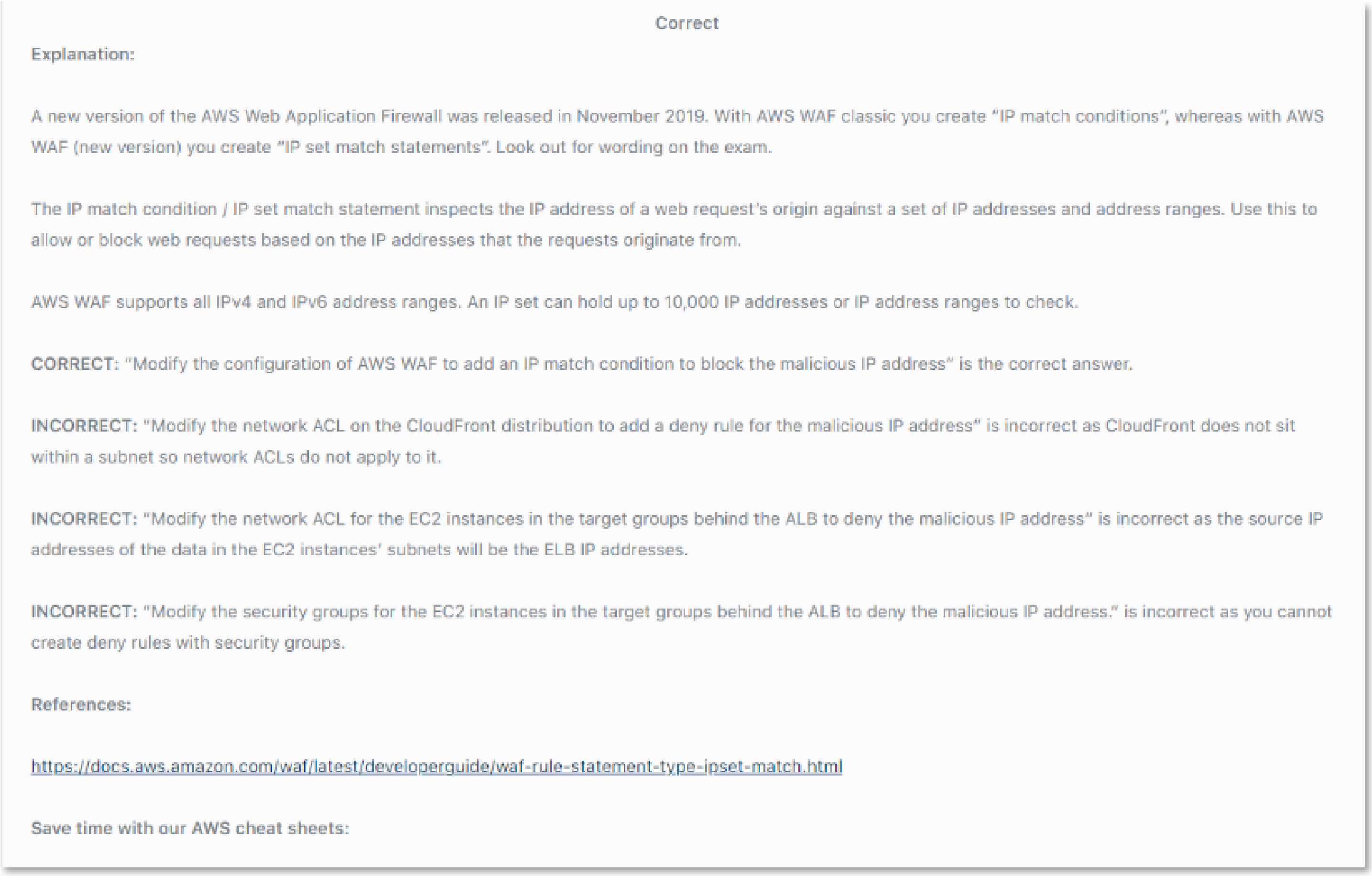
What should a solutions architect do to protect the application?

○ Modify the network ACL on the CloudFront distribution to add a deny rule for the malicious IP address

● Modify the configuration of AWS WAF to add an IP match condition to block the malicious IP address

○ Modify the network ACL for the EC2 instances in the target groups behind the ALB to deny the malicious IP address

○ Modify the security groups for the EC2 instances in the target groups behind the ALB to deny the malicious IP address.



Correct

Explanation:

A new version of the AWS Web Application Firewall was released in November 2019. With AWS WAF classic you create "IP match conditions", whereas with AWS WAF (new version) you create "IP set match statements". Look out for wording on the exam.

The IP match condition / IP set match statement inspects the IP address of a web request's origin against a set of IP addresses and address ranges. Use this to allow or block web requests based on the IP addresses that the requests originate from.

AWS WAF supports all IPv4 and IPv6 address ranges. An IP set can hold up to 10,000 IP addresses or IP address ranges to check.

CORRECT: "Modify the configuration of AWS WAF to add an IP match condition to block the malicious IP address" is the correct answer.

INCORRECT: "Modify the network ACL on the CloudFront distribution to add a deny rule for the malicious IP address" is incorrect as CloudFront does not sit within a subnet so network ACLs do not apply to it.

INCORRECT: "Modify the network ACL for the EC2 instances in the target groups behind the ALB to deny the malicious IP address" is incorrect as the source IP addresses of the data in the EC2 instances' subnets will be the ELB IP addresses.

INCORRECT: "Modify the security groups for the EC2 instances in the target groups behind the ALB to deny the malicious IP address." is incorrect as you cannot create deny rules with security groups.

References:

https://docs.aws.amazon.com/waf/latest/developerguide/waf-rule-statement-type-ipset-match.html

Save time with our AWS cheat sheets:

Bibliography

I. Official

[Smith 2016]

Smith, Alex (2016). How to Prevent Hotlinking by Using AWS WAF, Amazon CloudFront, and Referer Checking. *AWS Security Blog*. April 21st 2016. Available at: https://aws.amazon.com/blogs/security/how-to-prevent-hotlinking-by-using-aws-waf-amazon-cloudfront-and-referer-checking/

[Brown 2015]

Brown, Travis (2015). How to Automatically Update your Security Groups for Amazon CloudFront and AWS WAF by Using AWS Lambda. *AWS Security Blog*. Dec 15th 2015. Available at: https://aws.amazon.com/blogs/security/how-to-automatically-update-your-security-groups-for-amazon-cloudfront-and-aws-waf-by-using-aws-lambda/

[Lovan 2021]

Lovan, Artem and Jesse Lepich (2021). The three most important AWS WAF rate-based rules. AWS Security Blog. July 22nd 2021. Available at: https://aws.amazon.com/blogs/security/three-most-important-aws-waf-rate-based-rules/

[Bolodeoku 2021]

Bolodeoku, Fola and Davidson Junior and Mário Pinho (2021). Automatically update AWS WAF IP sets with AWS IP ranges. *AWS Security Blog*. July 8th 2021. Available at: https://aws.amazon.com/blogs/security/automatically-update-aws-waf-ip-sets-with-aws-ip-ranges/

[Phatak 2021]

Phatak, Kaustubh and Chen, EJ (2021). Customize requests and responses with AWS WAF. Available at: https://aws.amazon.com/blogs/security/customize-requests-and-responses-with-aws-waf/

[Ramesh 2020]

Ramesh, Umesh and Mahek Pavagadhi (2020). Centrally manage AWS WAF (API v2) and AWS Managed Rules at scale with Firewall Manager. *AWS Security Blog*. Nov 17th 2020. Available at: https://aws.amazon.com/blogs/security/centrally-manage-aws-waf-api-v2-and-aws-managed-rules-at-scale-with-firewall-manager/

[Menon 2020]

Menon, Vijay and Gaurav Sahi (2020). Field Notes: How to Identify and Block Fake Crawler Bots Using AWS WAF. AWS Architecture Blog. Nov 4th 2020. Available at: https://aws.amazon.com/blogs/architecture/field-notes-how-to-identify-and-block-fake-crawler-bots-using-aws-waf/

[Kumar 2020]

Kumar, Satheesh and Ramanan Kannan (2020). Automate AWS Firewall Manager onboarding using AWS Centralized WAF and VPC Security Group Management solution. *AWS Security Blog*. Oct 15th 2020. Available at: https://aws.amazon.com/blogs/security/automate-aws-firewall-manager-onboarding-using-aws-centralized-waf-and-vpc-security-group-management-solution/

[Cerini 2020]

Cerini, Adam (2020). Automatically updating AWS WAF Rule in real time using Amazon EventBridge. *AWS Security Blog*. Sept 21st 2020. Available at: https://aws.amazon.com/blogs/security/automatically-updating-aws-waf-rule-in-real-time-using-amazon-eventbridge/

[Swart 2020]

Defense in depth using AWS Managed Rules for AWS WAF (part 1). *AWS Security Blog*. Sept 2nd 2020. Available at: https://aws.amazon.com/blogs/security/defense-in-depth-using-aws-managed-rules-for-aws-waf-part-1/

[Swart 2020]

Swart, Daniel (2020). Deploying defense in depth using AWS Managed Rules for AWS WAF (part 2). *AWS Security Blog*. Sept 2nd 2020. Available at: https://aws.amazon.com/blogs/security/deploying-defense-in-depth-using-aws-managed-rules-for-aws-waf-part-2/

[Ramesh 2020]

Ramesh, Umesh and Kevin Lee. Migrating your rules from AWS WAF Classic to the new AWS WAF. *AWS Security Blog*. Aug 4th 2020. Available at: https://aws.amazon.com/blogs/security/migrating-rules-from-aws-waf-classic-to-new-aws-waf/

[Stachlewski 2020]

Deploy a dashboard for AWS WAF with minimal effort. *AWS Security Blog*. July 8th 2020. Available at: https://aws.amazon.com/blogs/security/deploy-dashboard-for-aws-waf-minimal-effort/

[George 2020]

George, Mike (2020). Enable automatic logging of web ACLs by using AWS Config. AWS Security Blog. Apr 10th 2020. Available at: https://aws.amazon.com/blogs/security/enable-automatic-logging-of-web-acls-by-using-aws-config/

[Aiken 2019]

Aiken, David (2019). Creating web access control lists using Fortinet Managed Rules set from AWS Marketplace. *AWS Marketplace* [blog]. August 2nd 2019. Available at: https://aws.amazon.com/blogs/awsmarketplace/creating-web-access-control-lists-using-fortinet-managed-rules-set-from-aws-marketplace/

[Tran 2019]

Tran, Tino (2019). Trimming AWS WAF logs with Amazon Kinesis Firehose transformations. *AWS Security Blog*. Apr 9th 2019. Available at: https://aws.amazon.com/blogs/security/trimming-aws-waf-logs-with-amazon-kinesis-firehose-transformations/

[Varuni 2019]

Varuni, Rajat Ravinder and Heitor Vital (2019). How to use AWS WAF to filter incoming traffic from embargoed countries. *AWS Security Blog*. Jan 9th 2019. Available at: https://aws.amazon.com/blogs/security/how-to-use-aws-waf-to-filter-incoming-traffic-from-embargoed-countries/

[Ramesh 2018]

Ramash, Umesh. Using AWS Firewall Manager and WAF to protect your web applications with master rules and application-specific rules. *AWS Security Blog*. Nov 12th 2018. Available at: https://aws.amazon.com/blogs/security/using-aws-firewall-manager-and-waf-to-protect-your-web-applications-with-master-rules-and-application-specific-rules/

[Adamski 2018]

Adamski, Tom (2018). How to analyze AWS WAF logs using Amazon Elasticsearch Service. *AWS Security Blog*. Oct 30th 2018. Available at: https://aws.amazon.com/blogs/security/how-to-analyze-aws-waf-logs-using-amazon-elasticsearch-service/

[Worrell 2018]

How to use Amazon GuardDuty and AWS Web Application Firewall to automatically block suspicious hosts. *AWS Security Blog*. Aug 3rd 2018. Available at: https://aws.amazon.com/blogs/security/how-to-use-amazon-guardduty-and-aws-web-application-firewall-to-automatically-block-suspicious-hosts/

[Barr 2017a]

Barr, Jeff (2017). Protect Web Sites and Services Using Rate-Based Rules for AWS WAF. *AWS News Blog*. June 21st 2017. Available at: https://aws.amazon.com/blogs/aws/protect-web-sites-services-using-rate-based-rules-for-aws-waf/

[Barr 2017b]

Barr, Jeff (2017). AWS Web Application Firewall (WAF) for Application Load Balancers. *AWS News Blog*. Jan 25th 2017. Available at: https://aws.amazon.com/blogs/aws/aws-web-application-firewall-waf-for-application-load-balancers/

https://aws.amazon.com/blogs/security/how-to-customize-behavior-of-aws-managed-rules-for-aws-waf/

II. Unofficial

Firewall factory. Available at: https://github.com/globaldatanet/aws-firewall-factory?ck_subscriber_id=1560524742

[Guru 2022]

A Cloud Guru. YouTube Channel: A Security Guru. Implement AWS Web Application Firewall. Available at: https://www.youtube.com/watch?v=lvj5XtljV3o&ab_channel=ASecurityGuru

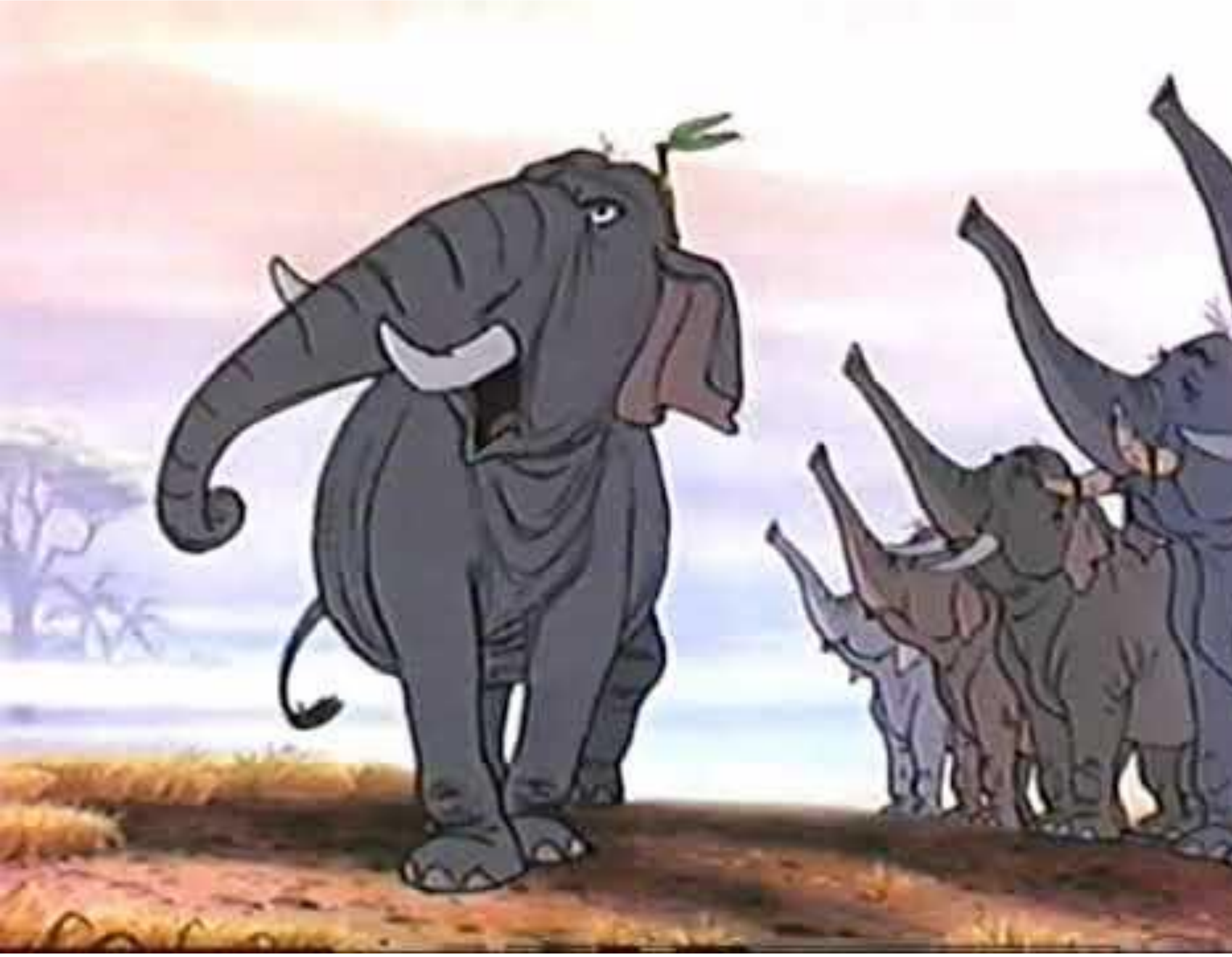
III. Critical

IV. General

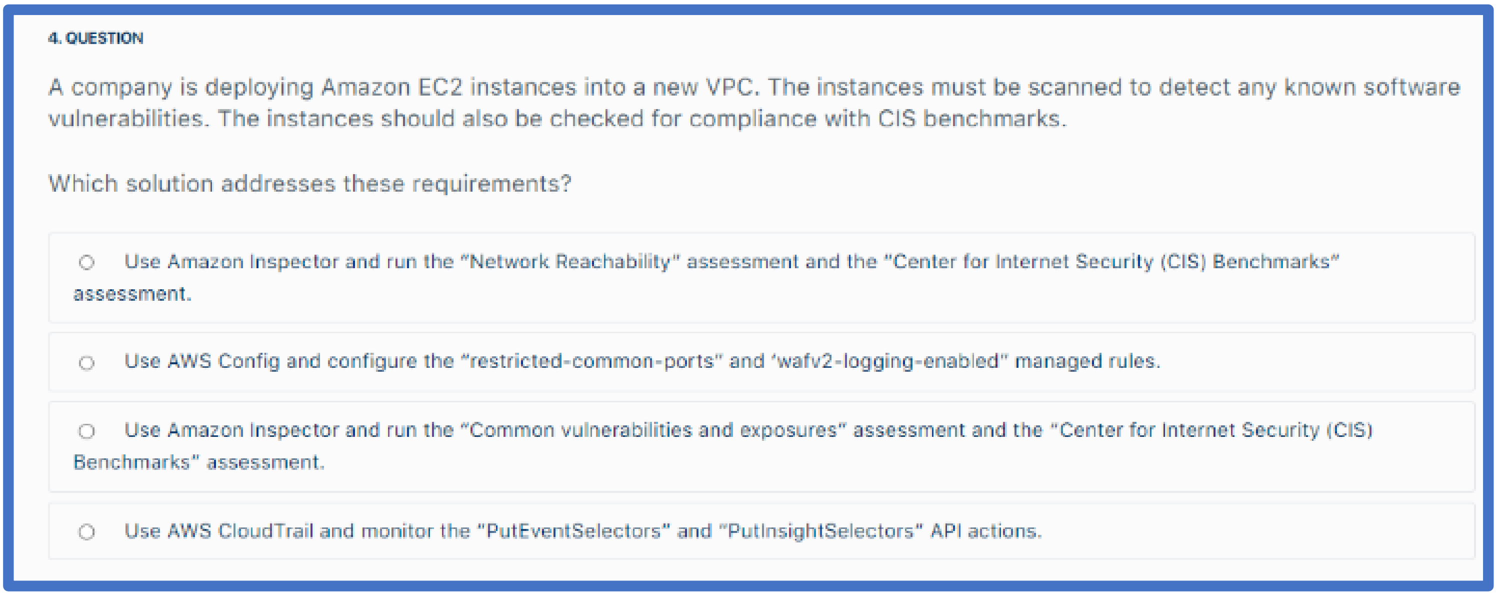
[Mickens 2017]

Mickens, James (2017). Securing Web Applications. [Lecture] YouTube Channel: MIT OpenCourseWare. Available at: https://www.youtube.com/watch?v=WlmKwIe9z1Q&ab_channel=MITOpenCourseWare

Inspector



INSPECTOR



4. QUESTION

A company is deploying Amazon EC2 instances into a new VPC. The instances must be scanned to detect any known software vulnerabilities. The instances should also be checked for compliance with CIS benchmarks.

