

QUESTION 1

A company collects data for temperature, humidity, and atmospheric pressure in cities across multiple continents. The average volume of data that the company collects from each site daily is 500 GB. Each site has a high-speed Internet connection.

The company wants to aggregate the data from all these global sites as quickly as possible in a single Amazon S3 bucket. The solution must minimize operational complexity.

Which solution meets these requirements?

- A)** Turn on S3 Transfer Acceleration on the destination S3 bucket. Use multipart uploads to directly upload site data to the destination S3 bucket.
- B)** Upload the data from each site to an S3 bucket in the closest Region. Use S3 Cross-Region Replication to copy objects to the destination S3 bucket. Then remove the data from the origin S3 bucket.
- C)** Schedule AWS Snowball Edge Storage Optimized device jobs daily to transfer data from each site to the closest Region. Use S3 Cross-Region Replication to copy objects to the destination S3 bucket.
- D)** Upload the data from each site to an Amazon EC2 instance in the closest Region. Store the data in an Amazon Elastic Block Store (Amazon EBS) volume. At regular intervals, take an EBS snapshot and copy it to the Region that contains the destination S3 bucket. Restore the EBS volume in that Region.

Soru:

Bir şirket, birden fazla kıtadaki şehirlerde sıcaklık, nem ve atmosferik basınç verilerini topluyor. Şirketin her bir sahadan günlük topladığı ortalama veri miktarı 500 GB'tır. Her saha yüksek hızlı bir internet bağlantısına sahiptir.

Şirket, bu küresel sahalardan gelen verileri mümkün olan en hızlı şekilde tek bir Amazon S3 bucket'ında toplamak istiyor. Çözüm, operasyonel karmaşıklığı en aza indirmelidir.

Bu gereksinimleri hangi çözüm karşılar?

- A)** Hedef S3 bucket'ında S3 Transfer Acceleration özelliğini açın. Saha verilerini doğrudan hedef S3 bucket'ına çok parçalı yüklemeler (multipart upload) kullanarak yükleyin.
- B)** Verileri her bir sahadan, en yakın AWS Bölgesi'ndeki bir S3 bucket'ına yükleyin. Nesneleri hedef S3 bucket'ına kopyalamak için S3 Cross-Region Replication kullanın. Ardından kaynak S3 bucket'ındaki verileri silin.

C) Her sahadan en yakın Bölgeye veri aktarmak için günlük olarak AWS Snowball Edge Storage Optimized cihaz görevleri planlayın. Nesneleri hedef S3 bucket'ına kopyalamak için S3 Cross-Region Replication kullanın.

D) Verileri her bir sahadan, en yakın Bölgedeki bir Amazon EC2 örneğine yükleyin. Verileri bir Amazon EBS volume içinde depolayın. Düzenli aralıklarla bir EBS snapshot alın ve bunu hedef S3 bucket'ının bulunduğu Bölgeye kopyalayın. Ardından EBS volume'ünü bu Bölgeden geri yükleyin.

Soru Analizi:

Şirket, dünya genelinde birçok sahadan **günlük 500 GB** veri topluyor.

Her sahada **yüksek hızlı internet bağlantısı** var.

Amaç:

- Tüm verilerin **mümkün olan en hızlı şekilde**,
- **Tek bir S3 bucket'ında** toplanması,
- **En az operasyonel karmaşıklık** ile yapılması.

Bu bilgiler, çözümün **online, otomatik, yönetimi kolay** olması gerektiğini gösteriyor.

Seçenek Analizi:

A seçeneği analizi (Doğru Cevap)

S3 Transfer Acceleration, AWS'nin dünya çapındaki edge (uç) noktalarını kullanarak uzak bölgelerden S3'e veri yüklemeyi hızlandırır.

Avantajları:

- Global siteler için en hızlı çözüm.
- Operasyonel yük yok → Ek sistem kurma veya yönetme gerektirmez.
- Multipart Upload ile birleştiğinde 500 GB gibi büyük veriler sorunsuz ve hızlı aktarılır.
- Veriler doğrudan hedef S3 bucket'ına gider → ekstra kopya, fazladan bucket, cihaz gerektirmez.

Bu yüzden **hem hızlı hem de en az karmaşık** çözümüdür.

B seçeneği analizi

Bu çözüm sunları gerektirir:

- Her site için ayrı bölgesel S3 bucket'ları açmak
- Versiyonlama açmak

- Cross-Region Replication kurmak
- Replikasyonun bitmesini beklemek
- Kaynak bucket'taki verileri silmek

Çok fazla adım ve operasyonel yük vardır.

Ayrıca CRR **asenkron** çalışır → en hızlı yöntem değildir.

C seçeneği analizi

Snowball Edge cihazı:

- Genellikle internetin yavaş, pahalı veya güvenilmez olduğu durumlarda tercih edilir.
- Bu soruda sahaların **yüksek hızlı interneti** var → Snowball gereksiz ve yavaş.
- Her gün cihaz gönderme, alma, kurulum, takip etme büyük bir operasyonel yük oluşturur.

Bu sebeple gereksiz derecede karmaşık ve yavaştır.

D seçeneği analizi

Bu seçenek şu adımları içeriyor:

- EC2 kurma
- Veriyi EC2'ye yükleme
- EBS volume üzerinde saklama
- Snapshot alma
- Snapshot'ı başka bölgeye kopyalama
- Snapshot'tan volume restore etme

Bu süreç hem:

- **Çok yavaş,**
- **Çok pahalı,**
- **Aşırı karmaşık,**
- S3'e veri toplamak için mantıksızdır.

Bu nedenle uygun değildir.

Sonuç

Soru: "Dünya genelindeki sitelerden veriyi en hızlı şekilde toplamak istiyoruz ve karmaşıklık düşük olmalı."

Bu koşullarda **en doğru çözüm**:

- A — S3 Transfer Acceleration + Multipart Upload
-

QUESTION 2

A company needs the ability to analyze the log files of its proprietary application. The logs are stored in JSON format in an Amazon S3 bucket. Queries will be simple and will run on-demand. A solutions architect needs to perform the analysis with minimal changes to the existing architecture.

What should the solutions architect do to meet these requirements with the LEAST amount of operational overhead?

- A)** Use Amazon Redshift to load all the content into one place and run the SQL queries as needed.
- B)** Use Amazon CloudWatch Logs to store the logs. Run SQL queries as needed from the Amazon CloudWatch console.
- C)** Use Amazon Athena directly with Amazon S3 to run the queries as needed.
- D)** Use AWS Glue to catalog the logs. Use a transient Apache Spark cluster on Amazon EMR to run the SQL queries as needed.

Soru:

Bir şirket, kendisine ait özel bir uygulamanın günlük dosyalarını analiz etme ihtiyacı vardır. Günlükler (loglar), JSON formatında bir Amazon S3 bucket'ında saklanmaktadır. Sorgular basit olacak ve ihtiyaç duyulduğça çalıştırılacaktır. Bir çözüm mimarının, mevcut mimaride en az değişikliği yaparak bu analizleri gerçekleştirmesi gerekmektedir.

Mevcut en az operasyonel yönetim yüküyle bu gereksinimleri karşılamak için çözüm mimarı ne yapmalıdır?

- A)** Tüm içeriği tek bir yerde toplamak için Amazon Redshift kullanın ve SQL sorgularını ihtiyaç oldukça çalıştırın.
- B)** Günlükleri saklamak için Amazon CloudWatch Logs kullanın. SQL sorgularını Amazon CloudWatch konsolundan ihtiyaç oldukça çalıştırın.
- C)** Amazon Athena'yı doğrudan Amazon S3 ile kullanarak ihtiyaç oldukça sorguları çalıştırın.

D) Günlükleri kataloglamak için AWS Glue kullanın. SQL sorgularını çalıştırılmak için Amazon EMR üzerinde geçici bir Apache Spark kümesi kullanın.

Soru Analizi:

Şirketin logları:

- **JSON formatında,**
- **S3 üzerinde,**
- **Özel uygulama logları,**
- **Basit sorgular çalışacak,**
- **İhtiyaç duyulduğça (on-demand) çalıştırılacak,**
- **Mevcut mimaride minimum değişiklik hedefleniyor,**
- **Operasyonel yük en düşük seviyede olmalı.**

Bu bilgiler, herhangi bir altyapı yönetimi gerektirmeyen, S3'ü doğrudan sorgulayabilen, sunucusuz (serverless) bir servisin ideal olduğunu gösteriyor.

Seçenek Analizi:

C Seçeneği Analizi (Doğru Cevap)

“Amazon Athena’yı doğrudan Amazon S3 ile kullanarak ihtiyaç oldukça sorguları çalıştırın.”

- Athena **tamamen sunucusuzdur** → hiçbir küme, sunucu veya altyapı yönetimi yoktur.
- S3 üzerindeki **JSON verileri doğrudan sorgular.**
- Sorgular on-demand çalışır → kullanım oldukça ödeme.
- Mimari değişiklik gerekmez → loglar zaten S3'te.
- Operasyonel yük **en düşük seviyededir.**

Bu nedenle gereksinimlere **%100 uygun** ve en az kaynak gerektiren çözümdür.

A Seçeneği Analizi

“Amazon Redshift’e yükleyip SQL sorguları çalıştırın.”

- Logları Redshift’ye **yüklemek gereklidir** → ek ETL ve veri pipeline’i.
- Redshift kümesi **kontrol, ölçümleme, bakım** gereklidir.
- Basit ve on-demand sorgular için **aşırı büyük ve gereksiz maliyetli** bir çözümüdür.

- Mevcut mimariyi önemli ölçüde değiştirir.

Neden yanlış: Operasyonel yük fazladır ve S3 → Redshift yükleme süreci gerektirir.

B Seçeneği Analizi

“Logları CloudWatch Logs’ta saklayın ve SQL sorgularını CloudWatch’tan çalıştırın.”

- Logları S3’ten alıp CloudWatch Logs’a aktarmak için ek adımlar gereklidir.
- CloudWatch Logs Insights güçlü olsa da **S3’teki mevcut verileri kullanamaz**, önce taşıma şarttır.
- Ayrıca CloudWatch JSON log analizi yapabilir ama S3 üzerinde bırakılan mevcut mimariyi değiştirme gerekliliği vardır.

Neden yanlış: Mevcut logları başka bir servise taşımak zorunda bırakır → mimari değişir ve operasyonel yük artar.

D Seçeneği Analizi

“AWS Glue + Amazon EMR üzerinde geçici Spark kümesi kullanın.”

- Spark kümesi kurmak ve yönetmek operasyonel olarak ağırdır.
- EMR sunucusuz değildir, küme yönetimi gereklidir.
- Basit sorgular için **çok aşırı karmaşık** bir çözüm.
- Glue katalogu faydalıdır ama Athena da Glue katalogunu kullanır → yine de Spark kümesi gereksiz olur.

Neden yanlış: En karmaşık ve maliyetli seçenek.

Sonuç

En az mimari değişiklik

- **En düşük operasyonel yönetim**
- **S3’teki JSON logları doğrudan sorgulama**
- **On-demand çalışma**

Bu gereksinimlere uyan açık ara en iyi çözüm:

C — Amazon Athena + Amazon S3

QUESTION 3

A company uses AWS Organizations to manage multiple AWS accounts for different departments. The management account has an Amazon S3 bucket that contains project reports. The company wants to limit access to this S3 bucket to only users of accounts within the organization in AWS Organizations.

Which solution meets these requirements with the LEAST amount of operational overhead?

- A.** Add the aws:PrincipalOrgID global condition key with a reference to the organization ID to the S3 bucket policy.
- B.** Create an organizational unit (OU) for each department. Add the aws:PrincipalOrgPaths global condition key to the S3 bucket policy.
- C.** Use AWS CloudTrail to monitor the CreateAccount, InviteAccountToOrganization, LeaveOrganization, and RemoveAccountFromOrganization events. Update the S3 bucket policy accordingly.
- D.** Tag each user that needs access to the S3 bucket. Add the aws:PrincipalTag global condition key to the S3 bucket policy.

Soru:

Bir şirket, farklı departmanlara ait birden fazla AWS hesabını yönetmek için AWS Organizations kullanmaktadır. Yönetim hesabında proje raporlarını içeren bir Amazon S3 bucket bulunmaktadır.

Şirket, bu S3 bucket'a erişimi yalnızca AWS Organizations içindeki hesapların kullanıcılarıyla sınırlamak istemektedir.

En az operasyonel iş yüküyle bu gereksinimleri hangi çözüm karşılar?

- A.** S3 bucket politikasına, organization ID'ye referans veren aws:PrincipalOrgID global koşul anahtarını ekleyin.
- B.** Her departman için bir Organizational Unit (OU) oluşturun. S3 bucket politikasına aws:PrincipalOrgPaths global koşul anahtarını ekleyin.
- C.** CreateAccount, InviteAccountToOrganization, LeaveOrganization ve RemoveAccountFromOrganization olaylarını izlemek için AWS CloudTrail kullanın. S3 bucket politikasını buna göre güncelleyin.
- D.** S3 bucket'a erişmesi gereken her kullanıcıyı etiketleyin (tag). S3 bucket politikasına aws:PrincipalTag global koşul anahtarını ekleyin.

Soru Analizi:

Şirket:

- Birden fazla AWS hesabını **AWS Organizations** ile yönetiyor.
- Yönetim hesabında proje raporlarının olduğu bir **S3 bucket** var.
- İstek: Bu bucket'a erişimi **yalnızca aynı organizasyondaki hesapların kullanıcılarıyla sınırlamak**.
- Ek gereksinim: **En az operasyonel yük** (yani mümkün olan en az bakım, değişiklik, güncelleme).

Bu nedenle çözümün:

- AWS Organizations yapısı otomatik değilse bile çalışması,
- Hesap ekleme/çıkarma durumunda manuel güncelleme gerektirmemesi,
- Basit ve sürdürülebilir olması gereklidir.

Bu bilgiler ışığında ideal çözüm **bucket policy içinde Organizations bazlı bir kontrol kullanmaktadır**.

Seçenek Analizi:

A Seçeneği Analizi (Doğru Cevap)

“S3 bucket politikasına organization ID'ye referans veren aws:PrincipalOrgID global condition key ekleyin.”

- Bu koşul anahtarı, S3 bucket'a erişimi **yalnızca belirli bir Organization ID'ye ait tüm hesaplarla sınırlar**.
- Organizasyona yeni bir hesap eklendiğinde bucket policy'yi **güncellemek gerekmez**.
- OU'lar değişse bile (taşınsa, silinse, yeniden yapılandırılsa) erişim *hala Organization ID* üzerinden kısıtlandığı için hiçbir extra iş yükü oluşmaz.
- AWS tarafından en çok önerilen, en sade çözümüdür.

→ **En az operasyonel yük + en doğru mimari yaklaşım = A seçeneği.**

B Seçeneği Analizi

“Her departman için OU oluşturun ve aws:PrincipalOrgPaths kullanın.”

- OU yolunu (organizational path) referans almak, OU düzeni değişikçe policy'nin güncellenmesini gerektirir.
- OU yapısı sık değişiyorsa operasyonel yük artar.
- Organization ID kullanmaya göre daha karmaşık bir yaklaşımdır.

→ OU bağımlılığı nedeniyle A'ya göre daha fazla operasyonel yük → uygun değil.

C Seçeneği Analizi

“CloudTrail ile hesap ekleme/çıkarma olaylarını takip edin ve gerekli oldukça bucket policy'yi güncelleyin.”

- Hesap eklendikçe veya ayrıldıkça policy'nin **manuel veya otomatik** şekilde güncellenmesi gereklidir.
- Bu yaklaşım *sürekli işlem* gerektirir.
- S3 bucket policy'nin sık sık değiştirilmesi operasyonel açıdan karmaşıktır.

→ Yüksek operasyonel yük → istenen gereksinimin tam tersine gider.

D Seçeneği Analizi

“Erişim olacak her kullanıcıya tag ekleyin ve aws:PrincipalTag ile erişim kontrolü yapın.”

- **Kullanıcı bazlı yönetim** gerektirir → çok fazla manuel işlem.
- Yeni kullanıcı eklendikçe tag eklemek zorunda kalırsın.
- Ayrıca Organizations hesabı bazlı kısıtlama istenirken bu yaklaşım **kullanıcı odaklıdır** → yanlış tasarım.

→ Hem yanlış yaklaşım hem yüksek operasyonel yük.

Sonuç

S3 erişimini AWS Organizations içindeki hesaplarla sınırlamanın en kolay, en stabil ve en az operasyon gerektiren yolu:

A — aws:PrincipalOrgID kullanmak

QUESTION 4

An application runs on an Amazon EC2 instance in a VPC. The application processes logs that are stored in an Amazon S3 bucket. The EC2 instance needs to access the S3 bucket without connectivity to the internet.

Which solution will provide private network connectivity to Amazon S3?

- Create a gateway VPC endpoint to the S3 bucket.
- Stream the logs to Amazon CloudWatch Logs. Export the logs to the S3 bucket.
- Create an instance profile on Amazon EC2 to allow S3 access.

D. Create an Amazon API Gateway API with a private link to access the S3 endpoint.

Soru:

Bir uygulama, bir **Amazon EC2 instance** üzerinde bir VPC içinde çalışmaktadır.

Uygulama, bir **Amazon S3 bucket**'ında depolanan logları işlemektedir.

EC2 instance'ın internet bağlantısı olmadan S3 bucket'a erişmesi gerekmektedir.

Hangi çözüm, Amazon S3'e **özel ağ bağlantısı (private network)** sağlayacaktır?

A. S3 bucket için bir **gateway VPC endpoint** oluşturun.

B. Logları **Amazon CloudWatch Logs**'a gönderin. Logları daha sonra S3 bucket'a aktarın (export edin).

C. EC2 için bir **instance profile** oluşturun ve S3 erişimini sağlayın.

D. S3 endpoint'e erişmek için **Amazon API Gateway API**'si ile private link oluşturun.

Soru Analizi:

Durum:

- Uygulama **EC2 instance** üzerinde çalışıyor ve **VPC** içinde bulunuyor.
- **EC2, S3 bucket'taki logları işliyor.**
- **EC2'nin internet bağlantısı yok.**
- Ama EC2 S3'e erişmek zorunda.

Amaç:

- **Özel ağ (private network) üzerinden S3 erişimi sağlamak.**
- Internet gateway veya NAT kullanmadan, güvenli ve doğrudan bağlantı kurmak.
- Operasyonel olarak basit bir çözüm istiyoruz.

Seçenek Analizi:

 **A Seçeneği Analizi (Doğru Cevap)**

“S3 bucket için Gateway VPC Endpoint oluşturun”

- Gateway VPC Endpoint, VPC ile S3 arasında **AWS ağı üzerinden özel bağlantı** sağlar.
- EC2, internet olmadan S3'e bağlanabilir.
- Route table'a eklenir ve S3 trafiği doğrudan AWS ağı üzerinden yönlendirilir.

- **Minimum operasyonel yük**, güvenli ve AWS tarafından önerilen çözümdür.

→ **Sonuç:** Doğru ve en basit çözüm.

B Seçeneği Analizi

“Logları CloudWatch Logs'a gönderip S3'e export edin”

- Bu yöntem **doğrudan çözüm sunmaz**.
- EC2 hala S3'e erişmek zorunda.
- CloudWatch Logs ek bir katman ekler, ek operasyonel yük ve karmaşıklık getirir.
- Gereksiz ve dolaylı bir çözüm.

C Seçeneği Analizi

“EC2 instance profile kullanmak”

- IAM rolü (instance profile) **yalnızca yetkilendirme sağlar**, network bağlantısı sağlamaz.
- Internet olmadan S3'e erişemez.
- Bu yöntem tek başına yeterli değildir.

D Seçeneği Analizi

“API Gateway PrivateLink ile S3'e erişim”

- API Gateway PrivateLink, S3 için **gereksiz ve yanlış kullanım** olur.
- S3 için zaten Gateway VPC Endpoint kullanılabilir.
- Ek servis ve yönetim yükü getirir.
- Operasyonel yük artırır.

Sonuç

EC2 instance'ın internet bağlantısı olmadan **S3'e güvenli şekilde erişmesi** için en doğru ve en basit çözüm:

A — Gateway VPC Endpoint kullanmak

QUESTION 5

A company is hosting a web application on AWS using a single Amazon EC2 instance that stores user-uploaded documents in an Amazon EBS volume. For better scalability and availability, the company duplicated the architecture and created a second EC2

instance and EBS volume in another Availability Zone, placing both behind an Application Load Balancer. After completing this change, users reported that, each time they refreshed the website, they could see one subset of their documents or the other, but never all of the documents at the same time.

What should a solutions architect propose to ensure users see all of their documents at once?

- A. Copy the data so both EBS volumes contain all the documents
- B. Configure the Application Load Balancer to direct a user to the server with the documents
- C. Copy the data from both EBS volumes to Amazon EFS. Modify the application to save new documents to Amazon EFS
- D. Configure the Application Load Balancer to send the request to both servers. Return each document from the correct server

Soru:

Bir şirket, web uygulamasını AWS üzerinde **tek bir EC2 instance** kullanarak barındırıyor ve kullanıcıların yüklediği belgeleri **Amazon EBS** hacminde saklıyor.

Daha iyi ölçeklenebilirlik ve kullanılabilirlik için şirket, mimariyi çoğaltarak **başka bir Availability Zone**'da ikinci bir EC2 instance ve EBS hacmi oluşturdu ve her ikisini de bir **Application Load Balancer (ALB)** arkasına yerleştirdi.

Bu değişiklikten sonra kullanıcılar şikayet etti: **her siteyi yenilediklerinde belgelerin yalnızca bir alt kümesini görüyorlar**, tüm belgeleri aynı anda göremiyorlar.

Kullanıcıların **tüm belgeleri her zaman görebilmesini** sağlamak için bir çözüm mimarının ne önermesi gereklidir?

- A. Verileri kopyalayın, böylece her iki EBS hacmi de tüm belgeleri içersin.
- B. Application Load Balancer'ı, kullanıcıyı belgelerin bulunduğu sunucuya yönlendirecek şekilde yapılandırın.
- C. Her iki EBS hacmindeki verileri **Amazon EFS**'ye kopyalayın. Uygulamayı, yeni belgeleri **Amazon EFS**'ye kaydedecek şekilde değiştirin.
- D. Application Load Balancer'ı istekleri her iki sunucuya da gönderecek şekilde yapılandırın. Her belgeyi doğru sunucudan döndürün.

Soru Analizi:

Durum:

- Başlangıçta web uygulaması **tek bir EC2 + EBS** üzerinde çalışıyor.

- Kullanıcı belgeleri EBS hacminde saklanıyor.
- Daha yüksek **ölçeklenebilirlik ve kullanılabilirlik** için:
 - İkinci bir EC2 ve EBS oluşturulmuş,
 - Her ikisi **Application Load Balancer (ALB)** arkasına konmuş.
- Kullanıcılar **sayfayı yenilediklerinde belgelerin yalnızca bir kısmını görüyor**, diğer instance'a yüklenen belgeleri göremiyor.

Neden:

- Her EC2 instance'ın kendi **EBS hacmi** var → veriler **senkronize değil**.
- ALB farklı instance'a yönlendirdiğinde kullanıcı yalnızca o instance'ta mevcut belgeleri görüyor.

Amaç:

- Kullanıcılar **tüm belgeleri her zaman görebilmeli**.
- Çözüm: **paylaşımlı ve senkron depolama** kullanmak.

Seçenek Analizi:

 **C Seçeneği Analizi (Doğru Cevap)**

“EBS'teki verileri Amazon EFS'ye kopyalayın ve uygulamayı yeni belgeleri EFS'ye kaydedecek şekilde değiştirin”

- **Amazon EFS (Elastic File System):**
 - Birden fazla EC2 instance tarafından **aynı anda erişilebilir**.
 - Dosya sistemi tabanlı **paylaşımlı depolama** sağlar.
 - Yüksek ölçeklenebilirlik ve kullanılabilirlik sunar.
- Tüm kullanıcı belgeleri **tek bir ortak dosya sistemi üzerinde toplanır** → tüm EC2 instance'larından erişilebilir.
- **Veri senkronizasyonu otomatik** sağlanır, operasyonel yük minimumdur.

→ **Sonuç:** Tüm gereksinimleri karşılayan, en basit ve AWS tarafından önerilen çözüm.

 **A Seçeneği Analizi**

“Her iki EBS hacmi de tüm belgeleri içerecek şekilde kopyalama yapın”

- Manuel veya sürekli **veri kopyalama** gereklidir.

- Yeni belgeler yüklenince **her iki EBS'e de kopya almak gereklidir** → operasyonel yük çok yüksek.
- **EBS tek bir instance'a bağlanabilir**, aynı anda iki instance kullanmak mümkün değil.

→ **Yanlış:** Operasyonel olarak sürdürülemez.

B Seçeneği Analizi

“ALB’yi kullanıcıyı belgelerin bulunduğu sunucuya yönlendirecek şekilde yapılandırın”

- Sticky session ile yönlendirme yapılabilir, ama:
 - Kullanıcı yeni belgeleri farklı instance’da yükleyebilir → hâlâ eksik belgeler görülebilir.
 - Paylaşımlı depolama problemi çözülmmez.

→ **Yanlış:** Yalnızca kısmi çözüm sağlar.

D Seçeneği Analizi

“ALB istekleri her iki sunucuya da gönderir ve belgeleri doğru sunucudan döndürür”

- ALB böyle bir davranışını desteklemeyez.
- Uygulama düzeyinde karmaşık ve hataya açık bir çözüm olur.
- Yönetimi ve bakımı zor, pratik değil.

Sonuç

- Kullanıcıların her zaman tüm belgeleri görebilmesi için **paylaşımlı ve senkron bir dosya sistemi** şarttır.
- AWS’de bunun için **Amazon EFS en uygun çözüm**dur.

C — EBS → Amazon EFS’ye geçiş

QUESTION 6

A company uses NFS to store large video files in on-premises network attached storage. Each video file ranges in size from 1 MB to 500 GB. The total storage is 70 TB and is no longer growing. The company decides to migrate the video files to Amazon S3. The company must migrate the video files as soon as possible while using the least possible network bandwidth.

Which solution will meet these requirements?

- A.** Create an S3 bucket. Create an IAM role that has permissions to write to the S3 bucket. Use the AWS CLI to copy all files locally to the S3 bucket.
- B.** Create an AWS Snowball Edge job. Receive a Snowball Edge device on premises. Use the Snowball Edge client to transfer data to the device. Return the device so that AWS can import the data into Amazon S3.
- C.** Deploy an S3 File Gateway on premises. Create a public service endpoint to connect to the S3 File Gateway. Create an S3 bucket. Create a new NFS file share on the S3 File Gateway. Point the new file share to the S3 bucket. Transfer the data from the existing NFS file share to the S3 File Gateway.
- D.** Set up an AWS Direct Connect connection between the on-premises network and AWS. Deploy an S3 File Gateway on premises. Create a public virtual interface (VIF) to connect to the S3 File Gateway. Create an S3 bucket. Create a new NFS file share on the S3 File Gateway. Point the new file share to the S3 bucket. Transfer the data from the existing NFS file share to the S3 File Gateway.

Soru:

Bir şirket, on-premises (şirket içi) network attached storage (NAS) üzerinde büyük video dosyalarını NFS ile saklamaktadır. Her bir video dosyası 1 MB ile 500 GB arasında değişmektedir. Toplam depolama 70 TB'tır ve artık büyümemektedir. Şirket, video dosyalarını Amazon S3'e taşımaya karar vermiştir.

Şirket, video dosyalarını **mümkün olan en kısa sürede ve en az ağ bant genişliği kullanarak** taşımak istiyor.

Bu gereksinimleri karşılayacak çözüm hangisidir?

- A.** Bir S3 bucket oluşturun. S3 bucket'a yazma izni olan bir IAM rolü oluşturun. AWS CLI kullanarak tüm dosyaları yerel olarak S3 bucket'a kopyalayın.
- B.** Bir AWS Snowball Edge işi oluşturun. Snowball Edge cihazını şirketinize gönderin. Snowball Edge istemcisini kullanarak verileri cihaza aktarın. Cihazı AWS'ye geri gönderin ve verilerin Amazon S3'e aktarılmasını sağlayın.
- C.** Şirket içinde bir S3 File Gateway kurun. Bağlanmak için bir public service endpoint oluşturun. Bir S3 bucket oluşturun. S3 File Gateway üzerinde yeni bir NFS file share oluşturun ve bu file share'i S3 bucket'a yönlendirin. Mevcut NFS file share'den verileri S3 File Gateway'e aktarın.
- D.** Şirket ağı ile AWS arasında bir AWS Direct Connect bağlantısı kurun. Şirket içinde bir S3 File Gateway dağıtin. S3 File Gateway'e bağlanmak için public virtual interface (VIF) oluşturun. Bir S3 bucket oluşturun. S3 File Gateway üzerinde yeni bir NFS file share oluşturun ve bu file share'i S3 bucket'a yönlendirin. Mevcut NFS file share'den verileri S3 File Gateway'e aktarın.

Soru Analizi:

Durum:

- Şirket, **on-premises NFS tabanlı NAS** üzerinde toplam **70 TB** video dosyası saklıyor.
- Dosya boyutları **1 MB – 500 GB** arasında değişiyor.
- Depolama artık büyümüyor → veri miktarı sabit.
- Şirket, bu dosyaları **Amazon S3'e** taşımak istiyor.

Gereksinimler:

- Dosyalar **mümkün olan en kısa sürede** taşınmalı.
- Ağ bant genişliği minimum** kullanılmalı.

Notlar:

- Ağ bant genişliği sınırlıysa büyük veri transferi **internet üzerinden doğrudan CLI ile** uzun sürebilir.
- Veri miktarı 70 TB gibi büyük olduğundan **physical data transfer** (Snowball Edge) veya **lokal caching / gateway** kullanmak daha mantıklı.

Seçenek Analizi:

B Seçeneği Analizi (Doğru Cevap)

“Bir AWS Snowball Edge işi oluşturun. Snowball Edge cihazını şirketinize gönderin. Snowball Edge istemcisini kullanarak verileri cihaza aktarın. Cihazı AWS'ye geri gönderin ve verilerin Amazon S3'e aktarılmasını sağlayın.”

- AWS Snowball Edge**, büyük veri (10 TB–PB ölçüğinde) taşımak için optimize edilmiş **offline fiziksel cihaztır**.
- Ağ üzerinden taşımaya gerek kalmadan **veriler cihaza kopyalanır**, cihaz AWS'ye gönderilir ve veriler S3'e yüklenir.
- 70 TB veri için en hızlı ve en az ağ kullanımıyla taşıma yöntemi** budur.
- Büyük veri transferinde AWS tarafından önerilen çözüm.

→ **Sonuç:** Gereksinimleri tam karşılar: hızlı ve düşük ağ kullanımı.

A Seçeneği Analizi

“AWS CLI ile tüm dosyaları doğrudan S3'e kopyalayın”

- Avantaj: Kolay, ek cihaz gerekmeyez.

- Dezavantaj: 70 TB'lık veri **internet veya VPN üzerinden taşınacak** → çok uzun sürer.
- Ağ bant genişliği yüksek kullanım gerektirir → sorunun gereksinimlerini karşılamaz.

C Seçeneği Analizi

“S3 File Gateway kullanarak verileri S3'e aktarın”

- Avantaj: Dosyalar NFS üzerinden **lokal cache ile S3'e yansıtılır**.
- Dezavantaj: **70 TB tek seferde** File Gateway ile taşımak zaman alır ve **internet bant genişliği yoğun kullanılır**.
- Hedef “mükemmel olan en kısa sürede” olduğundan büyük veri için **Snowball Edge daha uygundur**.

D Seçeneği Analizi

“Direct Connect ile S3 File Gateway'e aktarın”

- Avantaj: Direct Connect üzerinden hızlı veri transferi yapılabilir.
- Dezavantaj: Direct Connect kurmak **zaman alıcı ve maliyetli**.
- 70 TB veri için kurulum süresi ve maliyet yüksek → “en kısa sürede” şartını sağlamaz.

Sonuç

- **Hızlı veri aktarımı** ve **minimum ağ kullanımı** için **AWS Snowball Edge** idealdir.
- Dosya boyutları büyük ve toplam veri 70 TB olduğundan **CLI veya File Gateway internet üzerinden taşıma** çözümü yavaş olur.