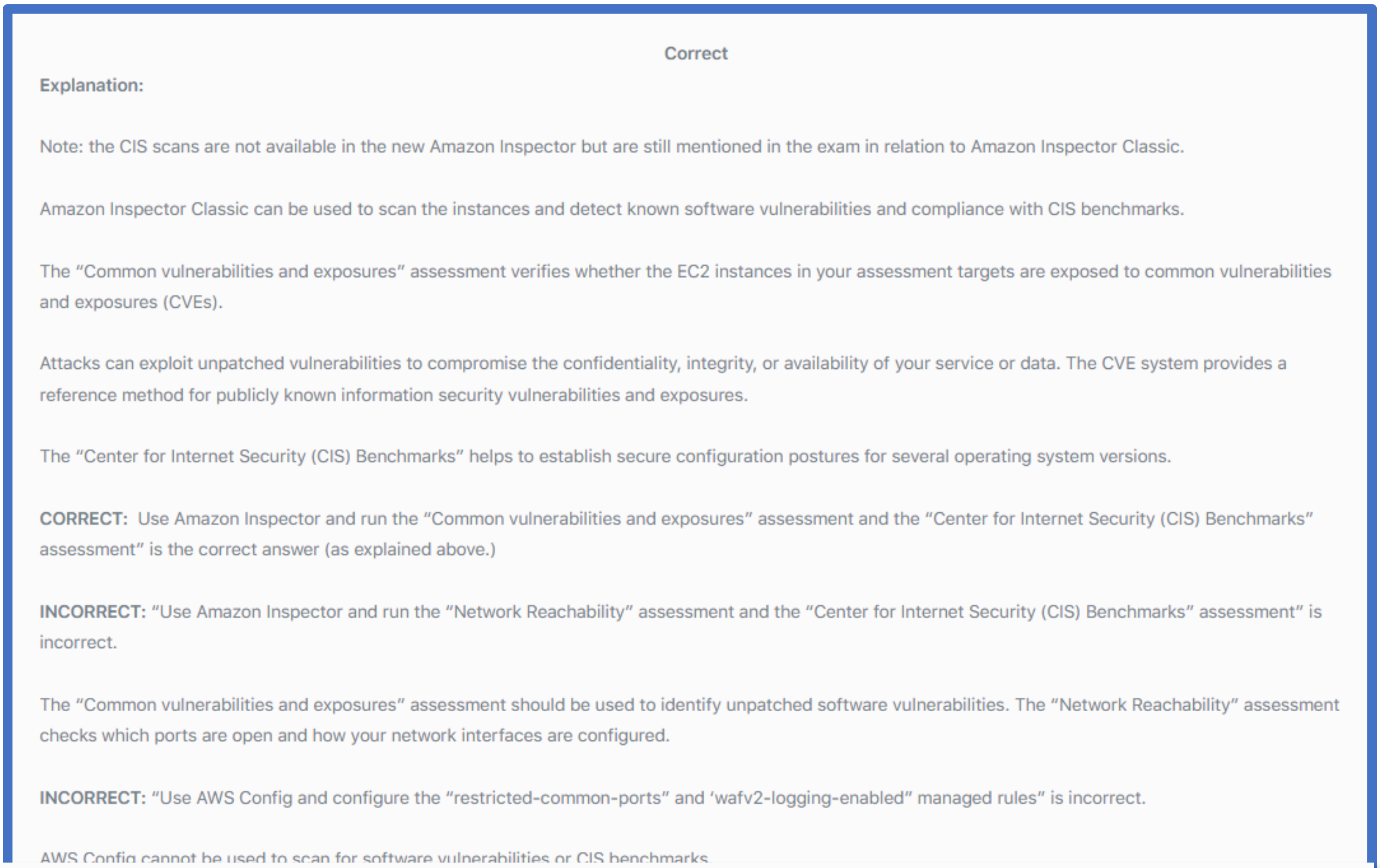
Which solution addresses these requirements?

○ Use Amazon Inspector and run the "Network Reachability" assessment and the "Center for Internet Security (CIS) Benchmarks" assessment.

○ Use AWS Config and configure the "restricted-common-ports" and 'wafv2-logging-enabled" managed rules.

○ Use Amazon Inspector and run the "Common vulnerabilities and exposures" assessment and the "Center for Internet Security (CIS) Benchmarks" assessment.

○ Use AWS CloudTrail and monitor the "PutEventSelectors" and "PutInsightSelectors" API actions.



Correct

Explanation:

Note: the CIS scans are not available in the new Amazon Inspector but are still mentioned in the exam in relation to Amazon Inspector Classic.

Amazon Inspector Classic can be used to scan the instances and detect known software vulnerabilities and compliance with CIS benchmarks.

The "Common vulnerabilities and exposures" assessment verifies whether the EC2 instances in your assessment targets are exposed to common vulnerabilities and exposures (CVEs).

Attacks can exploit unpatched vulnerabilities to compromise the confidentiality, integrity, or availability of your service or data. The CVE system provides a reference method for publicly known information security vulnerabilities and exposures.

The "Center for Internet Security (CIS) Benchmarks" helps to establish secure configuration postures for several operating system versions.

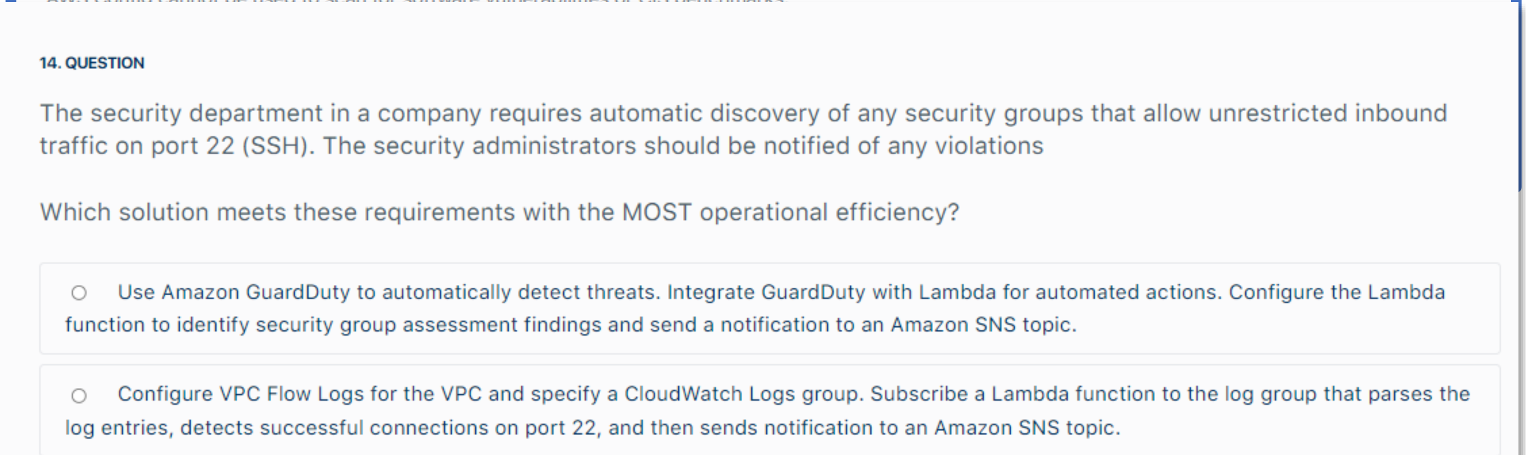
CORRECT: Use Amazon Inspector and run the "Common vulnerabilities and exposures" assessment and the "Center for Internet Security (CIS) Benchmarks" assessment" is the correct answer (as explained above.)

INCORRECT: "Use Amazon Inspector and run the "Network Reachability" assessment and the "Center for Internet Security (CIS) Benchmarks" assessment" is incorrect.

The "Common vulnerabilities and exposures" assessment should be used to identify unpatched software vulnerabilities. The "Network Reachability" assessment checks which ports are open and how your network interfaces are configured.

INCORRECT: "Use AWS Config and configure the "restricted-common-ports" and 'wafv2-logging-enabled" managed rules" is incorrect.

AWS Config cannot be used to scan for software vulnerabilities or CIS benchmarks



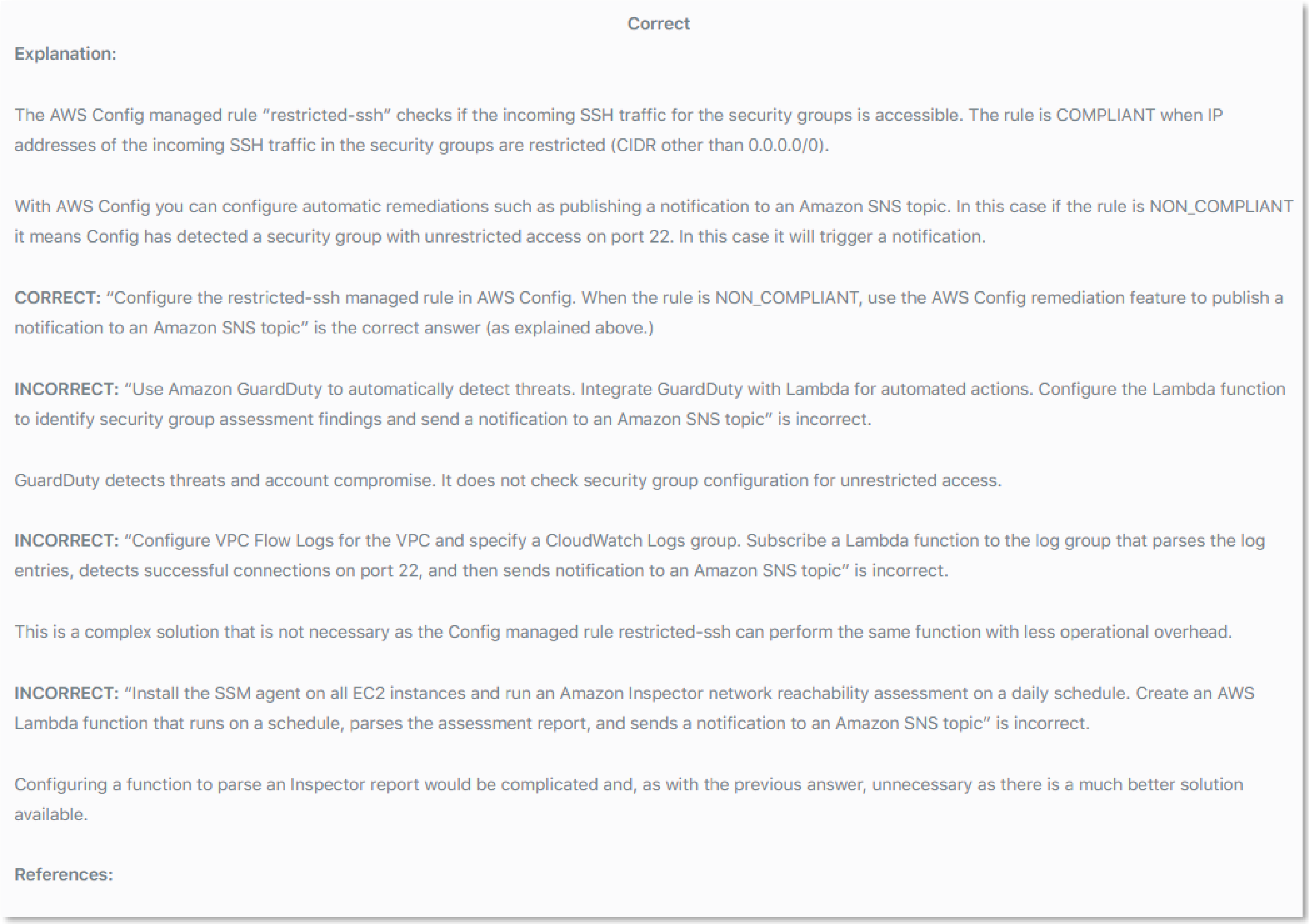
14. QUESTION

The security department in a company requires automatic discovery of any security groups that allow unrestricted inbound traffic on port 22 (SSH). The security administrators should be notified of any violations

Which solution meets these requirements with the MOST operational efficiency?

○ Use Amazon GuardDuty to automatically detect threats. Integrate GuardDuty with Lambda for automated actions. Configure the Lambda function to identify security group assessment findings and send a notification to an Amazon SNS topic.

○ Configure VPC Flow Logs for the VPC and specify a CloudWatch Logs group. Subscribe a Lambda function to the log group that parses the log entries, detects successful connections on port 22, and then sends notification to an Amazon SNS topic.



Correct

Explanation:

The AWS Config managed rule "restricted-ssh" checks if the incoming SSH traffic for the security groups is accessible. The rule is COMPLIANT when IP addresses of the incoming SSH traffic in the security groups are restricted (CIDR other than 0.0.0.0/0).

With AWS Config you can configure automatic remediations such as publishing a notification to an Amazon SNS topic. In this case if the rule is NON_COMPLIANT it means Config has detected a security group with unrestricted access on port 22. In this case it will trigger a notification.

CORRECT: "Configure the restricted-ssh managed rule in AWS Config. When the rule is NON_COMPLIANT, use the AWS Config remediation feature to publish a notification to an Amazon SNS topic" is the correct answer (as explained above.)

INCORRECT: "Use Amazon GuardDuty to automatically detect threats. Integrate GuardDuty with Lambda for automated actions. Configure the Lambda function to identify security group assessment findings and send a notification to an Amazon SNS topic" is incorrect.

GuardDuty detects threats and account compromise. It does not check security group configuration for unrestricted access.

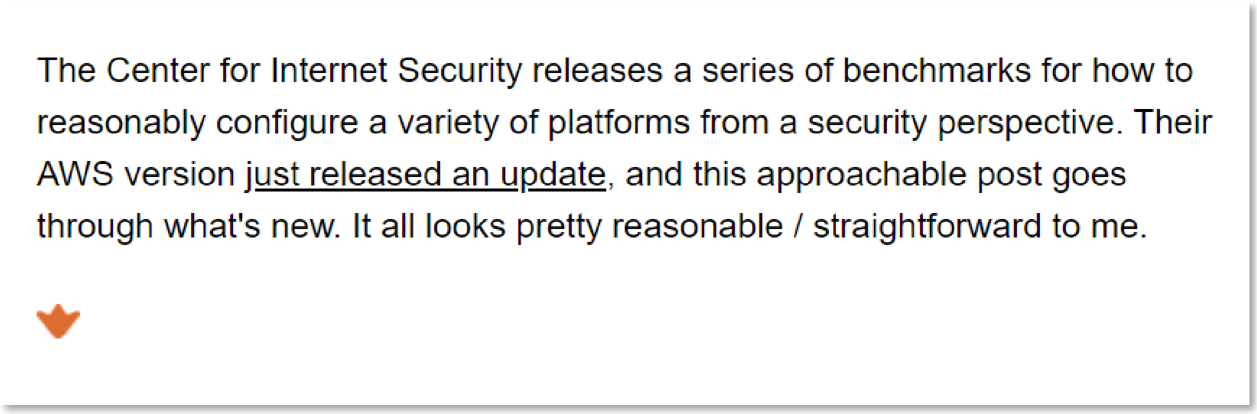
INCORRECT: "Configure VPC Flow Logs for the VPC and specify a CloudWatch Logs group. Subscribe a Lambda function to the log group that parses the log entries, detects successful connections on port 22, and then sends notification to an Amazon SNS topic" is incorrect.

This is a complex solution that is not necessary as the Config managed rule restricted-ssh can perform the same function with less operational overhead.

INCORRECT: "Install the SSM agent on all EC2 instances and run an Amazon Inspector network reachability assessment on a daily schedule. Create an AWS Lambda function that runs on a schedule, parses the assessment report, and sends a notification to an Amazon SNS topic" is incorrect.

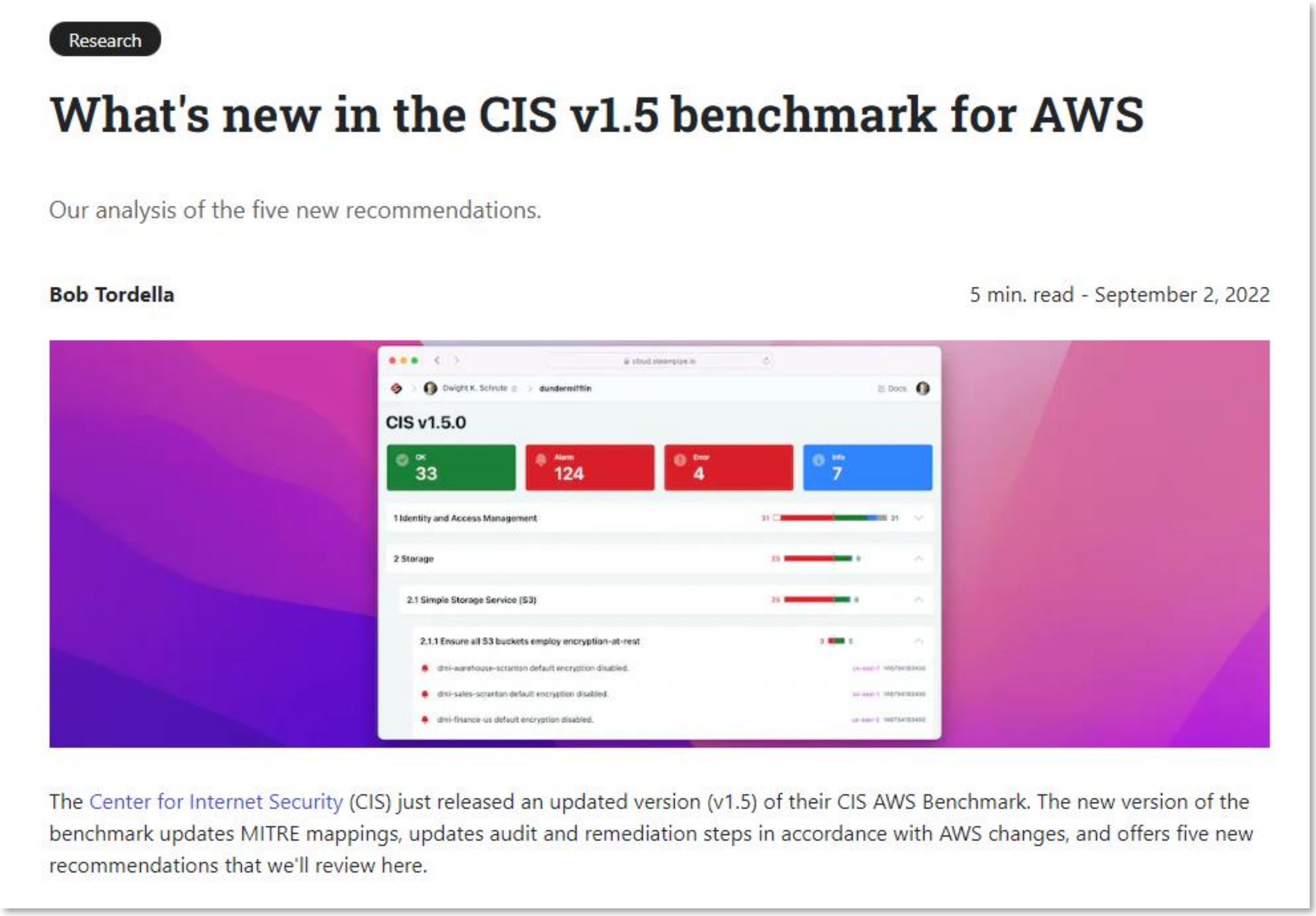
Configuring a function to parse an Inspector report would be complicated and, as with the previous answer, unnecessary as there is a much better solution available.

References:



The Center for Internet Security releases a series of benchmarks for how to reasonably configure a variety of platforms from a security perspective. Their AWS version just released an update, and this approachable post goes through what's new. It all looks pretty reasonable / straightforward to me.

Snippet from Corey Quinn's newsletter on 15th September 2022



Research

What's new in the CIS v1.5 benchmark for AWS

Our analysis of the five new recommendations.

Bob Tordella

5 min. read - September 2, 2022

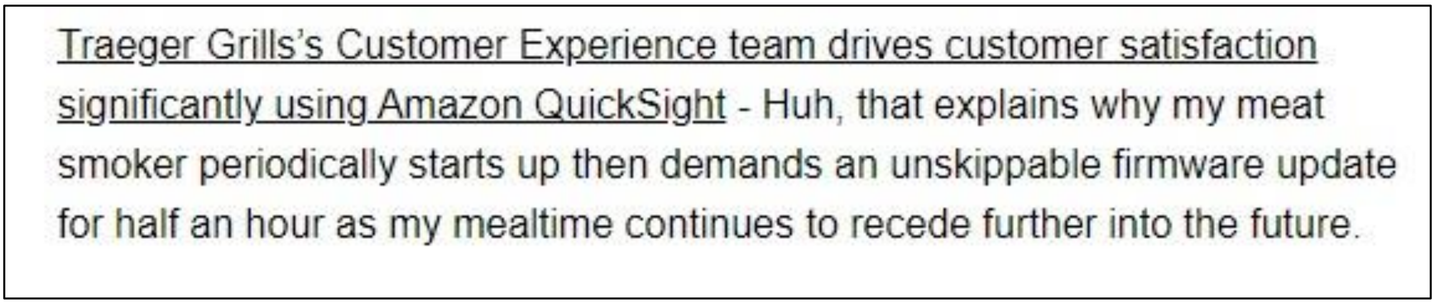
cloud.steampipe.io
Dwight K. Schrute
dundermifflin
Docs
CIS v1.5.0
OK
33
Alarm
124
Error
4
Info
7
1 Identity and Access Management
31
31
2 Storage
2.1 Simple Storage Service (S3)
2.1.1 Ensure all S3 buckets employ encryption-at-rest

dmi-warehouse-scranton default encryption disabled.

dmi-sales-scranton default encryption disabled.

dmi-finance-us default encryption disabled.

The Center for Internet Security (CIS) just released an updated version (v1.5) of their CIS AWS Benchmark. The new version of the benchmark updates MITRE mappings, updates audit and remediation steps in accordance with AWS changes, and offers five new recommendations that we'll review here.



Traeger Grills's Customer Experience team drives customer satisfaction significantly using Amazon QuickSight - Huh, that explains why my meat smoker periodically starts up then demands an unskippable firmware update for half an hour as my mealtime continues to recede further into the future.