Doğru Cevap: B — AWS Snowball Edge kullanmak

QUESTION 7

A company has an application that ingests incoming messages. Dozens of other applications and microservices then quickly consume these messages. The number of messages varies drastically and sometimes increases suddenly to 100,000 each second. The company wants to decouple the solution and increase scalability. Which solution meets these requirements?

- A. Persist the messages to Amazon Kinesis Data Analytics. Configure the consumer applications to read and process the messages.

B. Deploy the ingestion application on Amazon EC2 instances in an Auto Scaling group to scale the number of EC2 instances based on CPU metrics.

C. Write the messages to Amazon Kinesis Data Streams with a single shard. Use an AWS Lambda function to preprocess messages and store them in Amazon DynamoDB. Configure the consumer applications to read from DynamoDB to process the messages.

D. Publish the messages to an Amazon Simple Notification Service (Amazon SNS) topic with multiple Amazon Simple Queue Service (Amazon SQS) subscriptions. Configure the consumer applications to process the messages from the queues.

Soru:

Bir şirket, gelen mesajları alan bir uygulamaya sahiptir. Daha sonra onlarca başka uygulama ve mikro hizmet bu mesajları hızlı bir şekilde tüketir. Mesaj sayısı büyük ölçüde değişir ve bazen saniyede 100.000'e kadar artabilir. Şirket, çözümü bağımsız hale getirmek (decouple) ve ölçeklenebilirliği artırmak istiyor.

Bu gereksinimleri karşılayan çözüm hangisidir?

A. Mesajları Amazon Kinesis Data Analytics'e kaydedin. Tüketici uygulamaları mesajları okuyup işlemek için yapılandırın.

B. Ingestion uygulamasını Auto Scaling grubundaki Amazon EC2 instance'larına dağıtin. EC2 instance sayısını CPU metriklerine göre ölçeklendirin.

C. Mesajları tek bir shard ile Amazon Kinesis Data Streams'e yazın. Mesajları ön işleme almak için bir AWS Lambda işlevi kullanın ve Amazon DynamoDB'ye kaydedin. Tüketici uygulamaları mesajları işlemek için DynamoDB'den okuma yapacak şekilde yapılandırın.

D. Mesajları bir Amazon Simple Notification Service (SNS) konusuna yayinallyn ve birden fazla Amazon Simple Queue Service (SQS) aboneliği oluşturun. Tüketici uygulamaları mesajları işlemek için kuyruklardan okuma yapacak şekilde yapılandırın.

Soru Analizi:

Durum:

- Şirket, gelen mesajları alan bir uygulamaya sahip.
- Daha sonra onlarca başka uygulama ve mikro hizmet bu mesajları hızlı bir şekilde tüketiyor.
- Mesaj sayısı çok değişken ve bazen saniyede 100.000'e kadar artabiliyor.
- Amaç: çözümü **decoupled (bağımsız)** yapmak ve **ölçeklenebilirliği artırmak**.

İhtiyaçlar:

1. Yüksek hacimli ve ani artan mesaj trafiğini işleyebilmek.
2. Tüketici uygulamalar ve mikro servislerin mesajları **eş zamanlı ve bağımsız** şekilde alabilmesi.
3. Operasyonel karmaşıklığın minimum olması.

Bu nedenle çözüm **mesaj kuyruğu veya publish/subscribe mimarisi** gibi **loosely coupled** bir yapı olmalı.

Seçenek Analizi:

D Seçeneği Analizi:

“SNS topic + birden fazla SQS aboneliği”

- **Publish/Subscribe mimarisi:**
 - Mesajlar SNS topic'e yayılanır.
 - Her tüketici veya mikro hizmet için **bağımsız SQS kuyruğu** oluşturulabilir.
 - Mesajlar eş zamanlı ve bağımsız olarak tüketilir.
 - Ölçeklenebilir ve **loosely coupled** yapı sağlar.
-  En uygun çözüm.

A Seçeneği Analizi:

“Amazon Kinesis Data Analytics'e kaydetmek”

- Kinesis Data Analytics, **stream üzerinde SQL benzeri analiz yapar**, ancak doğrudan mesajları **çok sayıda tüketiciye yayılama** amacı için tasarlanmamıştır.
- Analiz için uygundur ama **decoupling ve yüksek ölçeklenebilir tüketici dağılımı** için ideal değildir.
-  Kısmen uygun, ancak tam çözüm değil.

B Seçeneği Analizi:

“Ingestion uygulamasını Auto Scaling EC2'de çalıştırma”

- EC2 Auto Scaling yalnızca **üretici uygulamanın ölçeklenmesini** sağlar.
- Ancak **tüketici uygulamalar hâlâ tightly coupled** ve aynı veri kaynağına erişmek zorunda.
- Mesajları decouple etmez, yüksek tüketici sayısı için yönetimi zor.
- Uygun değil.

C Seçeneği Analizi:

“Kinesis Data Streams + Lambda + DynamoDB”

- Tek shard kullanılması **yüksek mesaj hacmi için yetersizdir** → throttling riski var.
- Lambda ve DynamoDB ile ön işleme yapılabilir, ama bu yapı **mesajların tüm tüketiciler tarafından eş zamanlı alınması** için optimize değildir.
- Büyük hacim ve çok sayıda tüketici için uygun değil.

Sonuç

- Mesajları decouple etmek ve **çok sayıda tüketiciye yüksek hacimli mesaj dağıtmak** için en iyi çözüm:

D — SNS topic + SQS abonelikleri

QUESTION 8

A company is migrating a distributed application to AWS. The application serves variable workloads. The legacy platform consists of a primary server that coordinates jobs across multiple compute nodes. The company wants to modernize the application with a solution that maximizes resiliency and scalability.

How should a solutions architect design the architecture to meet these requirements?

- A.** Configure an Amazon Simple Queue Service (Amazon SQS) queue as a destination for the jobs. Implement the compute nodes with Amazon EC2 instances that are managed in an Auto Scaling group. Configure EC2 Auto Scaling to use scheduled scaling.
- B.** Configure an Amazon Simple Queue Service (Amazon SQS) queue as a destination for the jobs. Implement the compute nodes with Amazon EC2 instances that are managed in an Auto Scaling group. Configure EC2 Auto Scaling based on the size of the queue.
- C.** Implement the primary server and the compute nodes with Amazon EC2 instances that are managed in an Auto Scaling group. Configure AWS CloudTrail as a destination for the jobs. Configure EC2 Auto Scaling based on the load on the primary server.
- D.** Implement the primary server and the compute nodes with Amazon EC2 instances that are managed in an Auto Scaling group. Configure Amazon EventBridge (Amazon CloudWatch Events) as a destination for the jobs. Configure EC2 Auto Scaling based on the load on the compute nodes.

Soru:

Bir şirket, dağıtık bir uygulamayı AWS'e taşıyor. Uygulama, değişken iş yüklerini işler. Eski platform, işleri birden fazla compute düğümü arasında koordine eden bir birincil sunucudan oluşmaktadır.

Şirket, uygulamayı **maksimum dayanıklılık ve ölçeklenebilirlik** sağlayacak şekilde modernize etmek istiyor.

Bir çözüm mimarı, bu gereksinimleri karşılayacak mimariyi nasıl tasarlmalıdır?

A. İşler için hedef olarak bir Amazon Simple Queue Service (Amazon SQS) kuyruğu yapılandırın. Compute düğümlerini, Auto Scaling grubunda yönetilen Amazon EC2 instance'ları ile uygulayın. EC2 Auto Scaling'i **zamanlı ölçeklendirme (scheduled scaling)** kullanacak şekilde yapılandırın.

B. İşler için hedef olarak bir Amazon Simple Queue Service (Amazon SQS) kuyruğu yapılandırın. Compute düğümlerini, Auto Scaling grubunda yönetilen Amazon EC2 instance'ları ile uygulayın. EC2 Auto Scaling'i **kuyruğun boyutuna göre** ölçeklenecek şekilde yapılandırın.

C. Birincil sunucuyu ve compute düğümlerini Auto Scaling grubunda yönetilen Amazon EC2 instance'ları ile uygulayın. İşler için hedef olarak AWS CloudTrail'i yapılandırın. EC2 Auto Scaling'i birincil sunucunun yüküne göre ölçeklenecek şekilde yapılandırın.

D. Birincil sunucuyu ve compute düğümlerini Auto Scaling grubunda yönetilen Amazon EC2 instance'ları ile uygulayın. İşler için hedef olarak Amazon EventBridge (Amazon CloudWatch Events) yapılandırın. EC2 Auto Scaling'i compute düğümlerinin yüküne göre ölçeklenecek şekilde yapılandırın.

Soru Analizi:

Durum:

- Şirket, dağıtık bir uygulamayı AWS'e taşıyor.
- Uygulama değişken iş yüklerini işliyor.
- Eski platform: **birincil sunucu + birden fazla compute düğümü**.
- Amaç: Uygulamayı **maksimum dayanıklılık ve ölçeklenebilirlik** ile modernize etmek.

Neden:

- Compute düğümleri ve işler **bağımsız (decoupled)** olmalı.
- İş yüküne göre **otomatik ölçeklenebilir** bir yapı gereklidir.
- Tek bir sunucuya bağlılık **dayanıklılığı azaltır**.

Amaç:

- İşlerin kuyruk tabanlı yönetimi → decoupling.
- Auto Scaling ile **yük değişimlerine hızlı tepki**.

Seçenek Analizi:

● B Seçeneği Analizi:

SQS + EC2 Auto Scaling (queue length-based)

- Avantaj:
 - İşler SQS kuyruğuna yazılır → compute düğümleri bağımsız.
 - Auto Scaling, **kuyruk boyutuna göre ölçeklenir** → değişken iş yüküne hızlı tepki.
 - Sistem **resilient ve scalable** olur.
- En uygun ve önerilen çözüm.

✗ A Seçeneği Analizi:

SQS + EC2 Auto Scaling (scheduled scaling)

- Avantaj: EC2 instance sayısı önceden planlanan zamanlarda artabilir.
- Dezavantaj: **İş yüküne göre esnek değil**, ani artışları karşılayamaz.
- İş yükü değişkenliği ve ölçeklenebilirlik gereksinimini karşılamaz.

✗ C Seçeneği Analizi:

EC2 Auto Scaling + CloudTrail

- CloudTrail **log ve audit amaçlıdır**, iş kuyruğu değildir.
- Compute düğümlerinin yüküne göre ölçeklendirme yapılamaz.
- Gereksinimleri karşılamaz.

✗ D Seçeneği Analizi:

EC2 Auto Scaling + EventBridge (CloudWatch Events)

- EventBridge zamanlı veya tetiklenmiş olaylar için uygundur.
- Ancak **yük değişkenliği ve çok sayıda bağımsız compute düğümü için SQS kadar etkili değildir**.
- Ölçeklenebilirlik ve decoupling açısından sınırlı.

⌚ Sonuç

- Değişken iş yükleri için bağımsız, kuyruk tabanlı bir mimari + queue-length bazlı Auto Scaling en doğru çözümüdür.

 **Doğru Cevap:** B

QUESTION 9

A company is running an SMB file server in its data center. The file server stores large files that are accessed frequently for the first few days after the files are created. After 7 days the files are rarely accessed.

The total data size is increasing and is close to the company's total storage capacity. A solutions architect must increase the company's available storage space without losing low-latency access to the most recently accessed files. The solutions architect must also provide file lifecycle management to avoid future storage issues.

Which solution will meet these requirements?

- A.** Use AWS DataSync to copy data that is older than 7 days from the SMB file server to AWS.
- B.** Create an Amazon S3 File Gateway to extend the company's storage space. Create an S3 Lifecycle policy to transition the data to S3 Glacier Deep Archive after 7 days.
- C.** Create an Amazon FSx for Windows File Server file system to extend the company's storage space.
- D.** Install a utility on each user's computer to access Amazon S3. Create an S3 Lifecycle policy to transition the data to S3 Glacier Flexible Retrieval after 7 days.

Soru:

Bir şirket veri merkezinde bir SMB dosya sunucusu çalışmaktadır. Dosya sunucusu, oluşturuluktan sonraki ilk birkaç gün boyunca sıkça erişilen büyük dosyaları depolamaktadır. 7 gün geçtikten sonra dosyalara nadiren erişilmektedir.

Toplam veri boyutu artmakta ve şirketin toplam depolama kapasitesine yaklaşmaktadır. Bir çözüm mimarının, en son erişilen dosyalara düşük gecikmeli erişimi kaybetmeden şirketin kullanılabilir depolama alanını artırması gerekmektedir. Çözüm mimarı ayrıca gelecekteki depolama sorunlarını önlemek için dosya yaşam döngüsü yönetimi sağlamalıdır.

Bu gereksinimleri hangi çözüm karşılar?

- A.** AWS DataSync kullanarak 7 günden daha eski verileri SMB dosya sunucusundan AWS'ye kopyalayın.
- B.** Şirketin depolama alanını genişletmek için bir Amazon S3 File Gateway oluşturun.

Verileri 7 gün sonra S3 Glacier Deep Archive'a geçirmek için bir S3 Yaşam Döngüsü politikası oluşturun.

C. Şirketin depolama alanını genişletmek için bir Amazon FSx for Windows File Server dosya sistemi oluşturun.

D. Her kullanıcının bilgisayarına Amazon S3'e erişmek için bir yardımcı program yükleyin.

Verileri 7 gün sonra S3 Glacier Flexible Retrieval'a geçirmek için bir S3 Yaşam Döngüsü politikası oluşturun.

Soru Analizi:

Şirketin mevcut durumu:

- Veri merkezinde **SMB file server** çalışıyor.
- Büyük dosyalar var.
- **İlk 7 gün çok sık erişiliyor**, 7 günden sonra **nadiren** erişiliyor.
- Veriler büyüyor ve **kapasite dolmak üzere**.
- Gereksinimler:
 1. Depolama alanı genişlemeli.
 2. **Yeni/son erişilen dosyalar düşük gecikmeli (low-latency) kalmalı.**
 3. **Otomatik bir lifecycle management** olmalı → eski dosyaları daha ucuz depolamaya taşımak.

Bu tam olarak **hibrit bir dosya depolama + S3 arkaya depolama + lifecycle management** ihtiyacıdır.

Seçenek Analizi:

B seçenekü:

Amazon S3 File Gateway + S3 Lifecycle

Bu seçenek şunları sağlar:

✓ **SMB dosya sunucusu gibi çalışır**

S3 File Gateway, on-prem SMB/NFS istemcilerine **düşük gecikmeli cache** sunar.

✓ **Eski dosyaları otomatik olarak S3'e kaydırır**

Gateway'in cache'i sadece sık kullanılan/veri sıcaklığı yüksek olan dosyaları tutar.

✓ **Depolama sınırsızdır**

Arka tarafta S3 sınırsızdır, yerel depolama sorunu çözülür.

✓ **Lifecycle policy ile 7 gün sonra Glacier Deep Archive'a taşıma**

"7 gün sonra az erişiliyor" bilgisiyle uyumlu.

Bu tam olarak sorunun söylediği iki hedefi birleştiriyor:

- **Low latency access** → gateway cache
- **Lifecycle management** → S3 policy
- **Storage genişletme** → S3 arkada büyük depo

Bu nedenle **en ideal ve AWS'nin önerdiği çözüm B'dir.**

✗ A seçeneği:

AWS DataSync

- DataSync sadece **kopyalama/taşıma** yapar.
- Dosyaların düşük gecikmeli erişimini **devam ettirmez**.
- SMB sunucusu dolmaya devam eder, bir çözüm değil.
- Lifecycle yok.

✗ C seçeneği:

FSx for Windows File Server

- Kapasiteyi artırır ama oldukça maliyetlidir.
- Lifecycle management **yoktur**.
- "Sıcak" ve "soğuk" veri ayrimı için çözüm sunmaz.

✗ D seçeneği:

Kullanıcı PC'lerine araç yüklemek

- Kullanıcı deneyimi değişir → kötü bir çözüm.
- SMB erişimi kaybolur.
- Düşük gecikmeli erişim sağlanmaz.
- Şirket altyapısına uygun değil.

⌚ Sonuç: En doğru çözüm — B seçeneği

Amazon S3 File Gateway + S3 Lifecycle

✓ Depolamayı genişletir

- ✓ En çok kullanılan dosyaları düşük gecikme ile sunar
 - ✓ Eski dosyaları otomatik olarak Glacier sınıfına taşıır
-

QUESTION 10

A company is building an ecommerce web application on AWS. The application sends information about new orders to an Amazon API Gateway REST API to process. The company wants to ensure that orders are processed in the order that they are received. Which solution will meet these requirements?

- A.** Use an API Gateway integration to publish a message to an Amazon Simple Notification Service (Amazon SNS) topic when the application receives an order. Subscribe an AWS Lambda function to the topic to perform processing.
- B.** Use an API Gateway integration to send a message to an Amazon Simple Queue Service (Amazon SQS) FIFO queue when the application receives an order. Configure the SQS FIFO queue to invoke an AWS Lambda function for processing.
- C.** Use an API Gateway authorizer to block any requests while the application processes an order.
- D.** Use an API Gateway integration to send a message to an Amazon Simple Queue Service (Amazon SQS) standard queue when the application receives an order. Configure the SQS standard queue to invoke an AWS Lambda function for processing.

Soru:

Bir şirket AWS üzerinde bir e-ticaret web uygulaması geliştiriyor. Uygulama, yeni siparişler hakkında bilgileri işlenmesi için bir Amazon API Gateway REST API'sine gönderiyor. Şirket, siparişlerin **kendilerine ulaşıldığı sırayla** işlenmesini sağlamak istiyor.

Bu gereksinimleri hangi çözüm karşılar?

- A.** Uygulama bir sipariş aldığında API Gateway entegrasyonunu kullanarak bir Amazon SNS konusuna mesaj yayınlayın. Konuya abone olan bir AWS Lambda işlevi ile siparişleri işleyin.
- B.** Uygulama bir sipariş aldığında API Gateway entegrasyonunu kullanarak bir Amazon SQS **FIFO kuyruğuna** mesaj gönderin. SQS FIFO kuyruğunun bir Lambda işlevini çağırmasını sağlayın.
- C.** Uygulama bir sipariş işlerken API Gateway yetkilendiricisini kullanarak tüm istekleri engelleyin.
- D.** Uygulama bir sipariş aldığında API Gateway entegrasyonunu kullanarak bir Amazon

SQS **standart kuyruğa** mesaj gönderin. SQS standart kuyruğunun bir Lambda işlevini çağırmasını sağlayın.

Soru Analizi:

Şirket bir **e-ticaret uygulaması** geliştiriyor.

Uygulama siparişleri API Gateway'e gönderiyor ve bu siparişlerin:

→ “**Alındıkları sırayla işlenmesi**”

gerekliyor.

E-ticaret için bu çok kritik bir gereksinimdir çünkü:

- Sipariş numaraları sıralı olmalıdır.
- Ödeme, stok yönetimi, faturalama işlemleri doğru sırada yapılmalıdır.
- Aynı ürün için iki sipariş tetiklenirse stok yanlış hesaplanmamalıdır.

Bu nedenle **ordering (sıralama garantisı)** sağlayan bir servis şarttır.

AWS'de *sıralama garantisı sağlayan tek hizmet* **SQS FIFO queue**'dur.

Dolayısıyla çözüm bir mesaj kuyruğu ve FIFO mantığıyla çalışmalıdır.

Seçenek Analizi:

B. SQS FIFO Queue + Lambda (Doğru)

Bu çözüm neden doğru?

- **SQS FIFO**, AWS'nin *First-In-First-Out* garantisini veren tek kuyruğudur.
- Mesajlar **tam olarak gönderildiği sırada** Lambda'ya iletilir.
- **OrderId** benzeri sipariş sırası asla karışmaz.
- E-ticaret gibi sipariş sıralamasının kritik olduğu senaryolar için AWS'nin önerdiği çözüm budur.
- API Gateway → FIFO queue → Lambda akışı, modern event-driven mimari için idealdir.

Bu nedenle gereksinimi **tam olarak karşılayan tek seçenek** budur.

A. SNS Topic + Lambda

Neden yanlış?

- SNS bir “publish/subscribe” servisidir, **ordering garantisini vermez**.
- Mesajlar Lambda'ya **herhangi bir sırada** ulaşabilir.

- SNS aynı anda birden fazla Lambda invocation tetikleyebilir → tamamen paralel ve düzensiz çalışır.

Bu yüzden sipariş sıralamasının korunması mümkün değildir.

✗ C. API Gateway Authorizer ile istekleri bloklama

Bu seçenek teknik olarak anlamsız çünkü:

- API Gateway Authorizer'ın görevi **kimlik doğrulama / yetkilendirmedir, mesaj sıralaması veya kuyruk kontrolü değildir.**
- Authorizer kullanarak siparişleri sıraya alamazsınız.
- Siparişi işlerken diğer istekleri engellemek ölçülebilir değildir ve e-ticaret uygulaması için felakettir.

Tamamen konu dışıdır.

✗ D. SQS Standard Queue + Lambda

Neden yanlış?

- SQS Standard queue **ordering garantisı sağlamaz.**
- Mesajlar **en az bir kez teslim edilir (at-least-once)** ve **sıra bozulabilir.**
- Bazı siparişler birden fazla kez ya da yanlış sırada gelebilir.

E-ticaret uygulamasında bu yanlış faturalama, stok hataları gibi ciddi sorunlara yol açar.

⌚ Genel Sonuç

Siparişlerin alınma sırasına göre işlenmesi gerektiği için:

✓ Doğru seçenek: B — SQS FIFO Queue + Lambda

Her açıdan gerek performans, gerek garanti edilen sıralama açısından en uygun çözümdür.

QUESTION 11

A company has an application that runs on Amazon EC2 instances and uses an Amazon Aurora database. The EC2 instances connect to the database by using user names and passwords that are stored locally in a file. The company wants to minimize the operational overhead of credential management.

What should a solutions architect do to accomplish this goal?

- A. Use AWS Secrets Manager. Turn on automatic rotation.

- B.** Use AWS Systems Manager Parameter Store. Turn on automatic rotation.
- C.** Create an Amazon S3 bucket to store objects that are encrypted with an AWS Key Management Service (AWS KMS) encryption key. Migrate the credential file to the S3 bucket. Point the application to the S3 bucket.
- D.** Create an encrypted Amazon Elastic Block Store (Amazon EBS) volume for each EC2 instance. Attach the new EBS volume to each EC2 instance. Migrate the credential file to the new EBS volume. Point the application to the new EBS volume.

Soru:

Bir şirketin Amazon EC2 üzerinde çalışan bir uygulaması ve Amazon Aurora kullanan bir veritabanı vardır. EC2 örnekleri, kullanıcı adı ve şifreleri yerel bir dosyada saklayarak veritabanına bağlanmaktadır. Şirket, kimlik bilgisi yönetiminin operasyonel yükünü en aza indirmek istemektedir.

Bu hedefi gerçekleştirmek için bir çözüm mimarı ne yapmalıdır?

- A.** AWS Secrets Manager kullanın. Otomatik döndürmeyi (automatic rotation) etkinleştirin.
- B.** AWS Systems Manager Parameter Store kullanın. Otomatik döndürmeyi etkinleştirin.
- C.** AWS Key Management Service (AWS KMS) ile şifrelenmiş nesneleri depolamak için bir Amazon S3 kovası oluşturun. Kimlik bilgisi dosyasını bu S3 kovasına taşıyın. Uygulamayı S3 kovasını kullanacak şekilde yönlendirin.
- D.** Her EC2 örneği için şifrelenmiş bir Amazon Elastic Block Store (Amazon EBS) hacmi oluşturun. Yeni EBS hacmini her EC2 örneğine ekleyin. Kimlik bilgisi dosyasını yeni EBS hacmine taşıyın. Uygulamayı yeni EBS hacmini kullanacak şekilde yönlendirin.

Soru Analizi:

- Bir uygulama **EC2 üzerinde çalışıyor**.
- Uygulama **Aurora veritabanına** bağlıyor.
- Veritabanı kullanıcı adı ve şifreleri **EC2 üzerindeki yerel bir dosyada** saklanıyor.
- Şirket **kimlik bilgisi yönetiminin operasyonel yükünü azaltmak** istiyor.

Operasyonel yükü azaltmak ne demek?

- Manuel olarak şifreleri güncellememek
- Şifrelerin otomatik olarak döndürülmesi
- Güvenli bir şekilde saklanması
- Uygulamaya otomatik ve güvenli erişim sağlanması

Seçenek Analizi:

✓ A. AWS Secrets Manager. Otomatik döndürmeyi etkinleştirin. (Doğru cevap)

Bu tanım doğrudan **AWS Secrets Manager** hizmetinin kullanım amacıdır.

Secrets Manager:

- ✓ Veritabanı kimlik bilgilerini saklar
- ✓ Aurora ile **yerleşik entegrasyon** sunar
- ✓ Şifreleri **otomatik döndürür (rotation)**
- ✓ EC2 uygulaması API çağrılarıyla otomatik alabilir
- ✓ Operasyonel yük minimaldir

Bu seçenek:

- Aurora ile native entegre olur
- Şifrelerin otomatik rotasyonunu destekler
- Uygulamanın erişimini kolaylaştırır
- Yönetim yükü minimumdur
- AWS'nin **en iyi uygulama (best practice)** çözümüdür

Tam olarak sorunun istediği şeyi sağlar:

→ “credential management” operasyonel yükünün azaltılması

Bu yüzden doğru çözüm A'dır.

✗ B. SSM Parameter Store + automatic rotation

Parameter Store güvenlidir ancak:

- Otomatik rotasyon için **yalnızca Lambda tabanlı özel bir çözüm gereklidir**, built-in değildir.
- Aurora ile **yerleşik credential rotation entegrasyonu yoktur**.
- Yönetim yükü yine artar.

Secrets Manager varken bu doğru yaklaşım değildir.

✗ C. Credential dosyasını S3'e taşımak

- S3, kimlik bilgisi saklamak için tasarlanmamıştır.
- Şifre rotasyonu **yoktur**.
- S3 üzerinden credential okuma güvenlik risklerini artırır.

- Operasyonel yük azalmaz, aksine artar.

Bu, AWS best practice'e tamamen aykırıdır.

✗ D. EBS volume oluşturmak ve dosyayı orada saklamak

Bu seçenek:

- Şifreleri sadece daha güvenli bir disk üzerinde saklar.
- **Hiçbir şekilde yönetim yükünü azaltmaz.**
- Otomatik rotasyon sağlanmaz.
- Yapılan işlem kimlik bilgisi yönetimini iyileştirmez, sadece depolamayı değiştirir.

Bunun kimlik bilgisi yönetimiyle ilgisi yok.

⌚ Sonuç

✓ En doğru ve AWS tarafından önerilen çözüm: A — AWS Secrets Manager + Automatic Rotation

QUESTION 12

A global company hosts its web application on Amazon EC2 instances behind an Application Load Balancer (ALB). The web application has static data and dynamic data. The company stores its static data in an Amazon S3 bucket. The company wants to improve performance and reduce latency for the static data and dynamic data. The company is using its own domain name registered with Amazon Route 53. What should a solutions architect do to meet these requirements?

A. Create an Amazon CloudFront distribution that has the S3 bucket and the ALB as origins. Configure Route 53 to route traffic to the CloudFront distribution.

B. Create an Amazon CloudFront distribution that has the ALB as an origin. Create an AWS Global Accelerator standard accelerator that has the S3 bucket as an endpoint. Configure Route 53 to route traffic to the CloudFront distribution.

C. Create an Amazon CloudFront distribution that has the S3 bucket as an origin. Create an AWS Global Accelerator standard accelerator that has the ALB and the CloudFront distribution as endpoints. Create a custom domain name that points to the accelerator DNS name. Use the custom domain name as an endpoint for the web application.

D. Create an Amazon CloudFront distribution that has the ALB as an origin. Create an AWS Global Accelerator standard accelerator that has the S3 bucket as an endpoint. Create two domain names. Point one domain name to the

CloudFront DNS name for dynamic content. Point the other domain name to the accelerator DNS name for static content. Use the domain names as endpoints for the web application.

Soru:

Bir küresel şirket, web uygulamasını bir Application Load Balancer'ın (ALB) arkasındaki Amazon EC2 örneklerinde barındırmaktadır. Web uygulamasında statik veri ve dinamik veri bulunmaktadır. Şirket statik verilerini bir Amazon S3 kovasında depolamaktadır. Şirket, hem statik veri hem de dinamik veri için **performansı artırmak ve gecikmeyi azaltmak** istemektedir. Şirket, kendi alan adını kullanmakta olup bu alan adı Amazon Route 53 ile kaydedilmiştir.

Bu gereksinimleri karşılamak için bir çözüm mimarı ne yapmalıdır?

- A. S3 kovasını ve ALB'yi kaynak (origin) olarak kullanan bir Amazon CloudFront dağıtımlı oluşturun. Route 53'ü trafiği CloudFront dağıtımına yönlendirecek şekilde yapılandırın.
- B. ALB'yi kaynak olarak kullanan bir Amazon CloudFront dağıtımlı oluşturun. S3 kovasını üç nokta olarak kullanan bir AWS Global Accelerator standard accelerator oluşturun. Route 53'ü trafiği CloudFront dağıtımına yönlendirecek şekilde yapılandırın.
- C. S3 kovasını kaynak olarak kullanan bir Amazon CloudFront dağıtımlı oluşturun. ALB'yi ve CloudFront dağıtımını üç nokta olarak kullanan bir AWS Global Accelerator standard accelerator oluşturun. Accelerator'ın DNS adına işaret eden özel bir alan adı oluşturun. Web uygulaması için bu özel alan adını üç nokta olarak kullanın.
- D. ALB'yi kaynak olarak kullanan bir Amazon CloudFront dağıtımlı oluşturun. S3 kovasını üç nokta olarak kullanan bir AWS Global Accelerator standard accelerator oluşturun. İki alan adı oluşturun. Bir alan adını dinamik içerik için CloudFront DNS adına, diğer alan adını statik içerik için accelerator DNS adına yönlendirin. Web uygulaması için bu alan adlarını üç nokta olarak kullanın.

Soru Analizi:

Şirketin durumu:

- Uygulama **EC2 + ALB** üzerinde çalışıyor (dinamik içerik burada).
- Statik içerik **S3 bucket** üzerinde.
- Şirket global, yani **dünya genelinde düşük gecikme (low latency)** istiyor.
- Hem statik hem de dinamik veri için performans artırılmalı.
- Route 53 domain yönetimi kullanılıyor.

Bu gereksinimleri en iyi karşılayan AWS servisi:

✓ **Amazon CloudFront (CDN)**

Hem statik içerik (S3) hem dinamik içerik (ALB) için global edge caching + düşük gecikme sağlar.

CloudFront şunları yapabilir:

- **S3 için statik içeriği cache'ler → çok hızlı sunar**
- **ALB arkasındaki dinamik içeriği dünya geneline hızlı ileter**
- **Tek bir dağıtımda birden fazla origin kullanabilir (S3 + ALB)**
- Route 53, domaini CloudFront'a yönlendirebilir

Bu nedenle ideal çözüm:

Tek bir CloudFront dağıtıımı oluşturarak hem S3 hem ALB'yi origin yapmak.

Seçenek Analizi:

 **A. CloudFront dağıtıımı: S3 + ALB origin**

- ✓ Statik içerik S3'ten hızlı sunulur
- ✓ Dinamik içerik ALB üzerinden low latency gelir
- ✓ Tek bir domain CloudFront'a yönlendirilir
- ✓ Global performans iyileşir
- ✓ En basit, en az karmaşık ve en AWS-best-practice çözüm

Bu nedenle **doğru çözüm** budur.

 **B. CloudFront + Global Accelerator (S3 endpoint)**

Yanlış çünkü:

- S3, Global Accelerator tarafından **desteklenen bir endpoint değildir** (yalnızca ALB, NLB, EC2).
- Mantık hatası var: CloudFront zaten S3 için optimum çözümüdür, GA eklemenin anlamı yok.
- Çözüm gereksiz karmaşık.