Corey Quinn writing on 25th March 2024

Glossary

Bibliography

I. Official

II. Unofficial

III. Critical

IV. General

I. Official

[Waikar 2017]

Waikar, Kanchan and David Aiken (2017). How to set up Continuous Golden AMI Vulnerability Assessments with Amazon Inspector. *AWS Security Blog*. Available at: <https://aws.amazon.com/blogs/security/how-to-set-up-continuous-golden-ami-vulnerability-assessments-with-amazon-inspector/>

II. Unofficial

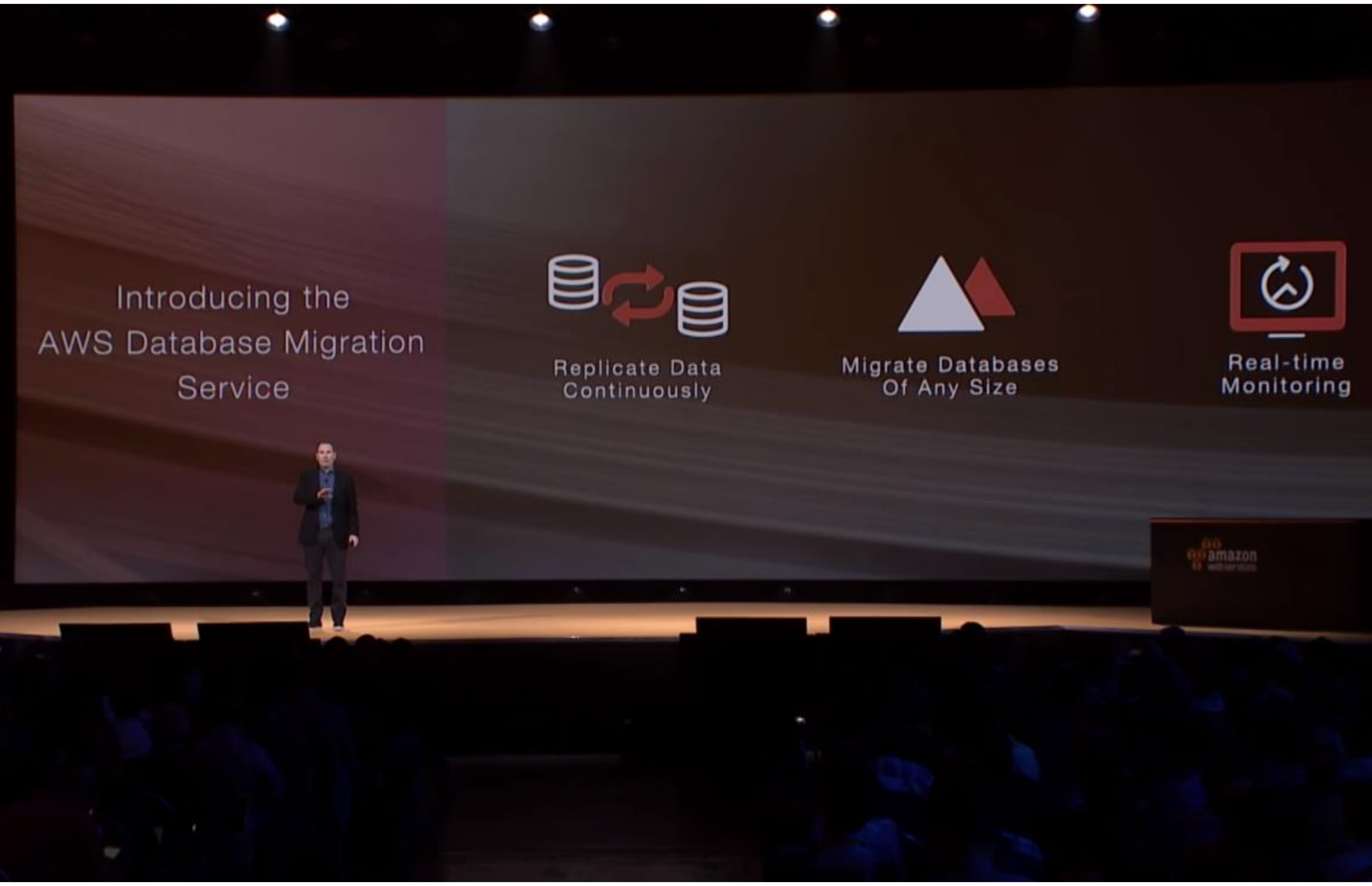
[Tordella 2022]

Tordella, Bob (2022). What's new in the CIS v1.5 benchmark for AWS. Available at: <https://steampipe.io/blog/cis-v15-aws-benchmark?ck_subscriber_id=1560524742>

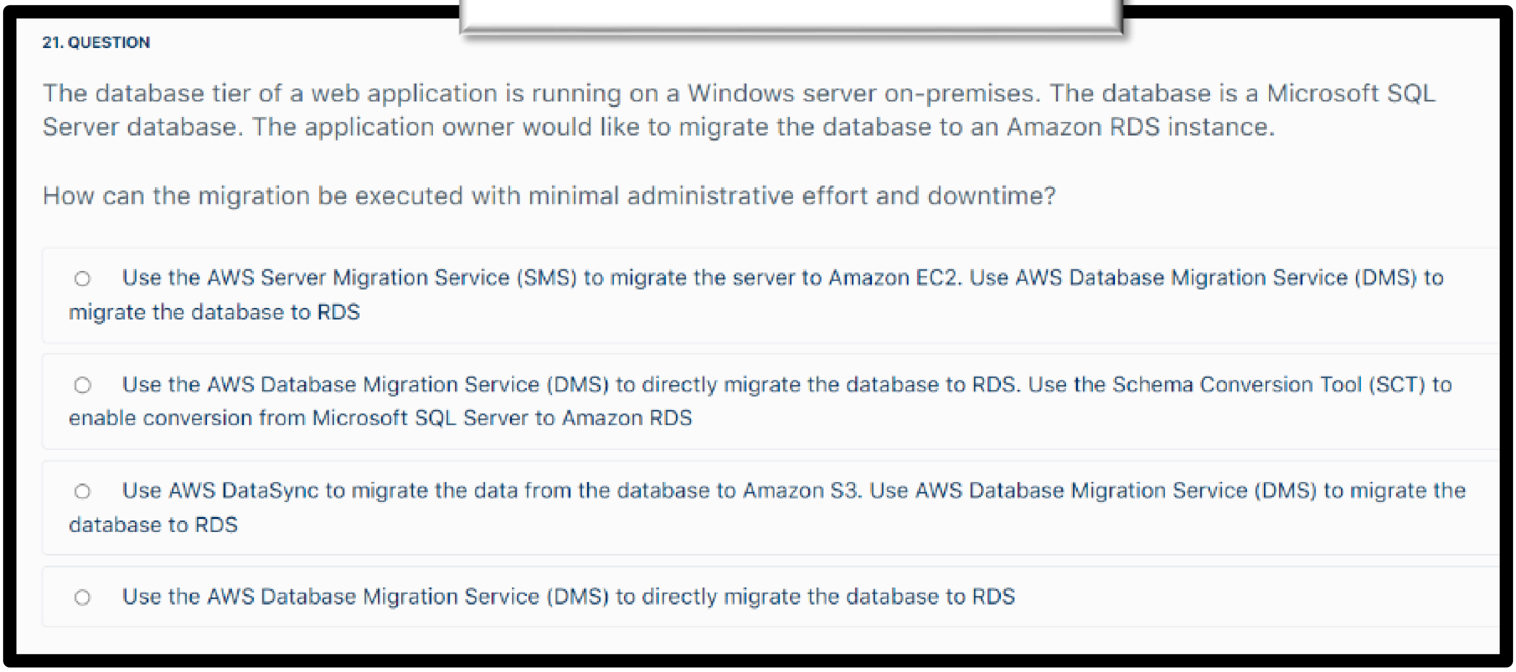
QuickSight

Pictured (middle) is Etienne Pradier, a great close-up magician

Database Migration Service



Andy Jassy (CEO of AWS) announcing the Database Migration Service in 2015



21. QUESTION

The database tier of a web application is running on a Windows server on-premises. The database is a Microsoft SQL Server database. The application owner would like to migrate the database to an Amazon RDS instance.

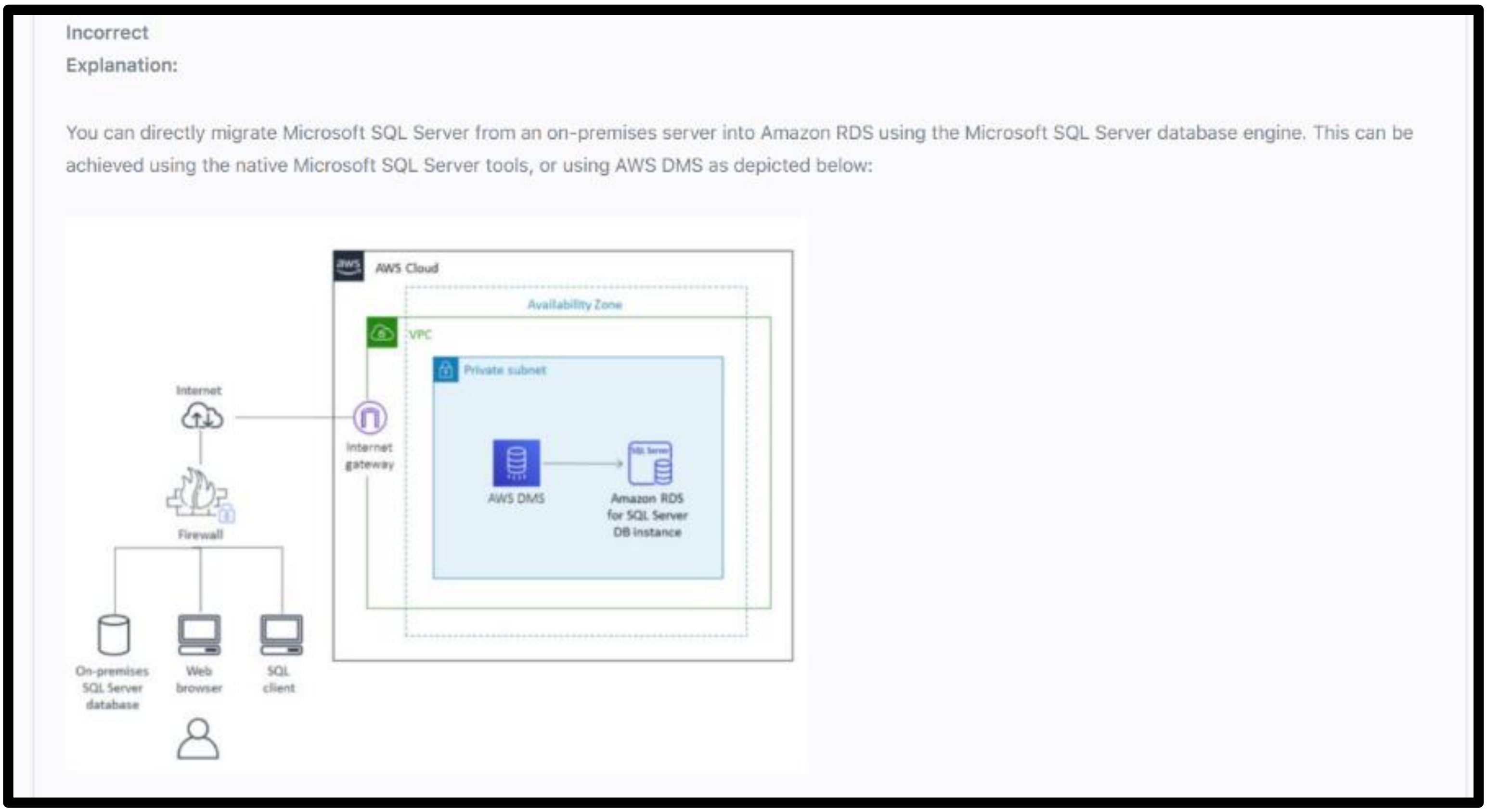
How can the migration be executed with minimal administrative effort and downtime?

○ Use the AWS Server Migration Service (SMS) to migrate the server to Amazon EC2. Use AWS Database Migration Service (DMS) to migrate the database to RDS

○ Use the AWS Database Migration Service (DMS) to directly migrate the database to RDS. Use the Schema Conversion Tool (SCT) to enable conversion from Microsoft SQL Server to Amazon RDS

○ Use AWS DataSync to migrate the data from the database to Amazon S3. Use AWS Database Migration Service (DMS) to migrate the database to RDS

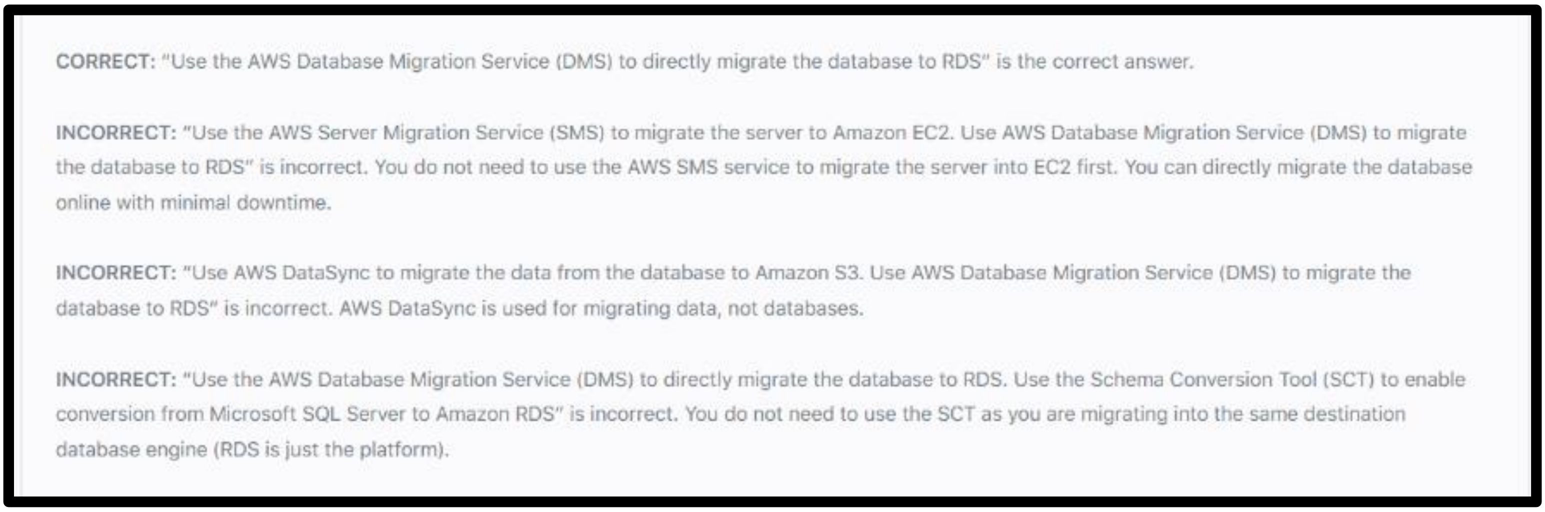
○ Use the AWS Database Migration Service (DMS) to directly migrate the database to RDS



Incorrect

Explanation:

You can directly migrate Microsoft SQL Server from an on-premises server into Amazon RDS using the Microsoft SQL Server database engine. This can be achieved using the native Microsoft SQL Server tools, or using AWS DMS as depicted below:



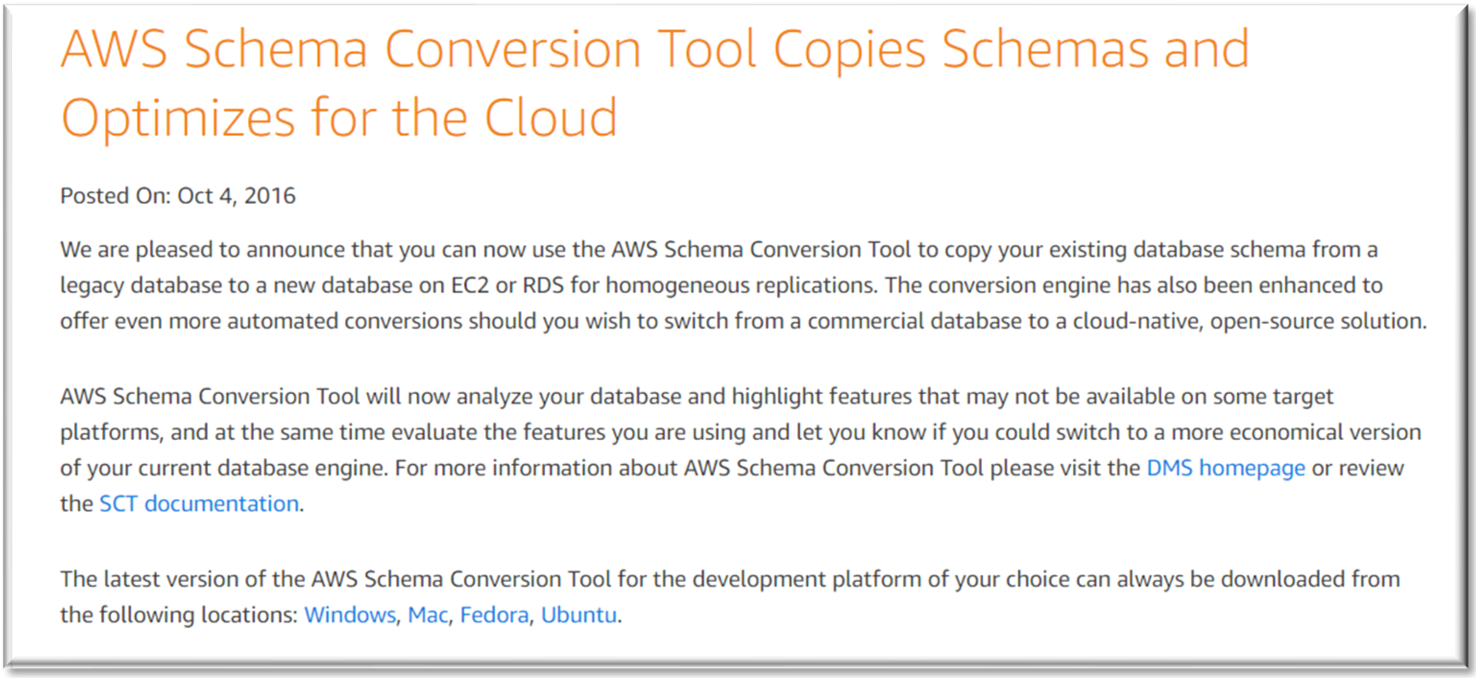
CORRECT: "Use the AWS Database Migration Service (DMS) to directly migrate the database to RDS" is the correct answer.

INCORRECT: "Use the AWS Server Migration Service (SMS) to migrate the server to Amazon EC2. Use AWS Database Migration Service (DMS) to migrate the database to RDS" is incorrect. You do not need to use the AWS SMS service to migrate the server into EC2 first. You can directly migrate the database online with minimal downtime.

INCORRECT: "Use AWS DataSync to migrate the data from the database to Amazon S3. Use AWS Database Migration Service (DMS) to migrate the database to RDS" is incorrect. AWS DataSync is used for migrating data, not databases.

INCORRECT: "Use the AWS Database Migration Service (DMS) to directly migrate the database to RDS. Use the Schema Conversion Tool (SCT) to enable conversion from Microsoft SQL Server to Amazon RDS" is incorrect. You do not need to use the SCT as you are migrating into the same destination database engine (RDS is just the platform).

Schema Conversion Tool (SCT)



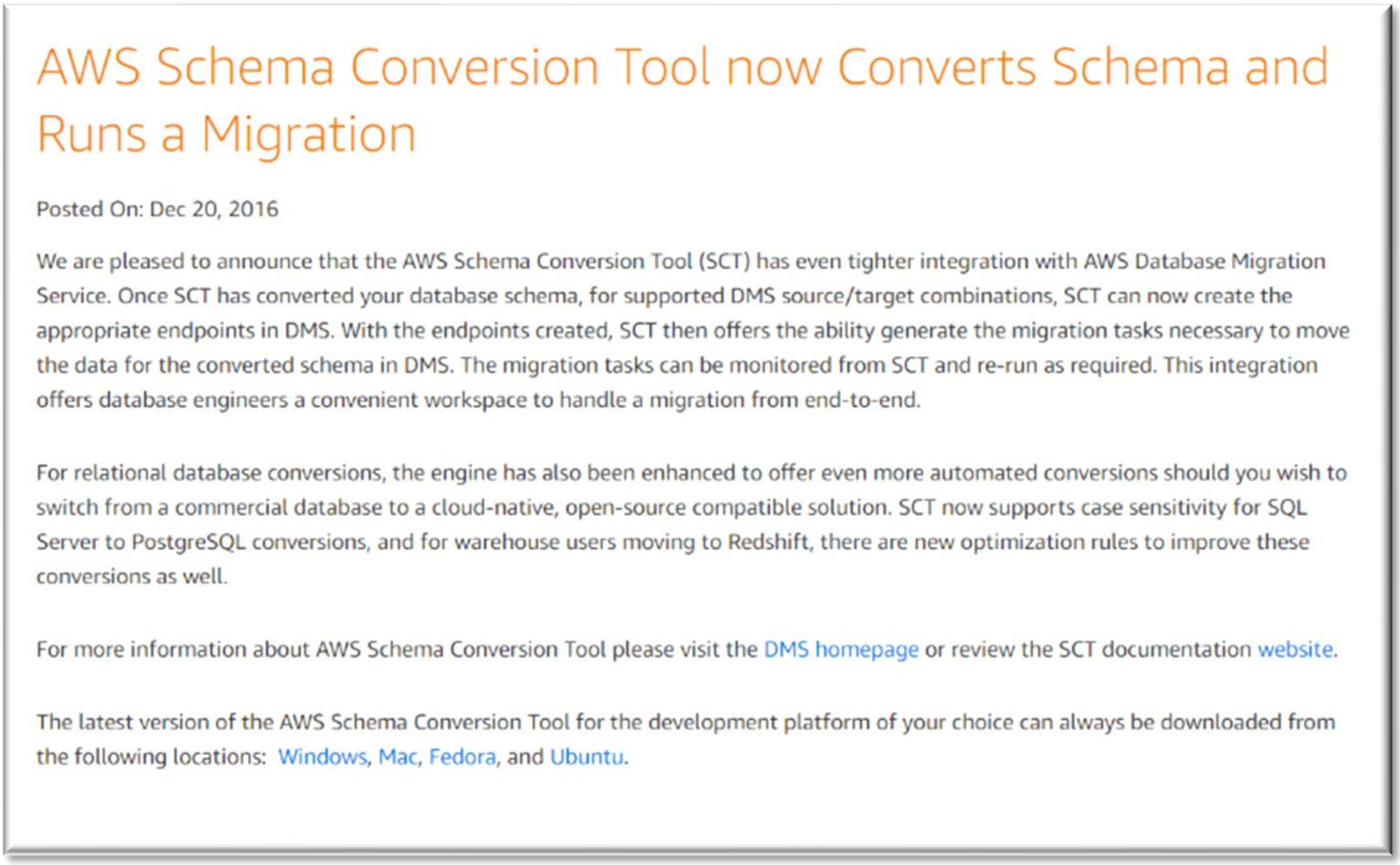
AWS Schema Conversion Tool Copies Schemas and Optimizes for the Cloud

Posted On: Oct 4, 2016

We are pleased to announce that you can now use the AWS Schema Conversion Tool to copy your existing database schema from a legacy database to a new database on EC2 or RDS for homogeneous replications. The conversion engine has also been enhanced to offer even more automated conversions should you wish to switch from a commercial database to a cloud-native, open-source solution.

AWS Schema Conversion Tool will now analyze your database and highlight features that may not be available on some target platforms, and at the same time evaluate the features you are using and let you know if you could switch to a more economical version of your current database engine. For more information about AWS Schema Conversion Tool please visit the DMS homepage or review the SCT documentation.

The latest version of the AWS Schema Conversion Tool for the development platform of your choice can always be downloaded from the following locations: Windows, Mac, Fedora, Ubuntu.



AWS Schema Conversion Tool now Converts Schema and Runs a Migration

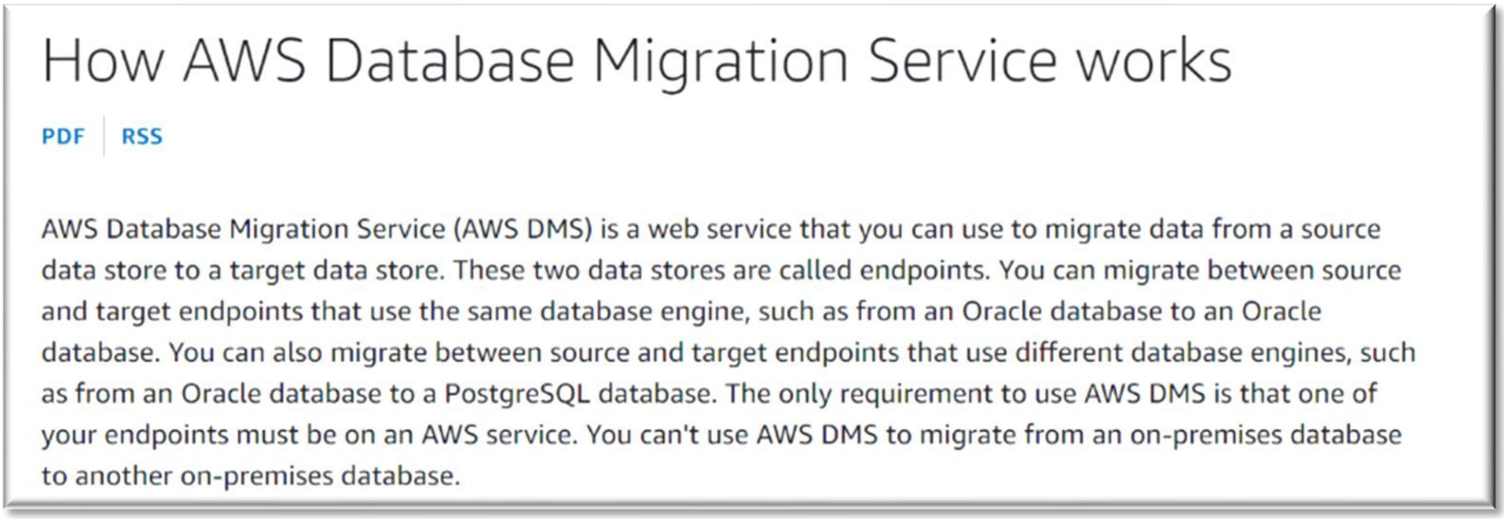
Posted On: Dec 20, 2016

We are pleased to announce that the AWS Schema Conversion Tool (SCT) has even tighter integration with AWS Database Migration Service. Once SCT has converted your database schema, for supported DMS source/target combinations, SCT can now create the appropriate endpoints in DMS. With the endpoints created, SCT then offers the ability generate the migration tasks necessary to move the data for the converted schema in DMS. The migration tasks can be monitored from SCT and re-run as required. This integration offers database engineers a convenient workspace to handle a migration from end-to-end.

For relational database conversions, the engine has also been enhanced to offer even more automated conversions should you wish to switch from a commercial database to a cloud-native, open-source compatible solution. SCT now supports case sensitivity for SQL Server to PostgreSQL conversions, and for warehouse users moving to Redshift, there are new optimization rules to improve these conversions as well.

For more information about AWS Schema Conversion Tool please visit the DMS homepage or review the SCT documentation website.

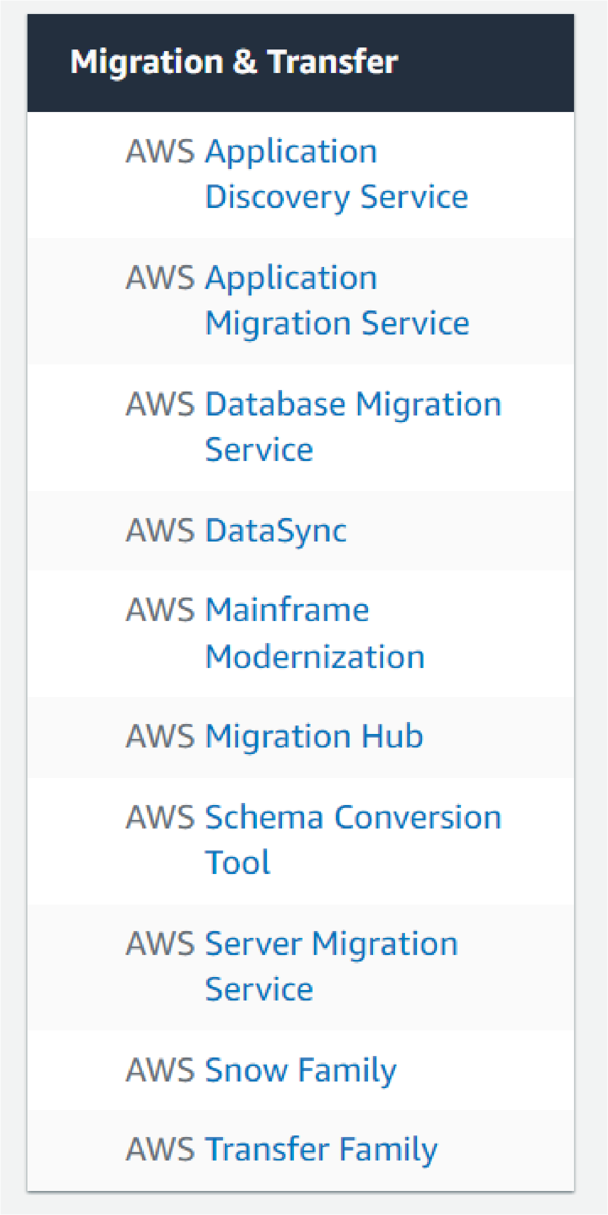
The latest version of the AWS Schema Conversion Tool for the development platform of your choice can always be downloaded from the following locations: Windows, Mac, Fedora, and Ubuntu.



How AWS Database Migration Service works

AWS Database Migration Service (AWS DMS) is a web service that you can use to migrate data from a source data store to a target data store. These two data stores are called endpoints. You can migrate between source and target endpoints that use the same database engine, such as from an Oracle database to an Oracle database. You can also migrate between source and target endpoints that use different database engines, such as from an Oracle database to a PostgreSQL database. The only requirement to use AWS DMS is that one of your endpoints must be on an AWS service. You can't use AWS DMS to migrate from an on-premises database to another on-premises database.

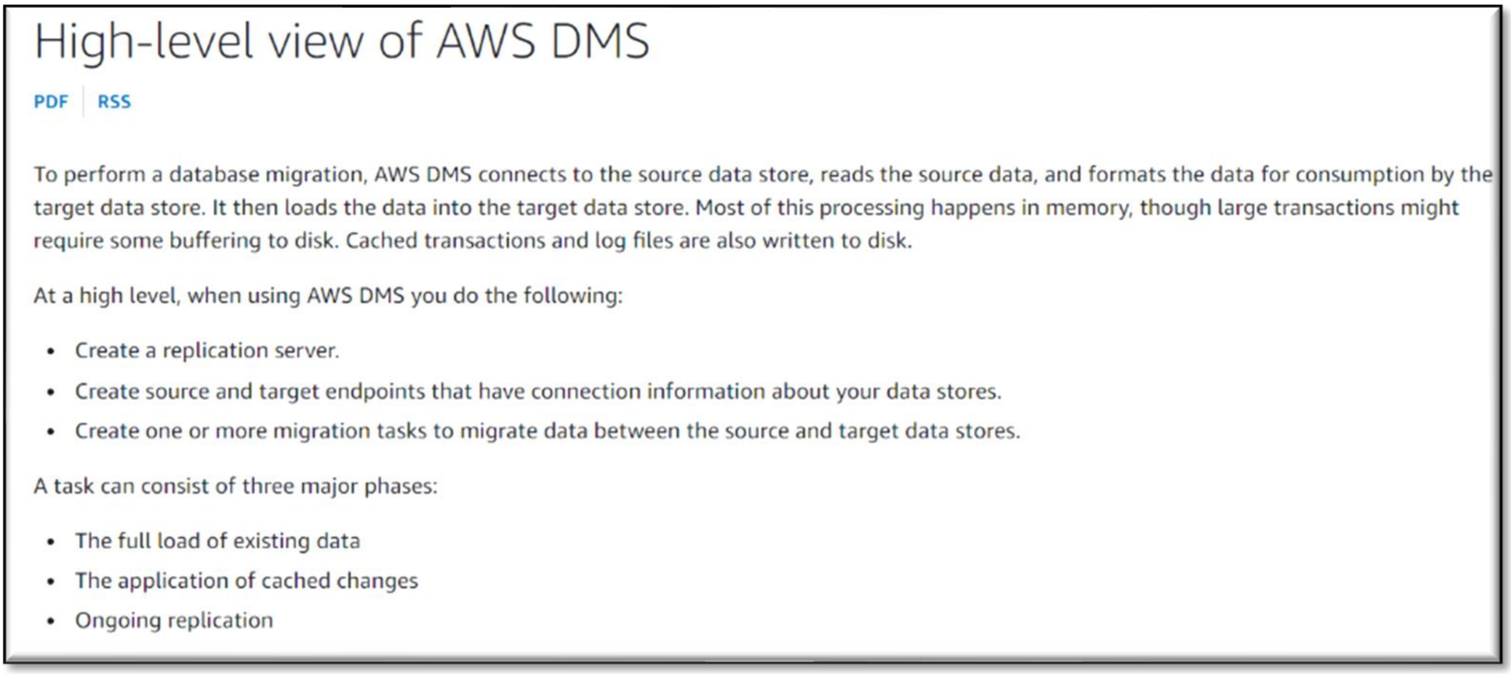
At the time of writing, the Schema Conversion Tool has a *distinct* User Guide from the DMS:



Migration & Transfer

- AWS Application Discovery Service
- AWS Application Migration Service
- AWS Database Migration Service
- AWS DataSync
- AWS Mainframe Modernization
- AWS Migration Hub
- AWS Schema Conversion Tool
- AWS Server Migration Service
- AWS Snow Family
- AWS Transfer Family

Let's start by looking at the DMS. Then, we will move onto the SCT (Schema Conversion Tool).



High-level view of AWS DMS

PDF | RSS

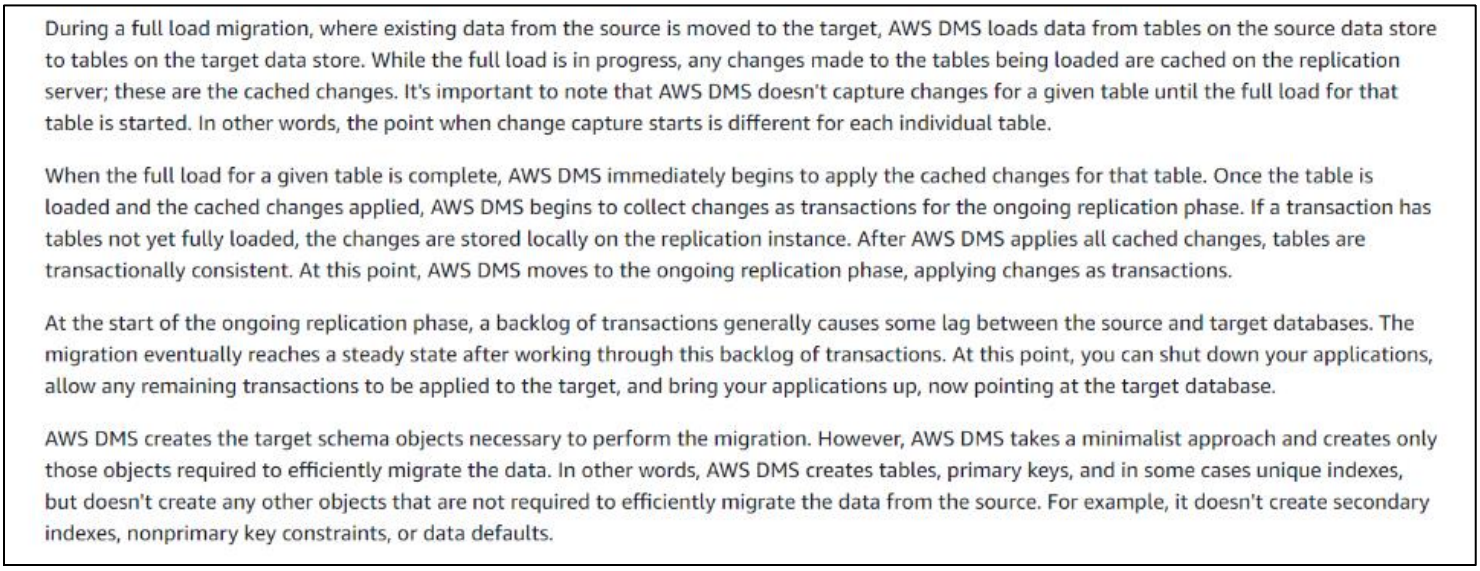
To perform a database migration, AWS DMS connects to the source data store, reads the source data, and formats the data for consumption by the target data store. It then loads the data into the target data store. Most of this processing happens in memory, though large transactions might require some buffering to disk. Cached transactions and log files are also written to disk.

At a high level, when using AWS DMS you do the following:

- Create a replication server.
- Create source and target endpoints that have connection information about your data stores.
- Create one or more migration tasks to migrate data between the source and target data stores.

A task can consist of three major phases:

- The full load of existing data
- The application of cached changes
- Ongoing replication

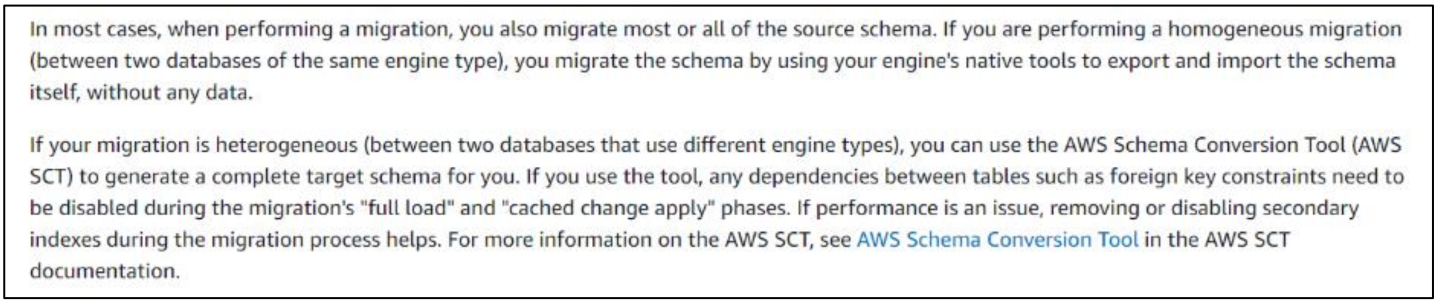


During a full load migration, where existing data from the source is moved to the target, AWS DMS loads data from tables on the source data store to tables on the target data store. While the full load is in progress, any changes made to the tables being loaded are cached on the replication server; these are the cached changes. It's important to note that AWS DMS doesn't capture changes for a given table until the full load for that table is started. In other words, the point when change capture starts is different for each individual table.

When the full load for a given table is complete, AWS DMS immediately begins to apply the cached changes for that table. Once the table is loaded and the cached changes applied, AWS DMS begins to collect changes as transactions for the ongoing replication phase. If a transaction has tables not yet fully loaded, the changes are stored locally on the replication instance. After AWS DMS applies all cached changes, tables are transactionally consistent. At this point, AWS DMS moves to the ongoing replication phase, applying changes as transactions.

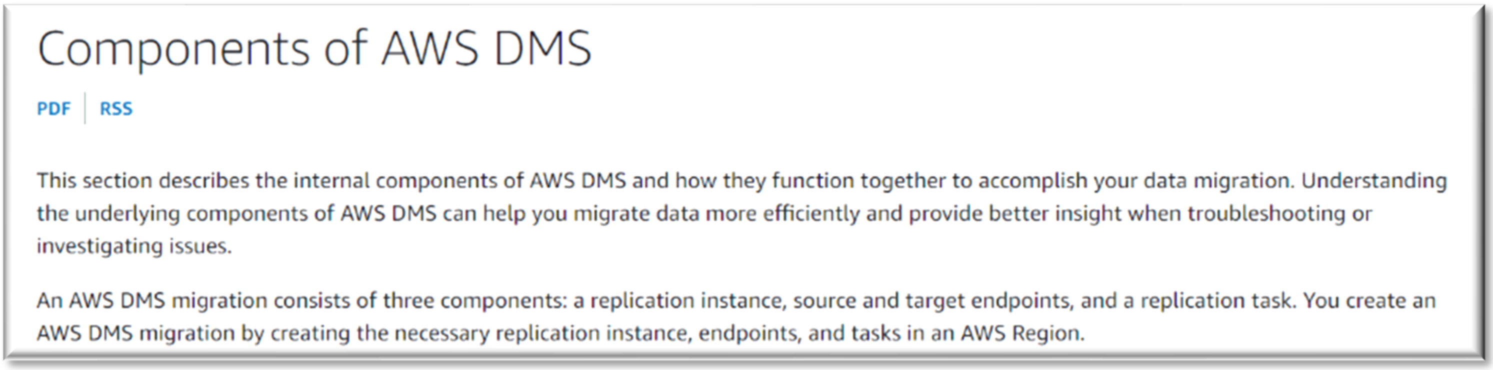
At the start of the ongoing replication phase, a backlog of transactions generally causes some lag between the source and target databases. The migration eventually reaches a steady state after working through this backlog of transactions. At this point, you can shut down your applications, allow any remaining transactions to be applied to the target, and bring your applications up, now pointing at the target database.

AWS DMS creates the target schema objects necessary to perform the migration. However, AWS DMS takes a minimalist approach and creates only those objects required to efficiently migrate the data. In other words, AWS DMS creates tables, primary keys, and in some cases unique indexes, but doesn't create any other objects that are not required to efficiently migrate the data from the source. For example, it doesn't create secondary indexes, nonprimary key constraints, or data defaults.



In most cases, when performing a migration, you also migrate most or all of the source schema. If you are performing a homogeneous migration (between two databases of the same engine type), you migrate the schema by using your engine's native tools to export and import the schema itself, without any data.

If your migration is heterogeneous (between two databases that use different engine types), you can use the AWS Schema Conversion Tool (AWS SCT) to generate a complete target schema for you. If you use the tool, any dependencies between tables such as foreign key constraints need to be disabled during the migration's "full load" and "cached change apply" phases. If performance is an issue, removing or disabling secondary indexes during the migration process helps. For more information on the AWS SCT, see AWS Schema Conversion Tool in the AWS SCT documentation.

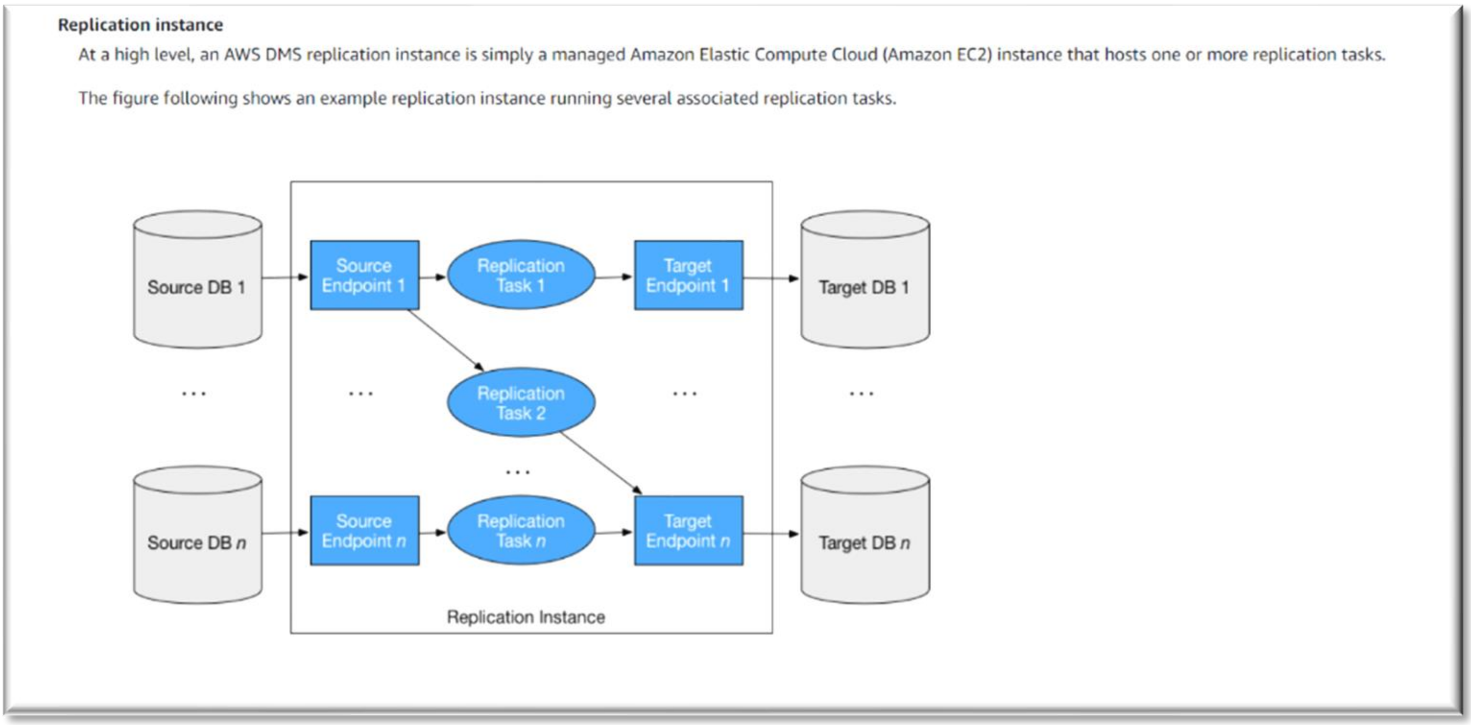


Components of AWS DMS

PDF | RSS

This section describes the internal components of AWS DMS and how they function together to accomplish your data migration. Understanding the underlying components of AWS DMS can help you migrate data more efficiently and provide better insight when troubleshooting or investigating issues.