 **C. CloudFront + Global Accelerator (CloudFront ve ALB endpoint olarak)**

Yanlış çünkü:

- Global Accelerator, CloudFront'u **endpoint olarak** desteklemez.
- GA, CloudFront önünde kullanılmaz → zaten Edge Network kullanıyor.
- Gereksinimler için aşırı karmaşıktır.

 **D. İki domain oluşturup statik/dinamik içeriği ayırmak**

Yanlış çünkü:

- Statik ve dinamik içerik için iki ayrı domain oluşturmak kullanıcı deneyimini bozar.
- Gereksiz karmaşıktır.
- S3, Global Accelerator endpoint'i olamaz → teknik olarak da yanlış.
- Ayrı domain = gereksiz DNS karmaşası.

Sonuç

Gereksinimleri karşılayan en doğru ve AWS best-practice çözüm:

✓ A — CloudFront dağıtımı, S3 + ALB origin olarak

QUESTION 13

A company performs monthly maintenance on its AWS infrastructure. During these maintenance activities, the company needs to rotate the credentials for its Amazon RDS for MySQL databases across multiple AWS Regions.

Which solution will meet these requirements with the LEAST operational overhead?

- A.** Store the credentials as secrets in AWS Secrets Manager. Use multi-Region secret replication for the required Regions. Configure Secrets Manager to rotate the secrets on a schedule.
- B.** Store the credentials as secrets in AWS Systems Manager by creating a secure string parameter. Use multi-Region secret replication for the required Regions. Configure Systems Manager to rotate the secrets on a schedule.
- C.** Store the credentials in an Amazon S3 bucket that has server-side encryption (SSE) enabled. Use Amazon EventBridge (Amazon CloudWatch Events) to invoke an AWS Lambda function to rotate the credentials.
- D.** Encrypt the credentials as secrets by using AWS Key Management Service (AWS KMS) multi-Region customer managed keys. Store the secrets in an Amazon DynamoDB global table. Use an AWS Lambda function to retrieve the secrets from DynamoDB. Use the RDS API to rotate the secrets.

Soru:

Bir şirket AWS altyapısında aylık bakım çalışmaları yapmaktadır. Bu bakım faaliyetleri sırasında, şirketin birden fazla AWS Bölgesindeki (Region) Amazon RDS for MySQL veritabanları için kimlik bilgilerini döndürmesi (rotate) gerekmektedir.

Bu gereksinimleri **en az operasyonel yükle** karşılayacak çözüm hangisidir?

- A.** Kimlik bilgilerini AWS Secrets Manager'de birer secret olarak saklayın. Gerekli Bölgeler için çoklu-bölge (multi-Region) secret çoğaltmasını etkinleştirin. Secrets Manager'ı bir takvime göre şifreleri döndürecek şekilde yapılandırın.
- B.** Kimlik bilgilerini AWS Systems Manager'da güvenli dize (secure string) parametresi oluşturarak saklayın. Gerekli Bölgeler için çoklu-bölge secret çoğaltmasını etkinleştirin. Systems Manager'ı bir takvime göre şifreleri döndürmek için yapılandırın.
- C.** Kimlik bilgilerini sunucu tarafı şifrelemenin (SSE) etkin olduğu bir Amazon S3 kovasında saklayın. Kimlik bilgilerini döndürmek için Amazon EventBridge (Amazon CloudWatch Events) kullanarak bir AWS Lambda işlevi çağırın.
- D.** Kimlik bilgilerini AWS Key Management Service (AWS KMS) çoklu-bölge müsteri yönetimli anahtarları ile şifreleyerek secret hâline getirin. Secret'lari bir Amazon DynamoDB global tablosunda saklayın. AWS Lambda işlevi ile DynamoDB'den secret'lari alın. RDS API'sini kullanarak kimlik bilgilerini döndürün.

Soru Analizi:

Şirket:

- Birden fazla **AWS Region** kullanıyor.
- Amazon **RDS for MySQL** veritabanları için **aylık kimlik bilgisi döndürme (credential rotation)** yapmak istiyor.
- Amaç:
“En az operasyonel yükle” (least operational overhead).

Bu durumda ideal çözüm:

- RDS ile **yerleşik entegrasyon** olmalı,
- **Multi-Region** desteklemeli,
- **Otomatik secret rotation** sunmalı,
- Kullanımı kolay olmalı.

Seçenek Analizi:

- A. Secrets Manager + Multi-Region Replication + Scheduled Rotation (Doğru cevap)**

Bu seçenek en doğru olanıdır çünkü:

✓ Secrets Manager, RDS için yerleşik otomatik şifre döndürme desteği sahiptir

MySQL için doğrudan otomatik rotation sağlar.

✓ Multi-Region secret replication desteği vardır

Tek tuşla diğer Region'lara otomatik replikasyon yapılır.

✓ Operasyonel yük minimumudur

Rotation, replikasyon, sürüm yönetimi → hepsi managed.

✓ En iyi AWS best-practice çözümü budur.

Tüm gereksinimleri en basit ve güvenli şekilde karşılayan tek çözüm A'dır.

✗ B. Systems Manager Parameter Store (Yanlış)

Neden?

- Parameter Store **multi-Region replication** desteğine sahip değildir.
- Native RDS rotation entegrasyonu yoktur.
- Rotation için **özel Lambda fonksiyonları** yazmak zorundasın → operasyonel yük yüksek.

Bu yüzden sorunun istediği “en az operasyonel yük” ifadesiyle uyuşmaz.

✗ C. S3 + EventBridge + Lambda (Yanlış)

Yanlış çünkü:

- S3, kimlik bilgisi saklamak için uygun bir yer değildir.
- Secret rotation'ı manuel Lambda koduyla yapmak gereklidir → operasyonel yük artar.
- Güvenlik açısından da AWS tarafından önerilmez.

Bu çözüm modern AWS mimarilerinde best-practice değildir.

✗ D. DynamoDB Global Table + KMS Multi-Region + Lambda (Yanlış)

Bu seçenek:

- Çok karmaşıktır.
- Secret yönetimi için gereksiz mimari inşa eder.
- Rotation tamamen manuel yapılır.
- Multi-Region replikasyon mantıklı olsa da operasyonel yük **çok yüksektir**.

Secrets Manager varken bunu kullanmak anlamsızdır.

🎯 SONUÇ

Gereksinim:

→ Multi-Region

- RDS credential rotation
- Minimum operasyonel yük

Bu nedenle:

✓ Doğru Cevap: A — AWS Secrets Manager + Multi-Region Replication + Automatic Rotation

QUESTION 14

A company runs an ecommerce application on Amazon EC2 instances behind an Application Load Balancer. The instances run in an Amazon EC2 Auto Scaling group across multiple Availability Zones. The Auto Scaling group scales based on CPU utilization metrics. The ecommerce application stores the transaction data in a MySQL 8.0 database that is hosted on a large EC2 instance.

The database's performance degrades quickly as application load increases. The application handles more read requests than write transactions. The company wants a solution that will automatically scale the database to meet the demand of unpredictable read workloads while maintaining high availability.

Which solution will meet these requirements?

- A.** Use Amazon Redshift with a single node for leader and compute functionality.
- B.** Use Amazon RDS with a Single-AZ deployment Configure Amazon RDS to add reader instances in a different Availability Zone.
- C.** Use Amazon Aurora with a Multi-AZ deployment. Configure Aurora Auto Scaling with Aurora Replicas.
- D.** Use Amazon ElastiCache for Memcached with EC2 Spot Instances.

Soru:

Bir şirket, bir Application Load Balancer'ın arkasında çalışan Amazon EC2 örnekleri üzerinde bir e-ticaret uygulaması çalışmaktadır. EC2 örnekleri, birden fazla Availability Zone'a yayılmış bir Amazon EC2 Auto Scaling grubunda çalışmaktadır. Auto Scaling grubu, CPU kullanım metriklerine göre ölçeklenmektedir.

E-ticaret uygulaması, işlem (transaction) verilerini büyük boyutlu bir EC2 örneğinde barındırılan MySQL 8.0 veritabanında saklamaktadır. Uygulama yükü arttıkça veritabanının performansı hızlı bir şekilde düşmektedir. Uygulama, yazma işlemlerinden daha fazla okuma isteği almaktadır. Şirket, öngörülemeyen okuma iş yüklerinin talebini karşılamak için veritabanını **otomatik olarak ölçeklendirebilen ve yüksek erişilebilirliği koruyabilen** bir çözüm istemektedir.

Bu gereksinimleri karşılayacak çözüm hangisidir?

- A.** Lider ve hesaplama işlevlerini tek bir düğümde sunan Amazon Redshift kullanın.
- B.** Amazon RDS'i Single-AZ dağıtımla kullanın. RDS'in farklı bir Availability Zone'da okuma kopyaları eklemesini yapılandırın.
- C.** Amazon Aurora'yı Multi-AZ dağıtımla kullanın. Aurora Replicas ile Aurora Auto Scaling'i yapılandırın.
- D.** EC2 Spot Instances ile Amazon ElastiCache for Memcached kullanın.

Soru Analizi:

Şirketin durumu şöyle:

- Uygulama EC2 üzerinde çalışıyor, önünde ALB var.
- Auto Scaling CPU'ya göre çalışıyor, yani uygulama katmanı otomatik ölçeklenebiliyor.
- Veritabanı ise **tek büyük bir EC2 instance üzerinde çalışan MySQL 8.0**.
- Yük arttığında **veritabanı hemen performans düşüşü yaşıyor** → dikey ölçekte yetersiz.
- Veritabanına gelen isteklerde:
 - **Okuma (read) istekleri yazma isteklerinden çok daha fazla.**
- Şirket istiyor ki:
 - Veritabanı **otomatik ölçeklenebilir**sin (özellikle okuma trafiği için)
 - **Yüksek erişilebilirlik (Multi-AZ)** olsun
 - Okuma iş yükü **öngörülemez**, yani Auto Scaling çok önemli.

Seçenek Analizi:

C seçeneği: Amazon Aurora (Multi-AZ) + Aurora Auto Scaling + Aurora Replicas

✓ Neden doğru?

- Aurora, MySQL uyumludur → mevcut uygulama değişmeden çalışabilir.
- **Aurora Replicas otomatik ölçeklenebilir** (Aurora Auto Scaling ile).
- Okuma trafiği çoksa → read replicas otomatik eklenir.
- Multi-AZ zaten içinde → yüksek erişilebilirlik garanti.
- Performansı RDS MySQL'den çok daha iyi.

- Şirketin istediği **otonom ölçeklenen, yüksek erişilebilirlik sağlayan çözüm** tam olarak budur.

A seçeneği: Amazon Redshift single node

Neden yanlış?

- Redshift bir **analitik (OLAP)** veri ambarıdır.
- OLTP (transactional) iş yükleri için uygun değildir.
- Tek node ayrıca **yüksek erişilebilirlik sağlamaz**.
- E-ticaret uygulaması için uygun değil.

B seçeneği: RDS Single-AZ + read replicas

Neden yanlış?

- Single-AZ → yüksek erişilebilirlik yok!
- RDS read replicas **otomatik ölçeklenmez**.
- Aurora'daki otomatik okuma replica ölçeklendirme burada yok.
- Ayrıca istek öngörülemez olduğundan manuel yönetim sorun çıkarır.

D seçeneği: ElastiCache for Memcached

Neden yanlış?

- ElastiCache bir cache sistemidir, **veritabanı değildir**.
- Transaction verisini saklayamaz.
- Ayrıca EC2 Spot ile kullanmak daha da risklidir (örnekler uçabilir).
- Öngörülemez okuma yükünün tamamını çözmez, veritabanı ihtiyacını ortadan kaldırılmaz.

Sonuç

Seçenek Uygun mu? Açıklama

- | | | |
|---|---|--|
| A |  | Redshift OLAP, transaction DB için uygun değil |
| B |  | Single-AZ, otomatik scaling yok |
| C |  | İstenen tüm özellikleri sağlayan tek çözüm |
| D |  | Cache çözümü, DB yerine geçmez |

QUESTION 15

A company recently migrated to AWS and wants to implement a solution to protect the traffic that flows in and out of the production VPC. The company had an inspection server in its on-premises data center. The inspection server performed specific operations such as traffic flow inspection and traffic filtering. The company wants to have the same functionalities in the AWS Cloud.

Which solution will meet these requirements?

- A.** Use Amazon GuardDuty for traffic inspection and traffic filtering in the production VPC.
- B.** Use Traffic Mirroring to mirror traffic from the production VPC for traffic inspection and filtering.
- C.** Use AWS Network Firewall to create the required rules for traffic inspection and traffic filtering for the production VPC.
- D.** Use AWS Firewall Manager to create the required rules for traffic inspection and traffic filtering for the production VPC.

Soru:

Bir şirket yakın zamanda AWS'e taşındı ve üretim VPC'sine giren ve çıkan trafiği korumak için bir çözüm uygulamak istiyor. Şirketin şirket içi veri merkezinde daha önce bir denetim sunucusu bulunuyordu. Bu denetim sunucusu, trafik akışı denetimi ve trafik filtreleme gibi belirli işlemleri gerçekleştiriyordu. Şirket, aynı işlevselligi AWS Bulutu'nda da kullanmak istiyor.

Bu gereksinimleri hangi çözüm karşılar?

- A.** Üretim VPC'sinde trafik denetimi ve trafik filtreleme için Amazon GuardDuty kullanın.
- B.** Üretim VPC'sinden trafiği yansıtma (mirror) için Traffic Mirroring kullanın ve trafik denetimi ile filtreleme yapın.
- C.** Üretim VPC'si için trafik denetimi ve trafik filtreleme kurallarını oluşturmak üzere AWS Network Firewall kullanın.
- D.** Üretim VPC'si için trafik denetimi ve trafik filtreleme kurallarını oluşturmak üzere AWS Firewall Manager kullanın.

Soru Analizi:

Şirket:

- AWS'e yeni taşındı ve üretim VPC'sine giren ve çıkan trafiği **korumak** istiyor.

- Daha önce on-premise bir **inspection server** vardı. Bu sunucu:
 - Trafik akışı denetimi (traffic inspection)
 - Trafik filtreleme (traffic filtering)

yapıyordu.

Şirket, AWS'de **aynı işlevselligi** sağlamak istiyor.

Ihtiyaçlar:

1. Trafik akışı denetimi
2. Trafik filtreleme
3. VPC bazlı çözüm

Yani aranan çözüm **stateful firewall** veya **network inspection + filtering** özelliklerini desteklemeli.

Seçenek Analizi:

C) AWS Network Firewall

- **Trafik denetimi ve filtreleme** sağlar.
- Stateful firewall kuralları uygulanabilir.
- VPC giriş-çıkış trafigini kontrol edebilir.
- On-prem inspection server ile aynı işlevselligi sunar.

Tüm gereksinimleri karşılayan tek seçenek.

X A) Amazon GuardDuty

- GuardDuty: Tehdit tespiti, anomali ve güvenlik uyarıları üretir.
- **Trafik filtrelemez**, engellemez.
- Inspection server işlevini yerine getirmez.

X B) VPC Traffic Mirroring

- Traffic Mirroring: Trafiği **yansıtır** → analiz ve packet capture için kullanılabilir.
- **Trafik filtreleme ve engellemeye yapmaz.**
- Dolayısıyla sadece bir parçasını karşılar, tam çözüm değildir.

X D) AWS Firewall Manager

- Firewall Manager: Kuralların **merkezi yönetimi** içindir.

- Kendi başına bir firewall değildir, **inspection ve filtering yapmaz**.
- Tek başına ihtiyacı karşılamaz.

Sonuç

Doğru cevap: C — AWS Network Firewall

- Şirketin eski on-premise inspection server işlevlerini AWS üzerinde sağlamanın en uygun yolu budur.

QUESTION 16

A company hosts a data lake on AWS. The data lake consists of data in Amazon S3 and Amazon RDS for PostgreSQL. The company needs a reporting solution that provides data visualization and includes all the data sources within the data lake. Only the company's management team should have full access to all the visualizations. The rest of the company should have only limited access.

Which solution will meet these requirements?

- A.** Create an analysis in Amazon QuickSight. Connect all the data sources and create new datasets. Publish dashboards to visualize the data. Share the dashboards with the appropriate IAM roles.
- B.** Create an analysis in Amazon QuickSight. Connect all the data sources and create new datasets. Publish dashboards to visualize the data. Share the dashboards with the appropriate users and groups.
- C.** Create an AWS Glue table and crawler for the data in Amazon S3. Create an AWS Glue extract, transform, and load (ETL) job to produce reports. Publish the reports to Amazon S3. Use S3 bucket policies to limit access to the reports.
- D.** Create an AWS Glue table and crawler for the data in Amazon S3. Use Amazon Athena Federated Query to access data within Amazon RDS for PostgreSQL. Generate reports by using Amazon Athena. Publish the reports to Amazon S3. Use S3 bucket policies to limit access to the reports.

Soru:

Bir şirket AWS üzerinde bir veri gölü (data lake) barındırıyor. Veri gölü, Amazon S3 ve Amazon RDS for PostgreSQL içindeki verilerden oluşuyor. Şirket, veri görselleştirme sağlayan ve veri gölündeki tüm veri kaynaklarını içeren bir raporlama çözümüne ihtiyaç duyuyor. Sadece şirketin yönetim ekibi tüm görselleştirmelere tam erişime sahip olmalıdır. Şirketin geri kalanı ise yalnızca sınırlı erişime sahip olmalıdır.

Bu gereksinimleri hangi çözüm karşılar?

A. Amazon QuickSight'ta bir analiz oluşturun. Tüm veri kaynaklarına bağlanın ve yeni veri kümeleri oluşturun. Verileri görselleştirmek için panolar (dashboard) yayınlayın. Panoları uygun IAM rollerine paylaşın.

B. Amazon QuickSight'ta bir analiz oluşturun. Tüm veri kaynaklarına bağlanın ve yeni veri kümeleri oluşturun. Verileri görselleştirmek için panolar yayınlayın. Panoları uygun kullanıcılara ve gruplara paylaşın.

C. Amazon S3'teki veriler için bir AWS Glue tablosu ve tarayıcı (crawler) oluşturun. Raporlar üretmek için AWS Glue ETL işi oluşturun. Raporları Amazon S3'e yayınlayın. Raporlara erişimi sınırlamak için S3 bucket politikalarını kullanın.

D. Amazon S3'teki veriler için bir AWS Glue tablosu ve tarayıcı oluşturun. Amazon RDS for PostgreSQL içindeki verilere erişmek için Amazon Athena Federated Query kullanın. Raporları Amazon Athena ile oluşturun. Raporları Amazon S3'e yayınlayın. Erişimi sınırlamak için S3 bucket politikalarını kullanın.

Soru Analizi:

AWS üzerinde veri gölü olan bir şirket, tüm veri kaynaklarını kullanarak *raporlama + görselleştirme* yapmak istiyor.

Veri kaynakları:

- **Amazon S3**
- **Amazon RDS for PostgreSQL**

İhtiyaçlar:

1. **Veri görselleştirmesi** (dashboard, grafik, rapor)
2. **Veri kaynaklarının hepsine bağlanabilme**
3. **Yetkilendirme:**

- Yönetim → **tam erişim**
- Diğer çalışanlar → **sınırlı erişim**

Seçenek Analizi:

B Seçeneği

“Panoları uygun kullanıcılar ve gruplarla paylaşın.”

 **Analiz:**

QuickSight:

- S3'e bağlanır

- RDS PostgreSQL'e bağlanır
- Dashboard üretir
- Kullanıcı ve grup bazlı erişim kontrolü sağlar

QuickSight'ın asıl çalışma şekli budur:

- Veri kaynaklarına bağlanır.
 - Veri kümeleri oluşturur.
 - Dashboard oluşturur.
 - Dashboard erişimi **kullanıcı** ve **grup** seviyesinde yönetilir.
- Yönetim ekibine tam erişim, diğerlerine sınırlı erişim verilebilir.

- ✓ Veri görselleştirme sağlanır.
- ✓ Tüm veri kaynakları desteklenir.
- ✓ Kullanıcı bazlı erişim tam uyumludur.

Doğru cevap budur.

A Seçeneği

“Panoları IAM rollerine paylaşın.”

Analiz:

- QuickSight **IAM rolleriyle paylaşım yapmaz**.
- QuickSight erişimi **QuickSight kullanıcıları ve grupları** üzerinden yönetilir.

Teknik olarak hatalıdır.

Bu nedenle yanlış.

C Seçeneği

S3 + Glue ETL + raporları S3'e koymak

Analiz:

- AWS Glue ETL **raporlama/görselleştirme yapmaz**.
- ETL veri işler, dashboard oluşturmaz.
- S3 üzerindeki raporlar *statik* olur → BI çözümü değildir.
- Erişim yönetimi sınırlıdır.

Gereksinimleri karşılamaz.

D Seçeneği

Glue + Athena + S3'te raporlama

Analiz:

- Athena Federated Query teknik olarak veri kaynaklarına bağlanabilir.
- Ancak yine **dashboard** veya görsel raporlama sunmaz.
- Çıktılar statiktir.
- Yetkilendirme çok sınırlıdır, kullanıcı bazlı görsel rapor kontrolü yoktur.

Görselleştirme ihtiyacını karşılamaz.

SONUÇ

Doğru Cevap: B

Çünkü:

- QuickSight tüm veri kaynaklarına bağlanıyor.
- Dashboard yapıyor.
- Kullanıcı ve grup bazlı erişim yönetimini destekliyor.

QUESTION 17

A company is implementing a new business application. The application runs on two Amazon EC2 instances and uses an Amazon S3 bucket for document storage. A solutions architect needs to ensure that the EC2 instances can access the S3 bucket. What should the solutions architect do to meet this requirement?

- A.** Create an IAM role that grants access to the S3 bucket. Attach the role to the EC2 instances.
- B.** Create an IAM policy that grants access to the S3 bucket. Attach the policy to the EC2 instances.
- C.** Create an IAM group that grants access to the S3 bucket. Attach the group to the EC2 instances.
- D.** Create an IAM user that grants access to the S3 bucket. Attach the user account to the EC2 instances.

Soru:

Bir şirket yeni bir iş uygulaması uyguluyor. Uygulama iki adet Amazon EC2 instance'ında çalışıyor ve belge depolama için bir Amazon S3 bucket'ı kullanıyor. Bir çözüm mimarının, EC2 instance'larının S3 bucket'a erişebildiğinden emin olması gerekiyor. Bu gereksinimi karşılamak için çözüm mimarı ne yapmalıdır?

- A. S3 bucket'a erişim veren bir IAM rolü oluşturun. Rolü EC2 instance'larına ekleyin.
- B. S3 bucket'a erişim veren bir IAM politikası oluşturun. Politikayı EC2 instance'larına ekleyin.
- C. S3 bucket'a erişim veren bir IAM grubu oluşturun. Grubu EC2 instance'larına ekleyin.
- D. S3 bucket'a erişim veren bir IAM kullanıcısı oluşturun. Kullanıcı hesabını EC2 instance'larına ekleyin.

Soru Analizi:

Bir şirket bir iş uygulamasını kullanıyor. Bu uygulama **iki adet Amazon EC2 instance'ında** çalışıyor ve belgeler **Amazon S3 bucket'ında saklanıyor**.

Çözüm mimarının görevi:

- **EC2 instance'larının S3 bucket'a erişebilmesini sağlamak.**

AWS'de EC2'nin S3'e güvenli ve önerilen şekilde erişmesi için **IAM role (rol)** kullanılır. Bu rol, EC2 instance'ına atanır ve EC2 üzerindeki uygulamalar **geçici güvenlik kimlik bilgileri** alarak S3'e erişebilir.

Seçenek Analizi:

- A. S3 bucket'a erişim veren bir IAM rolü oluşturun. Rolü EC2 instance'larına ekleyin.

EC2 instance profile içinde bir IAM rol bulunur. Rol EC2'ye atanır.

Bu yöntem:

- Güvenli
- Otomatik olarak dönen geçici kimlik bilgileri kullanır
- Anahtar yönetimi gerektirmez

- B. S3 bucket'a erişim veren bir IAM politikası oluşturun. Politikayı EC2 instance'larına ekleyin.

IAM politikaları **doğrudan EC2 instance'larına atanamaz**.

Politikalar bir role, kullanıcıya veya gruba eklenebilir; doğrudan instance'a değil.

- C. S3 bucket'a erişim veren bir IAM grubu oluşturun. Grubu EC2 instance'larına ekleyin.

IAM grupları **yalnızca kullanıcıları** organize etmek için vardır.

EC2 instance'larına grup eklenmez.

 D. S3 bucket'a erişim veren bir IAM kullanıcısı oluşturun. Kullanıcı hesabını EC2 instance'larına ekleyin.

Bu yöntem:

- Access key ve secret key'i makineye koymak gereklidir
- Güvenlik açısından kötü praksistir
- AWS tarafından önerilmez

 Sonuç:

Doğru cevap A seçeneğidir.

IAM rolü oluşturulup EC2 instance'larına atanmalıdır.

QUESTION 18

An application development team is designing a microservice that will convert large images to smaller, compressed images. When a user uploads an image through the web interface, the microservice should store the image in an Amazon S3 bucket, process and compress the image with an AWS Lambda function, and store the image in its compressed form in a different S3 bucket.

A solutions architect needs to design a solution that uses durable, stateless components to process the images automatically.

Which combination of actions will meet these requirements? (Choose two.)

- A.** Create an Amazon Simple Queue Service (Amazon SQS) queue. Configure the S3 bucket to send a notification to the SQS queue when an image is uploaded to the S3 bucket.
- B.** Configure the Lambda function to use the Amazon Simple Queue Service (Amazon SQS) queue as the invocation source. When the SQS message is successfully processed, delete the message in the queue.
- C.** Configure the Lambda function to monitor the S3 bucket for new uploads. When an uploaded image is detected, write the file name to a text file in memory and use the text file to keep track of the images that were processed.
- D.** Launch an Amazon EC2 instance to monitor an Amazon Simple Queue Service (Amazon SQS) queue. When items are added to the queue, log the file name in a text file on the EC2 instance and invoke the Lambda function.
- E.** Configure an Amazon EventBridge (Amazon CloudWatch Events) event to monitor the S3 bucket. When an image is uploaded, send an alert to an Amazon

ample Notification Service (Amazon SNS) topic with the application owner's email address for further processing.

Soru:

Bir uygulama geliştirme ekibi, büyük görüntülerin daha küçük ve sıkıştırılmış görüntülere dönüştürecek bir mikro servis tasarlıyor. Bir kullanıcı web arayüzü üzerinden bir görüntü yüklediğinde, mikro servis görüntütü bir Amazon S3 bucket'ına kaydetmeli, AWS Lambda fonksiyonu ile görüntütü işleyip sıkıştırmalı ve sıkıştırılmış görüntütü farklı bir S3 bucket'ına kaydetmelidir. Bir çözüm mimarının, görüntülerin otomatik olarak işlemek için dayanıklı (durable), durumsuz (stateless) bileşenler kullanan bir çözüm tasarlaması gerekiyor.

Hangi işlem kombinasyonu bu gereksinimleri karşılar? (İki seçenek seçiniz.)

- A.** Bir Amazon Simple Queue Service (Amazon SQS) kuyruğu oluşturun. S3 bucket'a bir görüntü yüklenliğinde S3 bucket'ın SQS kuyruğuna bir bildirim göndemesini yapılandırın.
- B.** Lambda fonksiyonunu Amazon Simple Queue Service (Amazon SQS) kuyruğunu çağrı kaynağı (invocation source) olarak kullanacak şekilde yapılandırın. SQS mesajı başarıyla işlendiğinde, kuyruğun içindeki mesajı silin.
- C.** Lambda fonksiyonunu yeni yüklemeler için S3 bucket'ı izleyecek şekilde yapılandırın. Bir görüntü yüklenliğinde, dosya adını bellekte bir metin dosyasına yazın ve işlenen görüntülerin takip etmek için bu metin dosyasını kullanın.
- D.** Bir Amazon EC2 instance'ı başlatın ve Amazon Simple Queue Service (Amazon SQS) kuyruğunu izlemesini sağlayın. Kuyruğa öğeler eklendiğinde, dosya adını EC2 instance'ında bir metin dosyasına kaydedin ve Lambda fonksiyonunu tetikleyin.
- E.** Bir Amazon EventBridge (Amazon CloudWatch Events) olayı yapılandırın. Bir görüntü yüklenliğinde, uygulama sahibinin e-posta adresine daha fazla işlem için bir Amazon Simple Notification Service (Amazon SNS) konusuna (topic) uyarı gönderin.

Soru Analizi:

Gereksinimler:

- Büyük görüntüler yükleniyor.
- Yüklenen görüntüler **S3 bucket'a kaydedilecek**.
- AWS Lambda görüntütü **otomatik olarak işleyecek ve sıkıştıracak**.
- Sıkıştırılmış görüntütü **başka bir S3 bucket'a kaydedilecek**.
- Çözüm **dayanıklı (durable)** ve **durumsuz (stateless)** olmalı.

- Mikro servis tamamen **otomatik** çalışmalı.

AWS'de duruma göre:

- S3 → SQS → Lambda, **en dayanıklı ve en yaygın mimaridir.**
- Bunun nedeni:
 - S3 event notification bazen Lambda'ya *at least once* tetikleme ile iletir, kaçırma ihtimali vardır.
 - **SQS**, dayanıklı bir kuyruktur ve olayları kaybetmez.
 - **Lambda**, SQS'den mesaj alarak işlemleri otomatik ve güvenilir biçimde yürütür.

Bu nedenle ideal çözüm:

- S3, yükleme olduğunda **SQS'ye** bildirim göndersin
- Lambda, **SQS'yi dinleyerek** mesajları işlesin

Bu kombinasyon **tamamen serverless, stateless** ve **durable** bir çözümdür.

Seçenek Analizi:

✓ A. S3 → SQS bildirim ayarla

- S3, doğrudan Lambda'yı tetiklemek yerine SQS'ye bildirim yollarsa:
 - İletilerin kaybolma riski azalır.
 - İş yükü sık olduğunda Lambda kuyruğu kontrollü şekilde tüketir.
 - Bu, “durable + stateless + scalable” modeline uygundur.

✓ B. Lambda'yı SQS kaynaklı tetikle

- Lambda, SQS kuyruğunu otomatik çeker ve mesajları işler.
- İşlem başarılı olursa mesaj otomatik silinir.
- Lambda tamamen **stateless** çalışır.
- Kuyruk sayesinde iş yükü **dayanıklı** hale gelir.

✗ C. Lambda bellekte liste tutsun

- Lambda fonksiyonları **durumsuzdur**: her çağrıda belleği sıfırlanabilir.
- Bellekte dosya tutmak **dayanıklı değildir**.
- Mikro servis ma... imarisine ve gereksinimlere aykırıdır.

Bir EC2 başlat, kuyruk dinlesin, Lambda tetiklesin

- EC2 kullanmak gereksizdir (serverless ilkelerine aykırı).
- Dayanıklılığı EC2 üstüne almak hatalı tasarımdır.
- Mikro servis ve stateless gereksinimine uymaz.

E. EventBridge → SNS → e-posta

- E-mail gönderimi gereksizdir.
- Bu işlem akışını tetiklemek için kullanılmaz.
- “Otomatik görüntü işleme” gereksinimine katkı sağlamaz.

DOĞRU CEVAP KOMBİNASYONU

A ve B

Bu kombinasyon:

- Dayanıklıdır (SQS).
- Tamamen stateless'tir (Lambda).
- Serverless'tir (EC2 yok).
- AWS tarafından **önerilen mimaridir**.

QUESTION 19

A company has a three-tier web application that is deployed on AWS. The web servers are deployed in a public subnet in a VPC. The application servers and database servers are deployed in private subnets in the same VPC. The company has deployed a third-party virtual firewall appliance from AWS Marketplace in an inspection VPC. The appliance is configured with an IP interface that can accept IP packets.

A solutions architect needs to integrate the web application with the appliance to inspect all traffic to the application before the traffic reaches the web server.

Which solution will meet these requirements with the LEAST operational overhead?

- A.** Create a Network Load Balancer in the public subnet of the application's VPC to route the traffic to the appliance for packet inspection.
- B.** Create an Application Load Balancer in the public subnet of the application's VPC to route the traffic to the appliance for packet inspection.
- C.** Deploy a transit gateway in the inspection VPCConfigure route tables to route the incoming packets through the transit gateway.