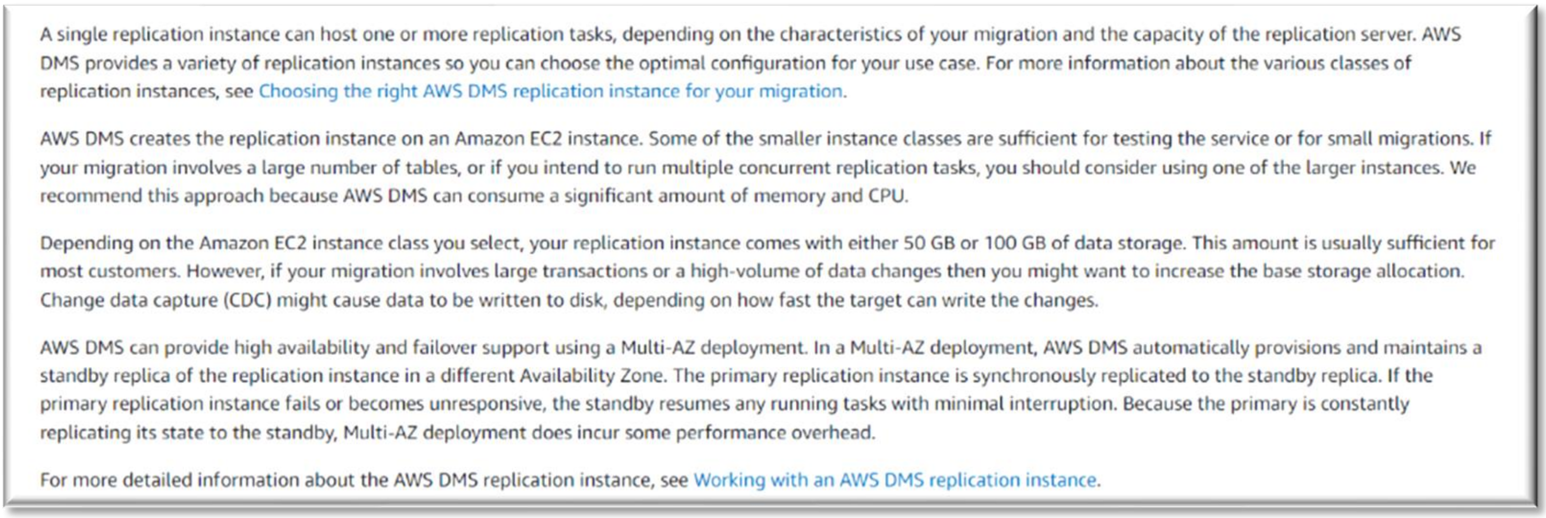
An AWS DMS migration consists of three components: a replication instance, source and target endpoints, and a replication task. You create an AWS DMS migration by creating the necessary replication instance, endpoints, and tasks in an AWS Region.



Replication instance

At a high level, an AWS DMS replication instance is simply a managed Amazon Elastic Compute Cloud (Amazon EC2) instance that hosts one or more replication tasks.

The figure following shows an example replication instance running several associated replication tasks.



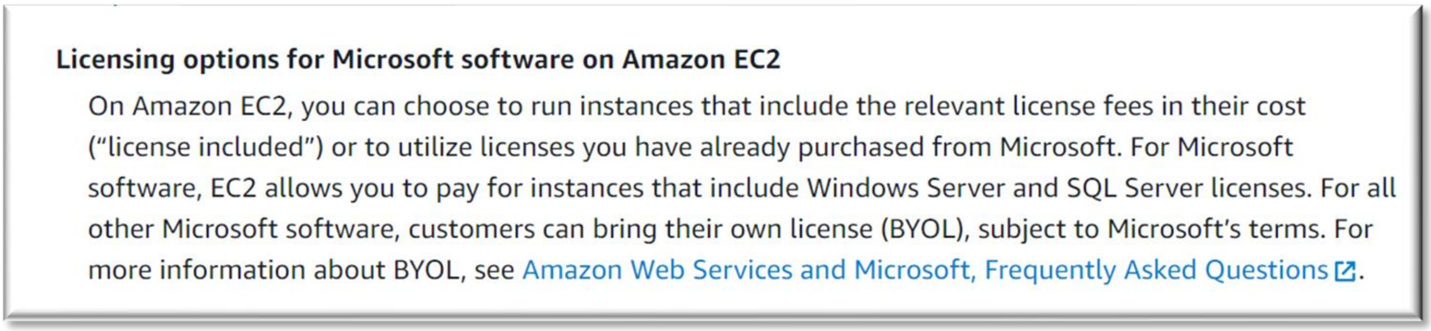
A single replication instance can host one or more replication tasks, depending on the characteristics of your migration and the capacity of the replication server. AWS DMS provides a variety of replication instances so you can choose the optimal configuration for your use case. For more information about the various classes of replication instances, see Choosing the right AWS DMS replication instance for your migration.

AWS DMS creates the replication instance on an Amazon EC2 instance. Some of the smaller instance classes are sufficient for testing the service or for small migrations. If your migration involves a large number of tables, or if you intend to run multiple concurrent replication tasks, you should consider using one of the larger instances. We recommend this approach because AWS DMS can consume a significant amount of memory and CPU.

Depending on the Amazon EC2 instance class you select, your replication instance comes with either 50 GB or 100 GB of data storage. This amount is usually sufficient for most customers. However, if your migration involves large transactions or a high-volume of data changes then you might want to increase the base storage allocation. Change data capture (CDC) might cause data to be written to disk, depending on how fast the target can write the changes.

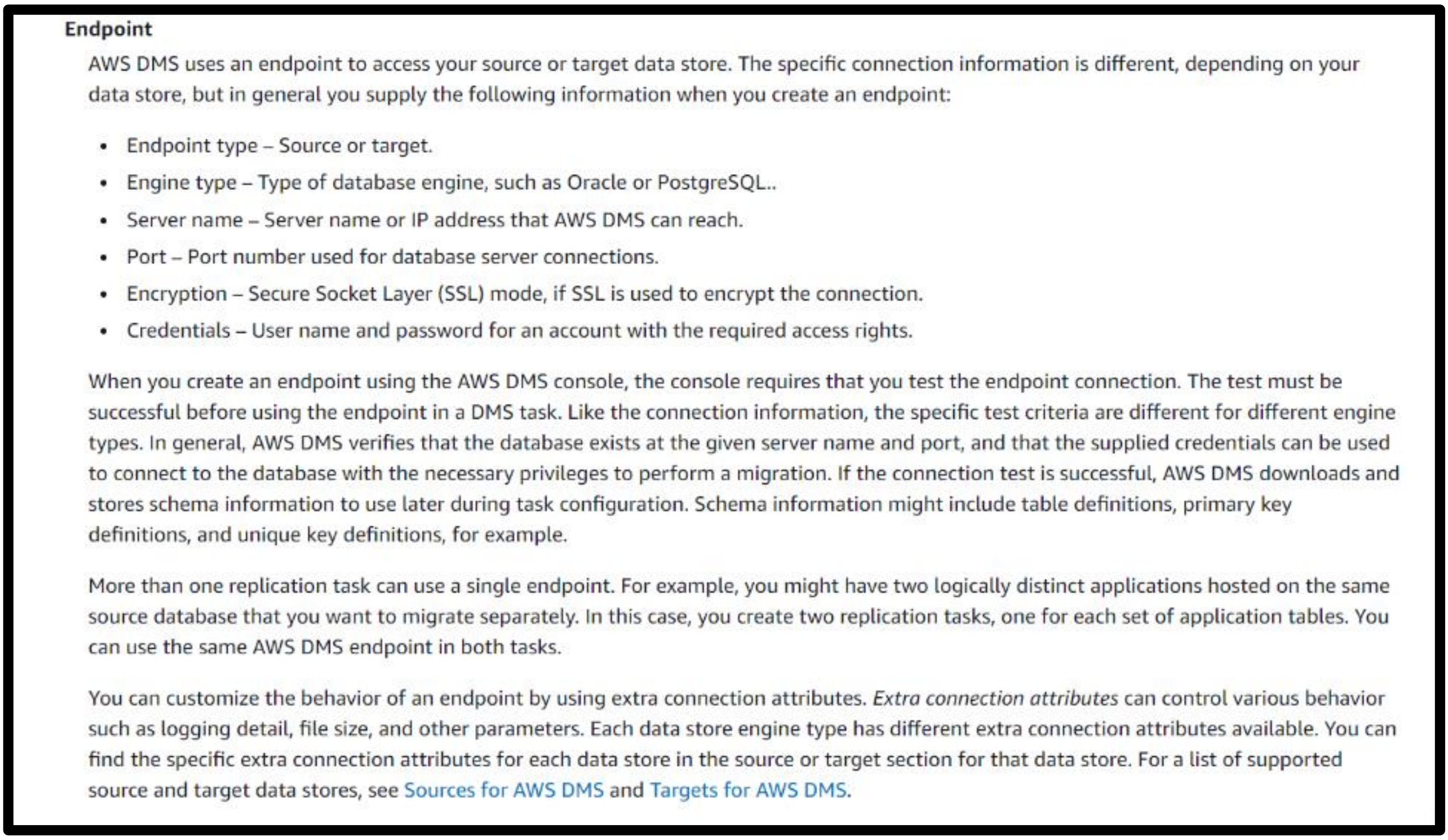
AWS DMS can provide high availability and failover support using a Multi-AZ deployment. In a Multi-AZ deployment, AWS DMS automatically provisions and maintains a standby replica of the replication instance in a different Availability Zone. The primary replication instance is synchronously replicated to the standby replica. If the primary replication instance fails or becomes unresponsive, the standby resumes any running tasks with minimal interruption. Because the primary is constantly replicating its state to the standby, Multi-AZ deployment does incur some performance overhead.

For more detailed information about the AWS DMS replication instance, see Working with an AWS DMS replication instance.



Licensing options for Microsoft software on Amazon EC2

On Amazon EC2, you can choose to run instances that include the relevant license fees in their cost ("license included") or to utilize licenses you have already purchased from Microsoft. For Microsoft software, EC2 allows you to pay for instances that include Windows Server and SQL Server licenses. For all other Microsoft software, customers can bring their own license (BYOL), subject to Microsoft's terms. For more information about BYOL, see Amazon Web Services and Microsoft, Frequently Asked Questions.



Endpoint

AWS DMS uses an endpoint to access your source or target data store. The specific connection information is different, depending on your data store, but in general you supply the following information when you create an endpoint:

- Endpoint type – Source or target.
- Engine type – Type of database engine, such as Oracle or PostgreSQL..
- Server name – Server name or IP address that AWS DMS can reach.
- Port – Port number used for database server connections.
- Encryption – Secure Socket Layer (SSL) mode, if SSL is used to encrypt the connection.
- Credentials – User name and password for an account with the required access rights.

When you create an endpoint using the AWS DMS console, the console requires that you test the endpoint connection. The test must be successful before using the endpoint in a DMS task. Like the connection information, the specific test criteria are different for different engine types. In general, AWS DMS verifies that the database exists at the given server name and port, and that the supplied credentials can be used to connect to the database with the necessary privileges to perform a migration. If the connection test is successful, AWS DMS downloads and stores schema information to use later during task configuration. Schema information might include table definitions, primary key definitions, and unique key definitions, for example.

More than one replication task can use a single endpoint. For example, you might have two logically distinct applications hosted on the same source database that you want to migrate separately. In this case, you create two replication tasks, one for each set of application tables. You can use the same AWS DMS endpoint in both tasks.

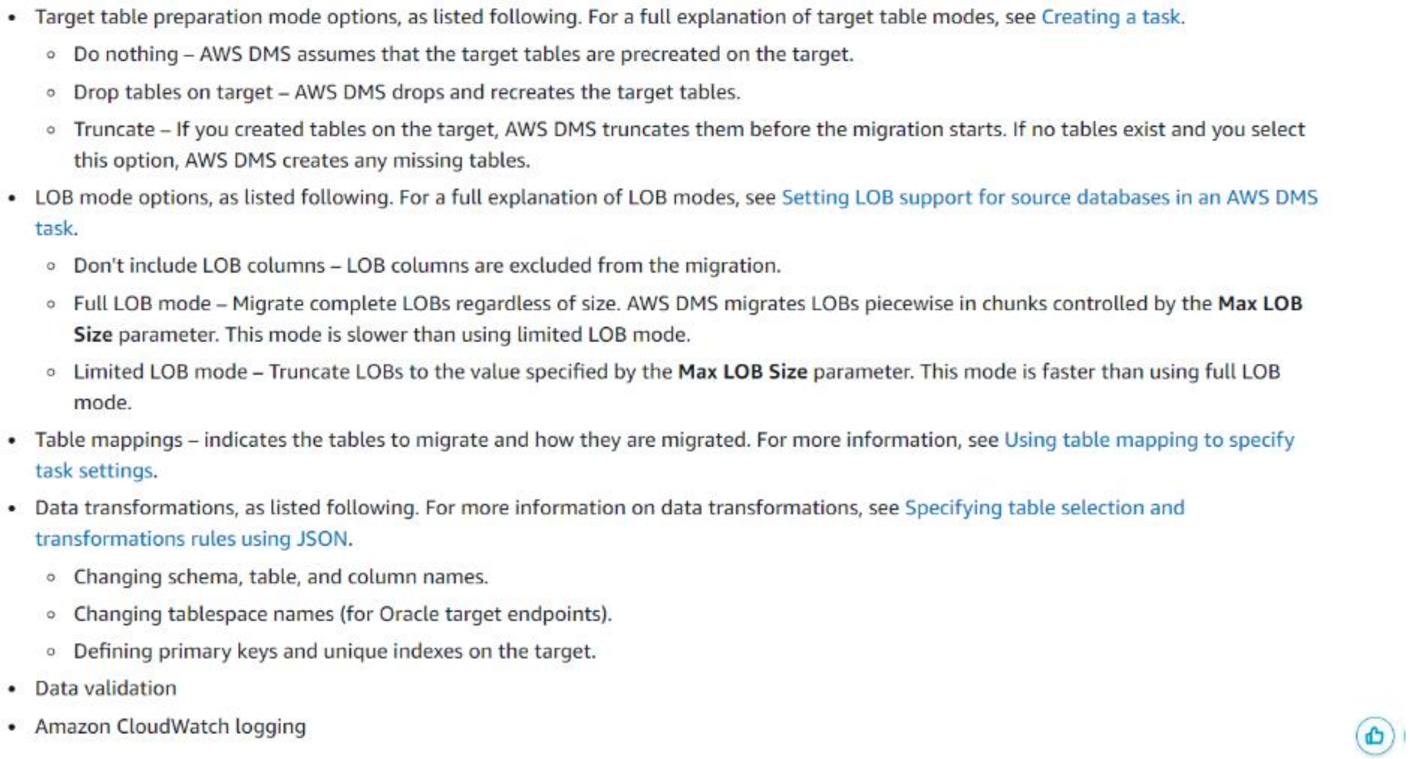
You can customize the behavior of an endpoint by using extra connection attributes. *Extra connection attributes* can control various behavior such as logging detail, file size, and other parameters. Each data store engine type has different extra connection attributes available. You can find the specific extra connection attributes for each data store in the source or target section for that data store. For a list of supported source and target data stores, see Sources for AWS DMS and Targets for AWS DMS.

Replication tasks

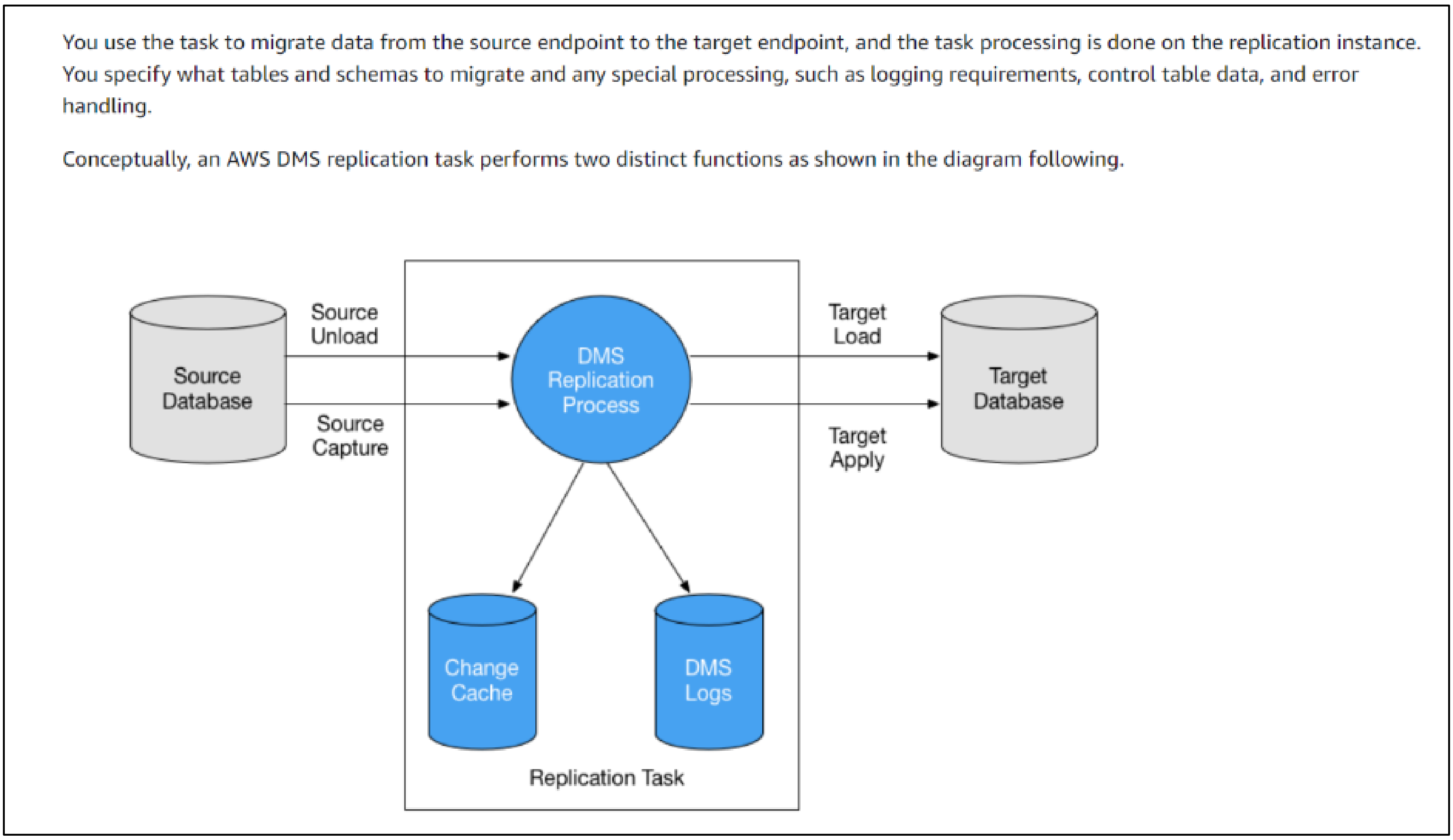
You use an AWS DMS replication task to move a set of data from the source endpoint to the target endpoint. Creating a replication task is the last step you need to take before you start a migration.

When you create a replication task, you specify the following task settings:

- Replication instance – the instance to host and run the task
- Source endpoint
- Target endpoint
- Migration type options, as listed following. For a full explanation of the migration type options, see Creating a task.
  - Full load (Migrate existing data) – If you can afford an outage long enough to copy your existing data, this option is a good one to choose. This option simply migrates the data from your source database to your target database, creating tables when necessary.
  - Full load + CDC (Migrate existing data and replicate ongoing changes) – This option performs a full data load while capturing changes on the source. After the full load is complete, captured changes are applied to the target. Eventually, the application of changes reaches a steady state. At this point, you can shut down your applications, let the remaining changes flow through to the target, and then restart your applications pointing at the target.
  - CDC only (Replicate data changes only) – In some situations, it might be more efficient to copy existing data using a method other than AWS DMS. For example, in a homogeneous migration, using native export and import tools might be more efficient at loading bulk data. In this situation, you can use AWS DMS to replicate changes starting when you start your bulk load to bring and keep your source and target databases in sync.

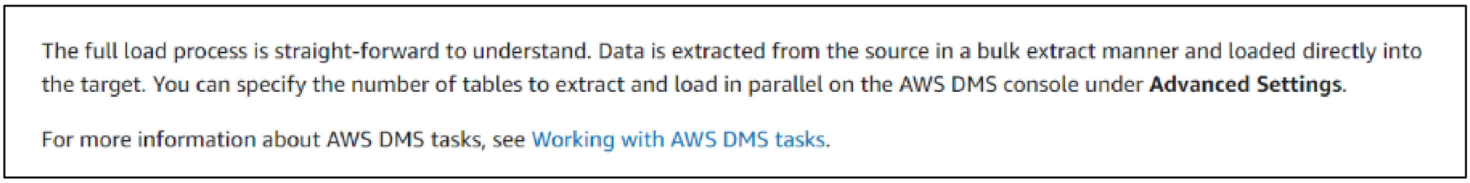


- Target table preparation mode options, as listed following. For a full explanation of target table modes, see Creating a task.
  - Do nothing – AWS DMS assumes that the target tables are precreated on the target.
  - Drop tables on target – AWS DMS drops and recreates the target tables.
  - Truncate – If you created tables on the target, AWS DMS truncates them before the migration starts. If no tables exist and you select this option, AWS DMS creates any missing tables.
- LOB mode options, as listed following. For a full explanation of LOB modes, see Setting LOB support for source databases in an AWS DMS task.
  - Don't include LOB columns – LOB columns are excluded from the migration.
  - Full LOB mode – Migrate complete LOBs regardless of size. AWS DMS migrates LOBs piecewise in chunks controlled by the **Max LOB Size** parameter. This mode is slower than using limited LOB mode.
  - Limited LOB mode – Truncate LOBs to the value specified by the **Max LOB Size** parameter. This mode is faster than using full LOB mode.
- Table mappings – indicates the tables to migrate and how they are migrated. For more information, see Using table mapping to specify task settings.
- Data transformations, as listed following. For more information on data transformations, see Specifying table selection and transformations rules using JSON.
  - Changing schema, table, and column names.
  - Changing tablespace names (for Oracle target endpoints).
  - Defining primary keys and unique indexes on the target.
- Data validation
- Amazon CloudWatch logging



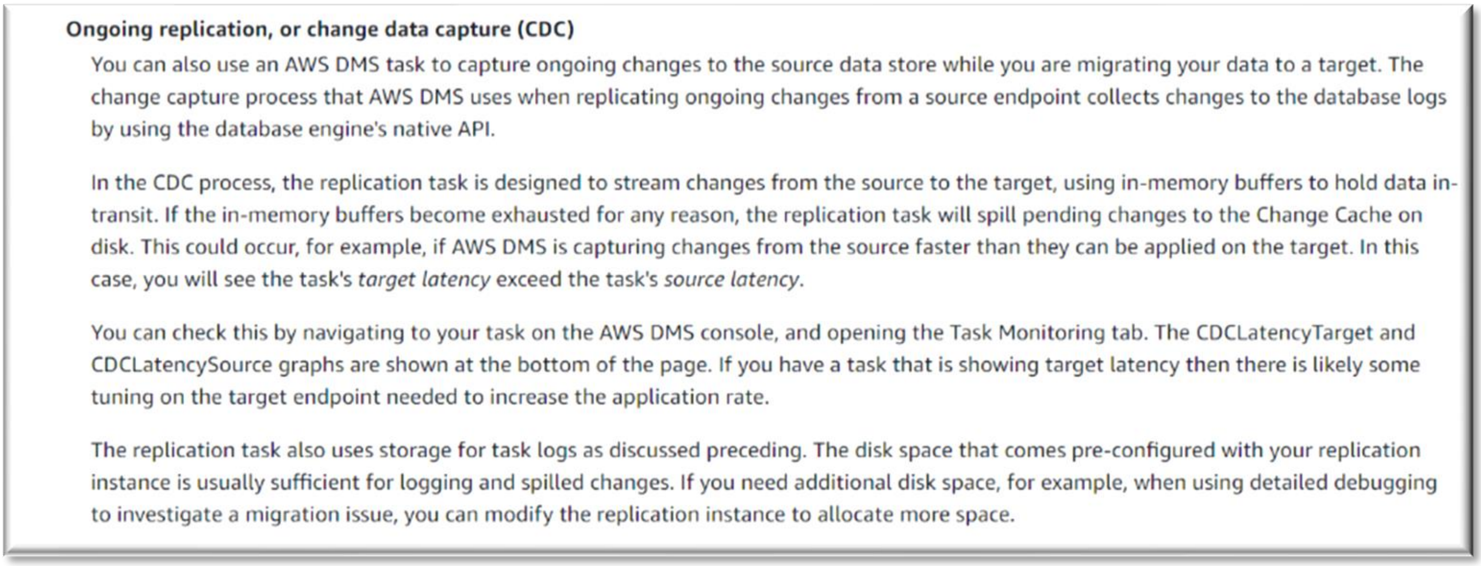
You use the task to migrate data from the source endpoint to the target endpoint, and the task processing is done on the replication instance. You specify what tables and schemas to migrate and any special processing, such as logging requirements, control table data, and error handling.

Conceptually, an AWS DMS replication task performs two distinct functions as shown in the diagram following.



The full load process is straight-forward to understand. Data is extracted from the source in a bulk extract manner and loaded directly into the target. You can specify the number of tables to extract and load in parallel on the AWS DMS console under **Advanced Settings**.

For more information about AWS DMS tasks, see Working with AWS DMS tasks.



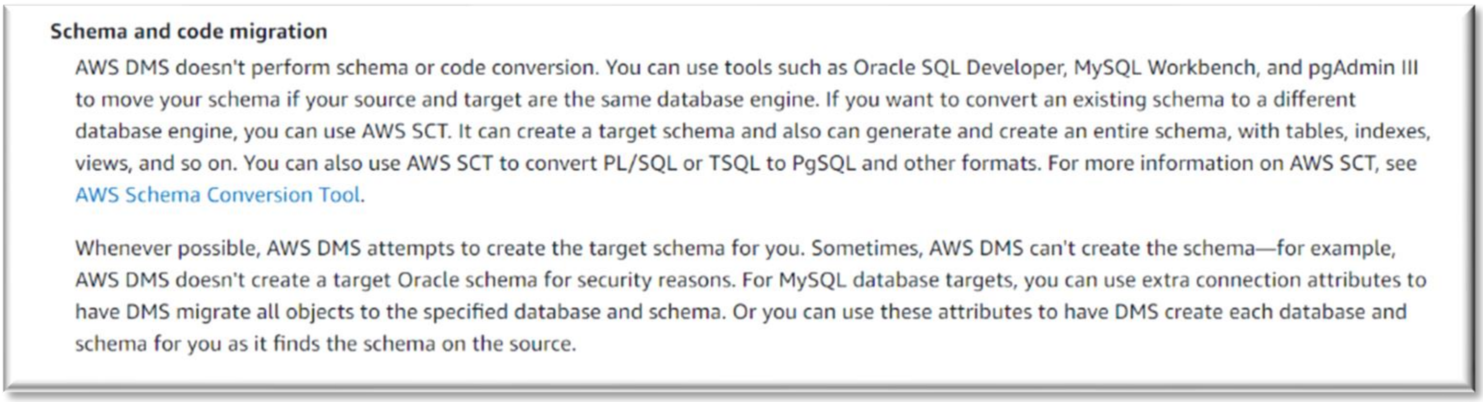
Ongoing replication, or change data capture (CDC)

You can also use an AWS DMS task to capture ongoing changes to the source data store while you are migrating your data to a target. The change capture process that AWS DMS uses when replicating ongoing changes from a source endpoint collects changes to the database logs by using the database engine's native API.

In the CDC process, the replication task is designed to stream changes from the source to the target, using in-memory buffers to hold data in-transit. If the in-memory buffers become exhausted for any reason, the replication task will spill pending changes to the Change Cache on disk. This could occur, for example, if AWS DMS is capturing changes from the source faster than they can be applied on the target. In this case, you will see the task's *target latency* exceed the task's *source latency*.

You can check this by navigating to your task on the AWS DMS console, and opening the Task Monitoring tab. The CDCLatencyTarget and CDCLatencySource graphs are shown at the bottom of the page. If you have a task that is showing target latency then there is likely some tuning on the target endpoint needed to increase the application rate.

The replication task also uses storage for task logs as discussed preceding. The disk space that comes pre-configured with your replication instance is usually sufficient for logging and spilled changes. If you need additional disk space, for example, when using detailed debugging to investigate a migration issue, you can modify the replication instance to allocate more space.



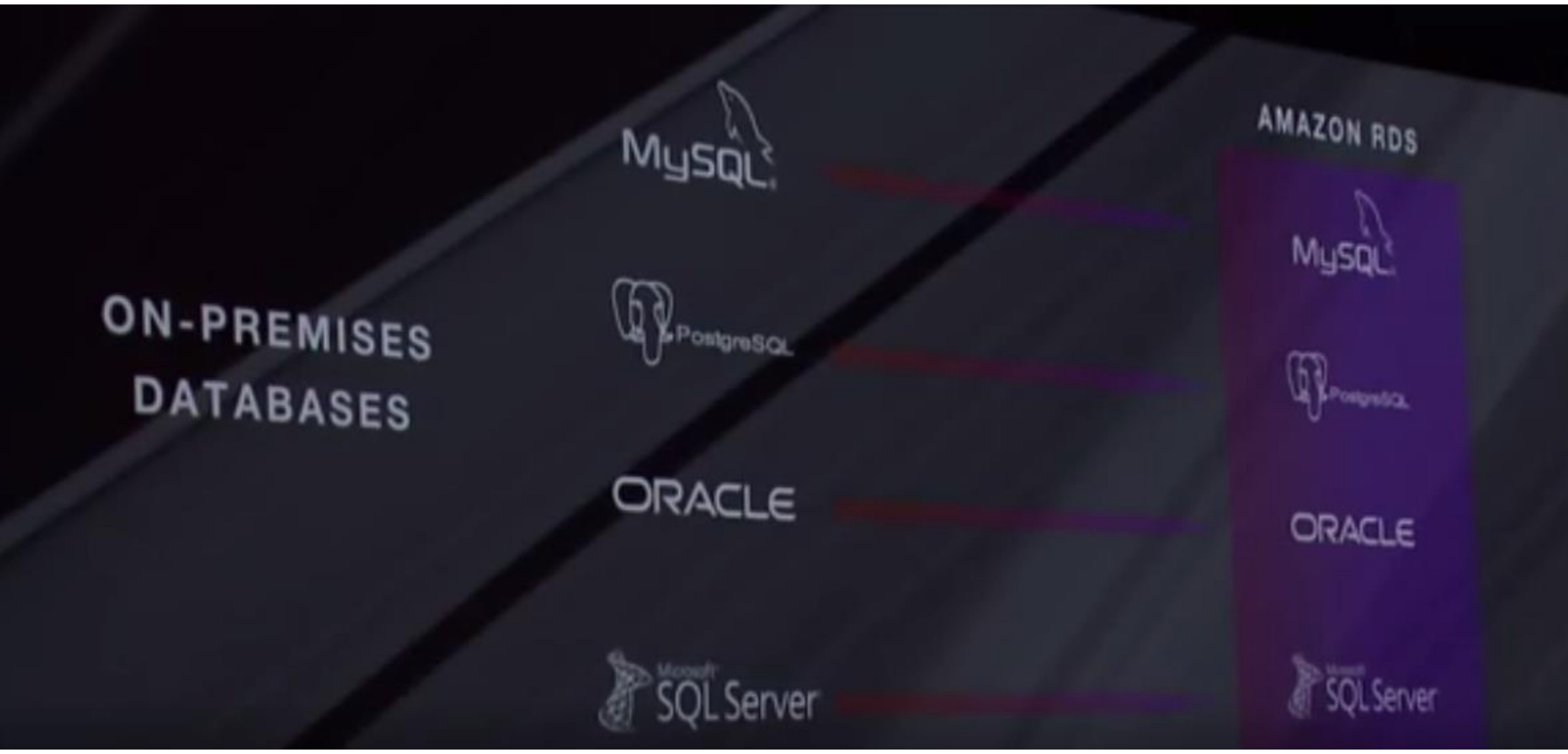
Schema and code migration

AWS DMS doesn't perform schema or code conversion. You can use tools such as Oracle SQL Developer, MySQL Workbench, and pgAdmin III to move your schema if your source and target are the same database engine. If you want to convert an existing schema to a different database engine, you can use AWS SCT. It can create a target schema and also can generate and create an entire schema, with tables, indexes, views, and so on. You can also use AWS SCT to convert PL/SQL or TSQL to PgSQL and other formats. For more information on AWS SCT, see AWS Schema Conversion Tool.

Whenever possible, AWS DMS attempts to create the target schema for you. Sometimes, AWS DMS can't create the schema—for example, AWS DMS doesn't create a target Oracle schema for security reasons. For MySQL database targets, you can use extra connection attributes to have DMS migrate all objects to the specified database and schema. Or you can use these attributes to have DMS create each database and schema for you as it finds the schema on the source.

Andy Hopper explaining the Database Migration Service (DMS)

Andy Hopper explaining the Database Migration Service (DMS)



This is a screenshot from Andy Jassy's announcement of DMS. Four database engines are displayed. The diagram shows them being moved onto Amazon RDS. Notice how they can continue to use that engine while on RDS (the logos on the righthand side of the image are the same as the lefthand).

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

Glossary

Commercial Database

Description of term here.

Open-source Database

Description of term here.

Homogeneous migration

Description of term here.

SCT

Stands for "Schema Conversion Tool".

DMS

Stands for Database Migration Service.

Bibliography

I. Official
II. Unofficial

III. Critical

IV. General

V. Official

https://aws.amazon.com/blogs/database/introducing-aws-schema-conversion-tool-version-1-0-502/

[Surname1]

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

[Surname1]

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

VI. Unofficial

[Davis 2022]

"Containers on AWS". *Medium*. Available at: https://neal-davis.medium.com/containers-on-aws-7160928c4339

[Surname1]

Fraser, Dominic (year). A beginner's guide to Amazon's Elastic Container Service. 20st May 2018. FreeCodeCamp. Available at: <https://www.freecodecamp.org/news/amazon-ecs-terms-and-architecture-807d8c4960fd/>.

[Nguyen 2017]

Nguyen, Teng (2017). "Gentle Introduction to how ECS works with example tutorial". 10th Sept 2017. Medium. Available at: https://medium.com/boltops/gentle-introduction-to-how-aws-ecs-works-with-example-tutorial-cea3d27ce63d

VII. Critical

[Surname1]

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

[Surname1]

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

VIII. General

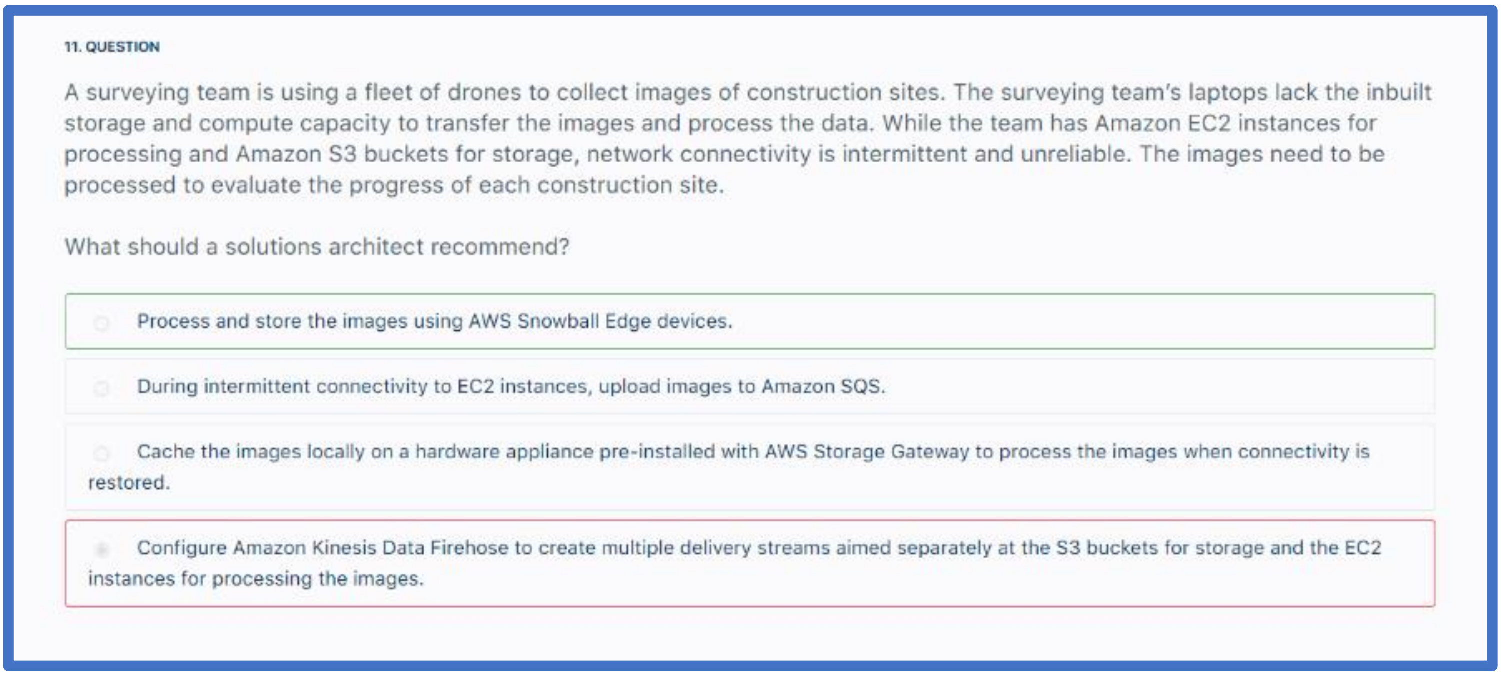
[Surname1]

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

[Surname1]

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

Snowball



11. QUESTION

A surveying team is using a fleet of drones to collect images of construction sites. The surveying team's laptops lack the inbuilt storage and compute capacity to transfer the images and process the data. While the team has Amazon EC2 instances for processing and Amazon S3 buckets for storage, network connectivity is intermittent and unreliable. The images need to be processed to evaluate the progress of each construction site.

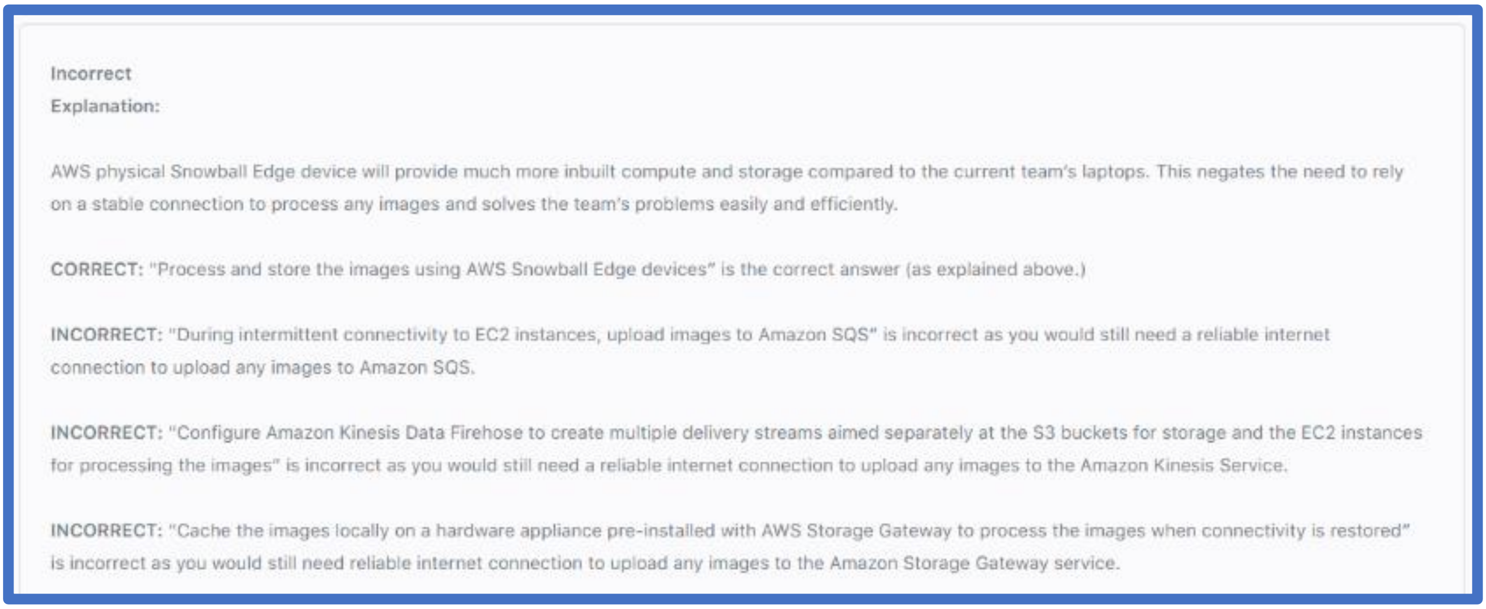
What should a solutions architect recommend?

○ Process and store the images using AWS Snowball Edge devices.

○ During intermittent connectivity to EC2 instances, upload images to Amazon SQS.

○ Cache the images locally on a hardware appliance pre-installed with AWS Storage Gateway to process the images when connectivity is restored.

● Configure Amazon Kinesis Data Firehose to create multiple delivery streams aimed separately at the S3 buckets for storage and the EC2 instances for processing the images.



Incorrect

Explanation:

AWS physical Snowball Edge device will provide much more inbuilt compute and storage compared to the current team's laptops. This negates the need to rely on a stable connection to process any images and solves the team's problems easily and efficiently.

CORRECT: "Process and store the images using AWS Snowball Edge devices" is the correct answer (as explained above.)