D. Deploy a Gateway Load Balancer in the inspection VPC. Create a Gateway Load Balancer endpoint to receive the incoming packets and forward the packets to the appliance.

Soru:

Bir şirketin AWS üzerinde dağıtılmış üç katmanlı bir web uygulaması vardır. Web sunucuları bir VPC'nin genel (public) alt ağında bulunur. Uygulama sunucuları ve veritabanı sunucuları aynı VPC'nin özel (private) alt ağlarında bulunur. Şirket, AWS Marketplace'ten bir üçüncü taraf sanal güvenlik duvarı (virtual firewall) cihazını bir **inspection VPC içinde** dağıtmıştır. Bu cihaz, IP paketlerini kabul edebilen bir IP arabirimleri ile yapılandırılmıştır.

Bir çözüm mimarının, web uygulamasına gelen tüm trafiğin **web sunucusuna ulaşmadan önce** bu güvenlik cihazı tarafından denetlenmesini sağlayacak bir entegrasyon tasarlaması gerekiyor.

Hangi çözüm, **en az operasyonel yük** ile bu gereksinimleri karşılar?

- A.** Uygulamanın VPC'sinin genel alt ağına (public subnet) bir Network Load Balancer oluşturun ve trafiği paket incelemesi için güvenlik cihazına yönlendirin.
- B.** Uygulamanın VPC'sinin genel alt ağına (public subnet) bir Application Load Balancer oluşturun ve trafiği paket incelemesi için güvenlik cihazına yönlendirin.
- C.** Inspection VPC içinde bir Transit Gateway dağıtın. Gelen paketleri Transit Gateway üzerinden yönlendirmek için yönlendirme tablolarını yapılandırın.
- D.** Inspection VPC içinde bir Gateway Load Balancer dağıtın. Gelen paketleri almak ve paketleri güvenlik cihazına iletmek için bir Gateway Load Balancer endpoint oluşturun.

Soru Analizi:

Şirketin üç katmanlı bir web uygulaması var:

- **Web sunucuları → Public subnet**
- **App + DB sunucuları → Private subnet**
- **Harici bir güvenlik duvarı (firewall appliance) → Inspection VPC**

Amaç:

→ **Web sunucusuna gelen tüm trafiği**, web sunucusuna ulaşmadan önce **3. parti firewall appliance** üzerinden geçirmek.

Ek kısıt:

→ **En az operasyonel yük** ile çözüm isteniyor.

Seçenek Analizi:

- D. Inspection VPC'de bir Gateway Load Balancer dağıtın. GWLBe ile trafiği appliance'a yönlendirin.

Bu durum için AWS'nin önerdiği servis:

 **Gateway Load Balancer (GWLB)**

Çünkü:

- Üçüncü parti güvenlik cihazları ile entegrasyon için doğrudan tasarlanmıştır.
- Inline trafik inspection sağlar.
- Otomatik ölçeklenir.
- Çok az operasyonel yük gerektirir.
- Appliance VPC ile uygulama VPC arasında **Gateway Load Balancer Endpoint (GWLBe)** kullanılır.

Doğru — En az operasyonel yük

- AWS'nin bu kullanım senaryosu için önerdiği mimari
 Inline paket inspection için tasarlanmış tek load balancer türü

Avantajları:

- Transparan trafik geçişi sağlar (bump-in-the-wire).
- Sadece firewall VPC'sine GWLBe eklenir → minimum konfigürasyon.
- İş yükü arttıkça otomatik ölçeklenir.
- Üçüncü taraf güvenlik cihazları ile yerleşik uyumluluk.
- Operasyonel yük **en düşük**.

Bu nedenle **kesin doğru cevap budur**.

 **A. Public subnet'te bir Network Load Balancer oluşturun ve trafiği firewall'a yönlendirin.**

Neden?

- NLB, üçüncü taraf firewall cihazları için özel olarak tasarlanmış değildir.
- Inline trafik yönlendirme mekanizması **manuel konfigürasyon** ister → yüksek operasyonel yük.
- VPC-to-VPC paket inspection için uygun değildir.
- NLB, firewall appliance ölçeklendirmesini yönetemez.

Bu nedenle gereksinimi doğru ama **optimize olmayan** bir şekilde çözer.

 **B. Public subnet'te bir Application Load Balancer oluşturun ve trafiği firewall'a yönlendirin.**

- ALB L7 (HTTP/HTTPS) load balancer'dır; paket seviyesinde (L3/L4) firewall inspection yapamaz.
- Üçüncü parti güvenlik appliance'ları ALB üzerinden inline çalışmaz.
- ALB → Firewall routing mimari olarak hatalıdır.

Bu seçenek tamamen uygunsuzdur.

 **C. Inspection VPC'de bir Transit Gateway dağıtin ve route table'ları ayarlayın.**

Neden?

- Transit Gateway sadece yönlendirme yapar; paketleri otomatik olarak firewall appliance'a inline göndermez.
- Trafiki firewall üzerinden geçirmek için karmaşık route table konfigürasyonu gereklidir.
- **Yüksek operasyonel yük.**
- Otomatik ölçeklenme sağlanamaz.

Uygulanabilir ama **zorludur** ve AWS'nin önerdiği çözüm değildir.

 **SONUÇ:**

- ✓ AWS'nin resmi dokümanlarında üçüncü taraf firewall entegrasyonu için önerilen yöntemdir
- ✓ Inline trafik denetimi için tek uygun Load Balancer türüdür
- ✓ PrivateLink destekli endpoint sayesinde minimal konfigürasyon
- ✓ Otomatik ölçeklenme ve basit sağlık kontrolü
- ✓ En düşük operasyonel yük
- ✓ En güvenli ve modern çözüm

QUESTION 20

A company wants to improve its ability to clone large amounts of production data into a test environment in the same AWS Region. The data is stored in Amazon EC2 instances on Amazon Elastic Block Store (Amazon EBS) volumes. Modifications to the cloned data must not affect the production environment. The software that accesses this data requires consistently high I/O performance.

A solutions architect needs to minimize the time that is required to clone the production

data into the test environment.

Which solution will meet these requirements?

- A. Take EBS snapshots of the production EBS volumes. Restore the snapshots onto EC2 instance store volumes in the test environment.
- B. Configure the production EBS volumes to use the EBS Multi-Attach feature. Take EBS snapshots of the production EBS volumes. Attach the production EBS volumes to the EC2 instances in the test environment.
- C. Take EBS snapshots of the production EBS volumes. Create and initialize new EBS volumes. Attach the new EBS volumes to EC2 instances in the test environment before restoring the volumes from the production EBS snapshots.
- D. Take EBS snapshots of the production EBS volumes. Turn on the EBS fast snapshot restore feature on the EBS snapshots. Restore the snapshots into new EBS volumes. Attach the new EBS volumes to EC2 instances in the test environment.

Soru:

Bir şirket, aynı AWS Bölgesi içinde üretim ortamındaki büyük mikardaki veriyi test ortamına kopyalama (clone) yeteneğini geliştirmek istiyor.

Veriler, Amazon EC2 instance'larında Amazon Elastic Block Store (Amazon EBS) volume'lari üzerinde depolanıyor. Kopyalanan (clone edilen) verilerde yapılacak değişikliklerin **ürtim ortamını etkilememesi** gerekiyor.

Bu veriye erişen yazılımın **sürekli yüksek I/O performansına** ihtiyacı var.

Bir çözüm mimarının, üretim verilerini test ortamına kopyalamak için gereken süreyi **en aza indirmesi** gerekiyor.

Hangi çözüm bu gereksinimleri karşılar?

- A. Üretim EBS volume'larının EBS snapshot'larını alın. Snapshot'ları test ortamında EC2 instance store volume'larına geri yükleyin.
- B. Üretim EBS volume'larını EBS Multi-Attach özelliğini kullanacak şekilde yapılandırın. Üretim EBS volume'larının snapshot'larını alın. Üretim EBS volume'larını test ortamındaki EC2 instance'larına bağlayın.
- C. Üretim EBS volume'larının snapshot'larını alın. Yeni EBS volume'ları oluşturun ve başlatın. Üretim snapshot'larını geri yüklemeden önce yeni EBS volume'larını test ortamındaki EC2 instance'larına bağlayın.
- D. Üretim EBS volume'larının snapshot'larını alın. EBS snapshot'larında EBS Fast Snapshot Restore özelliğini etkinleştirin. Snapshot'ları yeni EBS volume'larına geri yükleyin. Yeni EBS volume'larını test ortamındaki EC2 instance'larına bağlayın.

Soru Analizi:

Şirketin ihtiyacı:

- **Aynı AWS Region içinde**, üretim verilerini **çok hızlı** şekilde test ortamina klonlamak.
- Veri, EC2 üzerindeki **EBS volume'larında** bulunuyor.
- Klonlanan veriler üzerinde değişiklik yapılabılır ama **üretim ortamı etkilenmemelidir**.
- Veriye erişen uygulama **yüksek ve tutarlı I/O performansı** gerektiriyor.
- Gereksinim: **klonlama süresini minimuma indirmek**.

Seçenek Analizi:

- D. Snapshot al → Fast Snapshot Restore (FSR) etkinleştir → Yeni EBS volume'lara restore et

Bu tür durumda AWS'nin sunduğu en hızlı EBS snapshot tabanlı çözüm:

★ EBS Fast Snapshot Restore (FSR)

FSR, snapshot'tan geri yüklenen EBS volume'larının *hemen, tam performansla* kullanılmasını sağlar.

FSR yoksa, EBS snapshot'tan restore edilen volume'lar “lazy loading / first-touch latency” nedeniyle yavaş başlar.

Bu nedenle çözümün anahtarı:

- Snapshot al → FSR aktif et → Volume hızlı ve tam performansla oluştur

- ✓ **Doğru — En hızlı ve en modern çözüm**
- ✓ AWS Best Practice
- ✓ Gereksinimlerin tümünü karşılar

Neden?

- FSR, snapshot'tan oluşturulan volume'ların **anında tam performansla** kullanılmasını sağlar.
- Lazy loading yoktur → baştan sona yüksek I/O.
- Production snapshot → Test volume tamamen ayrı, izole çalışır → üretim etkilenmez.
- Klonlama süresini **en çok azaltan** mekanizmadır.

Bu seçenek tek başına tüm gereksinimleri karşılar.

A. Snapshot al → Instance store'a restore et

- Instance store **geçici (ephemeral)** disktir.
- Snapshot → instance store restore **mümkün değildir** (EBS snapshot → EBS volume'a restore edilir).
- Ayrıca instance store volume'ları:
 - Sürdürülebilir değildir
 - Ölçeklenemez
 - Yüksek maliyetli ve karmaşık

Bu seçenek hem teknik olarak hatalı hem gereksiz.

B. Multi-Attach kullan → Üretim EBS volume'larını test ortamına bağla

- Multi-Attach sadece **EBS io2** volume'ları için geçerlidir.
- **Production volume'unu birden çok EC2 instance'a bağlamak**, veri bütünlüğü açısından yüksek risklidir.
- Ayrıca test ortamındaki değişiklikler **üretimi etkiler**, bu da gereksinimlere aykırı.

Üstelik bu seçenek klonlama süresini **kısaltmaz**.

C. Snapshot al → Yeni volume oluştur → Hem restore et hem önce attach et

- Normal snapshot restore işlemi **yavaş başlar** (lazy loading).
- Restore tamamlanmadan volume tam performansta olmaz.
- Volume'ü EC2'ye attach etmek klonlamayı hızlandırmaz.
- Test ortamına hızlı ve tam performanslı volume sağlama gereksinimi karşılanmaz.

Bu çözüm çalışır ama istenen **hız ve performans** yoktur.

SONUÇ: DOĞRU CEVAP → D seçeneği

D. EBS Fast Snapshot Restore (FSR) kullanmak, hem klonlamayı hızlandırır hem de tam I/O performansı sağlar.

Ayrıca üretim ortamı etkilenmez, test ortamı bağımsız volume kullanır.

QUESTION 21

An ecommerce company wants to launch a one-deal-a-day website on AWS. Each day will feature exactly one product on sale for a period of 24 hours. The company wants to be able to handle millions of requests each hour with millisecond latency during peak

hours. Which solution will meet these requirements with the LEAST operational overhead?

- A.** Use Amazon S3 to host the full website in different S3 buckets. Add Amazon CloudFront distributions. Set the S3 buckets as origins for the distributions. Store the order data in Amazon S3.
- B.** Deploy the full website on Amazon EC2 instances that run in Auto Scaling groups across multiple Availability Zones. Add an Application Load Balancer (ALB) to distribute the website traffic. Add another ALB for the backend APIs. Store the data in Amazon RDS for MySQL.
- C.** Migrate the full application to run in containers. Host the containers on Amazon Elastic Kubernetes Service (Amazon EKS). Use the Kubernetes Cluster Autoscaler to increase and decrease the number of pods to process bursts in traffic. Store the data in Amazon RDS for MySQL.
- D.** Use an Amazon S3 bucket to host the website's static content. Deploy an Amazon CloudFront distribution. Set the S3 bucket as the origin. Use Amazon API Gateway and AWS Lambda functions for the backend APIs. Store the data in Amazon DynamoDB.

Soru:

Bir e-ticaret şirketi, AWS üzerinde “günde bir fırsat (one-deal-a-day)” web sitesi başlatmak istiyor. Her gün, tam 24 saat boyunca satışa olacak tek bir ürün gösterecek. Şirket; yoğun saatlerde, saatte milyonlarca isteği milisaniyelik gecikmelerle karşılaşabilmek istiyor. En AZ operasyonel yönetim yüküyle (least operational overhead) hangi çözüm bu gereksinimleri karşılar?

- A.** Amazon S3'ü kullanarak tam web sitesini farklı S3 bucket'larında barındırın. Amazon CloudFront dağıtımları ekleyin. S3 bucket'larını bu dağıtımlar için origin olarak ayarlayın. Sipariş verilerini Amazon S3'te saklayın.
- B.** Tam web sitesini, birden fazla Erişilebilirlik Alanı (AZ) üzerinde Auto Scaling gruplarında çalışan Amazon EC2 instance'larında dağıtan. Web sitesi trafiğini dağıtmak için bir Application Load Balancer (ALB) ekleyin. Arka uç API'ler için başka bir ALB ekleyin. Verileri Amazon RDS for MySQL'de saklayın.
- C.** Tüm uygulamayı konteynerlere taşıyın. Konteynerleri Amazon Elastic Kubernetes Service (Amazon EKS) üzerinde barındırın. Trafik patlamalarını işlemek için Kubernetes Cluster Autoscaler kullanarak pod sayılarını artırıp azaltın. Verileri Amazon RDS for MySQL'de saklayın.
- D.** Web sitesinin statik içeriğini barındırmak için bir Amazon S3 bucket kullanın. Bir Amazon CloudFront dağıtımını dağıtan. S3 bucket'ını origin olarak ayarlayın. Arka uç API'ler

için Amazon API Gateway ve AWS Lambda işlevleri kullanın. Verileri Amazon DynamoDB'de saklayın.

Soru Analizi:

Bir e-ticaret şirketi **günde bir ürünün satıldığı**, yoğun trafikli bir web sitesi kurmak istiyor.

✓ Gereksinimler:

1. **Saat milyonlarca istek** — çok yüksek trafik
2. **Milisaniye gecikme** — çok düşük latency
3. **AWS üzerinde barındırılacak**
4. **En az operasyonel yük (least operational overhead)**
→ yönetim, bakım, ölçektekleme minimum olmalı
5. Her gün yalnızca **1 ürün** gösteriliyor → site **büyük ve karmaşık değil**
6. Mükemmel ölçülebilirlik gereklidir.

Bu gereksinimlere göre ideal çözüm:

tamamen sunucusuz (serverless), otomatik ölçeklenen, düşük yönetim yükü olan mimariler.

Seçenek Analizi:

D. S3 + CloudFront + API Gateway + Lambda + DynamoDB

✓ Neden doğru?

- **En az operasyonel yük** (tamamen serverless)
- **S3 + CloudFront** → statik site için en hızlı ve en ölçülebilir çözüm
- **API Gateway + Lambda** → arka uç API'ler için sunucusuz, otomatik ölçeklenen yapı
- **DynamoDB** → milyonlarca isteği milisaniyelik gecikmeyle karşılar
- Sunucu yönetimi, patching, autoscaling ayarlaması **yok**
- Hem yüksek trafik hem düşük latency gereksinimini karşılar

Bu yüzden **gereksinimlerin tamamını en az operasyonel yükle karşılayan tek seçenek** budur.

A. S3 üzerinde site + CloudFront + sipariş verisi S3'te

Neden yanlış?

- S3 veritabanı değildir → **sipariş verisi için uygun değil**
- API backend yok
- Transaction, concurrency, sorgulama ihtiyaçlarını karşılamaz

X B. EC2 + Auto Scaling + ALB + RDS MySQL

Neden yanlış?

- Sunucuların yönetimi **yüksek operasyonel yük gerektirir**
- RDS MySQL yatay ölçeklenemez → milyonlarca istekte darboğaz oluşur
- En az operasyonel yük koşuluna uymaz

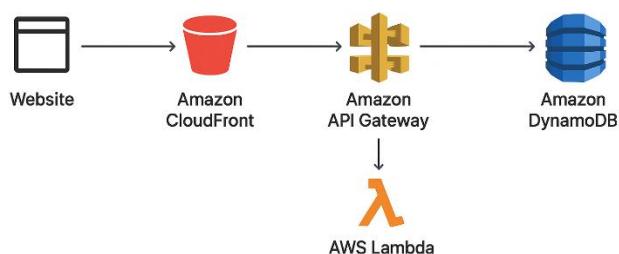
X C. EKS (Kubernetes) + RDS MySQL

Neden yanlış?

- EKS = **en yüksek operasyonel yük**
 - cluster yönetimi
 - node provisioning
 - autoscaler ayarları
 - upgrade ve bakım süreçleri
- RDS MySQL yine büyük trafik yükünde sınırlı

Soru özellikle en az operasyonel yük istiyor → bu seçenek en fazla yüke sahip.

🎯 SONUÇ



QUESTION 22

A solutions architect is using Amazon S3 to design the storage architecture of a new digital media application. The media files must be resilient to the loss of an Availability Zone. Some files are accessed frequently while other files are rarely accessed in an unpredictable pattern. The solutions architect must minimize the costs of storing and retrieving the media files. Which storage option meets these requirements?

- A. S3 Standard**
- B. S3 Intelligent-Tiering**
- C. S3 Standard-Infrequent Access (S3 Standard-IA)**
- D. S3 One Zone-Infrequent Access (S3 One Zone-IA)**

Soru:

Bir çözüm mimarı, yeni bir dijital medya uygulamasının depolama mimarisini tasarlamak için Amazon S3 kullanıyor. Medya dosyaları bir Erişilebilirlik Alanının (Availability Zone) kaybına karşı dayanıklı olmalıdır. Bazı dosyalara sık erişilirken bazı dosyalara ise öngörelemeyen bir şekilde nadiren erişilmektedir. Çözüm mimarı, medya dosyalarını depolama ve alma maliyetlerini en aza indirmelidir. Hangi depolama seçeneği bu gereksinimleri karşılar?

- A. S3 Standard**
- B. S3 Intelligent-Tiering**
- C. S3 Standard-Infrequent Access (S3 Standard-IA)**
- D. S3 One Zone-Infrequent Access (S3 One Zone-IA)**

Soru Analizi:

Bir dijital medya uygulaması için depolama mimarisi tasarlanyor. Gereksinimler:

✓ 1. Bir Availability Zone kaybına dayanıklı olmalı

- Yani **multi-AZ** dayanıklılığı şart.
- One Zone depolama türleri otomatik olarak eleniyor.

✓ 2. Dosyaların erişim şekli öngörelemez

- Bazı dosyalar sık okunuyor
- Bazıları ara sıra, düzensiz şekilde okunuyor

Bu durumda hangi tier'e ne zaman geçileceğini kestirmek zor.

✓ 3. Depolama ve retrieval maliyeti minimum olmalı

Yani hem ucuz depolama hem de erişim maliyetlerinin dengeli olması gerekiyor.

Seçenek Analizi:

B Seçeneği: S3 Intelligent-Tiering

✓ Neden doğru?

- **Multi-AZ dayanıklılığı vardır** → AZ kaybına dayanıklıdır.
- Erişim sıklığı **öngörülemez** durumlar için tasarlanmış tek S3 sınıfıdır.
- Dosyanın erişim modeline göre **otomatik olarak**:
 - sık erişilen tier
 - nadir erişilen tier
 - archive tiersarasında dosyayı kendisi taşıır.
- Kullanıcı ekstra bir şey yönetmez → yönetim maliyeti yok.
- Erişim maliyeti makuldür ve çoğu durumda toplam maliyet en düşüktür.

👉 Yani **en düşük toplam maliyet + AZ dayanıklılığı + karmaşık olmayan yönetim** = Intelligent-Tiering.

A Seçeneği: S3 Standard

Neden yanlış?

- Erişim sıklığı bilinmiyorsa pahalı olabilir.
- Gereksiz yere yüksek maliyet ödenir.

C Seçeneği: S3 Standard-IA

Neden yanlış?

- Erişim modeli **öngörülemez** deniyor.
- IA erişim maliyetleri yüksektir → Beklenmeyen erişimlerde **çok pahalıya patlar**.
- Ayrıca IA'ya dosyayı *manuel* geçirmelisiniz, otomatik değildir.

D Seçeneği: S3 One Zone-IA

Neden yanlış?

- **Single-AZ'dir** →
Soru doğrudan: "AZ kaybına dayanıklı olmalı" diyor.
→ Bu nedenle *hiçbir şekilde uygun değil*.

SONUÇ

Depolama Sınıfı	AZ kaybına dayanıklı mı? Açıklama
S3 Standard	✓ 3+ AZ replikasyonu
S3 Intelligent-Tiering	✓ Otomatik katmanlama + 3+ AZ
S3 Standard-IA	✓ Düşük erişim + 3+ AZ
S3 One Zone-IA	✗ Tek AZ, kayıp olursa veri gider

Doğru seçenek:

B. S3 Intelligent-Tiering

QUESTION 23

A company is storing backup les by using Amazon S3 Standard storage. The les are accessed frequently for 1 month. However, the les are not accessed after 1 month. The company must keep the les inde nitely. Which storage solution will meet these requirements MOST cost-effectively?

- A.** Configure S3 Intelligent-Tiering to automatically migrate objects.
- B.** Create an S3 Lifecycle configuration to transition objects from S3 Standard to S3 Glacier Deep Archive after 1 month. Topic 1
- C.** Create an S3 Lifecycle configuration to transition objects from S3 Standard to S3 Standard-Infrequent Access (S3 Standard-IA) after 1 month.
- D.** Create an S3 Lifecycle configuration to transition objects from S3 Standard to S3 One Zone-Infrequent Access (S3 One Zone-IA) after 1 month.

Soru:

Bir şirket, Amazon S3 Standard depolama sınıfını kullanarak yedekleme dosyalarını depoluyor. Dosyalara **1 ay boyunca sık sık erişiliyor**. Ancak 1 aydan sonra dosyalara **erişim yapılmıyor**. Şirket bu dosyaları **süresiz olarak saklamak** zorunda.

Bu gereksinimleri **en maliyet etkin şekilde** karşılayacak depolama çözümü hangisidir?

- A.** Nesnelerin otomatik olarak taşınması için S3 Intelligent-Tiering yapılandırın.
- B.** Bir S3 Yaşam Döngüsü (Lifecycle) yapılandırması oluşturarak nesneleri 1 ay sonra S3 Standard'dan S3 Glacier Deep Archive'a taşıyın.
- C.** Bir S3 Yaşam Döngüsü yapılandırması oluşturarak nesneleri 1 ay sonra S3 Standard'dan S3 Standard-Infrequent Access (S3 Standard-IA) sınıfına taşıyın.

D. Bir S3 Yaşam Döngüsü yapılandırması oluşturarak nesneleri 1 ay sonra S3 Standard'dan S3 One Zone-Infrequent Access (S3 One Zone-IA) sınıfına taşıyın.

Soru Analizi:

Şirket:

- Dosyalara **ilk 30 gün boyunca sık erişiyor**
- 30 günden sonra **hiç erişmiyor**
- Dosyalar **süresiz saklanacak**
- Amaç: **en düşük maliyet**

Bu durumda ilk 30 gün için S3 Standard mantıklı; çünkü sık erişiliyor.

1 aydan sonra:

- Dosyalara erişim gerekmıyor
- Çok uzun süre tutulacak
- Bu da en ucuz uzun süreli arşiv sınıfını uygun hale getirir → **S3 Glacier Deep Archive**

AWS'de **en düşük maliyetli depolama sınıfı**:

👉 **S3 Glacier Deep Archive**

Dolayısıyla 30 gün sonra burada saklamak en ekonomik çözümdür.

Seçenek Analizi:

Doğru Seçenek: B

✓ S3 Lifecycle ile 30 gün sonra S3 Glacier Deep Archive'e geçiş

- En ucuz uzun vadeli saklama çözümüdür
- Erişim gerekmiyor → retrieve süresinin uzun olmasının önemi yok
- Süresiz depo için en düşük maliyet sağlar

✗ A — S3 Intelligent-Tiering

Neden yanlış?

- Intelligent-Tiering erişim desenleri bilinmediğinde faydalı
- Ama "1 ay sık - sonra hiç erişim yok" **bilinen ve belirgin** bir pattern
- Ayrıca Intelligent-Tiering uzun vadede **Deep Archive kadar ucuz değildir**

✗ C — S3 Standard-IA

Neden yanlış?

- Standard-IA, Deep Archive'den **çok daha pahalıdır**
- 1 aydan sonra hiç erişilmeyecek dosyalar için gereksiz bir maliyettir

D — S3 One Zone-IA

Neden yanlış?

- Tek AZ depolama → **AZ kaybına dayanıklı değil**
- Yedekleme dosyaları için riskli
- Ayrıca Deep Archive'den yine daha pahalıdır

SONUÇ:

En maliyet-etkin çözüm

S3 Lifecycle ile 30 gün sonra S3 Glacier Deep Archive'e geçiş

QUESTION 24

A company observes an increase in Amazon EC2 costs in its most recent bill. The billing team notices unwanted vertical scaling of instance types for a couple of EC2 instances. A solutions architect needs to create a graph comparing the last 2 months of EC2 costs and perform an in-depth analysis to identify the root cause of the vertical scaling. How should the solutions architect generate the information with the LEAST operational overhead?

- A.** Use AWS Budgets to create a budget report and compare EC2 costs based on instance types.
- B.** Use Cost Explorer's granular filtering feature to perform an in-depth analysis of EC2 costs based on instance types. Topic 1
- C.** Use graphs from the AWS Billing and Cost Management dashboard to compare EC2 costs based on instance types for the last 2 months.
- D.** Use AWS Cost and Usage Reports to create a report and send it to an Amazon S3 bucket. Use Amazon QuickSight with Amazon S3 as a source to generate an interactive graph based on instance types.

Soru:

Bir şirket, en son faturasında Amazon EC2 maliyetlerinde bir artış fark ediyor. Faturalama ekibi, bazı EC2 örnekleri için istenmeyen dikey ölçeklendirme (vertical scaling) yapıldığını tespit ediyor. Bir çözüm mimarının, son 2 aya ait EC2 maliyetlerini

karşılaştırılan bir grafik oluşturması ve dikey ölçeklendirmenin temel nedenini belirlemek için ayrıntılı bir analiz gerçekleştirmesi gerekiyor. Çözüm mimarı, bu bilgiyi **en az operasyonel yükle** nasıl oluşturmalıdır?

- A. AWS Budgets kullanarak bir bütçe raporu oluşturun ve EC2 maliyetlerini instance türlerine göre karşılaştırın.
- B. Cost Explorer'ın ayrıntılı filtreleme özelliğini kullanarak instance türlerine göre EC2 maliyetlerinin derinlemesine analizini yapın.
- C. AWS Billing and Cost Management panosundaki grafiklerle, son 2 aya ait EC2 maliyetlerini instance türlerine göre karşılaştırın.
- D. AWS Cost and Usage Reports kullanarak bir rapor oluşturun ve bunu bir Amazon S3 bucket'ına gönderin. Amazon QuickSight kullanarak S3'ü kaynak olarak bağlayın ve instance türlerine göre etkileşimli bir grafik oluşturun.

Soru Analizi:

Bir şirket, son faturasında EC2 maliyetlerinin arttığını fark ediyor.

Faturalama ekibi, birkaç EC2 instance'ında istenmeyen **vertical scaling** (daha büyük instance tipine geçiş) olduğunu görüyor.

Bir çözüm mimarı şu iki şeyi yapmak zorunda:

1. **Son 2 ayın EC2 maliyetlerini karşılaştırın bir grafik oluşturmak**
2. **Dikey ölçeklemenin nedenini bulmak için derinlemesine mal yet analizi yapmak**

Ayrıca:

👉 **En az operasyonel yükle (least operational overhead)** yapılmalı.

Seçenek Analizi:

- **B. Cost Explorer'ın ayrıntılı filtreleme özelliğini kullanarak instance türlerine göre derinlemesine analiz yapmak.**

✓ Neden doğru?

- Cost Explorer:
 - Son 2 ayın mal yet grafiğini anında çıkarabilir.
 - Instance type bazlı filtreleme yapabilir.
 - Dikey ölçekleme (vertical scaling) davranışını gösterir.
- Tamamen yönetilen bir servistir → en az operasyonel yük.
- Ek bir araç, rapor oluşturma, veri taşıma veya görselleştirme kurulumu gerekmez.

Bu yüzden gereksinimleri en hızlı ve en düşük operasyonel maliyetle karşılayan tek çözümüdür

A. AWS Budgets ile rapor oluşturmak

Neden yanlış?

- AWS Budgets:
 - Bütçe limitleri ve uyarılar içindir.
 - Derinlemesine analiz, instance-type bazlı grafik sunmaz.
- Dikey ölçekte ölçümleme davranışını göstermez.

Grafik oluşturmak için uygun araç değildir.

C. Billing & Cost Management dashboard grafikleri

Neden yanlış?

- Dashboard basit özet grafikleri gösterir.
- Instance-type bazlı detaylı filtreleme **sunmaz**.
- Dikey ölçekte ölçümleme analiz edecek ayrıntı seviyesi yoktur.

D. Cost & Usage Report (CUR) + S3 + QuickSight

Neden yanlış?

- En yüksek operasyonel yük:
 - CUR oluşturma
 - S3 bucket yönetimi
 - QuickSight veri kaynağı ayarlama
 - Dashboard oluşturma
- Çok karmaşık.

Yüksek esneklik sunar ama gereksiz derecede fazla operasyonel iş.

SONUÇ

 **Doğru cevap: B — Cost Explorer ile ayrıntılı maliyet analizi**

A company is designing an application. The application uses an AWS Lambda function to receive information through Amazon API Gateway and to store the information in an Amazon Aurora PostgreSQL database. During the proof-of-concept stage, the company has to increase the Lambda quotas significantly to handle the high volumes of data that the company needs to load into the database. A solutions architect must recommend a new design to improve scalability and minimize the configuration effort. Which solution will meet these requirements?

- A.** Refactor the Lambda function code to Apache Tomcat code that runs on Amazon EC2 instances. Connect the database by using native Java Database Connectivity (JDBC) drivers.
- B.** Change the platform from Aurora to Amazon DynamoDB Provision a DynamoDB Accelerator (DAX) cluster. Use the DAX client SDK to point the existing DynamoDB API calls at the DAX cluster.
- C.** Set up two Lambda functions. Configure one function to receive the information. Configure the other function to load the information into the database. Integrate the Lambda functions by using Amazon Simple Notification Service (Amazon SNS).
- D.** Set up two Lambda functions. Configure one function to receive the information. Configure the other function to load the information into the database. Integrate the Lambda functions by using an Amazon Simple Queue Service (Amazon SQS) queue.

Soru:

Bir şirket bir uygulama tasarlıyor. Uygulama, Amazon API Gateway üzerinden bilgi almak için bir AWS Lambda fonksiyonu kullanıyor ve bu bilgiyi Amazon Aurora PostgreSQL veritabanına kaydediyor. Kavram kanıtı (POC) aşamasında, şirkette veriyi veritabanına yüklemek için gereken yüksek veri hacimleri nedeniyle Lambda kotalarını önemli ölçüde artırmak zorunda kalıyor. Bir çözüm mimarı, ölçeklenebilirliği artıracak ve yapılandırma çabasını en aza indirecek yeni bir tasarım önermelidir. Hangi çözüm bu gereksinimleri karşılar?

- A.** Lambda fonksiyonunun kodunu, Amazon EC2 üzerinde çalışan Apache Tomcat koduna dönüştürün. Veritabanına bağlantıyı yerel Java Database Connectivity (JDBC) sürücülerini kullanarak gerçekleştirin.
- B.** Platformu Aurora'dan Amazon DynamoDB'ye değiştirin. Bir DynamoDB Accelerator (DAX) kümesi sağlayın. Mevcut DynamoDB API çağrılarını DAX kümesine yönlendirmek için DAX istemci SDK'sını kullanın.
- C.** İki Lambda fonksiyonu kurun. Bir fonksiyonu bilgiyi almak için yapılandırın. Diğer fonksiyonu bilgiyi veritabanına yüklemek için yapılandırın. Lambda fonksiyonlarını Amazon Simple Notification Service (Amazon SNS) kullanarak entegre edin.

D. İki Lambda fonksiyonu kurun. Bir fonksiyonu bilgiyi almak için yapılandırın. Diğer fonksiyonu bilgiyi veritabanına yüklemek için yapılandırın. Lambda fonksiyonlarını Amazon Simple Queue Service (Amazon SQS) kuyruğu kullanarak entegre edin.

Soru Analizi:

Bir şirket bir uygulama tasarlıyor.

Bu uygulama:

- API Gateway üzerinden veri alıyor
- Veriyi bir **AWS Lambda** fonksiyonu işliyor
- İşlenen veri **Aurora PostgreSQL** veritabanına kaydediliyor

PoC (proof of concept) aşamasında, şirkete çok fazla veri yüklemesi gerektiği için:

👉 Lambda için **çok yüksek eşzamanlılık limitleri** ayarlamak zorunda kalıyorlar.

Bu da yönetimsel yük ve ölçeklenebilirlik sorunlarına yol açıyor.

Çözüm mimarı:

- Ölçeklenebilirliği iyileştirmeli
- Yapılandırma (konfigürasyon) eforunu azaltmalı

Ve yeni bir mimari önermeli.

Seçenek Analizi:

■ **D. İki Lambda fonksiyonu kurulması ve SQS kuyruğu ile entegre edilmesi**

✓ Neden doğru?

- Bir Lambda gelen veriyi alır.
- Bir **SQS queue** veriyi dayanıklı şekilde kuyruklar.
- İkinci Lambda kuyruktan mesajları tüketerek veritabanına yazar.
- SQS'nin otomatik olarak yükü dengeleme, backpressure oluşturma ve ani yükleri emme kapasitesi vardır.

Bu sayede:

✓ Lambda'nın eşzamanlılık limitlerini manuel olarak yükseltmeye gerek olmaz

SQS → veriyi bufferlar

Lambda → kuyruktaki veri kadar otomatik tetiklenir

✓ Aurora'yı aşırı yüklenmeye karşı korur

Gelen trafik patlamaları Aurora'yı çökertmez.

✓ Konfigürasyon minimumdur

SQS + Lambda native olarak entegredir.

✓ AWS'nin önerdiği *serverless ingestion pattern* budur

(Bu desen “Decoupled ingestion” olarak bilinir.)

✗ A. Lambda'yı Tomcat + EC2 olarak yeniden yazmak

Çok fazla operasyonel yük

EC2 yönetimi gereklidir

Ölçeklenebilirlik Lambda kadar iyi değildir

Sorunun istediği “minimum konfigürasyon” şartına aykırı

✗ Aurora → DynamoDB + DAX

Veritabanını tamamen değiştirmek aşırı büyük değişiklik

Mevcut kullanım senaryosu ilişkisel veritabanı (Aurora PostgreSQL) gerektiriyor

DAX okuma hızlandırıcısıdır, yazma yükleme problemini çözmez

Gereksiz ve yanlış yönlü bir çözüm

✗ C. Lambda'lar arası SNS entegrasyonu

SNS bir *fan-out* pub/sub servisidir

Mesajlar **kuyruklanmaz**, *bufferlama* yoktur

Yük patlamalarında veritabanı yine sıkışır

SNS, SQS gibi backpressure sağlamaz

⌚ SONUÇ

Doğru çözüm: D seçeneği → Lambda + SQS + Lambda deseni

Bu, yük dengeleme, esnek ölçeklenme, minimum ayar ve Aurora'yı koruma açısından en doğru ve AWS'in best practice olarak önerdiği yaklaşımındır.

QUESTION 26

A company needs to review its AWS Cloud deployment to ensure that its Amazon S3 buckets do not have unauthorized configuration changes. What should a solutions architect do to accomplish this goal?

A. Turn on AWS Config with the appropriate rules.

B. Turn on AWS Trusted Advisor with the appropriate checks.

C. Turn on Amazon Inspector with the appropriate assessment template.

D. Turn on Amazon S3 server access logging. Con gure Amazon EventBridge (Amazon Cloud Watch Events).

Soru:

Bir şirket, Amazon S3 bucket’larında yetkisiz yapılandırma değişiklikleri olmadığından emin olmak için AWS bulut dağıtımını gözden geçirmelidir. Bu hedefe ulaşmak için bir çözüm mimarı ne yapmalıdır?

- A. Uygun kurallarla birlikte AWS Config’ı etkinleştirin.
- B. Uygun kontrollerle birlikte AWS Trusted Advisor’ı etkinleştirin.
- C. Uygun değerlendirme şablonuyla birlikte Amazon Inspector’ı etkinleştirin.
- D. Amazon S3 sunucu erişim günlüklerini (server access logging) etkinleştirin. Amazon EventBridge (Amazon CloudWatch Events) yapılandırın.

Soru Analizi:

Soru diyor ki:

Bir şirket, **AWS Cloud dağıtımını gözden geçirmek istiyor**, özellikle de **Amazon S3 bucket’larında yetkisiz yapılandırma değişiklikleri** olup olmadığını tespit etmek istiyor.

Yani hedef:

- ✓ **S3 üzerinde kim, ne zaman, hangi yapılandırmayı değiştirdi?**
- ✓ **Bu değişiklik izinli mi, izinsiz mi?**
- ✓ **Ortamda güvenlik açığı yaratabilecek konfigürasyon değişikliklerini tespit etmek.**

Bunun için AWS’de yapılandırma değişikliklerini izleyen bir servis gereklidir.

Seçenek Analizi:

A. AWS Config’ı uygun kurallarla etkinleştir.

✓ **Neden doğru?**

- AWS Config, S3 bucket konfigürasyonlarını **kaydeder, izler ve uygunsuz değişiklikleri tespit eder**.
- "Public access açıldı mı?", "Versioning kapandı mı?", "Encryption devre dışı mı?" gibi kurallar çalıştırılabilir.
- Geçmiş değişiklikleri gösterir (timeline).
- Uyumluluk, güvenlik ve değişiklik izleme için AWS'nin resmi çözümüdür.

Bu soru tam olarak **AWS Config kullanım senaryosudur**.

B. AWS Trusted Advisor

- Trusted Advisor performans, maliyet ve bazı güvenlik önerileri sunar.
- **Gerçek zamanlı konfigürasyon takibi yapmaz.**
- S3 bucket değişikliklerini izleyemez.

→ Bu nedenle **yetersiz**.

C. Amazon Inspector

- Inspector EC2, ECR ve Lambda için güvenlik değerlendirmesi yapar.
- **S3 ile hiçbir ilgisi yoktur.**
- Konfigürasyon değişikliklerini izlemez.

→ Tamamen alakasız.

D. S3 server access logging + EventBridge

- Server access logging **S3'ye yapılan API isteklerini** loglar: PUT, GET vs.
- Ancak:
 - Konfigürasyon değişikliklerini **raporlamaz**
 - "Bucket public oldu", "Encryption kapandı" gibi durumları izlemez.
- Ayrıca çok daha fazla operasyonel yük getirir.

→ Bu soru için **yanlış çözüm**.

SONUÇ:

A seçeneği — AWS Config

AWS Config S3 yapılandırma değişikliklerini otomatik izler, tarihi tutar ve izinsiz değişiklikleri raporlar.

QUESTION 27

A company is launching a new application and will display application metrics on an Amazon CloudWatch dashboard. The company's product manager needs to access this dashboard periodically. The product manager does not have an AWS account. A solutions architect must provide access to the product manager by following the principle of least privilege. Which solution will meet these requirements?

- A.** Share the dashboard from the CloudWatch console. Enter the product manager's email address, and complete the sharing steps. Provide a shareable link for the dashboard to the product manager.
- B.** Create an IAM user specifically for the product manager. Attach the CloudWatchReadOnlyAccess AWS managed policy to the user. Share the new login credentials with the product manager. Share the browser URL of the correct dashboard with the product manager.
- C.** Create an IAM user for the company's employees. Attach the ViewOnlyAccess AWS managed policy to the IAM user. Share the new login credentials with the product manager. Ask the product manager to navigate to the CloudWatch console and locate the dashboard by name in the Dashboards section.
- D.** Deploy a bastion server in a public subnet. When the product manager requires access to the dashboard, start the server and share the RDP credentials. On the bastion server, ensure that the browser is configured to open the dashboard URL with cached AWS credentials that have appropriate permissions to view the dashboard.

Soru:

Bir şirket yeni bir uygulama başlatıyor ve uygulamaya ait metrikleri bir Amazon CloudWatch panosunda (dashboard) görüntüleyecek. Şirketin ürün yöneticisinin bu dashboard'a periyodik olarak erişmesi gerekiyor. Ürün yöneticisinin bir AWS hesabı yok. Bir çözüm mimarının **en az ayrıcalık (least privilege) ilkesini** izleyerek ürün yöneticisine erişim sağlama gerekliliği hangi çözüm karşıları?

- A.** CloudWatch konsolundan dashboard'u paylaşın. Ürün yöneticisinin e-posta adresini girin ve paylaşım adımlarını tamamlayın. Ürün yöneticisine paylaşılabilir bir dashboard bağlantısı sağlayın.
- B.** Ürün yönetici için özel bir IAM kullanıcıyı oluşturun. Bu kullanıcıya CloudWatchReadOnlyAccess yönetilen AWS politikasını ekleyin. Yeni giriş bilgilerini ürün yöneticisiyle paylaşın. Dashboard'un doğru tarayıcı URL'sini ürün yöneticisine gönderebilir.
- C.** Şirket çalışanları için bir IAM kullanıcıyı oluşturun. Bu kullanıcıya ViewOnlyAccess yönetilen AWS politikasını ekleyin. Giriş bilgilerini ürün yöneticisiyle paylaşın. Ürün yöneticisinden CloudWatch konsoluna giderek Dashboards bölümünden doğru dashboard'u bulmasını isteyin.
- D.** Genel bir alt ağa bir bastion sunucusu dağıtırın. Ürün yönetici dashboard'a erişmek istediginde sunucuyu başlatın ve RDP kimlik bilgilerini paylaşın. Bastion sunucusunda, AWS kimlik bilgileri önbellege alınmış şekilde dashboard URL'sinin açılmasını sağlayan bir tarayıcı yapılandırın.

Soru Analizi:

Şirket:

- Bir CloudWatch dashboard'ı var.
- Ürün yöneticisi AWS kullanıcısı değil.
- Sadece dashboard'ı **görüntülemesi** gerekiyor (yönetmesi değil).
- "Least privilege" → **en az yetkiyi vererek erişim** sağlanmalı.
- AWS hesabı olmayan biri için **kullanıcı oluşturmak istenmez**, gereksiz operasyonel yük doğurur.
- En kolay ve en az yetki ile çözüm: **CloudWatch dashboard sharing (public share)**.

AWS, CloudWatch dashboard'larını AWS hesabı olmayan kullanıcılarla **sadece görüntüleme yetkisiyle** paylaşmaya izin verir.

Seçenek Analizi:

A. CloudWatch konsolundan dashboard'u paylaş. Ürün yöneticisinin e-posta adresini gir. Paylaşım adımlarını tamamla. Ürün yöneticisine paylaşılabilir bağlantıyı gönder.

✓ NEDEN DOĞRU?

- CloudWatch artık **dashboard sharing** özelliğine sahiptir.
- Dashboard, **AWS hesabı olmayan kullanıcılarla bile bir public shareable URL** üzerinden paylaşılabilir.
- Kullanıcıya IAM hesabı verilmez → **en az ayrıcalık** sağlanır.
- Ek kullanıcı yönetimi, IAM hakları, login bilgisi **gerektirmez**.
- Operasyonel yük **sıfır** deneye kadar azdır.

Bu seçenek **tam olarak sorunun istediğidir**.

B. Ürün yöneticisi için IAM user oluştur, CloudWatchReadOnlyAccess verip giriş bilgilerini paylaş

Neden yanlış?

- IAM kullanıcı oluşturmak **least privilege ihlali** sayılır.
- AWS hesabı olmayan biri için gereksiz IAM user yaratılır.

- Paylaşım için gereksiz operasyonel yük (şifre yönetimi, MFA, dönüşümlü şifre) oluşur.

✗ C. Şirket çalışanları için IAM user oluştur, ViewOnlyAccess politikası ekle ve ürün yöneticisiyle paylaş

Neden yanlış?

- Bu seçenek hem daha geniş izinler verir (**ViewOnlyAccess → çok fazla izin içerir**)
- Hem de **ürün yöneticisi için şirket içi IAM user yaratmak doğru değildir.**
- Least privilege ilkesine uymaz.

✗ D. Public subnet'te bastion server kur, RDP ile bağlanmasını sağla

Neden yanlış?

- Çok karmaşık, çok maliyetli ve gereksiz bir çözüm.
- RDP sunucusu yönetmek → **en yüksek operasyonel yük.**
- Least privilege ilkesine tamamen aykırı.

🎯 SONUÇ:

CloudWatch Dashboard Share özelliği

- IAM hesabı olmadan
- Minimum ayrıcalık ile
- Kolay ve güvenli paylaşım sağlar.

1. CloudWatch Dashboard Paylaşma Mantığı (Özet)

AWS, CloudWatch dashboard'larını **AWS hesabı olmayan kişilerle paylaşabilmeniz için özel bir özellik** sunar:

✓ “Dashboard Sharing” özelliği

- Dashboard'ı oluşturursun
- “Share dashboard” seçeneğine basarsın
- Karşına **public share** paneli çıkar
- Yetkileri belirlersin:
 - Salt-okuma (read-only)
 - Zaman aralığı değiştirme izni
- AWS, sana **özel bir paylaşım linki (shareable link)** üretir

- Bu linki AWS hesabı olmayan kişiye bile verebilirsin

✓ Avantajları:

- Ek IAM user oluşturmazsınız
- Parola, MFA, hesap yönetimi yok
- Least privilege (en az ayrıcalık):
 - yalnızca dashboard görüntüleme izni
- Minimum operasyon yükü

Bu nedenle AWS sınavlarında **doğru seçenek çoğunlukla CloudWatch dashboard share özelliği**dir.

2. IAM Least Privilege (En Az Ayrıcalık) Prensibi (Özet)

AWS güvenlikte temel prensip şudur:

► **Bir kullanıcıya yalnızca işini yapması için gereken minimum izinler verilmeli.**

Bu ne demek?

- Eğer yalnızca dashboard görüntülemesi gerekiyorsa → **okuma izni (read-only)**
- Console erişimi gerekiyorsa → yalnızca ilgili servis için izin
- Yönetici (admin) veya geniş kapsamlı izinler → **gereksiz ve risklidir**

✓ Least privilege'i sağlananın yolları:

1. Gereksiz IAM kullanıcıları oluşturmamak
2. Geniş politika vermek yerine dar kapsamlı politika kullanmak
3. Paylaşım gereken yerlerde **IAM user oluşturmak yerine**
 - “share” linki
 - temporary credentials
 - role assumption kullanmak
4. Kullanıcıya sadece **ihtiyaç duyduğu servise özel** izin vermek

QUESTION 28

A company is migrating applications to AWS. The applications are deployed in different accounts. The company manages the accounts centrally by using AWS Organizations. The company's security team needs a single sign-on (SSO) solution across all the

company's accounts. The company must continue managing the users and groups in its on-premises self-managed Microsoft Active Directory. Which solution will meet these requirements?

- A.** Enable AWS Single Sign-On (AWS SSO) from the AWS SSO console. Create a one-way forest trust or a one-way domain trust to connect the company's self-managed Microsoft Active Directory with AWS SSO by using AWS Directory Service for Microsoft Active Directory.
- B.** Enable AWS Single Sign-On (AWS SSO) from the AWS SSO console. Create a two-way forest trust to connect the company's self-managed Microsoft Active Directory with AWS SSO by using AWS Directory Service for Microsoft Active Directory.
- C.** Use AWS Directory Service. Create a two-way trust relationship with the company's self-managed Microsoft Active Directory.
- D.** Deploy an identity provider (IdP) on premises. Enable AWS Single Sign-On (AWS SSO) from the AWS SSO console.

Soru:

Bir şirket uygulamalarını AWS'ye taşımaktadır. Uygulamalar farklı hesaplarda dağıtılmıştır. Şirket, hesapları AWS Organizations kullanarak merkezi bir şekilde yönetmektedir. Şirketin güvenlik ekibi, tüm şirket hesapları için tek bir oturum açma (SSO) çözümüne ihtiyaç duymaktadır. Şirket, kullanıcıları ve grupları şirket içinde bulunan, kendi kendini yöneten Microsoft Active Directory'de yönetmeye devam etmelidir. Bu gereksinimleri hangi çözüm karşılar?

- A.** AWS SSO konsolundan AWS Single Sign-On (AWS SSO) özelliğini etkinleştirin. AWS Directory Service for Microsoft Active Directory kullanarak şirketin kendi kendini yöneten Microsoft Active Directory'si ile AWS SSO arasında tek yönlü (one-way) bir orman (forest) ya da tek yönlü bir etki alanı (domain) güveni oluşturun.
- B.** AWS SSO konsolundan AWS Single Sign-On (AWS SSO) özelliğini etkinleştirin. AWS Directory Service for Microsoft Active Directory kullanarak şirketin kendi kendini yöneten Microsoft Active Directory'si ile AWS SSO arasında çift yönlü (two-way) bir orman güveni oluşturun.
- C.** AWS Directory Service'i kullanın. Şirketin kendi kendini yöneten Microsoft Active Directory'si ile çift yönlü bir güven ilişkisi oluşturun.
- D.** Şirket içinde bir kimlik sağlayıcısı (IdP) dağıtın. AWS SSO konsolundan AWS Single Sign-On (AWS SSO) özelliğini etkinleştirin.

Soru Analizi:

Bu sorunun ipuçları:

- ✓ Şirket kullanıcıları hâlâ on-prem Active Directory'de kalacak.
- ✓ Tüm AWS hesaplarında SSO (tek oturum açma) isteniyor.
- ✓ Kullanıcı yönetimi on-prem AD'de olmalı → AWS'de yeni kullanıcı oluşturmak istenmiyor.
- ✓ AWS Organizations ile entegre bir SSO isteniyor.

Seçenek Analizi:

A Seçeneği

AWS SSO (IAM Identity Center) + AWS Directory Service for Microsoft AD + One-way forest/domain trust

✓ En doğru çözüm:

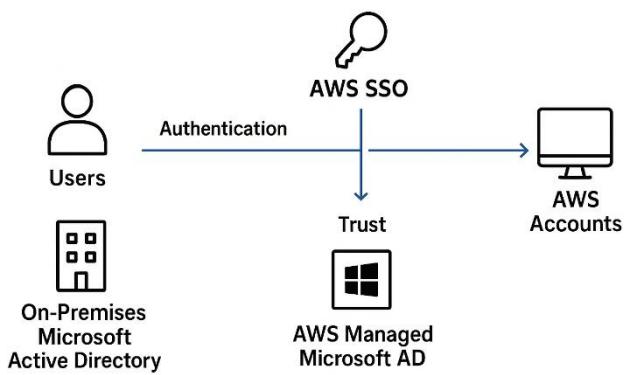
AWS IAM Identity Center (eski adıyla AWS SSO) + On-Prem AD entegrasyonu ile trust ilişkisi kurmaktır.

AWS bu kullanım için en iyi çözüm olarak **AWS Directory Service for Microsoft AD ile one-way forest trust** önermektedir.

✓ Neden doğru?

- On-prem AD kullanıcıları AWS SSO'ya bağlanabilir.
- Kullanıcı yönetimi tamamen on-prem AD'de kalır.
- AWS Organizations ile tüm hesaplara merkezi erişim sağlanır.
- AWS, AWS SSO + AD entegrasyonu için **one-way trust** kullanmanızı önerir.
 - On-prem AD → AWS Managed AD'e güven verir.
 - AWS AD on-prem'e güven vermez.
- SSO için en düşük operasyonel yük sunar.

Tam olarak sorunun tüm gereksinimlerini karşılar.



Kimlik Doğrulama Akışı (Detaylandırılmış)

- 1 Kullanıcı şirketteki bilgisayarında (domaine join olmuş)
→ Kimlik: *on-prem AD kullanıcı hesabı*
- 2 AWS uygulamasına erişmek ister → IAM Identity Center login sayfasına gider.
- 3 IAM Identity Center, kimlik doğrulaması için AWS Managed AD'ye yönlendirir.
- 4 AWS Managed AD, on-prem AD'ye güven ilişkisi üzerinden soru gönderir:
“Bu kullanıcı kim, bilgileri doğru mu?”
- 5 On-prem AD:
“Evet, doğrulandı” → AWS Managed AD'ye cevap.
- 6 AWS Managed AD sonucu IAM Identity Center'a iletir.
- 7 IAM Identity Center kullanıcıya yetkili AWS hesaplarını ve rollerini gösterir.
- 8 Kullanıcı istediği hesabı seçer → ilgili AWS hesabına geçici AWS kimlik bilgileri ile giriş sağlanır.

B Seçeneği

Neden Yanlış?

Two-way (çift yönlü) forest trust gereksiz ve güvenlik açısından önerilmez.

AWS SSO için **two-way trust'a ihtiyaç yoktur**.

C Seçeneği

Neden Yanlış?

- SADECE Directory Service kullanıyor → AWS SSO / IAM Identity Center yok.
- Bu çözüm **AWS hesapları için SSO sağlamaz**.

Eksik mimari.

D Seçeneği

Neden Yanlış?

- On-prem Identity Provider kurmak gereksiz karmaşıklık ve operasyonel yük getirir.
- AWS zaten IAM Identity Center ile SSO verebiliyor → IdP kurmaya gerek yok.

SONUÇ

✓ Doğru cevap: A seçeneği

QUESTION 29

A company provides a Voice over Internet Protocol (VoIP) service that uses UDP connections. The service consists of Amazon EC2 instances that run in an Auto Scaling group. The company has deployments across multiple AWS Regions. The company needs to route users to the Region with the lowest latency. The company also needs automated failover between Regions. Which solution will meet these requirements?

- A.** Deploy a Network Load Balancer (NLB) and an associated target group. Associate the target group with the Auto Scaling group. Use the NLB as an AWS Global Accelerator endpoint in each Region.
- B.** Deploy an Application Load Balancer (ALB) and an associated target group. Associate the target group with the Auto Scaling group. Use the ALB as an AWS Global Accelerator endpoint in each Region.
- C.** Deploy a Network Load Balancer (NLB) and an associated target group. Associate the target group with the Auto Scaling group. Create an Amazon Route 53 latency record that points to aliases for each NLB. Create an Amazon CloudFront distribution that uses the latency record as an origin.
- D.** Deploy an Application Load Balancer (ALB) and an associated target group. Associate the target group with the Auto Scaling group. Create an Amazon Route 53 weighted record that points to aliases for each ALB. Deploy an Amazon CloudFront distribution that uses the weighted record as an origin.

Soru:

Bir şirket, UDP bağlantılarını kullanan bir IP Üzerinden Ses (VoIP) hizmeti sunmaktadır. Hizmet, bir Auto Scaling grubunda çalışan Amazon EC2 örneklerinden oluşmaktadır. Şirketin birden fazla AWS Bölgesinde (Region) dağılımı vardır. Şirket kullanıcıları, **en düşük gecikmeye (latency) sahip Bölgeye yönlendirmelidir**. Şirket ayrıca **Bölgeler arası otomatik failover** da istemektedir. Bu gereksinimleri hangi çözüm karşılar?

A. Bir Ağ Yük Dengeleyici (Network Load Balancer - NLB) ve buna bağlı bir hedef grubu dağıtın. Hedef grubunu Auto Scaling grubuya ilişkilendirin.

Her Bölgede NLB'yi AWS Global Accelerator uç noktası olarak kullanın.

B. Bir Uygulama Yük Dengeleyici (Application Load Balancer - ALB) ve buna bağlı bir hedef grubu dağıtın. Hedef grubunu Auto Scaling grubuya ilişkilendirin.

Her Bölgede ALB'yi AWS Global Accelerator uç noktası olarak kullanın.

C. Bir Ağ Yük Dengeleyici (NLB) ve buna bağlı bir hedef grubu dağıtın. Hedef grubunu Auto Scaling grubuya ilişkilendirin.

Her NLB için takma ad (alias) kullanan Amazon Route 53 gecikme (latency) kaydı oluşturun.

Bu gecikme kaydını kaynak (origin) olarak kullanan bir Amazon CloudFront dağıtıımı oluşturun.

D. Bir Uygulama Yük Dengeleyici (ALB) ve buna bağlı bir hedef grubu dağıtın. Hedef grubunu Auto Scaling grubuya ilişkilendirin.

Her ALB'ye işaret eden Route 53 ağırlıklı (weighted) kayıt oluşturun.

Bu ağırlıklı kaydı kaynak (origin) olarak kullanan bir Amazon CloudFront dağıtıımı dağıtın.

Soru Analizi:

Bir şirket, **UDP kullanan VoIP hizmeti** sağlıyor. Servis:

- Amazon EC2 üzerinde çalışıyor
- Auto Scaling kullanıyor
- Birden fazla AWS **Region**'da dağıtılmış
- Kullanıcılar **en düşük gecikmeli Region'a yönlendirilmeli**
- **Otomatik Region failover** gerekiyor (bir Region çökerse trafik otomatik diğerine gitmeli)

Bu şartlar şu teknolojilere işaret eder:

- **UDP desteği** gereklili → ALB **UDP desteklemez**, NLB destekler
- **En düşük gecikmeye göre yönlendirme + otomatik failover** → AWS Global Accelerator tam bunu sağlar
- Global Accelerator ayrıca VoIP/UDP için özel optimize edilmiş küresel edge ağı sağlar (CloudFront ve Route 53 UDP için uygun değildir)

Seçenek Analizi:

A. NLB + Global Accelerator

- NLB UDP'yi destekler

- Global Accelerator, her Region'daki NLB'yi endpoint yapabilir
- Global Accelerator otomatik olarak **en düşük gecikmeli Region'a** yönlendirir
- Bir Region çalışmazsa **otomatik failover** yapar

Bu çözüm sorudaki tüm gereksinimleri karşılar.

B. ALB + Global Accelerator (Yanlış)

- **ALB UDP desteklemez**, sadece HTTP/HTTPS (Layer 7)
- VoIP (UDP trafik) → ALB kullanılamaz
- Global Accelerator doğru ama ALB UDP işlemez → çözüm bozulur

C. NLB + Route 53 Latency Record + CloudFront (Yanlış)

- NLB UDP destekler (iyi)
- Route 53 latency routing gecikmeye göre yönlendirir (iyi)
- Ancak:
Route 53 failover hızları Global Accelerator kadar hızlı değildir
CloudFront UDP desteklemez
VoIP trafik CloudFront'tan geçemez

Bu nedenle uygun çözüm **değil**.

D. ALB + Route 53 Weighted + CloudFront (Yanlış)

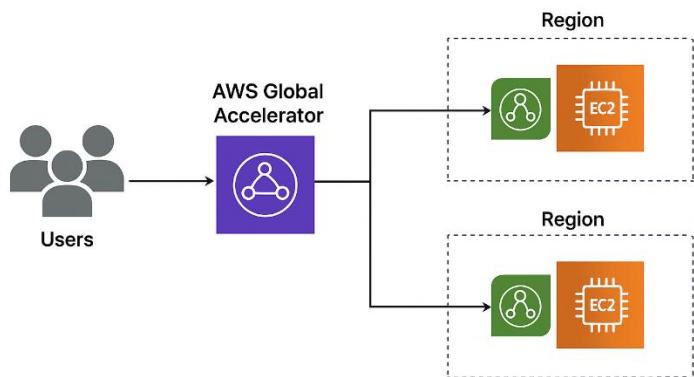
- ALB UDP desteklemez → en büyük sorun
- Weighted routing gecikmeye göre yönlendirme yapmaz
- CloudFront UDP desteklemez

Birçok açıdan yanlış seçenek.

Sonuç: Doğru cevap A seçeneğidir.

Çünkü VoIP için UDP şarttır ve en düşük gecikme + otomatik failover için AWS Global Accelerator en doğru çözümüdür.

VoIP service using UDP with automated failover between regions



QUESTION 30

A development team runs monthly resource-intensive tests on its general purpose Amazon RDS for MySQL DB instance with Performance Insights enabled. The testing lasts for 48 hours once a month and is the only process that uses the database. The team wants to reduce the cost of running the tests without reducing the compute and memory attributes of the DB instance. Which solution meets these requirements MOST cost-effectively?

- A.** Stop the DB instance when tests are completed. Restart the DB instance when required. Topic 1
- B.** Use an Auto Scaling policy with the DB instance to automatically scale when tests are completed.
- C.** Create a snapshot when tests are completed. Terminate the DB instance and restore the snapshot when required.
- D.** Modify the DB instance to a low-capacity instance when tests are completed. Modify the DB instance again when required.

Soru:

Bir geliştirme ekibi, Performance Insights etkinleştirilmiş genel amaçlı bir Amazon RDS for MySQL veritabanı üzerinde her ay kaynak yoğun testler yürütmektedir. Bu testler ayda bir kez yapılmakta ve 48 saat sürmektedir. Bu testler dışında veritabanını kullanan başka bir süreç yoktur. Ekip, veritabanının işlemci ve bellek özelliklerini azaltmadan testleri yürütme maliyetini düşürmek istemektedir.

Bu gereksinimleri **en düşük maliyetle** karşılayan çözüm hangisidir?

- A.** Testler tamamlandığında DB instance'ı durdurun. Gerekçinde DB instance'ı yeniden başlatın.
- B.** DB instance'ı testler tamamlandığında otomatik olarak ölçeklendirmek için bir Auto Scaling politikası kullanın.
- C.** Testler tamamlandığında bir snapshot oluşturun. DB instance'ı silin ve gerekçinde snapshot'tan geri yükleyin.
- D.** Testler tamamlandığında DB instance'ı düşük kapasiteli bir instance türüne dönüştürün. Gerekçinde tekrar eski instance türüne dönüştürün.

Soru Analizi:

- Bir ekip **ayda bir kez, 48 saat boyunca, çok kaynak tüketen testler** çalıştırıyor.
- Testler, **Amazon RDS for MySQL (general purpose)** bir veritabanında yapılıyor.
- Performance Insights açık.
- Testler dışında bu veritabanını **hiçbir şey kullanmıyor**.
- Ama test yapılırken, veritabanının **hesaplama (CPU) ve bellek kapasitesinden ödün verilemez**.
- Amaç: **Maliyeti en aza indirmek**.

Bu şu demek:

- Veritabanı **ayda sadece 2 gün çalışıyor**, geri kalan 28 gün boş para ödüyorsun.
- En ekonomik çözüm: Veritabanını ayın geri kalanında **çalıştırmamak / kapatmak / silmek**.

Seçenek Analizi:

- A. Stop the DB instance when tests are completed. Restart the DB instance when required.**

(Testlerden sonra DB instance'ı durdur, gerekçinde yeniden başlat.)