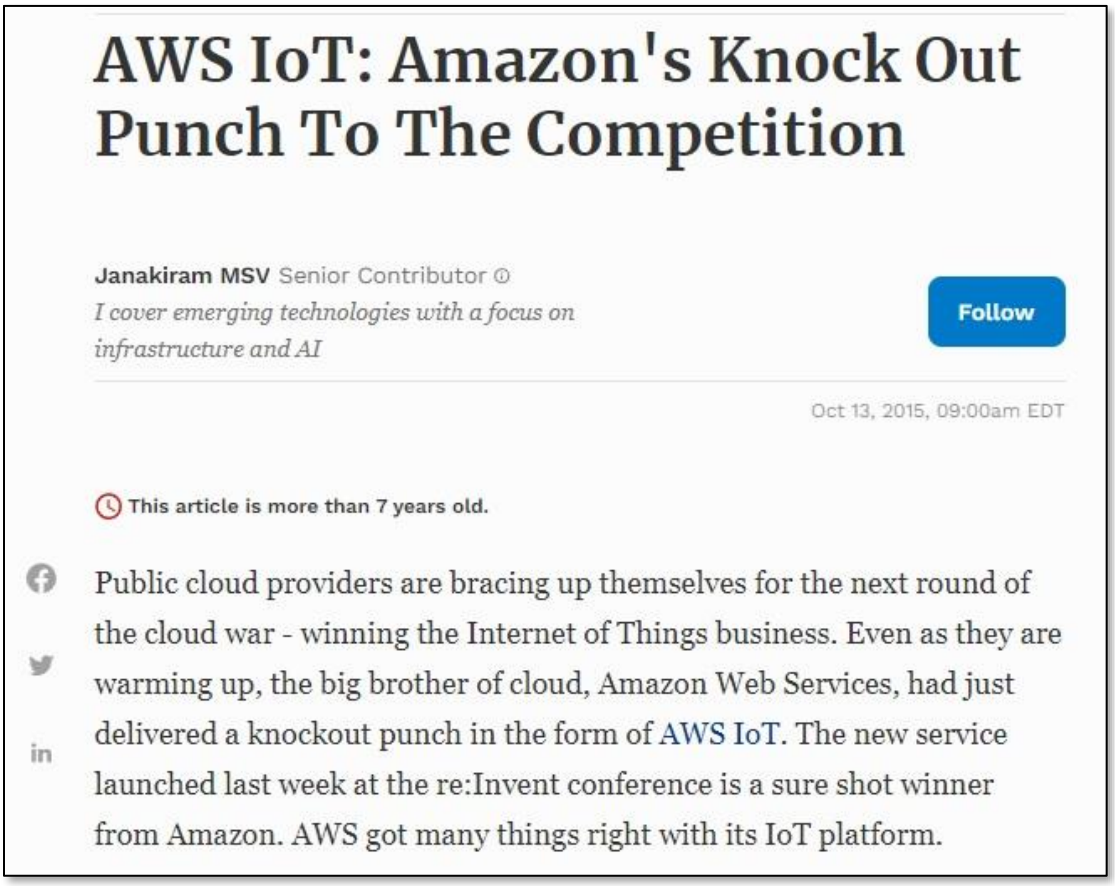
INCORRECT: "During intermittent connectivity to EC2 instances, upload images to Amazon SQS" is incorrect as you would still need a reliable internet connection to upload any images to Amazon SQS.

INCORRECT: "Configure Amazon Kinesis Data Firehose to create multiple delivery streams aimed separately at the S3 buckets for storage and the EC2 instances for processing the images" is incorrect as you would still need a reliable internet connection to upload any images to the Amazon Kinesis Service.

INCORRECT: "Cache the images locally on a hardware appliance pre-installed with AWS Storage Gateway to process the images when connectivity is restored" is incorrect as you would still need reliable internet connection to upload any images to the Amazon Storage Gateway service.

Elasticsearch Service

IoT Core



AWS IoT: Amazon's Knock Out Punch To The Competition

Janakiram MSV Senior Contributor

I cover emerging technologies with a focus on infrastructure and AI

Oct 13, 2015, 09:00am EDT

This article is more than 7 years old.

Public cloud providers are bracing up themselves for the next round of the cloud war - winning the Internet of Things business. Even as they are warming up, the big brother of cloud, Amazon Web Services, had just delivered a knockout punch in the form of AWS IoT. The new service launched last week at the re:Invent conference is a sure shot winner from Amazon. AWS got many things right with its IoT platform.

[Janakiram 2015]

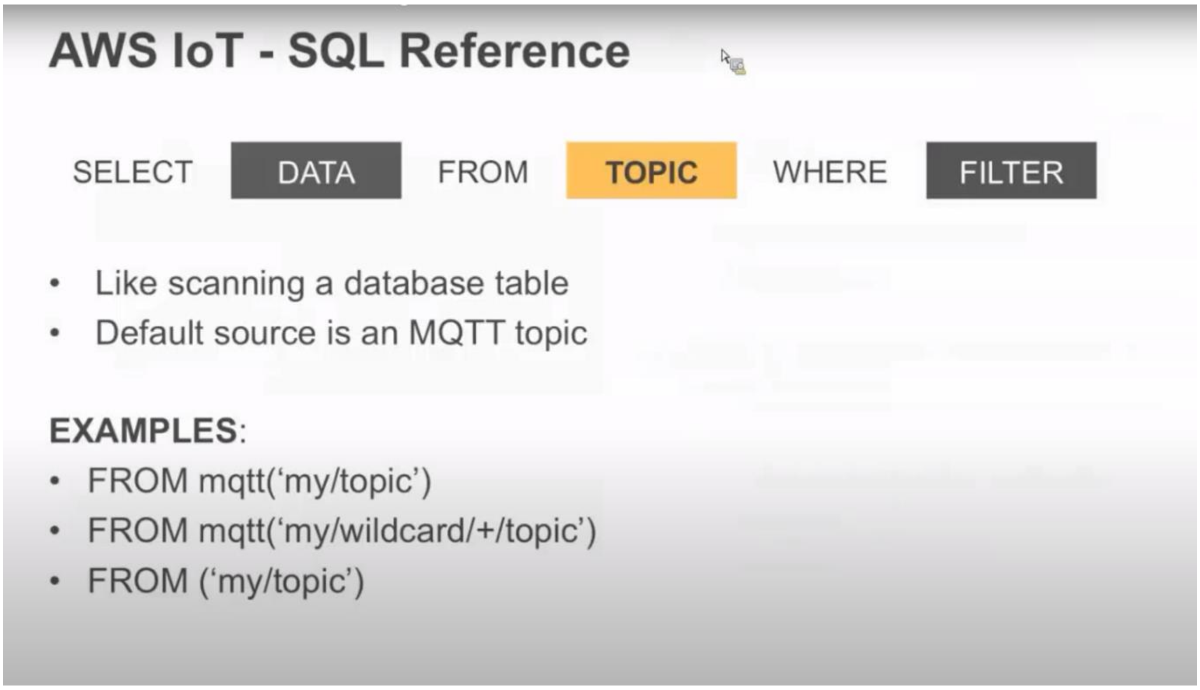
What does EIN stand for?

This is mentioned in the 2015 video by Kyle Roche.

What on earth is MQTT?

What is the Registry?

Why is it important to have a registry?



AWS IoT - SQL Reference

SELECT DATA FROM TOPIC WHERE FILTER

- Like scanning a database table
- Default source is an MQTT topic

EXAMPLES:

- FROM mqtt('my/topic')
- FROM mqtt('my/wildcard/+/topic')
- FROM ('my/topic')

From [Roche 2015]

Bibliography

I. Official
II. Unofficial
III. Critical

IV. General

I. Official

[Janakiram 2015]

Janakiram (2015). AWS IoT: Amazon's Knockout Punch to the Competition. *Forbes*. Oct 13th 2015. Available at: <https://www.forbes.com/sites/janakirammsv/2015/10/13/aws-iot-amazons-knock-out-punch-to-the-competition/>

II. Unofficial

https://paolopatierno.wordpress.com/2015/10/13/an-iot-platforms-match-microsoft-azure-iot-vs-amazon-aws-iot/

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8467692/

Great YouTube series introducing you to IoT

https://www.youtube.com/watch?v=rz73uG_ahzU&list=PL5_Rrj9tYQAml3cNxw4rlAJL-QyEoNG8b&ab_channel=MorethanCertified

https://www.forbes.com/sites/alexkonrad/2015/10/08/amazon-jumps-into-internet-of-things-frenzy-with-new-cloud-platform/?sh=2f23d82ade4d

III. Critical

IV. General

Cloud Adoption Framework (CAF)

AWS re:Invent

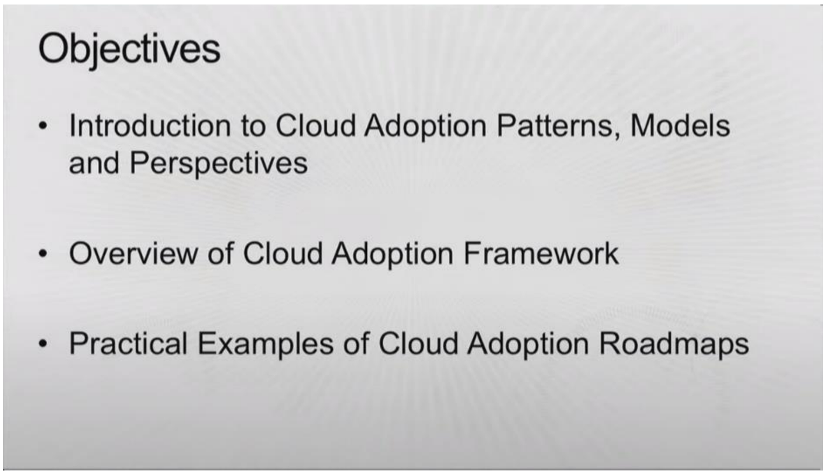
ENT305

Develop an Enterprise-wide Cloud Adoption Strategy

Miha Kralj, Principal Consultant, AWS Professional Services
Blake Chism, Senior Consultant, AWS Professional Services

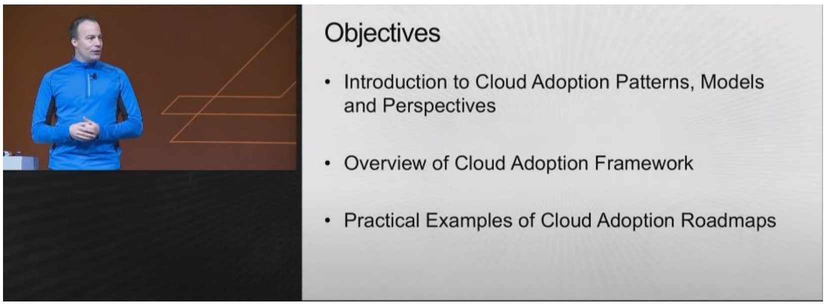
November 13, 2014 | Las Vegas, NV

amazon web services



Objectives

- Introduction to Cloud Adoption Patterns, Models and Perspectives
- Overview of Cloud Adoption Framework
- Practical Examples of Cloud Adoption Roadmaps

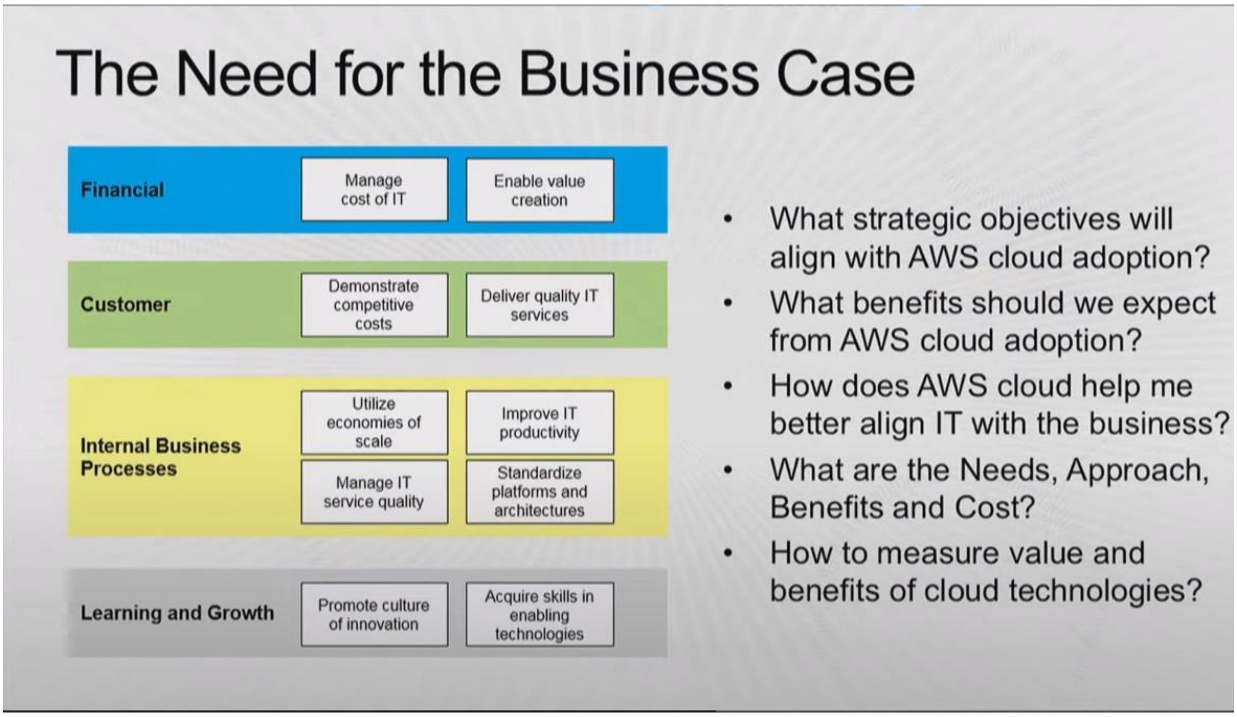


Objectives

- Introduction to Cloud Adoption Patterns, Models and Perspectives
- Overview of Cloud Adoption Framework
- Practical Examples of Cloud Adoption Roadmaps

Figure 1 Blake Chism speaking in 2014

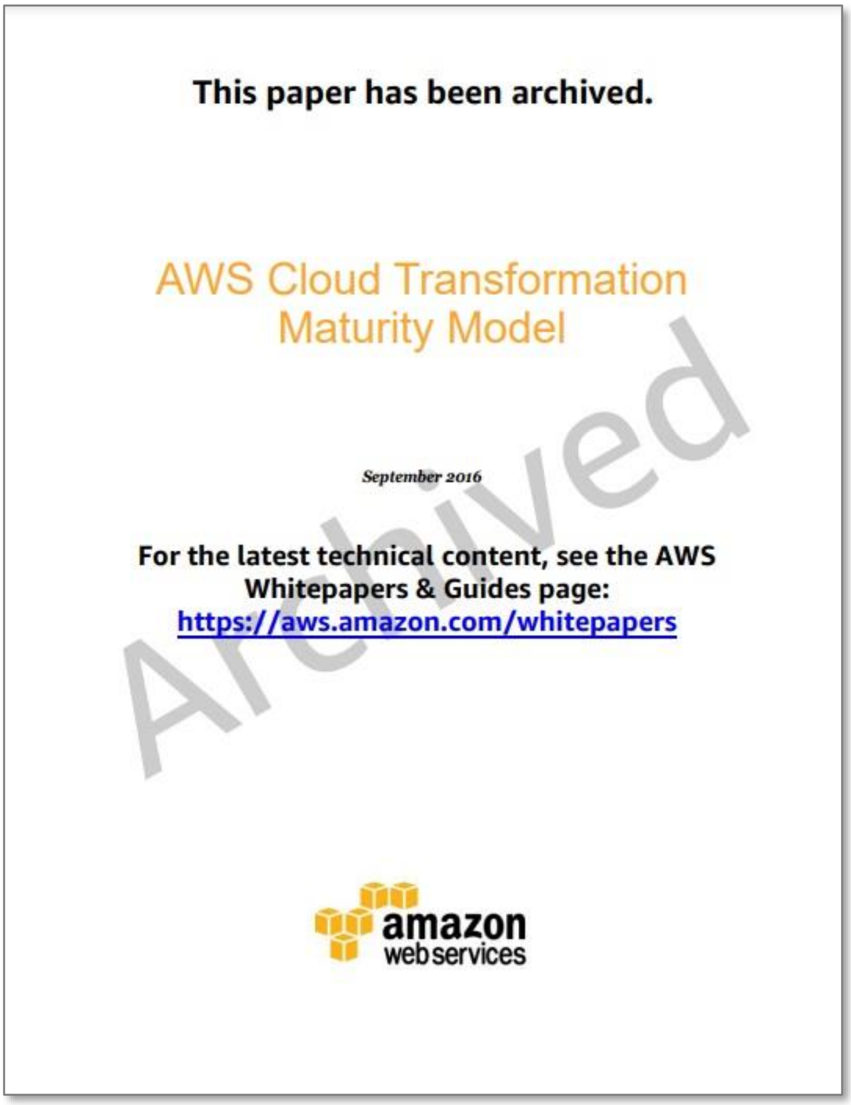
Figure 2 Miha Kralj speaking in 2014



The Need for the Business Case

Financial	Manage cost of IT	Enable value creation
Customer	Demonstrate competitive costs	Deliver quality IT services
Internal Business Processes	Utilize economies of scale	Improve IT productivity
	Manage IT service quality	Standardize platforms and architectures
Learning and Growth	Promote culture of innovation	Acquire skills in enabling technologies

- What strategic objectives will align with AWS cloud adoption?
- What benefits should we expect from AWS cloud adoption?
- How does AWS cloud help me better align IT with the business?
- What are the Needs, Approach, Benefits and Cost?
- How to measure value and benefits of cloud technologies?



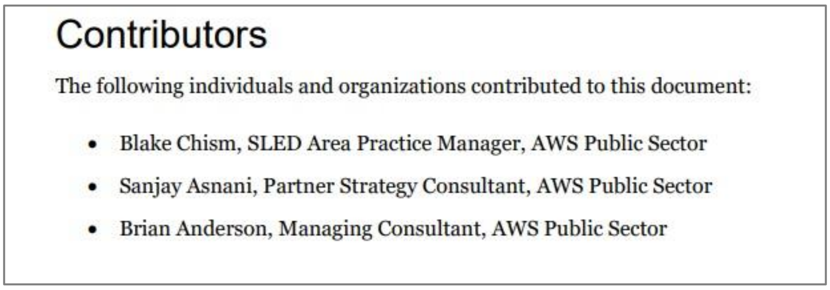
This paper has been archived.

AWS Cloud Transformation Maturity Model

September 2016

For the latest technical content, see the AWS Whitepapers & Guides page: https://aws.amazon.com/whitepapers

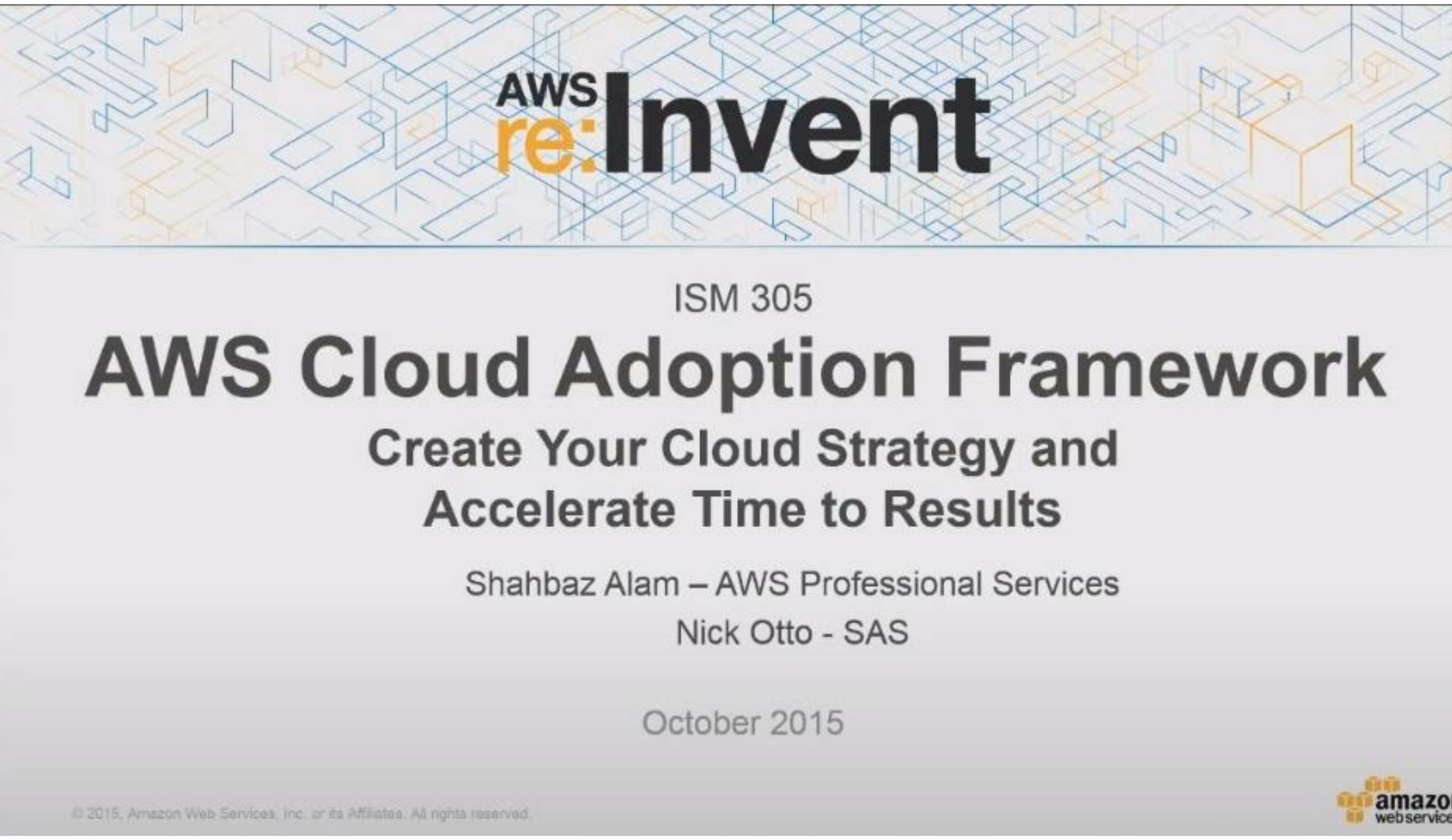
amazon web services



Contributors

The following individuals and organizations contributed to this document:

- Blake Chism, SLED Area Practice Manager, AWS Public Sector
- Sanjay Asnani, Partner Strategy Consultant, AWS Public Sector
- Brian Anderson, Managing Consultant, AWS Public Sector