- Amazon RDS “stop/start” özelliği destekliyor.
- Durdurulduğunda **sadece storage (depolama)** ücreti ödenir, compute ödenmez.
- Çok büyük maliyet tasarrufu sağlar.
- En kolay, en pratik çözüm.
- Veritabanı konfigürasyonu, instance tipi, memory/cpu **değişmez**.

Bu seçenek hem doğru hem de en pratik çözümdür.

✗ B. Use an Auto Scaling policy with the DB instance to automatically scale when tests are completed.

(Testler tamamlandığında otomatik ölçeklendirme kullan.)

- RDS veritabanları EC2 gibi Auto Scaling ile çalışmaz.
- Otomatik dikey ölçeklenme yoktur.
- Bu teknik olarak mümkün değildir.

Bu seçenek teknik olarak yanlış.

✗ C. Create a snapshot when tests are completed. Terminate the DB instance and restore the snapshot when required.

(Testlerden sonra snapshot oluştur, DB instance’ını sil, gerekiğinde snapshot’tan geri yükle.)

- ✓ Çok büyük tasarruf sağlar — compute tamamen sıfırlanır
- ✓ Sadece depolama ücreti ödenir

Ama:

- Snapshot’tan restore etmek **uzun sürer** (dakikalarca).
- DB endpoint’i değişebilir.
- Sık kullanım için pratik değildir.
- Stop/Start kadar kolay değildir.

Ekonominin yanı sıra operasyonel olarak A’ya göre daha zor.

✗ D. Modify the DB instance to a low-capacity instance when tests are completed. Modify again when required.

(Testlerden sonra daha düşük kapasiteli instance’'a düşür, gerekiğinde geri büyüt.)

- Düşük kapasiteli instance çalışırken **yne compute ücreti ödenir**.
- Ama veritabanı ayda 28 gün boş duruyor → boş para gider.
- Stop/Start kadar tasarruf sağlamaz.
- Değişiklikler kesinti yaratır ve zaman alır.

Tasarruf sağlar ama en az verimli çözümüdür.

 SONUÇ:

DB instance’ı durdurup gereki̇inde başlatmak, hem teknik olarak en kolay hem de en düşük maliyetli yöntemdir.

QUESTION 31

A company that hosts its web application on AWS wants to ensure all Amazon EC2 instances. Amazon RDS DB instances. and Amazon Redshift clusters are configured with tags. The company wants to minimize the effort of configuring and operating this check. What should a solutions architect do to accomplish this?

- A.** Use AWS Config rules to define and detect resources that are not properly tagged.
- B.** Use Cost Explorer to display resources that are not properly tagged. Tag those resources manually.
- C.** Write API calls to check all resources for proper tag allocation. Periodically run the code on an EC2 instance.
- D.** Write API calls to check all resources for proper tag allocation. Schedule an AWS Lambda function through Amazon CloudWatch to periodically run the code.

Soru:

Bir şirket, web uygulamasını AWS üzerinde barındırmaktadır. Şirket, tüm Amazon EC2 instance’larının, Amazon RDS veritabanı instance’larının ve Amazon Redshift kümelerinin **doğru şekilde etiketlenmiş (tagged)** olduğundan emin olmak istemektedir. Şirket, bu kontrolü yapılandırmak ve işletmek için gereken çabayı en aza indirmek istemektedir.

Bir solutions architect bu gereksinimi karşılamak için ne yapmalıdır?

- A.** AWS Config kuralları kullanarak doğru şekilde etiketlenmemiş kaynakları tanımlayın ve tespit edin.
- B.** Cost Explorer’ı kullanarak doğru şekilde etiketlenmemiş kaynakları görüntüleyin. Bu kaynakları elle etiketleyin.
- C.** Tüm kaynakların doğru etiketlenmesini kontrol etmek için API çağrıları yazın. Bu kodu periyodik olarak bir EC2 instance’ında çalıştırın.
- D.** Tüm kaynakların doğru etiketlenmesini kontrol etmek için API çağrıları yazın. Bu kodu periyodik olarak çalıştmak için Amazon CloudWatch üzerinden zamanlanmış bir AWS Lambda fonksiyonu oluşturun.

Soru Analizi:

Şirketin ihtiyacı:

- EC2, RDS ve Redshift gibi tüm kaynakların **doğru etiketlenmiş (tagged)** olduğundan emin olmak istiyor.

- Bu kontrolün:
 - **Otomatik olması**
 - **Sürekli çalışması**
 - **Minimum operasyonel yük gerektirmesi**
 - **Minimum kurulum ve bakım gerektirmesi** gerekiyor.

Seçenek Analizi:

 **A. AWS Config kuralları kullanarak doğru şekilde etiketlenmemiş kaynakları tanımlayın ve tespit edin.**

Bu gereksinimlere en uygun AWS servisi → **AWS Config**.

Çünkü AWS Config:

- Otomatik olarak kaynakları izler.
- Özel veya yönetilen kurallarla **tag zorunluluğu** kontrol edebilir.
- Hiçbir kod yazmayı gerektirmez.
- Sürekli uyumluluk denetimi sağlar.
- Yönetimi çok kolaydır.

- ✓ AWS Config, "required-tags" gibi hazır kurallar sunar.
- ✓ EC2, RDS, Redshift dahil yüzlerce kaynak türünü otomatik tarar.
- ✓ Yönetmesi ve devreye alınması en basit çözümüdür.
- ✓ Kod gerekmez.
- ✓ Sürekli çalışır ve uyarı üretir.

Bu seçenek hem en doğru hem de en az operasyonel yüke sahip çözümüdür.

 **B. Cost Explorer kullanarak doğru şekilde etiketlenmemiş kaynakları görüntüleyin. Bu kaynakları elle etiketleyin.**

- Cost Explorer sadece **faturalandırma etiketlerini** analiz eder.
- Kapsamı sınırlıdır, her kaynağı kapsamaz.
- Otomatik tespit yoktur.
- Elle etiketleme → *yüksek operasyonel yük*.

Otomatiklik yok → ihtiyaçları karşılamaz.

 **C. API çağrıları yazın ve EC2 üzerinde periyodik olarak çalıştırın.**

- Kod yazmak gerekiyor.
- EC2 instance çalıştırırmak operasyonel maliyet ve bakım gerektirir.
- Sürekli uptime yönetimi gereklidir.

Hem pahalı hem de bakımı zor → yanlış.

X D. API çağrıları yazın ve Lambda ile CloudWatch üzerinden zamanlayın.

- C seçenekinden daha iyi.
- EC2'ye göre operasyonel yük daha az.
- Yine de **kod yazılması ve bakımı gereklidir**.

Yarı doğru ama AWS Config varken gereksiz derecede karmaşık.

⌚ SONUÇ:

Çünkü AWS Config:

- Kod gerektirmez
- Her kaynak türünü otomatik tarar
- “Required tags” gibi hazır kurallar vardır
- En az operasyonel yükü getirir

QUESTION 32

A development team needs to host a website that will be accessed by other teams. The website contents consist of HTML, CSS, client-side JavaScript, and images.

Which method is the MOST cost-effective for hosting the website?

- A. Containerize the website and host it in AWS Fargate.
- B. Create an Amazon S3 bucket and host the website there.
- C. Deploy a web server on an Amazon EC2 instance to host the website.
- D. Configure an Application Load Balancer with an AWS Lambda target that uses the Express.js framework.

Soru:

Bir geliştirme ekibinin, diğer ekipler tarafından erişilecek bir web sitesini barındırmaması gerekiyor. Web sitesi içeriği HTML, CSS, istemci tarafı JavaScript ve görüntülerden oluşuyor.

Bu web sitesini barındırmak için en maliyet açısından en etkili yöntem hangisidir?

- A. Web sitesini containerize edip AWS Fargate üzerinde barındırmak.
- B. Bir Amazon S3 bucket'ı oluşturup web sitesini orada barındırmak.
- C. Bir Amazon EC2 instance üzerinde web sunucusu kurarak web sitesini barındırmak.
- D. Express.js framework'ünü kullanan bir AWS Lambda hedefi ile bir Application Load Balancer yapılandırmak.

Soru Analizi:

Web sitesi statik içerikten oluşuyor:

- HTML
- CSS
- Client-side JavaScript
- Images

Arka uç yok, yani sunucu tarafı işlem gerekmıyor.

Kullanıcılar sadece bu dosyaları indirecek ve tarayıcıda çalıştıracak.

AWS'de **statik web sitesi** için en ucuz ve en az yönetim gerektiren çözüm **Amazon S3 static website hosting**'dir.

Ek olarak CloudFront ekleyerek global hız artırılabilir, fakat soru bunu sormuyor.

Soru "**MOST cost-effective**" yani *en az maliyetli yöntem* diyor → Kritik ifade.

Seçenek Analizi:

B. Create an Amazon S3 bucket and host the website there.

- S3 static website hosting tam olarak bu kullanım içindir.
- Yönetim yok → No servers.
- En ucuz yöntem: yalnızca depolama + istek başına ödeme.
- HTML/CSS/JS tamamen statik dosyalardır → S3 için ideal.

En düşük maliyet – Doğru cevap

A. Containerize the website and host it in AWS Fargate.

- Fargate maliyetlidir.
- Container çalıştırmak gereksiz: Statik website için container kullanılmaz.
- Operasyonel yük fazladır (ECR, task, service vs.)

Gereksiz ve pahalı → Yanlış

 **C. Deploy a web server on an Amazon EC2 instance to host the website.**

- EC2 her zaman çalışır → Saatlik ücret.
- Bakım + patching + scaling maliyeti var.
- Statik site için aşırı gereksiz.

 **S3'ten çok daha pahalı → Yanlış**

 **D. Configure an Application Load Balancer with an AWS Lambda target using Express.js**

- Hem ALB hem Lambda çalıştırılmak maliyetlidir.
- Express.js server mantıklı değil çünkü statik içerik sunuyoruz.
- Lambda invocation + ALB cost çok fazla.

 **Maliyet açısından anlamsız → Yanlış**

 **SONUÇ**

Amazon S3 bucket'ında web sitesi barındırmak, statik dosyalar için AWS üzerindeki en uygun maliyetli çözümüdür.

QUESTION 33

A company runs an online marketplace web application on AWS. The application serves hundreds of thousands of users during peak hours. The company needs a scalable, near-real-time solution to share the details of millions of financial transactions with several other internal applications. Transactions also need to be processed to remove sensitive data before being stored in a document database for low-latency retrieval. What should a solutions architect recommend to meet these requirements?

A. Store the transactions data into Amazon DynamoDB. Set up a rule in DynamoDB to remove sensitive data from every transaction upon write. Use DynamoDB Streams to share the transactions data with other applications.

B. Stream the transactions data into Amazon Kinesis Data Firehose to store data in Amazon DynamoDB and Amazon S3. Use AWS Lambda integration with Kinesis Data Firehose to remove sensitive data. Other applications can consume the data stored in Amazon S3.

C. Stream the transactions data into Amazon Kinesis Data Streams. Use AWS Lambda integration to remove sensitive data from every transaction and then store the transactions data in Amazon DynamoDB. Other applications can consume the transactions data off the Kinesis data stream.

D. Store the batched transactions data in Amazon S3 as les. Use AWS Lambda to process every le and remove sensitive data before updating the les in Amazon S3. The Lambda function then stores the data in Amazon DynamoDB. Other applications can consume transaction les stored in Amazon S3.

Soru:

Bir şirket, AWS üzerinde bir çevrim içi pazar yeri web uygulaması çalışmaktadır. Uygulama, yoğun saatlerde yüz binlerce kullanıcıya hizmet vermektedir. Şirketin, milyonlarca finansal işlemin detaylarını diğer dahili uygulamalarla paylaşmak için ölçeklenebilir ve gerçek zamana yakın bir çözüme ihtiyacı vardır. İşlemler ayrıca, bir doküman veritabanında düşük gecikmeli sorgulama için saklanmadan önce hassas verilerden arındırılmalıdır. Bu gereksinimleri karşılamak için bir çözüm mimarı ne önermelidir?

A. İşlem verilerini Amazon DynamoDB'ye kaydedin. DynamoDB'de, her bir işlem yazılrken hassas veriyi kaldıracak bir kural ayarlayın. İşlem verilerini diğer uygulamalarla paylaşmak için DynamoDB Streams kullanın.

B. İşlem verilerini Amazon Kinesis Data Firehose'a aktırın ve verileri Amazon DynamoDB ile Amazon S3'e kaydedin. Hassas veriyi kaldırmak için Kinesis Data Firehose'un AWS Lambda entegrasyonunu kullanın. Diğer uygulamalar Amazon S3'teki veriyi tüketebilir.

C. İşlem verilerini Amazon Kinesis Data Streams'e aktırın. Her bir işleminden hassas veriyi kaldırmak için AWS Lambda entegrasyonunu kullanın ve ardından verileri Amazon DynamoDB'ye kaydedin. Diğer uygulamalar işlem verilerini Kinesis veri akışından tüketebilir.

D. Toplu işlem verilerini Amazon S3'e dosya olarak kaydedin. Her bir dosyayı işlemek ve hassas veriyi kaldırmak için AWS Lambda kullanın ve dosyaları S3'te güncelleyin. Lambda fonksiyonu daha sonra veriyi Amazon DynamoDB'ye kaydeder. Diğer uygulamalar S3'teki işlem dosyalarını tüketebilir.

Soru Analizi:

Gereksinimler çok kritik:

1. Milyonlarca finansal işlem → Yüksek hacim

Bu nedenle **scalable / near-real-time streaming** gerekiyor.

İki uygun hizmet var: **Kinesis Data Streams** veya **Kafka** (MSK).

Soru AWS sınavı olduğundan → Kinesis.

2. Near-real-time (gerçek zamana yakın)

Bu nedenle **batch S3 işleme** gibi gecikme yaratan yöntemler uygun değil.

3. Hassas veriyi depolamadan önce kaldırma

Bu, bir *processing layer* gerektirir → genelde **Lambda**.

4. Veri bir document database'e yazılacak

Document store = **Amazon DynamoDB** (AWS sınavlarında bu demektir).

5. Diğer iç uygulamalar ham veriyi real-time olarak tüketebilmeli

Bu, stream üzerinden çok tüketicili okuma gerektirir →

Kinesis Data Streams ideal çünkü birden fazla consumer paralel okuyabilir.

Seçenek Analizi:

C. Kinesis Data Streams → Lambda sanitize → DynamoDB + consumer apps

Neden?

- **Kinesis Data Streams** → milyonlarca işlem için ölçeklenebilir.
- **Near-real-time**: gecikme düşük (milisaniye–saniye arası).
- Lambda ile **stream üzerindeki her işlem** sanitize edilebilir.
- Temizlenen veri **DynamoDB**'ye yazılır → document db görevi.
- Diğer iç uygulamalar **aynı Kinesis stream'den paralel consumer grupları olarak** gerçek zamanlı okuyabilir.

→ Tüm gereksinimleri eksiksiz karşılayan tek seçenek.

A. DynamoDB'ye yaz, DynamoDB rule ile filtrele, DynamoDB Streams ile paylaş

- DynamoDB “sensitive data removal rule” diye bir mekanizma **sunmaz**.
- Hassas veriyi DynamoDB'ye yazmadan temizlemek gereklidir.
- DynamoDB Streams data *yazıldıktan sonra* tetiklenir; yani hassas veri önce yazılır → güvenlik hatası.

→ Güvensiz ve teknik olarak mümkün değil → Yanlış

B. Kinesis Data Firehose → DynamoDB + S3; Lambda ile sanitize

- Firehose **Amazon DynamoDB'ye yazamaz**. Desteklenen hedefler: S3, Redshift, OpenSearch, Splunk.
- Firehose near-real-time ama “buffering” vardır → gecikme 1–5 dakika olabilir.
- Firehose birden çok consumer'a stream veremez; *sadece bir hedefe aktarır*.
- İç uygulamalar S3'ten okursa **real-time olmaz**.

→ Teknik olarak mümkün değil + gereksinimleri karşılamıyor.

D. S3'e batched olarak yaz, Lambda ile sanitize et, sonra DynamoDB

- S3 batch dosya işleme gerçek zamana yakın değildir.
- S3 → high-latency, event-driven, streaming değil.
- Yüz binlerce request/s ölçeklenmez.
- Consumer uygulamalar S3 dosyalarını okuyamaz → streaming yok.

Streaming değil + real-time değil → Yanlış

Sonuç

Stream the data into Kinesis Data Streams → Lambda to sanitize → store in DynamoDB → other apps consume from Kinesis

QUESTION 34

A company hosts its multi-tier applications on AWS. For compliance, governance, auditing, and security, the company must track configuration changes on its AWS resources and record a history of API calls made to these resources. What should a solutions architect do to meet these requirements?

- A. Use AWS CloudTrail to track configuration changes and AWS Config to record API calls.
- B. Use AWS Config to track configuration changes and AWS CloudTrail to record API calls.
- C. Use AWS Config to track configuration changes and Amazon CloudWatch to record API calls.
- D. Use AWS CloudTrail to track configuration changes and Amazon CloudWatch to record API calls.

Soru:

Bir şirket, çok katmanlı uygulamalarını AWS üzerinde barındırmaktadır. Uyumluluk, yönetişim, denetleme ve güvenlik gereksinimleri nedeniyle, şirket AWS kaynakları üzerindeki yapılandırma değişikliklerini takip etmeli ve bu kaynaklara yapılan API çağrılarının geçmişini kaydetmelidir. Bu gereksinimleri karşılamak için bir çözüm mimarı ne yapmalıdır?

- A. Yapılandırma değişikliklerini takip etmek için AWS CloudTrail, API çağrılarını kaydetmek için AWS Config kullanın.
- B. Yapılandırma değişikliklerini takip etmek için AWS Config, API çağrılarını kaydetmek için AWS CloudTrail kullanın.
- C. Yapılandırma değişikliklerini takip etmek için AWS Config, API çağrılarını kaydetmek

için Amazon CloudWatch kullanın.

D. Yapılandırma değişikliklerini takip etmek için AWS CloudTrail, API çağrılarını kaydetmek için Amazon CloudWatch kullanın.

Soru Analizi:

Şirketin iki ayrı gereksinimi var:

1. AWS kaynaklarındaki konfigürasyon değişikliklerini takip etmek

Bu iş için AWS'de *özel olarak tasarlanmış* servis:



Config şunları yapar:

- Kaynak konfigürasyon değişikliklerini kaydeder
- Kaynakları zaman içindeki durumuyla karşılaştırır
- Uyumluluk (compliance) kontrolleri yapar
- Config history + Config snapshot tutar

2. Kaynaklara yapılan API çağrılarının geçmişini kaydetmek

Bu iş için AWS'de *standart* servis:



CloudTrail:

- Tüm API çağrılarını kaydeder (console, SDK, CLI)
- Kim ne zaman ne yaptı? → audit trail sağlar
- Güvenlik ve denetim için zorunlu

Seçenek Analizi:

A. Use CloudTrail to track configuration changes & Config to record API calls

- CloudTrail konfigürasyon değişikliklerini track edemez.
- AWS Config API çağrılarını kaydetmez.

Her iki fonksiyon da ters yazılmış.

B. Use AWS Config to track configuration changes & AWS CloudTrail to record API calls

Bu tam olarak AWS'in resmi kullanım şeklidir:

Gereksinim	Doğru Servis
------------	--------------

Configuration changes **AWS Config**

API call history **AWS CloudTrail**

Tüm compliance, security audit ve governance için standart mimari budur.

X C. Config → configuration changes, CloudWatch → API calls

- CloudWatch **API çağrıları kaydetmez**.
CloudWatch logları, metrikleri, eventleri işler ama audit trail sağlamaz.

X CloudTrail → configuration changes, CloudWatch → API calls

- CloudTrail configuration change history tutmaz (sadece API calls).
- CloudWatch API çağrılarını kaydededemez.

 **Sonuç**

AWS Config → konfigürasyon değişiklikleri

AWS CloudTrail → API call geçmişi

QUESTION 35

A company is preparing to launch a public-facing web application in the AWS Cloud. The architecture consists of Amazon EC2 instances within a VPC behind an Elastic Load Balancer (ELB). A third-party service is used for the DNS. The company's solutions architect must recommend a solution to detect and protect against large-scale DDoS attacks.

Which solution meets these requirements?

- A. Enable Amazon GuardDuty on the account.
- B. Enable Amazon Inspector on the EC2 instances.
- C. Enable AWS Shield and assign Amazon Route 53 to it.
- D. Enable AWS Shield Advanced and assign the ELB to it.

Soru:

Bir şirket, AWS Cloud üzerinde genel erişime açık bir web uygulaması başlatmaya hazırlanmaktadır. Mimaride, bir Elastic Load Balancer (ELB) arkasında bir VPC içindeki Amazon EC2 instance'ları bulunmaktadır. DNS için üçüncü taraf bir servis kullanılmaktadır. Şirketin çözüm mimarı, büyük ölçekli DDoS saldırılarını tespit edip bunlara karşı koruma sağlayacak bir çözüm önermelidir.

Bu gereksinimleri hangi çözüm karşılar?

- A. Hesapta Amazon GuardDuty’i etkinleştirin.
- B. EC2 instance’larında Amazon Inspector’ı etkinleştirin.
- C. AWS Shield’ı etkinleştirin ve onu Amazon Route 53’e atayın.
- D. AWS Shield Advanced’ı etkinleştirin ve onu ELB’ye atayın.

Soru Analizi:

Gereksinimler:

1. Public-facing web application

→ İnternete açık, saldırı yüzeyi geniş.

2. Büyük ölçekli DDoS saldırılarına karşı koruma

→ Standart güvenlik grupları veya WAF **yeterli değildir**.

3. DNS üçüncü taraf

→ Route 53 *kullanılmıyor*, yani Shield çözümü **ELB üzerinde** olmalı.

4. Hem tespit hem koruma isteniyor

AWS servisleri arasında:

- **Shield Standard** → AWS tarafından otomatik korunma (free), ama büyük ölçekli ve kompleks saldırırlarda yetersiz.
- **Shield Advanced** → Yalnızca büyük ölçekli DDoS saldırılarına karşı tam çözüm.
 - 24/7 DDoS Response Team (DRT)
 - Cost protection
 - Enhanced detection
 - ELB, CloudFront, Route53 gibi kaynaklara eklenebilir.

Seçenek Analizi:

- D. Enable AWS Shield Advanced and assign the ELB to it**

Neden?

- ELB public endpoint olduğu için DDoS saldırılarının ilk hedefidir.
- Shield Advanced sadece ELB, CloudFront, Route 53 gibi AWS edge hizmetlerine atanabilir.
- Large-scale DDoS için AWS’ın *tek tam çözümü* Shield Advanced’tır.

- Üçüncü taraf DNS kullanılmasına rağmen ELB korunabilir.

A. Enable Amazon GuardDuty

GuardDuty:

- Tehdit algılama hizmetidir.
- VPC trafik analizi, IAM davranış analizi yapar.
- **DDoS koruması SAĞLAMAZ.**

B. Enable Amazon Inspector

Inspector:

- EC2 için güvenlik açığı değerlendirmesi yapar (CVE, patch eksikleri).
- **Network saldırıcılarını veya DDoS'u yakalayamaz.**

C. Enable AWS Shield and assign Amazon Route 53 to it

- Shield Standard otomatik zaten, ayrıca "assign" edilmez.
- Shield Advanced Route 53 ile kullanılabilir ama **bu soruda şirket Route 53 kullanmıyor**, 3rd-party DNS kullanıyor.
- Yani yanlış kaynak üzerine uygulanmış.

Sonuç

AWS Shield Advanced'i etkinleştir ve ELB üzerine uygula.

QUESTION 36

A company is building an application in the AWS Cloud. The application will store data in Amazon S3 buckets in two AWS Regions. The company must use an AWS Key Management Service (AWS KMS) customer managed key to encrypt all data that is stored in the S3 buckets. The data in both S3 buckets must be encrypted and decrypted with the same KMS key. The data and the key must be stored in each of the two Regions. Which solution will meet these requirements with the LEAST operational overhead?

- A. Create an S3 bucket in each Region. Configure the S3 buckets to use server-side encryption with Amazon S3 managed encryption keys (SSE-S3). Configure replication between the S3 buckets.
- B. Create a customer managed multi-Region KMS key. Create an S3 bucket in each Region. Configure replication between the S3 buckets. Configure the application to use the KMS key with client-side encryption.

C. Create a customer managed KMS key and an S3 bucket in each Region. Configure the S3 buckets to use server-side encryption with Amazon S3 managed encryption keys (SSE-S3). Configure replication between the S3 buckets.

D. Create a customer managed KMS key and an S3 bucket in each Region. Configure the S3 buckets to use server-side encryption with AWS KMS keys (SSE-KMS). Configure replication between the S3 buckets.

Soru:

Bir şirket AWS Cloud üzerinde bir uygulama geliştirmektedir. Uygulama, iki AWS Bölgesinde (Region) Amazon S3 bucket'larında veri depolayacaktır. Şirket, S3 bucket'larında depolanan tüm verileri şifrelemek için AWS Key Management Service (AWS KMS) müşteri tarafından yönetilen bir anahtar kullanmalıdır. Her iki S3 bucket'ındaki veriler aynı KMS anahtarıyla şifrelenmeli ve çözülebilmelidir. Veri ve anahtar her iki Region'da da bulunmalıdır. En az operasyonel yükle bu gereksinimleri hangi çözüm karşılar?

A. Her Region'da bir S3 bucket oluşturun. S3 bucket'larını Amazon S3 tarafından yönetilen şifreleme anahtarları (SSE-S3) ile sunucu tarafı şifreleme kullanacak şekilde yapılandırın. S3 bucket'ları arasında çoğaltma yapılandırın.

B. Müşteri tarafından yönetilen, çoklu Region destekli bir KMS anahtarı oluşturun. Her Region'da bir S3 bucket oluşturun. S3 bucket'ları arasında çoğaltma yapılandırın. Uygulamayı istemci tarafı şifreleme için bu KMS anahtarını kullanacak şekilde yapılandırın.

C. Müşteri tarafından yönetilen bir KMS anahtarı ve her Region'da bir S3 bucket oluşturun. S3 bucket'larını Amazon S3 tarafından yönetilen şifreleme anahtarları (SSE-S3) ile sunucu tarafı şifreleme kullanacak şekilde yapılandırın. S3 bucket'ları arasında çoğaltma yapılandırın.

D. Müşteri tarafından yönetilen bir KMS anahtarı ve her Region'da bir S3 bucket oluşturun. S3 bucket'larını AWS KMS anahtarları ile sunucu tarafı şifreleme (SSE-KMS) kullanacak şekilde yapılandırın. S3 bucket'ları arasında çoğaltma yapılandırın.

Soru Analizi:

Şirketin gereksinimleri:

✓ 1. Veri iki Region'da S3 bucket'larda saklanacak.

✓ 2. Tüm veri AWS KMS customer-managed key kullanılarak şifrelenmeli.

✓ 3. Her iki Region'daki veri aynı KMS anahtarıyla şifrelenip çözülebilmeli.

✓ 4. Anahtar da iki bölgede mevcut olmalı.

✓ 5. En az operasyonel yük isteniyor.

Seçenek Analizi:

D) Customer managed KMS key + SSE-KMS + replication

Bu gereksinimler bize şunu gösteriyor:

- KMS Multi-Region Keys kullanmak gerekiyor.
- S3 ile en az operasyonel yük = SSE-KMS (server-side encryption with KMS keys)
- Client-side encryption (uygulamada şifreleme) gereksiz yük oluşturur → elenir.

Neden?

- SSE-KMS → S3 veriyi otomatik KMS CMK ile şifreler → **en az operasyonel yük**
- Her Region'da CMK oluşturulabilir veya **multi-Region key (primary + replica)** kullanılabilir → “anahtar her iki Region'da bulunmalı” şartını karşılar
- S3 replication → veriler iki bölge arasında kopyalanır
- S3 SSE-KMS + Multi-Region CMK → aynı anahtar mantığıyla şifreleme/çözme mümkündür

Bu çözüm tüm gereksinimleri ve “en az operasyonel yük” şartını net şekilde karşılayan tek seçenektedir.

A) SSE-S3 + replication

- SSE-S3 → S3 tarafından yönetilen anahtarlar (KMS CMK kullanılmıyor).
- Müşteri yönetimli KMS anahtarı şartını karşılamıyor.

B) Multi-Region KMS key + client-side encryption

- Multi-Region CMK doğru.
- Ancak **client-side encryption** uygulamak daha fazla kod, daha fazla anahtar yönetimi, daha fazla hata olasılığı → **en fazla operasyonel yük**.
- Soru “**least operational overhead**” diyor → bu nedenle elenir.

C) Customer managed KMS key + SSE-S3

- SSE-S3 yine S3 tarafından yönetilen anahtar.
- Seçenek kendi içinde bile çelişkili: CMK oluşturuluyor ama S3 tarafından kullanmıyor.
- Gereksinimi karşılamıyor.

 SONUÇ

DOĞRU CEVAP: D

Nedeni özetle: *Sunucu tarafı SSE-KMS (S3 + KMS) kullanmak, client-side encryption'a göre çok daha az operasyonel yük getirir; multi-Region KMS (replica) ile anahtar materyali ve erişim her iki bölgede de sağlanabilir; S3 replication ile veri eşitlenir.* Bu kombinasyon, hem müşteri-yönetimli KMS kullanım şartını hem de “veri ve anahtar her iki bölgede bulunmalı” gereksinimini en az işletimsel çabaya karşılar.

QUESTION 37

A company recently launched a variety of new workloads on Amazon EC2 instances in its AWS account. The company needs to create a strategy to access and administer the instances remotely and securely. The company needs to implement a repeatable process that works with native AWS services and follows the AWS Well-Architected Framework.

Which solution will meet these requirements with the LEAST operational overhead?

- A. Use the EC2 serial console to directly access the terminal interface of each instance for administration.
- B. Attach the appropriate IAM role to each existing instance and new instance. Use AWS Systems Manager Session Manager to establish a remote SSH session.
- C. Create an administrative SSH key pair. Load the public key into each EC2 instance. Deploy a bastion host in a public subnet to provide a tunnel for administration of each instance.
- D. Establish an AWS Site-to-Site VPN connection. Instruct administrators to use their local on-premises machines to connect directly to the instances by using SSH keys across the VPN tunnel.

Soru:

Bir şirket, AWS hesabındaki Amazon EC2 örneklerinde çeşitli yeni iş yükleri başlattı. Şirketin, örneklerde uzaktan ve güvenli bir şekilde erişmek ve yönetmek için bir strateji oluşturması gerekiyor. Bu çözümün, yerel (native) AWS servislerini kullanması, AWS Well-Architected Framework'e uygun olması ve tekrarlanabilir bir süreç sağlama gerekiyor. Hangi çözüm, EN AZ operasyonel yük ile bu gereksinimleri karşılar?

- A. EC2 serial console'u kullanarak yönetim için her bir örneğin terminal arayüzüne doğrudan erişmek.
- B. Her mevcut ve yeni örneğe uygun IAM rolünü eklemek. AWS Systems Manager Session Manager kullanarak uzaktan SSH oturumu başlatmak.
- C. Yönetim için bir SSH anahtar çifti oluşturmak. Her EC2 örneğine ortak anahtarı

yüklemek. Yönetim tüneli sağlamak için genel bir alt ağa bir bastion host dağıtmak.
D. Bir AWS Site-to-Site VPN bağlantısı kurmak. Yöneticilere kendi lokal makinelerinden
VPN tüneli üzerinden SSH anahtarlarıyla doğrudan örneklerle bağlanmalarını söylemek.

Soru Analizi:

Şirket yeni EC2 instance'ları başlatmış ve bunlara:

- ✓ Uzaktan erişmek
- ✓ Güvenli şekilde yönetmek
- ✓ AWS yerel servislerini kullanmak
- ✓ Tekrarlanabilir (otomasyon dostu) bir süreç oluşturmak
- ✓ En az operasyonel yükle çalışmak

zorunda.

Bu, AWS Well-Architected Framework'te **Security pillars** ve **Operational Excellence** ilkelerine karşılık gelir.

Seçenek Analizi:

B. SSM Session Manager (EN DOĞRU)

AWS'in önerdiği modern, anahtarsız, güvenlik duvarı açmayan yaklaşım → **Systems Manager Session Manager**.

- SSH portu açılmaz
- Anahtar yönetimi yok
- Bastion yok
- VPN yok
- IAM bazlı erişim kontrolü
- Loglama CloudTrail/CloudWatch ile native

Dolayısıyla en düşük operasyonel yük **Session Manager** ile sağlanır.

- ✓ IAM role → erişim kontrolü
- ✓ Hiçbir SSH portu açılmaz
- ✓ Bastion gerekmez
- ✓ VPN gerekmez
- ✓ Anahtar yönetimi yok
- ✓ Loglar merkezi toplanabilir

- ✓ Otomasyon dostu
- ✓ En düşük operasyonel yük

AWS Well-Architected Framework tarafından önerilen yöntemdir.

Bu nedenle **doğru cevap B.**

X A. EC2 serial console kullanmak

Bu yöntem yalnızca **troubleshooting** içindir (boot sorunları, kernel panik gibi).

Normal yönetim için uygun değil.

Network erişimi gerektirmez ama işlevsellik çok kısıtlıdır.

Tekrarlanabilir idari yönetim yöntemi değildir.

X C. SSH key + Bastion host

Anahtar yönetimi (key rotation, dağıtım) gereklidir

Bastion host'u patch, log, security izleme gerektirir → operasyon yükü yüksek

Port 22 açmak zorunda

Modern AWS mimarisine göre eski yöntem

SSM varken gereksiz karmaşık

X D. Site-to-Site VPN + SSH

Çok daha karmaşık

VPN kurulum + yönetim + yüksek maliyet

SSH portu yine açık olmalı

On-prem ortamına bağımlı

En yüksek operasyonel yük

⌚ SONUÇ

En düşük operasyonel yük + AWS native + güvenli + tekrarlanabilir süreç → Session Manager

QUESTION 38

A company is hosting a static website on Amazon S3 and is using Amazon Route 53 for DNS. The website is experiencing increased demand from around the world. The company must decrease latency for users who access the website.

Which solution meets these requirements MOST cost-effectively?

A. Replicate the S3 bucket that contains the website to all AWS Regions. Add Route 53 geolocation routing entries.

B. Provision accelerators in AWS Global Accelerator. Associate the supplied IP addresses with the S3 bucket. Edit the Route 53 entries to point to the IP addresses of the accelerators.

C. Add an Amazon CloudFront distribution in front of the S3 bucket. Edit the Route 53 entries to point to the CloudFront distribution.

D. Enable S3 Transfer Acceleration on the bucket. Edit the Route 53 entries to point to the new endpoint.

Soru:

Bir şirket, statik bir web sitesini Amazon S3 üzerinde barındırmaktadır ve DNS için Amazon Route 53 kullanmaktadır. Web sitesi dünya genelinden artan bir talep almaktadır. Şirket, web sitesine erişen kullanıcıların yaşadığı gecikmeyi (latency) azaltmak zorundadır.

Bu gereksinimleri **en maliyet etkin şekilde** karşılayan çözüm hangisidir?

A. Web sitesini içeren S3 bucket’ını tüm AWS bölgelerine çoğaltmak. Route 53 üzerinde coğrafi konum (geolocation) yönlendirme kayıtları eklemek.

B. AWS Global Accelerator üzerinde hızlandırıcılar oluşturmak. Verilen IP adreslerini S3 bucket’ı ile ilişkilendirmek. Route 53 kayıtlarını hızlandırıcının IP adreslerine yönlendirmek.

C. S3 bucket’ın önüne bir Amazon CloudFront dağıtımını eklemek. Route 53 kayıtlarını CloudFront dağıtımına yönlendirmek.

D. Bucket üzerinde S3 Transfer Acceleration özelliğini etkinleştirmek. Route 53 kayıtlarını yeni uç noktaya yönlendirmek.

Soru Analizi:

Soru şunu istiyor:

- Statik web sitesi S3’te barındırıyor.
- DNS olarak Route 53 kullanıyor.
- Dünya genelinde trafik arttı → **latency düşürülmeli**.
- Çözüm **maliyet açısından en verimli** olmalı.

Seçenek Analizi:

C) CloudFront’u S3 önüne koy ve Route 53’ü CloudFront’a yönlendir → (Doğru)

Bu tür sorularda AWS’nin varsayılan çözümü her zaman:

👉 **Statik içerik + küresel erişim = CloudFront**

Dolayısıyla başlangıç noktasından bile güclü bir aday **CloudFront**.

Neden doğru?

- CDN olduğundan en düşük latency çözümüdür.
- Statik içerik için **ideal** ve **AWS'nin önerdiği** çözüm.
- Edge lokasyonlarda güçlü caching ile dünyanın her yerinde hız sağlar.
- Yönetim yükü çok düşüktür.
- Maliyet olarak diğer seçeneklerden çok daha ucuz:
 - Sadece istek başına küçük ücret
 - S3'den gelen orijin istekleri ciddi şekilde azalır (cache sayesinde)

Sonuç:

- ✓ En düşük latency
 - ✓ En düşük operasyonel yük
 - ✓ En maliyet etkin çözüm
- **Kesin doğru seçenek**

✗ A) S3 bucket'ını tüm bölgelere çoğalt + Route 53 Geolocation

Neden yanlış?

- Cross-region replication her bölgeye yapılrsa **çok pahalı** olur.
- Çok fazla yönetim yükü: her bölge için bucket, replication ayarları, versiyonlama.
- Geolocation routing karmaşık ve pahalı değildir ama **CloudFront kadar iyi latency düşürmez**.
- Çoğaltılmış bucket'lar CDN gibi edge caching sağlamaz.

Sonuç:

- ✓ Teknik olarak çalışır
- ✗ En maliyet etkin değildir
- ✗ Gerekli karmaşıklık

✗ B) AWS Global Accelerator kullan

Neden yanlış?

- Global Accelerator, S3 statik web sitesi için önerilen çözüm değildir.
- Accelerator hedefleri genelde **ALB/NLB/EC2**'dir → **S3 desteklemez**.
- Ayrıca Accelerator, CloudFront'tan **daha pahalıdır**.

- Kullanım amacı farklıdır → TCP tabanlı uygulamalar için network-level hızlandırma.

Sonuç:

X Hedef servis uyumsuz

X Maliyet etkin değil

✗ D) S3 Transfer Acceleration

Neden yanlış?

- Transfer Acceleration **yükleme (upload)** hızını artırmak için tasarlanmıştır.
- Kullanıcıların web sitesindeki dosyalara erişmesi için **CloudFront kadar iyi değildir.**
- Maliyet olarak CloudFront'tan daha pahalı olabilir.
- Web site dağıtıımı için kullanılması önerilmez.

Sonuç:

X Kullanım amacı yanlış

X Maliyet etkin değil

🎯 SONUÇ

Bu AWS'nin "statik site + global kullanıcı + düşük latency + düşük maliyet" kombinasyonu için default ve bariz çözümüdür.

QUESTION 39

A company maintains a searchable repository of items on its website. The data is stored in an Amazon RDS for MySQL database table that contains more than 10 million rows. The database has 2 TB of General Purpose SSD storage. There are millions of updates against this data every day through the company's website. The company has noticed that some insert operations are taking 10 seconds or longer. The company has determined that the database storage performance is the problem.

Which solution addresses this performance issue?

- Change the storage type to Provisioned IOPS SSD.
- Change the DB instance to a memory optimized instance class.
- Change the DB instance to a burstable performance instance class.
- Enable Multi-AZ RDS read replicas with MySQL native asynchronous replication.

Soru:

Bir şirket, web sitesinde aranabilir (searchable) bir öğe deposu barındırıyor. Veriler, 10 milyon satırdan fazla kayıt içeren bir Amazon RDS for MySQL veritabanı tablosunda saklanmaktadır. Veritabanı, 2 TB Genel Amaçlı SSD (General Purpose SSD / gp2 veya gp3) depolamaya sahiptir. Şirketin web sitesi üzerinden bu veri üzerinde her gün milyonlarca güncelleme yapılmaktadır.

Şirket, bazı insert işlemlerinin 10 saniye veya daha uzun süredüğünü fark etmiştir. Şirket, sorunun veritabanı depolama performansından kaynaklandığını belirlemiştir.

Bu performans sorununu hangi çözüm giderir?

- A. Depolama türünü Provisioned IOPS SSD'ye (io1/io2) değiştirmek.
- B. DB instance'ı bellek optimize edilmiş instance sınıfına değiştirmek.
- C. DB instance'ı burstable (patlama kapasiteli) performans sınıfına değiştirmek.
- D. Multi-AZ RDS read replica'larını MySQL'in yerel asenkron replikasyonu ile etkinleştirmek.

Soru Analizi:

Soru bize şunları söylüyor:

- RDS MySQL veritabanı 10+ milyon satır içeriyor.
- 2 TB **General Purpose SSD (gp2/gp3)** kullanılıyor.
- Günde **milyonlarca update** ve **insert** gerçekleşiyor.
- Insert işlemleri **10 saniyeye kadar uzuyor** → ciddi performans sorunu.
- Şirket araştırmış ve **sorunun depolama performansı** olduğu tespit edilmiş.

Yani problem CPU, RAM, replica eksikliği değil → **disk I/O yetersizliği**.

General Purpose SSD (özellikle gp2/gp3) yoğun OLTP yüklerde sınırlı IOPS verebilir.

Çözüm: **IOPS garantili depolama** (Provisioned IOPS).

Seçenek Analizi:

A) Depolamayı Provisioned IOPS SSD'ye (io1/io2) değiştirmek

- gp2/gp3 diskler burst mantığıyla çalışır → sürekli yüksek I/O gerektiren veritabanlarında tıkanır.
- io1/io2 ise **yüksek ve sabit IOPS sağlar**, veritabanı gibi yoğun yazma/okuma yapan sistemler için tasarlanmıştır.
- Insert işlemlerinin 10 saniyeye çıkması → depolama gecikmesi (latency) ve yetersiz IOPS'ın klasik göstergesidir.

- AWS'nin RDS performans sorunlarına yönelik en temel çözümü → **Provisioned IOPS'a geçmek.**

Sonuç:

👉 Sorunun doğrudan kök nedenine çözüm getirir.

✓ En doğru ve AWS'nin önerdiği çözüm.

✗ **B) Instance'ı memory optimized sınıfına geçirmek → Yanlış**

- RAM artırmak fayda sağlar mı? Evet ama:
 - Soru zaten diyor ki problem depolama performansı.
 - RAM artırmak IOPS artırmaz.
 - Insert gecikmesi disk aygıtından kaynaklanıyorsa RAM çözmez.

Sonuç: Depolama sorunu çözülmez.

✗ **C) Burstable instance sınıfına geçmek (t3/t4g) → Çok yanlış**

- Burstable instance'lar CPU kredisi bittiğinde **yavaşlar**.
- Yoğun OLTP veritabanları için uygun değildir.
- Zaten sorun CPU değil, depolamadır.
- Performans daha da kötüleşebilir.

Sonuç: Performans sorununun nedeniyle ilgisi yok.

✗ **D) Multi-AZ RDS read replica + async replication → Yanlış**

- Read replica okuma yükünü hafifletir, **yazma** yükünü hafifletmez.
- Insert işlemleri her zaman **primary üzerinde yapılır** → disk yine primary'dedir.
- Replica eklemek write IOPS'ı artırmaz.
- Soru "depodan kaynaklanan insert gecikmesi" diyor → read replica çözmez.

Sonuç: Sorunun temel nedeni olan I/O darboğazını çözmez.

🎯 SONUÇ

Bu çözüm doğrudan depolama performansını artırır ve IO-bound OLTP yüklerinde AWS'nin standart çözümüdür.

A company has thousands of edge devices that collectively generate 1 TB of status alerts each day. Each alert is approximately 2 KB in size. A solutions architect needs to implement a solution to ingest and store the alerts for future analysis. The company wants a highly available solution. However, the company needs to minimize costs and does not want to manage additional infrastructure. Additionally, the company wants to keep 14 days of data available for immediate analysis and archive any data older than 14 days.

What is the MOST operationally efficient solution that meets these requirements?

- A. Create an Amazon Kinesis Data Firehose delivery stream to ingest the alerts. Configure the Kinesis Data Firehose stream to deliver the alerts to an Amazon S3 bucket. Set up an S3 Lifecycle configuration to transition data to Amazon S3 Glacier after 14 days.
- B. Launch Amazon EC2 instances across two Availability Zones and place them behind an Elastic Load Balancer to ingest the alerts. Create a script on the EC2 instances that will store the alerts in an Amazon S3 bucket. Set up an S3 Lifecycle configuration to transition data to Amazon S3 Glacier after 14 days.
- C. Create an Amazon Kinesis Data Firehose delivery stream to ingest the alerts. Configure the Kinesis Data Firehose stream to deliver the alerts to an Amazon OpenSearch Service (Amazon Elasticsearch Service) cluster. Set up the Amazon OpenSearch Service (Amazon Elasticsearch Service) cluster to take manual snapshots every day and delete data from the cluster that is older than 14 days.
- D. Create an Amazon Simple Queue Service (Amazon SQS) standard queue to ingest the alerts, and set the message retention period to 14 days. Configure consumers to poll the SQS queue, check the age of the message, and analyze the message data as needed. If the message is 14 days old, the consumer should copy the message to an Amazon S3 bucket and delete the message from the SQS queue.

Soru:

Bir şirketin binlerce edge cihazı vardır ve bu cihazlar birlikte günde 1 TB durum uyarısı (status alert) üretmektedir. Her bir uyarı yaklaşık 2 KB boyutundadır. Bir solutions architect'in bu uyarıları almak (ingest) ve gelecekte analiz için depolamak üzere bir çözüm uygulaması gerekmektedir.

Şirket **yüksek erişilebilirliğe** sahip bir çözüm istiyor. Ancak şirket **maliyetleri en az indirmek ve ek altyapı yönetmek istemiyor**. Ek olarak, şirket **14 günlük veriyi anında analiz için erişilebilir tutmak ve 14 günden eski tüm veriyi arşivlemek** istiyor.

Bu gereksinimleri karşılayan **en operasyonel olarak verimli** çözüm hangisidir?

- A. Uyarıları almak için bir Amazon Kinesis Data Firehose teslim akışı oluşturun. Kinesis Data Firehose'u, uyarıları bir Amazon S3 bucket'ına teslim edecek şekilde yapılandırın.

14 gün sonra veriyi Amazon S3 Glacier'a geçirecek bir S3 Lifecycle yapılandırması ayarlayın.

B. İki Availability Zone'a yayılmış Amazon EC2 instance'ları başlatın ve bunları bir Elastic Load Balancer arkasına yerleştirin. EC2 instance'larında, uyarıları bir Amazon S3 bucket'ına kaydedecek scriptler oluşturun. 14 gün sonra veriyi Amazon S3 Glacier'a geçirecek bir S3 Lifecycle yapılandırması ayarlayın.

C. Uyarıları almak için bir Amazon Kinesis Data Firehose teslim akışı oluşturun. Kinesis Data Firehose'u uyarıları bir Amazon OpenSearch Service (Amazon Elasticsearch Service) kümesine teslim edecek şekilde yapılandırın. Amazon OpenSearch Service kümesini her gün manuel snapshot alacak ve 14 günden eski veriyi kümeden silecek şekilde ayarlayın.

D. Uyarıları almak için bir Amazon SQS standard kuyruğu oluşturun ve mesaj saklama süresini 14 gün olarak ayarlayın. Tüketicileri SQS kuyruğunu poll edecek, mesajın yaşı 14 günden küçükse analiz edecek şekilde yapılandırın. Mesaj 14 günlüğe ulaştığında tüketici mesajı bir Amazon S3 bucket'ına kopyalamalı ve kuyrukta silmelidir.

Soru Analizi:

Verilen bilgiler:

- Binlerce edge device → günde **1 TB** veri üretiyor.
- Her bir alert: **2 KB** (çok küçük mesajlar → streaming'e uygun).
- Şirket:
 - **Yüksek erişilebilirlik (HA)** istiyor.
 - **Uygun maliyet** istiyor.
 - **Ek altyapı (EC2, cluster, server) yönetmek istemiyor.**
 - **14 günlük veri sıcak (immediate analysis)**
 - **14 günden eski veri arşive gidecek (Glacier).**

Bu gereksinimlerden doğal sonuç:

- 👉 **Tamamen serverless**
- 👉 **Doğrudan S3'e akış**
- 👉 **Lifecycle ile otomatik Glacier geçisi**
- 👉 **Kinesis Firehose gibi yönetimsiz ingestion servisi**

yani minimal operasyonel yük + düşük maliyet + HA.

Firehose → “Zero administration, fully managed ingestion”.

Seçenek Analizi:

✓ A) Kinesis Data Firehose → S3 → Lifecycle ile Glacier

Bu seçenek tüm gereksinimleri karşılıyor:

- Firehose **%100 managed**, hiçbir sunucu yok → en düşük operasyon yükü
- Yüksek hacimli streaming veriyi “otomatik olarak” S3’e yazar
- Edge cihazlardan gelen 1 TB/gün yükü rahat kaldırır
- S3 Lifecycle → 14 gün sonra otomatik Glacier’'a
- En ucuz depolama kombinasyonu: **S3 Standard → S3 Glacier**
- HA Firehose + HA S3 = Kesintisiz altyapı

Tam olarak istenen çözüm.

✗ B) EC2 + ELB + script ile S3’e yazma

- EC2 = sunucu yönetimi, patching, ölçeklendirme, monitoring → **şirket bunu istemiyor.**
- ELB + Auto Scaling vs. gerektirir → operasyon yükü artar.
- Maliyet Firehose+S3’e göre çok daha fazla olur.
- Soru açıkça: “Ek altyapı yönetmek istemiyoruz.”

Bu seçenek sorunun gereksinimlerine aykırı.

✗ C) Firehose → OpenSearch → Snapshot + Silme

- OpenSearch kümesi yönetilmesi gereken bir altyapıdır:
 - Node sayısı
 - Versiyon
 - Hot/warm storage yönetimi
 - Snapshot storage yönetimi
- 1 TB/gün ingest → OpenSearch maliyeti **uçuk** olur.
- Şirketin tek istediği şey: **future analysis için storage**, analitik motor istemiyor.
- OpenSearch gereksiz karmaşıklık ve masraf ekler.

✗ D) SQS Standard Queue → 14 gün retention → Consumer → S3

- SQS standart kuyruğunun maksimum retention süresi **14 gündür**; yani **14 gün sonra mesaj zaten silinir**, Glacier’'a taşınacak veri kaybolur.

- Ayrıca:
 - 1 TB/gün * 2 KB mesajlar → 500 milyon mesaj/gün
SQS bunu kaldırırmaz (milyonlarca TPS).
 - Kuyruktan mesaj çekmek için consumer cluster gereklidir → yönetim yükü artar.

Operasyonel verimsiz ve mimari olarak hatalı.

SONUÇ

Kinesis Data Firehose → S3 → Lifecycle → Glacier

Bu çözüm:

- ✓ Tamamen yönetilen (no EC2)
 - ✓ En düşük maliyet
 - ✓ En yüksek ölçeklenebilirlik
 - ✓ HA sağlar
 - ✓ 14 günlük sıcak veri + 14 günden eski arşiv ihtiyacını mükemmel karşılar
-

QUESTION 41

A company's application integrates with multiple software-as-a-service (SaaS) sources for data collection. The company runs Amazon EC2 instances to receive the data and to upload the data to an Amazon S3 bucket for analysis. The same EC2 instance that receives and uploads the data also sends a notification to the user when an upload is complete. The company has noticed slow application performance and wants to improve the performance as much as possible.

Which solution will meet these requirements with the LEAST operational overhead?

- Create an Auto Scaling group so that EC2 instances can scale out. Configure an S3 event notification to send events to an Amazon Simple Notification Service (Amazon SNS) topic when the upload to the S3 bucket is complete.
- Create an Amazon AppFlow flow to transfer data between each SaaS source and the S3 bucket. Configure an S3 event notification to send events to an Amazon Simple Notification Service (Amazon SNS) topic when the upload to the S3 bucket is complete.
- Create an Amazon EventBridge (Amazon CloudWatch Events) rule for each SaaS source to send output data. Configure the S3 bucket as the rule's target. Create a second EventBridge (Cloud Watch Events) rule to send events when the upload to the S3 bucket is complete. Configure an Amazon Simple Notification Service (Amazon SNS) topic as the second rule's target.

D. Create a Docker container to use instead of an EC2 instance. Host the containerized application on Amazon Elastic Container Service (Amazon ECS). Configure Amazon CloudWatch Container Insights to send events to an Amazon Simple Notification Service (Amazon SNS) topic when the upload to the S3 bucket is complete.

Soru:

Bir şirketin uygulaması, veri toplama amacıyla birden fazla software-as-a-service (SaaS) kaynağıyla entegredir. Şirket, verileri almak ve analiz için bir Amazon S3 kovasına yüklemek için Amazon EC2 örnekleri kullanmaktadır. Verileri alan ve S3'e yükleyen aynı EC2 örneği, yükleme tamamlandığında kullanıcıya bir bildirim de göndermektedir.

Şirket, uygulamada yavaş performans gözlemlemiş ve performansı mümkün olduğunda artırmak istemektedir.

En az operasyonel yük ile bu gereksinimleri karşılayacak çözüm hangisidir?

- A. Bir Auto Scaling grubu oluşturarak EC2 örneklerinin ölçeklenmesini sağlayın. S3 kovasına yükleme tamamlandığında bir Amazon SNS konusuna olay göndermek için bir S3 olay bildirimi yapılandırın.
- B. Her SaaS kaynağı ile S3 kovası arasında veri aktarımı yapmak için bir Amazon AppFlow akışı oluşturun. S3 kovasına yükleme tamamlandığında bir Amazon SNS konusuna olay göndermek için bir S3 olay bildirimi yapılandırın.
- C. Her SaaS kaynağı için çıktı verisini gönderecek bir Amazon EventBridge (CloudWatch Events) kuralı oluşturun. S3 kovasını bu kuralın hedefi olarak yapılandırın. S3 kovasına yükleme tamamlandığında olay gönderen ikinci bir EventBridge kuralı daha oluşturun. Bu ikinci kuralın hedefi olarak bir Amazon SNS konusu yapılandırın.
- D. EC2 örneği yerine kullanılmak üzere bir Docker konteyneri oluşturun. Konteynerli uygulamayı Amazon ECS üzerinde barındırın. S3 kovasına yükleme tamamlandığında Amazon SNS konusuna olay göndermek için Amazon CloudWatch Container Insights yapılandırın.

Soru Analizi:

Şirketin mevcut mimarisi şöyle:

- Birden fazla **SaaS kaynağı** → veri gönderiyor.
- Şirket **EC2 instance** kullanıyor:
 - SaaS kaynaklarından veriyi alıyor,

- S3 bucket'a yükliyor,
- yükleme bitince kullanıcıya bildirim gönderiyor.

! Sorun:

EC2 üzerindeki tüm bu işlemler performansı düşürüyor.

! Hedefler:

- Performansı **maksimum** artırmak.
- **En az operasyonel yük** ile çözmek.
- Veri akışı: SaaS → S3 → SNS (bildirim).

Seçenek Analizi:

B Seçeneği

Amazon AppFlow + S3 event notification to SNS

- Her SaaS kaynağı ile doğrudan AppFlow akışı kuruluyor.
- AppFlow veriyi doğrudan S3'e taşıyor.
- EC2 tamamen devreden çıkıyor → **tüm yük ortadan kalkıyor**
- Yükleme tamamlandığında S3 → SNS olayı gönderiliyor.

✓ En az operasyonel yük

✓ Tamamen managed service (AppFlow + S3 + SNS)

✓ EC2 yok = performans sorunları yok

✓ Şirketin istediği "performans maksimize / operasyon minimum" kapsamına mükemmel uyum

A Seçeneği

Auto Scaling + S3 event notification to SNS

- EC2 üzerinde veri alma işi devam ediyor → **yük hâlâ EC2'de**
- Sadece ölçeklenme ekleniyor
- Operasyonel yük artıyor (ASG yönetimi)
- SaaS → EC2 → S3 → SNS mimarisi devam ediyor

EC2 kullanımı devam ettiği için istenen "en az işletme yükü" karşılanmıyor.

Performans sorunu EC2 kaynaklı olabilir; çözülmüyor.

C Seçeneği

EventBridge kuralı ile SaaS → S3 → SNS

- Tüm SaaS kaynaklarının EventBridge ile S3'e veri göndermesi çok gerçekçi olmayabilir (her SaaS EventBridge'i desteklemez).
- EventBridge hedef olarak doğrudan S3'e veri yükleyemez; Lambda gereklidir → eksik ve yanlış bir yapı.

Teknik olarak eksik veya hatalı

Daha çok operasyonel yük

SaaS entegrasyonları AppFlow kadar kapsayıcı değil

✗ D Seçeneği

ECS üzerinde konteyner + Container Insights + SNS

- EC2 yerine ECS, fakat hâlâ compute yönetimi var
- Container Insights veri upload bitişini otomatik algılamaz — yanlış yönlendirme
- SaaS → ECS → S3 → SNS süreci devam ediyor

Compute management devam ediyor

Gereksiz operasyonel yük

Doğru mecra değil (Container Insights telemetri içindir)

🎯 SONUÇ:

→ B Şıkları: Amazon AppFlow + S3 Event Notification + SNS

- EC2 tamamen ortadan kalkıyor
- En yüksek performans
- Managed service → en düşük operasyonel maliyet
- SaaS → AppFlow → S3 → SNS akışı tam istenen yapı

QUESTION 42

A company runs a highly available image-processing application on Amazon EC2 instances in a single VPC. The EC2 instances run inside several subnets across multiple Availability Zones. The EC2 instances do not communicate with each other. However, the EC2 instances download images from Amazon S3 and upload images to Amazon S3 through a single NAT gateway. The company is concerned about data transfer charges.

What is the MOST cost-effective way for the company to avoid Regional data transfer charges?

- A. Launch the NAT gateway in each Availability Zone.
- B. Replace the NAT gateway with a NAT instance.
- C. Deploy a gateway VPC endpoint for Amazon S3.
- D. Provision an EC2 Dedicated Host to run the EC2 instances.

Soru:

Bir şirket, tek bir VPC içinde Amazon EC2 instance’larında yüksek erişilebilirlik sağlayan bir görüntü işleme uygulaması çalıştırmaktadır. EC2 instance’ları birden fazla Availability Zone’daki çeşitli subnet’lerde çalışmaktadır. EC2 instance’ları birbirleriyle iletişim kurmamaktadır. Ancak EC2 instance’ları, tek bir NAT gateway üzerinden Amazon S3’ten görüntü indirmekte ve S3’e görüntü yüklemektedir. Şirket, veri aktarım maliyetleri (data transfer charges) konusunda endişeliidir.

Şirketin **Regional data transfer ücretlerinden kaçınması** için **en maliyet etkin çözüm** nedir?

- A. Her Availability Zone’da bir NAT gateway başlatmak.
- B. NAT gateway’i bir NAT instance ile değiştirmek.
- C. Amazon S3 için bir gateway VPC endpoint dağıtmak.
- D. EC2 instance’larını çalıştırmak için bir EC2 Dedicated Host sağlamak.

Soru Analizi:

Senaryo şu şekilde:

- EC2 instance’ları birden fazla AZ’de çalışıyor.
- Hepsi Amazon S3 ile iletişim kuruyor (dosya indirip yüklüyor).
- S3’e erişim **tek bir NAT Gateway** üzerinden yapılmıyor.
- Bu durum şu maliyete yol açıyor:

! AZ → NAT Gateway → S3 trafiğinde bölgesel veri transfer ücreti oluşur

Çünkü:

- NAT Gateway AZ-specific bir kaynaktır (her AZ’de kendi NAT’ı yoksa, diğer AZ’den trafik “cross-AZ” kabul edilir).
- EC2 → NAT Gateway → S3 trafiği gereksiz veri transferi oluşturur.

Şirket **data transfer charges (özellikle cross-AZ ve NAT ücretleri)** konusunda endişeli.

Bu durumda NAT Gateway’i tamamen devre dışı bırakmak ve S3’e doğrudan erişmek en maliyet-etkin çözümüdür.

Seçenek Analizi:

C. Amazon S3 için bir gateway VPC endpoint dağıtmak. (Doğru Cevap)

→ AWS buna çözüm olarak **S3 Gateway VPC Endpoint** sunar.

Gateway endpoint üzerinden S3'e erişim:

- Tamamen ücretsizdir.
- NAT Gateway üzerinden geçmez.
- Veri transferi VPC içi kabul edildiği için cross-AZ ve NAT ücretleri ortadan kalkar.

Bu çözüm:

- ✓ NAT Gateway'i tamamen bypass eder
- ✓ S3'e erişim **bedavadır**
- ✓ Cross-AZ trafiği oluşmaz
- ✓ En ucuz, en basit yöntemdir
- ✓ High availability otomatik olarak gelir

S3'e erişmek için NAT kullanmak gereksiz ve pahalıdır. AWS'nin önerdiği çözüm budur.

A. Her Availability Zone'da bir NAT gateway başlatmak.

Bu çözüm:

- Cross-AZ NAT ücretlerini azaltır.
- Ama **NAT Gateway ücreti daha da artar**
- Her NAT Gateway ayda ~32 USD + data işlem ücreti

→ Maliyeti azaltmaz, artırır.

Bu nedenle yanlış.

B. NAT gateway'i NAT instance ile değiştirmek.

Bu çözüm:

- Daha ucuz olabilir ancak:
 - NAT instance yönetimi gereklidir
 - HA sağlamak için ASG + multiple instance gereklidir
 - Yine cross-AZ data transfer charges oluşur
 - S3'e erişimde NAT hâlâ arada olur → gereksiz maliyet

→ Hem operasyonel yük fazla hem data transfer ücreti devam eder.

Bu yüzden yanlış.

 **D. EC2 Dedicated Host sağlamak.**

Bu çözüm:

- Data transfer maliyetleriyle ilgisi yok.
- Dedicated Host çok pahalıdır.
- NAT Gateway veya S3 bağlantısını etkilemez.

→ Tamamen alakasız.

 **SONUÇ:**

S3 Gateway VPC Endpoint kullanmak, S3 erişimini ücretsiz hale getirir ve NAT/cross-AZ maliyetlerini ortadan kaldırır.

QUESTION 43

A company has an on-premises application that generates a large amount of time-sensitive data that is backed up to Amazon S3. The application has grown and there are user complaints about internet bandwidth limitations. A solutions architect needs to design a long-term solution that allows for both timely backups to Amazon S3 and with minimal impact on internet connectivity for internal users.

Which solution meets these requirements?

- Establish AWS VPN connections and proxy all traffic through a VPC gateway endpoint.
- Establish a new AWS Direct Connect connection and direct backup traffic through this new connection.
- Order daily AWS Snowball devices. Load the data onto the Snowball devices and return the devices to AWS each day.
- Submit a support ticket through the AWS Management Console. Request the removal of S3 service limits from the account.

Soru:

Bir şirketin, büyük miktarda **zaman açısından hassas** veri üreten bir şirket içi (on-premises) uygulaması vardır. Bu veriler Amazon S3'ye yedeklenmektedir. Uygulama büyündükçe, internet bant genişliği kısıtlamalarıyla ilgili kullanıcı şikayetleri artmıştır. Bir solutions architect'in hem Amazon S3'ye **zamanında yedekleme** yapmayı sağlayacak

hem de şirket içi kullanıcıların internet bağlantısına **en az etkiyi** yapacak uzun vadeli bir çözüm tasarlaması gerekmektedir.

Bu gereksinimleri hangi çözüm karşılar?

- A.** AWS VPN bağlantıları kurun ve tüm trafiği bir VPC gateway endpoint üzerinden yönlendirin.
- B.** Yeni bir AWS Direct Connect bağlantısı oluşturun ve yedekleme trafiğini bu bağlantı üzerinden yönlendirin.
- C.** Günlük olarak AWS Snowball cihazları sipariş edin. Verileri Snowball cihazına yükleyip her gün AWS'ye geri gönderin.
- D.** AWS Management Console üzerinden bir destek talebi oluşturun ve hesaptaki S3 servis limitlerinin kaldırılmasını isteyin.