AWS re:Invent

ISM 305

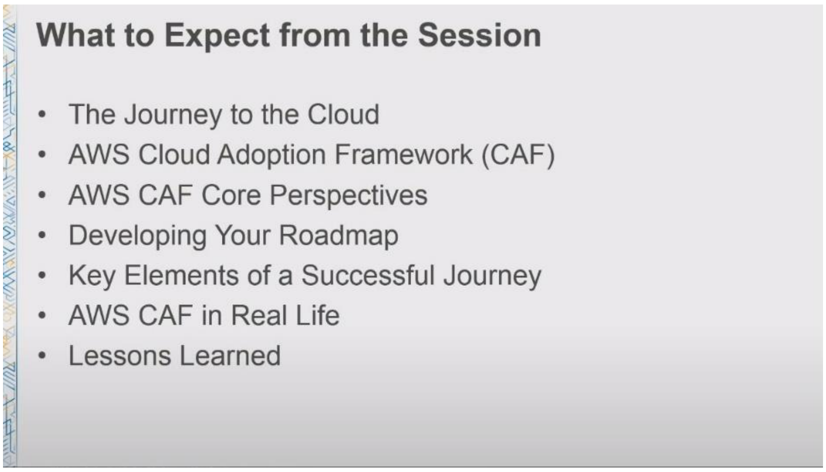
AWS Cloud Adoption Framework

Create Your Cloud Strategy and Accelerate Time to Results

Shahbaz Alam – AWS Professional Services

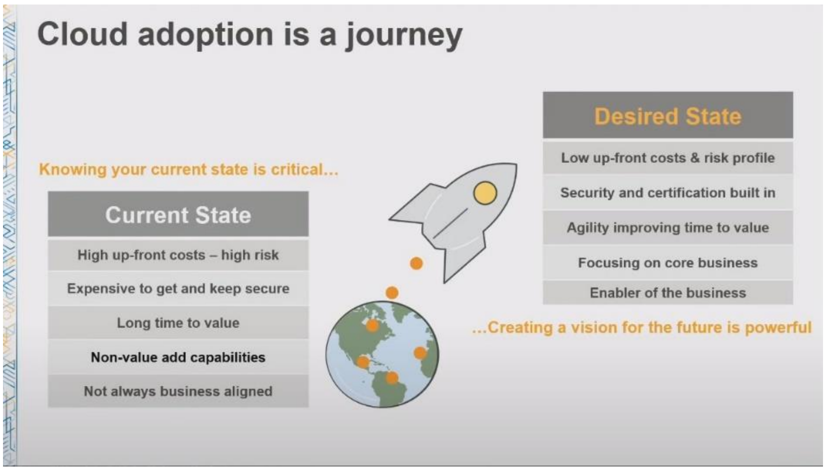
Nick Otto - SAS

October 2015



What to Expect from the Session

- The Journey to the Cloud
- AWS Cloud Adoption Framework (CAF)
- AWS CAF Core Perspectives
- Developing Your Roadmap
- Key Elements of a Successful Journey
- AWS CAF in Real Life
- Lessons Learned



Cloud adoption is a journey

Knowing your current state is critical…

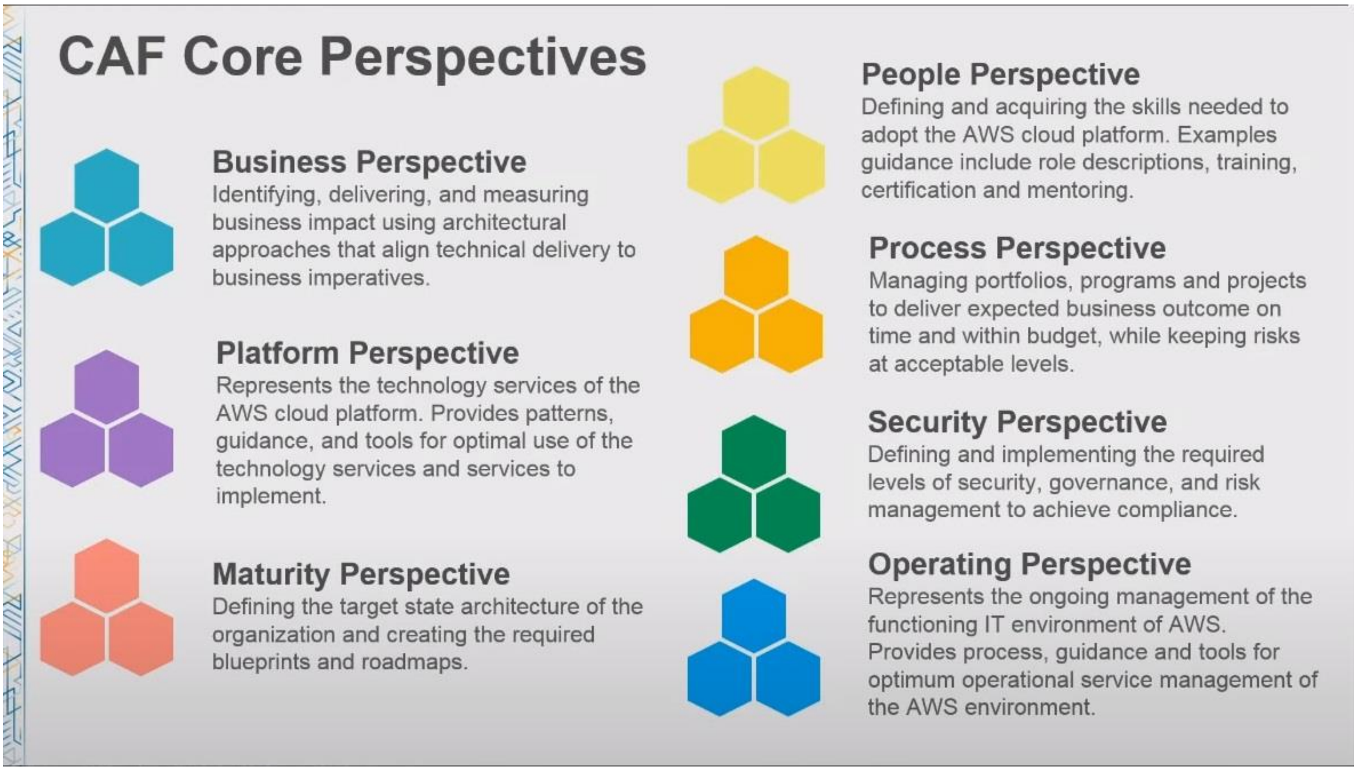
Current State

- High up-front costs – high risk
- Expensive to get and keep secure
- Long time to value
- Non-value add capabilities
- Not always business aligned

Desired State

- Low up-front costs & risk profile
- Security and certification built in
- Agility improving time to value
- Focusing on core business
- Enabler of the business

…Creating a vision for the future is powerful



CAF Core Perspectives

Business Perspective

Identifying, delivering, and measuring business impact using architectural approaches that align technical delivery to business imperatives.

Platform Perspective

Represents the technology services of the AWS cloud platform. Provides patterns, guidance, and tools for optimal use of the technology services and services to implement.

Maturity Perspective

Defining the target state architecture of the organization and creating the required blueprints and roadmaps.

People Perspective

Defining and acquiring the skills needed to adopt the AWS cloud platform. Examples guidance include role descriptions, training, certification and mentoring.

Process Perspective

Managing portfolios, programs and projects to deliver expected business outcome on time and within budget, while keeping risks at acceptable levels.

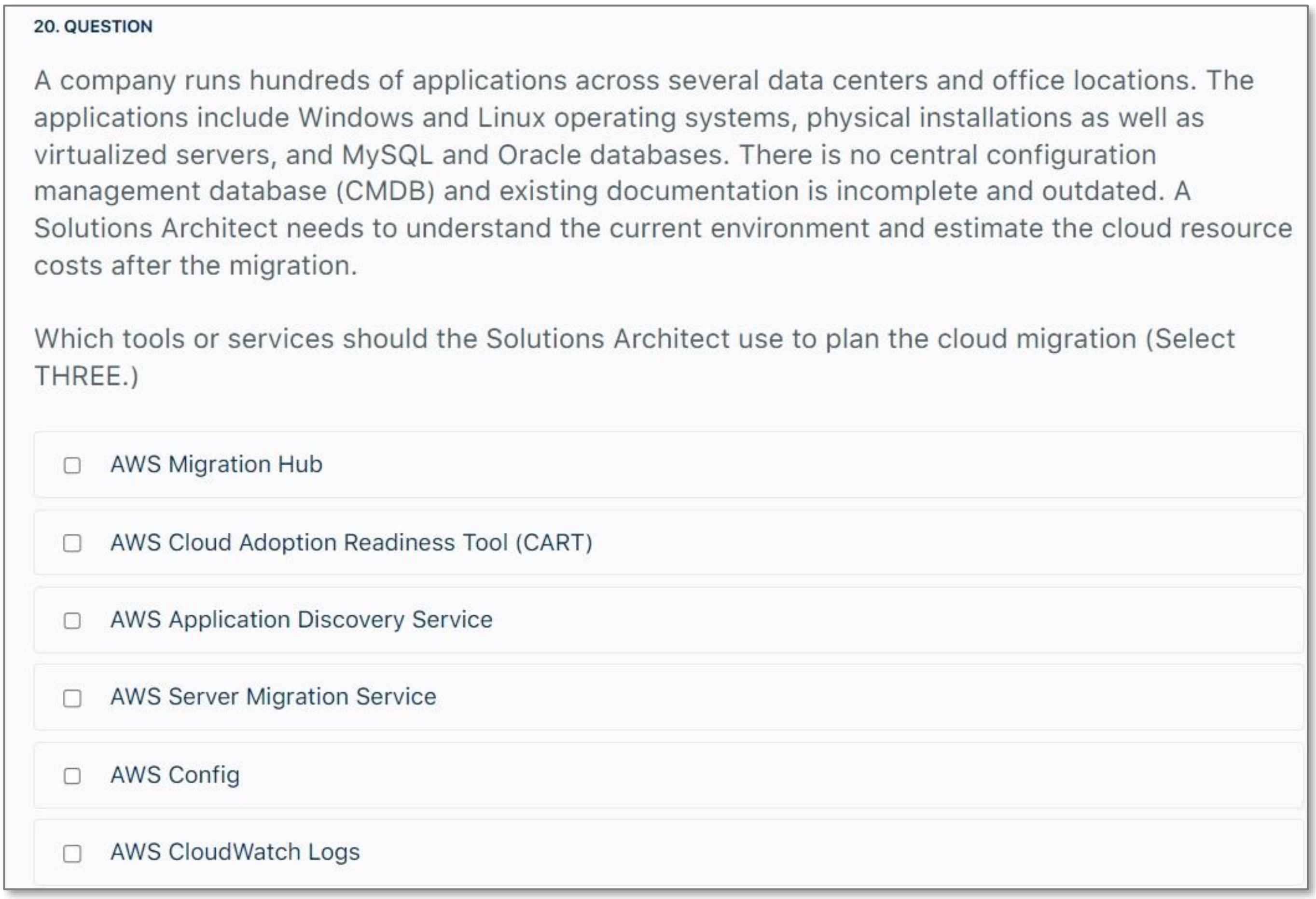
Security Perspective

Defining and implementing the required levels of security, governance, and risk management to achieve compliance.

Operating Perspective

Represents the ongoing management of the functioning IT environment of AWS. Provides process, guidance and tools for optimum operational service management of the AWS environment.

I came across the question below while training for the SA Professional examination. Note that it does not actually mention the Cloud Adoption Framework. It does, however, mention the Cloud Adoption Readiness Tool (CART).

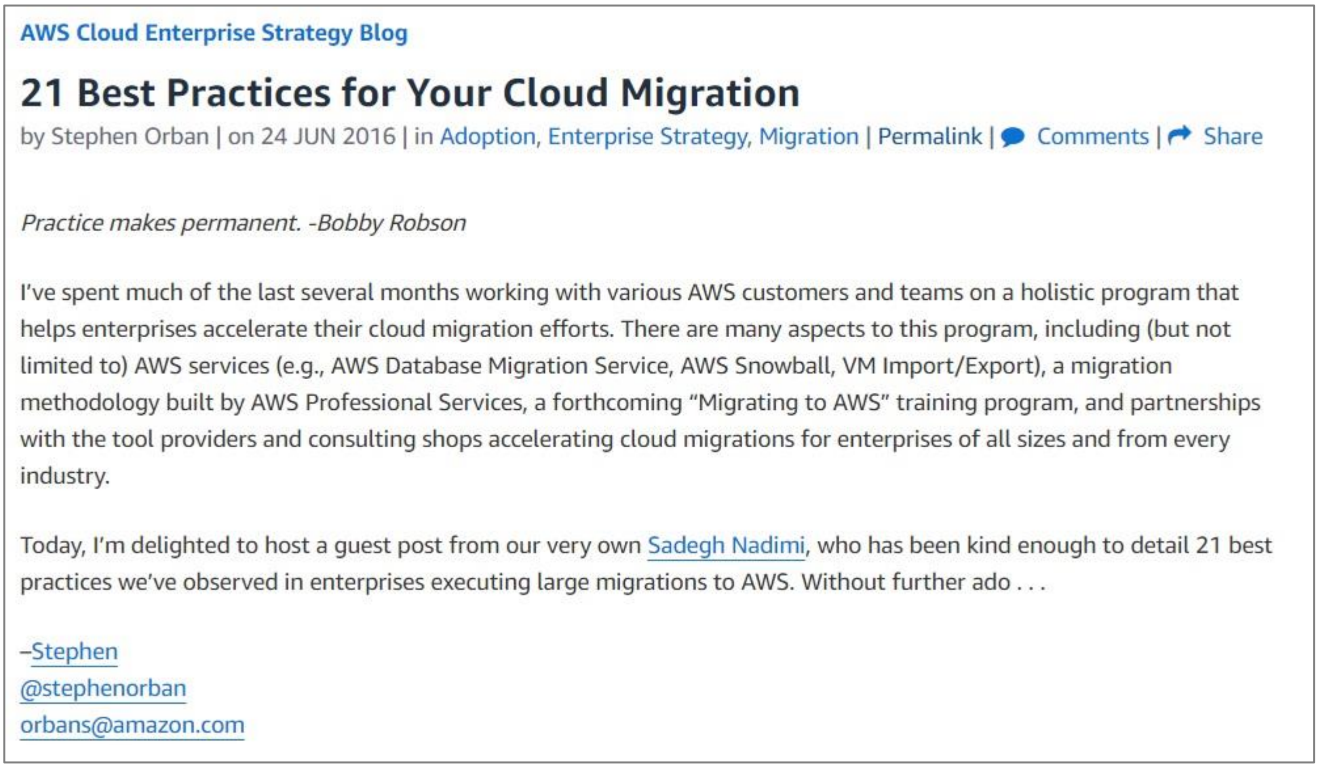


20. QUESTION

A company runs hundreds of applications across several data centers and office locations. The applications include Windows and Linux operating systems, physical installations as well as virtualized servers, and MySQL and Oracle databases. There is no central configuration management database (CMDB) and existing documentation is incomplete and outdated. A Solutions Architect needs to understand the current environment and estimate the cloud resource costs after the migration.

Which tools or services should the Solutions Architect use to plan the cloud migration (Select THREE.)

- □ AWS Migration Hub
- □ AWS Cloud Adoption Readiness Tool (CART)
- □ AWS Application Discovery Service
- □ AWS Server Migration Service
- □ AWS Config
- □ AWS CloudWatch Logs



AWS Cloud Enterprise Strategy Blog

21 Best Practices for Your Cloud Migration

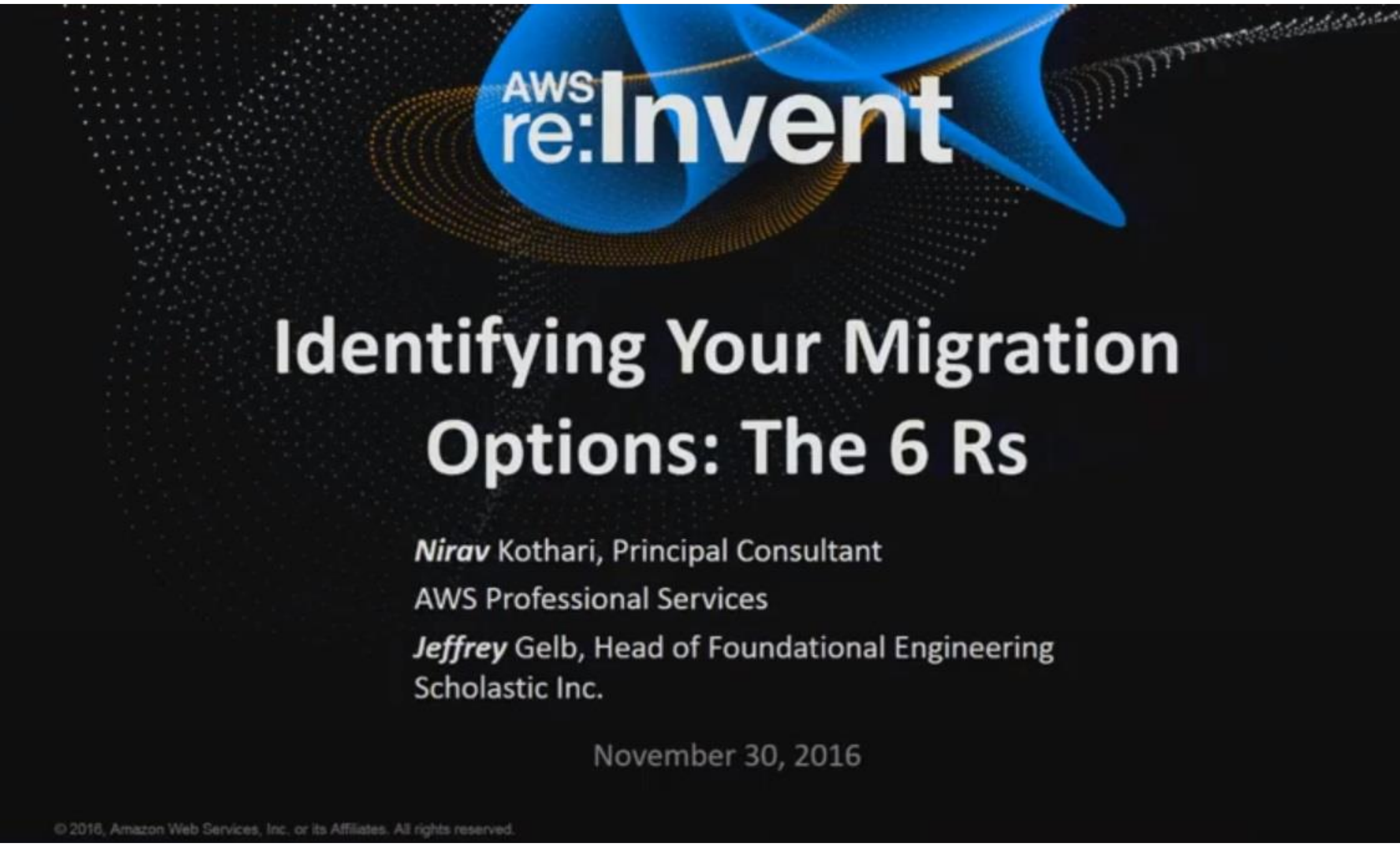
by Stephen Orban | on 24 JUN 2016 | in Adoption, Enterprise Strategy, Migration | Permalink | 💬 Comments | ➦ Share

Practice makes permanent. -Bobby Robson

I've spent much of the last several months working with various AWS customers and teams on a holistic program that helps enterprises accelerate their cloud migration efforts. There are many aspects to this program, including (but not limited to) AWS services (e.g., AWS Database Migration Service, AWS Snowball, VM Import/Export), a migration methodology built by AWS Professional Services, a forthcoming "Migrating to AWS" training program, and partnerships with the tool providers and consulting shops accelerating cloud migrations for enterprises of all sizes and from every industry.

Today, I'm delighted to host a guest post from our very own Sadegh Nadimi, who has been kind enough to detail 21 best practices we've observed in enterprises executing large migrations to AWS. Without further ado . . .

–Stephen
@stephenorban
orbans@amazon.com



AWS re:Invent

Identifying Your Migration Options: The 6 Rs

Nirav Kothari, Principal Consultant
AWS Professional Services
Jeffrey Gelb, Head of Foundational Engineering
Scholastic Inc.

November 30, 2016

Bibliography

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

I. Official

[Chism 2014]

Chism, Blake and Miha Kralj (2014). Develop an Enterprise-wide Cloud Adoption Strategy. Available at: <https://www.youtube.com/watch?v=wl3TsEpNlH0&ab_channel=AmazonWebServices>

[Alam 2015]

Alam, Shahbaz (2015). AWS Cloud Adoption Framework. YouTube Channel: Amazon Web Services. Available at: <https://www.youtube.com/watch?v=iLIOemi3wTg&ab_channel=AmazonWebServices>

[Chapple 2015]

A Framework for IT and Business Transformation. *ReInvent 2016* [Conference]. Available at: <https://www.youtube.com/watch?v=E-gEj8CkVs8&ab_channel=AmazonWebServices>

[Aylward 2016]

Aylward, Kevin (2016). Large-Scale AWS Migrations. *ReInvent 2016* [Conference]. YouTube Channel: Amazon Web Services. Available at: <https://www.youtube.com/watch?v=03y9lxtxD4g&ab_channel=AmazonWebServices>

[Blake 2016]

Chism, Blake and Sanjay Asnani and Brian Anderson (2016). AWS Cloud Transformation Maturity Model [Whitepaper]. September 2016.

[Becker 2020]

Becker, Mark (2020). Privacy conscious cloud migrations: mapping the AWS Cloud Adoption Framework to the NIST Privacy Framework. Available at: <https://aws.amazon.com/blogs/security/privacy-conscious-cloud-migrations-mapping-aws-cloud-adoption-framework-to-nist-privacy-framework/>

[Ivan 2019]

II. Unofficial

[Orban 2016]

Orban, Stephen (2016). 21 Best Practices for your Cloud Migration. Available at: <https://aws.amazon.com/blogs/enterprise-strategy/21-best-practices-for-your-cloud-migration/>

III. Critical

IV. General

AWS Cost and Usage Reports

Elastic Container Registry

Bibliography

I. Official

II. Unofficial

III. Critical

IV. General