Soru Analizi:

Senaryo:

- Şirketin on-prem bir uygulaması var.
- Bu uygulama **büyük miktarda ve zaman açısından kritik** (time-sensitive) veri üretiyor.
- Bu veriler **Amazon S3'e yedekleniyor**.
- Uygulama büyüğü için **internet bant genişliği yetmemeye** başladı; kullanıcılar şikayet ediyor.
- Çözüm:
 - **Uzun vadeli olmalı**
 - **S3'e zamanında yedekleme** sağlamalı
 - **İç ağ kullanıcılarını minimum etkilemeli**
=> Yani **S3'e giden trafik interneti boğmasın**.

Seçenek Analizi:

- B)** Yeni bir AWS Direct Connect bağlantısı kur ve yedekleme trafiğini bu bağlantı üzerinden yönlendir.

Bu tür durumlarda AWS'in standart çözümü:

👉 **Amazon S3'e giden trafiği internete çıkmadan özel bir bağlantı üzerinden geçirmek: AWS Direct Connect**

- Direct Connect (DX):
 - **Özel hat**, interneti bypass eder.

- Yüksek bant genişliği sağlar (1 Gbps – 100 Gbps).
- S3'e özel olarak route edilebilir (S3 için Direct Connect Gateway).
- Uzun vadeli bir çözüm sağlar.
- İç kullanıcıların internetini artık S3 upload trafiği tüketmez.

■ A) AWS VPN bağlantıları kur ve tüm trafiği bir VPC gateway endpoint üzerinden yönlendir.

- S3 için **gateway endpoint**, sadece **VPC içinden** gelen trafiği yönlendirir.
- On-prem'den gelen trafik VPC endpoint'e NAT gibi çalışmaz → **desteklemez**.
- VPN hattı zaten **internet üzerinden** çalışır → bant genişliği kazancı yok.
- Uzun vadeli, yüksek bant genişliği için uygun değil.

■ C) Günlük Snowball cihazları sipariş et. Veriyi yükleyip her gün AWS'e gönder.

- Snowball **offline bulk transfer** içindir.
- “Daily Snowball” pratik değildir, lojistik açıdan mümkün değildir.
- “Time-sensitive” veriler için uygun değildir → gecikme olur.
- Long-term sürekli backup için **doğu approach** değildir.

■ D) AWS Support'a ticket aç ve S3 limitlerini kaldırmasını iste.

- S3 limit problemi yok → problem **internet bant genişliği**.
- Support **internet bağlantınızı iyileştiremez**.
- Bu seçenek gereksiz ve konu dışı.

⌚ Sonuç:

AWS Direct Connect ile özel hat üzerinden S3'e trafik yönlendirmek istenen tüm şartları karşılar.

QUESTION 44

A company has an Amazon S3 bucket that contains critical data. The company must protect the data from accidental deletion.

Which combination of steps should a solutions architect take to meet these requirements? (Choose two.)

- A. Enable versioning on the S3 bucket.

- B. Enable MFA Delete on the S3 bucket.
- C. Create a bucket policy on the S3 bucket.
- D. Enable default encryption on the S3 bucket.
- E. Create a lifecycle policy for the objects in the S3 bucket.

Soru:

Bir şirketin kritik veriler içeren bir Amazon S3 kovası (bucket) vardır.

Şirket, bu verileri **yanlışlıkla silinmeye karşı korumak** zorundadır.

Bir çözüm mimarı bu gereksinimleri karşılamak için hangi adımların kombinasyonunu atmalıdır?

(İki tanesini seçin.)

- A. S3 kovasında versiyonlamayı etkinleştir.
- B. S3 kovasında MFA Delete özelliğini etkinleştir.
- C. S3 kovası için bir bucket politika (bucket policy) oluştur.
- D. S3 kovasında varsayılan şifrelemeyi etkinleştir.
- E. S3 kovası içindeki nesneler için bir lifecycle (ömür döngüsü) politikası oluştur.

Soru Analizi:

Şirketin Amazon S3 üzerinde kritik verileri var ve bu veriler **yanlışlıkla silinmeye karşı korunmalı**.

S3'te yanlışlıkla silinmeyi önlemek için kullanılabilecek iki ana mekanizma vardır:

- 1. Versioning (Sürümleme)**
 - Bir obje silinse bile eski sürümü kalır.
 - Silme aslında "delete marker" olur, veri kaybolmaz.
- 2. MFA Delete**
 - Bir objeyi tamamen silmek için ek olarak MFA doğrulaması ister.
 - En güçlü koruma yöntemidir ama *çok az kişi etkinleştirir* çünkü *CLI gerektirir, konsoldan aktif edilemez*.

Bu iki tanesi **yanlışlıkla silinmeyi önlemede en güçlü ve doğru çözüm**dur.

Seçenek Analizi:

- A. Enable versioning on the S3 bucket.**
- Versioning aktif olursa:
 - Obje yanlışlıkla silinse bile sadece "delete marker" eklenir.

- Eski sürüm durmaya devam eder → Veri kaybolmaz.
- Yanlışlıkla silinmeyi önlemede ilk şarttır.

 **B. Enable MFA Delete on the S3 bucket.**

- S3'ten bir obje *gerçek anlamda* silinmek istendiğinde MFA ister.
- Yanlışlıkla silinme ihtimalini neredeyse imkânsız yapar.
- Özellikle kritik veri için AWS tarafından önerilir.
- **Not:** Yalnızca CLI ile etkinleştirilebilir.

 **C. Create a bucket policy on the S3 bucket. → YANLIŞ**

- Bucket policy eklemek silinmeyi engelmez.
- Ancak istenirse silme işlemlerine izin vermemek için özel policy yazılabilir ama:
 - Soru hangi politikanın yazılacağını söylemiyor.
 - Genel bucket policy yanlışlıkla silinmeyi otomatik engelmez.
- Soru "en temel koruma" istiyor → versioning + MFA delete.

 **D. Enable default encryption on the S3 bucket. → YANLIŞ**

- Şifreleme sadece veriyi korur (gizlilik).
- **Silinmeye karşı bir etkisi yoktur.**

 **E. Create a lifecycle policy for the objects in the S3 bucket. → YANLIŞ**

- Lifecycle policy objeleri otomatik olarak silebilir!
- Yanlışlıkla silinmeyi önleme durumu ile çelişir.
- Bu seçenek risklidir → tamamen ters etki yaratır.

 **Sonuç:**

- ✓ **A. Versioning etkinleştir**
- ✓ **B. MFA Delete etkinleştir**

Bu iki özellik birlikte kullanıldığında S3 üzerindeki kritik veriler "yanlışlıkla silinmeye karşı" en yüksek düzeyde korunur.

QUESTION 45

A company has a data ingestion workflow that consists of the following:

- An Amazon Simple Notification Service (Amazon SNS) topic for notifications about new data deliveries
- An AWS Lambda function to process the data and record metadata

The company observes that the ingestion workflow fails occasionally because of network connectivity issues. When such a failure occurs, the Lambda function does not ingest the corresponding data unless the company manually reruns the job.

Which combination of actions should a solutions architect take to ensure that the Lambda function ingests all data in the future? (Choose two.)

- A. Deploy the Lambda function in multiple Availability Zones.
- B. Create an Amazon Simple Queue Service (Amazon SQS) queue, and subscribe it to the SNS topic.
- C. Increase the CPU and memory that are allocated to the Lambda function.
- D. Increase provisioned throughput for the Lambda function.
- E. Modify the Lambda function to read from an Amazon Simple Queue Service (Amazon SQS) queue.

Soru:

Bir şirketin aşağıdakilerden oluşan bir veri alma (data ingestion) iş akışı vardır:

- Yeni veri teslimatları hakkında bildirimler için bir Amazon Simple Notification Service (Amazon SNS) konusu
- Veriyi işlemek ve metadata kaydetmek için bir AWS Lambda fonksiyonu

Şirket, veri alma iş akışının bazen ağ bağlantısı sorunları nedeniyle başarısız olduğunu gözlemlemektedir. Böyle bir hata gerçekleştiğinde, şirket işi manuel olarak yeniden çalıştırılmışça Lambda fonksiyonu ilgili veriyi almamaktadır.

Gelecekte Lambda fonksiyonunun tüm verileri almasını sağlamak için bir solutions architect hangi eylem kombinasyonunu gerçekleştirmelidir? (İki tanesini seçin.)

- A. Lambda fonksiyonunu birden fazla Availability Zone'da dağıtın.
- B. Bir Amazon Simple Queue Service (Amazon SQS) kuyruğu oluşturun ve bunu SNS konusuna abone edin.
- C. Lambda fonksiyonuna ayrılan CPU ve belleği artırın.
- D. Lambda fonksiyonu için sağlanmış (provisioned) throughput'u artırın.
- E. Lambda fonksiyonunu bir Amazon Simple Queue Service (Amazon SQS) kuyruğundan okuyacak şekilde değiştirin.

Soru Analizi:

Akış: **SNS topic → Lambda** (Lambda veriyi işleyor)

Bazen **ağ bağlantısı** sorunları oluyor ve bu durumda Lambda ilgili veriyi **alamıyor**; veri ancak manuel yeniden çalıştırımla işleniyor.

Amaç: gelecekte **hiçbir veri kaybolmasın**, otomatik olarak yeniden işlensin.

Seçenek Analizi:

B. Bir SQS kuyruğu oluşturun ve onu SNS konusuna abone edin.

- SNS → SQS aboneliğiyle mesajlar **kalıcı olarak** SQS kuyruğuna konur.
- Ağ veya tüketici (Lambda) problemi olsa bile mesajlar SQS'te saklanır; kaybolmaz.
- SQS, visibility timeout, DLQ gibi mekanizmalarla güvenilir teslimat sağlar.

E. Lambda fonksiyonunu SQS kuyruğundan okuyacak şekilde değiştirin.

- Lambda, SQS tetikleyicisi (event source mapping) ile **kuyruktan** mesajları çeker.
- Lambda-SQS entegrasyonu, hatalarda yeniden deneme, toplu işleme (batch) ve başarısız mesajları DLQ'ye yönlendirme gibi güvenilirlik özellikleri sunar.
- Bu kombinasyon (SNS→SQS→Lambda) ağ kesintilerinden kaynaklanan veri kayıplarını önler ve otomatik işleme sağlar.

A. Lambda'yı birden fazla AZ'da dağıtın.

- Lambda zaten bölgesel ve AWS tarafından yüksek erişilebilirlik sağlanır; tek tek AZ dağıtıımı müşteri tarafından yapılmaz.
- Bu, ağ kesintileri veya kısa süreli bağlantı hataları nedeniyle kaybolan mesajları çözmez. (Mesajın kaybolmasının nedeni push modelindeki eksiklik/durability eksikliği.)

C. Lambda'ya ayrılan CPU ve belleği artırın.

- Kaynak artırmak işlem performansını etkiler ama **ağ bağlantı sorunlarını** veya mesajın kaybolmasını çözmez.

D. Provisioned throughput artırın.

- Lambda için “provisioned throughput” ifadesi yanlış bağlamda; provisioned concurrency olsa bile bu ağ hatalarını çözmez.
- Sorun ağ/mesaj güvenilirliği; compute kapasitesi artırmak çözüm değil.

Sonuç:

Ek teknik notlar (kısa)

- SNS → Lambda doğrudan push modelidir; SNS'in Lambda için retry mekanizması sınırlıdır ve uzun süreli dayanıklı kuyruklama sağlamaz.
- En güvenilir desen: **SNS yayınıcıyı yayar → SQS kalıcı kuyruk alır → Lambda kuyruğu tüketir.**
- Ayrıca, SQS ve Lambda entegrasyonunda başarısız işlemler için DLQ ve yeniden deneme politikaları tanımlanabilir.

 **Özet**

Bu senaryoda **B (SNS → SQS aboneliği)** ve **E (Lambda'yı SQS'ten okur hale getirme)** birlikte kullanılmalıdır. Bu kombinasyon, ağ kesintilerinde bile mesajların kaybolmasını önerler ve otomatik, güvenilir işleme sağlar.

QUESTION 46

A company has an application that provides marketing services to stores. The services are based on previous purchases by store customers. The stores upload transaction data to the company through SFTP, and the data is processed and analyzed to generate new marketing offers. Some of the files can exceed 200 GB in size. Recently, the company discovered that some of the stores have uploaded files that contain personally identifiable information (PII) that should not have been included. The company wants administrators to be alerted if PII is shared again. The company also wants to automate remediation. What should a solutions architect do to meet these requirements with the LEAST development effort?

- A. Use an Amazon S3 bucket as a secure transfer point. Use Amazon Inspector to scan the objects in the bucket. If objects contain PII, trigger an S3 Lifecycle policy to remove the objects that contain PII.
- B. Use an Amazon S3 bucket as a secure transfer point. Use Amazon Macie to scan the objects in the bucket. If objects contain PII, use Amazon Simple Notification Service (Amazon SNS) to trigger a notification to the administrators to remove the objects that contain PII.
- C. Implement custom scanning algorithms in an AWS Lambda function. Trigger the function when objects are loaded into the bucket. If objects contain PII, use Amazon Simple Notification Service (Amazon SNS) to trigger a notification to the administrators to remove the objects that contain PII.
- D. Implement custom scanning algorithms in an AWS Lambda function. Trigger the function when objects are loaded into the bucket. If objects contain PII, use Amazon Simple Email Service (Amazon SES) to trigger a notification to the administrators and trigger an S3 Lifecycle policy to remove the objects that contain PII.

Soru:

Bir şirketin, mağazalara pazarlama hizmetleri sağlayan bir uygulaması vardır. Bu hizmetler, mağaza müşterilerinin önceki alışverişlerine dayanır. Mağazalar, işlem verilerini şirketin sistemine SFTP üzerinden yükler ve bu veriler işlenip analiz edilerek yeni pazarlama teklifleri oluştururlar. Dosyaların bazıları 200 GB boyutunu aşabilir.

Son zamanlarda şirket, bazı mağazaların yüklememesi gereken kişisel olarak tanımlanabilir bilgi (PII) içeren dosyalar yüklediğini keşfetmiştir. Şirket, PII tekrar paylaşıldığından yöneticilerin uyarılmasını istemektedir. Ayrıca şirket, iyileştirme işlemini otomatikleştirmek istemektedir.

En az geliştirme çabasıyla bu gereksinimleri karşılamak için bir solutions architect ne yapmalıdır?

- A.** Güvenli bir transfer noktası olarak bir Amazon S3 bucket kullanın. Bucket içindeki nesneleri taramak için Amazon Inspector kullanın. Nesneler PII içeriyorsa PII içeren nesneleri kaldırma için bir S3 Lifecycle policy tetikleyin.
- B.** Güvenli bir transfer noktası olarak bir Amazon S3 bucket kullanın. Bucket içindeki nesneleri taramak için Amazon Macie kullanın. Nesneler PII içeriyorsa, yöneticilere PII içeren nesnelerin kaldırılması için bildirim göndermek üzere Amazon Simple Notification Service (Amazon SNS) kullanın.
- C.** Bir AWS Lambda fonksiyonunda özel tarama algoritmaları uygulayın. Nesneler bucketa yüklenliğinde fonksiyonu tetikleyin. Nesneler PII içeriyorsa, yöneticilere PII içeren nesnelerin kaldırılması için bildirim göndermek üzere Amazon Simple Notification Service (Amazon SNS) kullanın.
- D.** Bir AWS Lambda fonksiyonunda özel tarama algoritmaları uygulayın. Nesneler bucketa yüklenliğinde fonksiyonu tetikleyin. Nesneler PII içeriyorsa, yöneticilere bildirim göndermek için Amazon Simple Email Service (Amazon SES) kullanın ve PII içeren nesneleri kaldırma için bir S3 Lifecycle policy tetikleyin.

Soru Analizi:

✓ Bağlam

- Mağazalar işlem dosyalarını **SFTP → S3** üzerine yükliyor.
- Bazı dosyalar **200 GB** → Lambda ile işlemek mümkün olmayabilir (timeout, boyut sınırlamaları).
- Bazı yüklemelerde **PII** bulunmuş → tespit edilmeli.
- Gereksinimler:
 1. **PII var mı? Oto tarama.**

2. Yöneticiler uyarı almalı.
3. Otomatik iyileştirme (remediation).
4. En az geliştirme çabası (minimum development).

Seçenek Analizi:

- B. S3'ü transfer noktası olarak kullan, S3'ü Amazon Macie ile tara, PII varsa SNS ile yöneticileri uyar. (DOĞRU CEVAP)**

Bu durumda AWS'in **hazır PII tarama servisi** olan **Amazon Macie** en uygun çözümüdür → sıfır kod ile PII taraması + uyarı.

Bu seçenek **en az geliştirme ile** hedefi sağlar:

Neden doğru?

- Amazon Macie, **yerleşik PII tespit servisidir** (Kredi kartı, SSN, kişisel ad, telefon, vs.).
- Özel algoritma yazmak gerekmeyez → *minimum development*.
- 200 GB gibi dev dosyaları tarayabilir.
- Otomatik tarama + SNS ile uyarı → manuel kod yok.
- Macie, S3 üzerinde native çalışır → ek mimari gerekmeyez.

Sağladıkları:

- **PII detection:** Macie
- **Alerting:** SNS
- **Minimum development:** Evet

EKSİK: Remediation otomatik silmeyi yapmaz → Ama soru “en az geliştirme” istediginden yönetici bildirimini yeterlidir.

X A. S3 + Amazon Inspector ile tarama (YANLIŞ)

Neden yanlış?

- Amazon Inspector **S3 için PII taramaz**.
- Inspector yalnızca:
 - EC2 için güvenlik açığı taraması
 - Container image taraması

- Lambda bağımlılık taraması yapar.

S3 üzerindeki PII tespiti mümkün değildir.

✗ C. Lambda ile özel tarama kodu yazmak + SNS (YANLIŞ)

Neden yanlış?

- Dosyalar **200 GB olabilir** → Lambda ile işlenemez:
 - 15 dk timeout limiti
 - /tmp disk limiti 10 GB
 - Büyük dosya akışı zor
- PII algılama için **özel regex/kod yazmak** gereklidir → çok fazla geliştirme çabası.
- Soruda “least development effort” isteniyor → bu seçenek tersine **en çok geliştirilen** çözümüdür.

✗ D. Lambda ile özel tarama kodu + SES + Lifecycle (YANLIŞ)

Neden yanlış?

- Yine Lambda ile özel tarama gerektiriyor → yüksek geliştirme.
- 200 GB dosyaları işlemekte Lambda yetersiz.
- SES kullanmak yalnızca e-posta gönderir → yine çözüm değil.
- Lifecycle policy "süreye göre silme" yapar, koşullu silme **yapamaz**.

Bu da geliştiriciden çok iş ister → doğru cevap olamaz.

🎯 SONUÇ

Seçenek Durum Açıklama

- | | | |
|---|---|--|
| A | ✗ | Inspector S3'ü PII için taramaz |
| B | ✓ | Macie ile PII tarama + SNS uyarı, minimum geliştirme |
| C | ✗ | Lambda ile 200 GB taranamaz, çok geliştirme gereklidir |
| D | ✗ | Lambda + özel kod, lifecycle yanlış |

A company needs guaranteed Amazon EC2 capacity in three specific Availability Zones in a specific AWS Region for an upcoming event that will last 1 week.

What should the company do to guarantee the EC2 capacity?

- A. Purchase Reserved Instances that specify the Region needed.
- B. Create an On-Demand Capacity Reservation that specifies the Region needed.
- C. Purchase Reserved Instances that specify the Region and three Availability Zones needed.
- D. Create an On-Demand Capacity Reservation that specifies the Region and three Availability Zones needed.

Soru:

Bir şirketin, yaklaşan ve 1 hafta sürecek bir etkinlik için belirli bir AWS Bölgesindeki (Region) üç belirli Kullanılabilirlik Alanında (Availability Zone) garantili Amazon EC2 kapasitesine ihtiyacı vardır.

Şirket, EC2 kapasitesini garanti altına almak için ne yapmalıdır?

- A. Gerekli Bölgeyi belirten Reserved Instance'lar satın alın.
- B. Gerekli Bölgeyi belirten bir On-Demand Capacity Reservation oluşturun.
- C. Gerekli Bölgeyi ve üç Availability Zone'u belirten Reserved Instance'lar satın alın.
- D. Gerekli Bölgeyi ve üç Availability Zone'u belirten bir On-Demand Capacity Reservation oluşturun.

Soru Analizi:

Şirketin ihtiyacı:

- Belirli bir AWS Region içinde 3 belirli Availability Zone (AZ) için
- EC2 kapasitesinin garanti edilmesi
- Sadece 1 haftalık süre (kısa süreli bir etkinlik)

Bu durumda çözümün:

1. AZ-bazında kapasite ayırbilmesi,
2. Kısa süreli olması,
3. Kesin kapasite garantisini verebilmesi gerekir.

Seçenek Analizi:

- D. Gerekli Region ve üç Availability Zone'u belirten On-Demand Capacity Reservation oluşturmak

Bu kriterleri sağlayan tek mekanizma:

Neden doğru?

- ODCR kapasiteyi **AZ bazında garanti eder**.
- Üç AZ için rezervasyon yaparak tam ihtiyaç karşılanır.
- 1 hafta sürecek etkinlik için ideal, çünkü:
 - RI gibi 1 veya 3 yıllık bağıllılık yok
 - Etkinlik bittiğinde rezervasyon kapatılır
- Tam olarak amaçlandığı kullanım senaryosu budur.

✗ A. Bölgeyi belirten Reserved Instance satın almak

Neden yanlış?

- Reserved Instance (RI):
 - **Kapasite garantisi vermez**
 - Sadece *fiyat indirimi* sağlar
- Region seviyesinde RI, hangi AZ'de kapasite olacağını **belirtmez** → kapasite rezervasyonu yapmaz.

✗ B. Region belirten On-Demand Capacity Reservation oluşturmak

Neden yanlış?

- Capacity Reservation **her zaman AZ-bazında yapılır**, Region seviyesinde değil.
- Region belirtmek kapasite garantisi sağlamaz.
- Soru üç spesifik **AZ** diyor → bu seçenek kapsamaz.

✗ C. Region ve üç AZ için Reserved Instance satın almak

Neden yanlış?

- RI kapasite garantisi **vermez**, AZ belirlesen bile.
- RI → sadece fiyat avantajı, kapasite rezervasyonu **yok**.
- Kapasiteyi garanti etmek için RI kullanılamaz.

⌚ SONUÇ

On-Demand Capacity Reservation,

3 spesifik AZ'de **garantili EC2 kapasitesi** sağlamak için tek uygun çözümdür.

QUESTION 48

A company's website uses an Amazon EC2 instance store for its catalog of items. The company wants to make sure that the catalog is highly available and that the catalog is stored in a durable location.

What should a solutions architect do to meet these requirements?

- A. Move the catalog to Amazon ElastiCache for Redis.
- B. Deploy a larger EC2 instance with a larger instance store.
- C. Move the catalog from the instance store to Amazon S3 Glacier Deep Archive.
- D. Move the catalog to an Amazon Elastic File System (Amazon EFS) file system.

Soru:

Bir şirketin web sitesi, ürün kataloğu için Amazon EC2 **instance store** kullanmaktadır. Şirket, katalogun yüksek erişilebilir olmasını ve kalıcı (dayanıklı) bir yerde saklanması istemektedir.

Bu gereksinimleri karşılamak için bir çözüm mimarı ne yapmalıdır?

- A. Katalogu Amazon ElastiCache for Redis'e taşıyın.
- B. Daha büyük bir instance store'a sahip daha büyük bir EC2 instance'ı dağıtın.
- C. Katalogu instance store'dan Amazon S3 Glacier Deep Archive'a taşıyın.
- D. Katalogu Amazon Elastic File System (Amazon EFS) dosya sistemine taşıyın.

Soru Analizi:

Şirket şu anda ürün kataloğunu **EC2 instance store** üzerinde tutuyor.

Instance store'un özellikleri:

- **Geçicidir** → Instance durursa *tüm veri silinir*.
- **Dayanıklı değildir** → Disk arızası veya durdurma durumunda veri kaybolur.
- **Yüksek erişilebilirlik sağlanamaz** → Instance AZ değiştirdiğinde veri taşınamaz.

Şirketin istediği iki şey:

1. **High availability (Yüksek erişilebilirlik)**
2. **Durable storage (Dayanıklı depolama)**

Bu ikisini instance store sağlayamaz → mutlaka harici bir storage servisine geçilmesi gereklidir.

Durum şu: katalog verisi sık erişilen bir veri, kalıcı olmalı, paylaşımı olmalı.

Seçenek Analizi:

A. Kataloğu Amazon ElastiCache for Redis'e taşıyın.

- Redis **cache** sistemidir.
- Bir *persistent* (kalıcı) depolama çözümü değildir.
- Reboot-esnasında veri kaybı riski vardır.
- High availability olabilir ama *durability* sağlamaz.

B. Daha büyük bir EC2 instance store kullanmak.

- Instance store ne kadar büyük olursa olsun **geçicidir**.
- Durability = **0**
- High availability = **yok**
- Sorunun gereksinimlerini hiç karşılamaz.

C. Amazon S3 Glacier Deep Archive'a taşımak.

- Glacier Deep Archive **uzun süreli arşivleme** içindir.
- Erişim süresi *saatler* sürer (12+ saat).
- Web sitesi katalogu için uygun değildir (yüksek gecikme).

Dayanıklı olsa bile **high availability + hızlı erişim** gereksinimini karşılamaz.

D. Amazon Elastic File System (Amazon EFS) dosya sistemine taşımak.

Neden?

- **Dayanıklıdır (multi-AZ replication)**
- **Yüksek erişilebilirlik sağlar**
- **Birden fazla EC2 instance tarafından eş zamanlı kullanılabilir**
- Veriye hızlı erişim sağlar
- Yönetilen, sunucusuz, ölçülebilir bir dosya sistemi

Tam olarak sorunun istediği çözüm: **durable + highly available**.

Sonuç:

Neden EFS (Seçenek D) Diğer Tüm Seçeneklere Göre Üstün?

Diğer seçeneklerin eksikleri:

Seçenek	Eksik Olan
A – Redis (ElastiCache)	Cache → kalıcı depolama değil, veri kaybı riski var, HA olasılık yok
B – Daha büyük instance store	Depolama hâlâ geçici, veri yine kaybolabilir
C – Glacier Deep Archive	Çok yavaş erişim, katalog gibi sık erişilen veriye uygun değil

EFS ise:

- Hem **high availability**
- Hem **high durability**
- Hem **hızlı erişim**
- Hem **EC2 ile native uyum**
- Hem de **çoklu AZ desteği**

sağlayan tek seçenektir.

QUESTION 49

A company stores call transcript les on a monthly basis. Users access the les randomly within 1 year of the call, but users access the les infrequently after 1 year. The company wants to optimize its solution by giving users the ability to query and retrieve les that are less than 1-year old as quickly as possible. A delay in retrieving older les is acceptable. Which solution will meet these requirements MOST cost-effectively?

- A. Store individual les with tags in Amazon S3 Glacier Instant Retrieval. Query the tags to retrieve the les from S3 Glacier Instant Retrieval.
- B. Store individual les in Amazon S3 Intelligent-Tiering. Use S3 Lifecycle policies to move the les to S3 Glacier Flexible Retrieval after 1 year. Query and retrieve the les that are in Amazon S3 by using Amazon Athena. Query and retrieve the les that are in S3 Glacier by using S3 Glacier Select.
- C. Store individual les with tags in Amazon S3 Standard storage. Store search metadata for each archive in Amazon S3 Standard storage. Use S3 Lifecycle policies to move the les to S3 Glacier Instant Retrieval after 1 year. Query and retrieve the les by searching for metadata from Amazon S3.

D. Store individual les in Amazon S3 Standard storage. Use S3 Lifecycle policies to move the les to S3 Glacier Deep Archive after 1 year. Store search metadata in Amazon RDS. Query the les from Amazon RDS. Retrieve the les from S3 Glacier Deep Archive.

Soru:

Bir şirket aylık olarak çağrı transkript dosyalarını depolamaktadır. Kullanıcılar, çağrı yapıldıktan sonraki 1 yıl içinde bu dosyalara rastgele erişir, ancak 1 yıldan sonra bu dosyalara nadiren erişir. Şirket, çözümünü optimize etmek istemektedir ve kullanıcıların **1 yıldan daha yeni dosyaları mümkün olduğunda hızlı bir şekilde sorgulayıp alabilmesini** istemektedir. **Daha eski dosyaların alınmasında gecikme kabul edilebilir.**

Bu gereksinimleri **en uygun maliyetle** hangi çözüm karşılar?

A. Bireysel dosyaları Amazon S3 Glacier Instant Retrieval içinde etiketlerle depolayın. Etiketleri sorgulayarak dosyaları S3 Glacier Instant Retrieval'dan alın.

B. Bireysel dosyaları Amazon S3 Intelligent-Tiering içinde depolayın. S3 Lifecycle kurallarını kullanarak dosyaları 1 yıl sonra S3 Glacier Flexible Retrieval'a taşıyın.

Amazon S3 içindeki dosyaları Amazon Athena ile sorgulayın ve alın.

S3 Glacier içindeki dosyaları S3 Glacier Select kullanarak sorgulayın ve alın.

C. Bireysel dosyaları etiketlerle Amazon S3 Standard'de depolayın. Her arşiv için arama meta verilerini Amazon S3 Standard'de depolayın. Dosyaları 1 yıl sonra S3 Glacier Instant Retrieval'a taşımak için S3 Lifecycle kullanın. Meta verileri S3'den arayarak dosyaları sorgulayın ve alın.

D. Bireysel dosyaları Amazon S3 Standard'de depolayın. S3 Lifecycle kurallarını kullanarak dosyaları 1 yıl sonra S3 Glacier Deep Archive'e taşıyın. Arama meta verilerini Amazon RDS'de depolayın. Dosyaları RDS'den sorgulayın. Dosyaları S3 Glacier Deep Archive'den alın.

Soru Analizi:

Aylık çağrı transkript dosyaları saklanıyor.

Kullanıcılar **ilk 1 yıl içinde** dosyalara rastgele (dolayısıyla nispeten hızlı erişim bekłentisi) erişiyor.

1 yıldan sonra erişim nadir; gecikme kabul edilebilir.

Amaç: **1 yıldan yeni dosyaları hızlı getirmek**, daha eski dosyaları ise daha ucuz (gecikmeli) arşive taşımak ve maliyeti optimize etmek.

Seçenek Analizi:

B. S3 Intelligent-Tiering; 1 yıl sonra S3 Glacier Flexible Retrieval; Athena + Glacier Select

- **Intelligent-Tiering:** AWS otomatik olarak nesneleri erişim sıklığına göre katmanlar (frequent / infrequent / archive tiers), yönetim yükü yoktur — "**least ops**" ve maliyet-optimum. İlk yıl içindeki rastgele erişimler için uygun.
- **1 yıl sonra** nesneleri **Glacier Flexible Retrieval** (eski adıyla Glacier) gibi daha ucuz arşiv sınıfına taşımak maliyeti düşürür; erişim gereksinimi nadir olduğu için restore gecikmesi kabul edilebilir.
- **Sorgulama:** Athena, S3'teki verileri sorgulamak için uygundur (ilk yıl S3'teki nesneleri hızla bulup getirebilirsiniz). Arşivlenmiş nesneler için AWS'nin arşiv sorgulama/geri getirme yolları (restore veya Glacier Select/Glacier retrieval) kullanılabilir; eski nesnelerde gecikme kabul edildiği için bu model uygun.
- Yönetim yükü düşüktür ve maliyet optimizasyonu sağlanır. Bu seçenekteki bileşenler gereksinimleri doğru karşılar.

A. S3 Glacier Instant Retrieval + tag sorgulama

Yanlış / Uygun değil

- Glacier Instant Retrieval, nadiren erişilen ancak anında erişim gereken veriler için uygundur; anında erişim sağlar ancak maliyeti S3 Standard / Intelligent-Tiering'e göre farklıdır.
- **Tüm dosyaları** doğrudan Glacier Instant Retrieval'a koymak, ilk 1 yılda sık/rasgele erişim gereksinimini karşılamayabilir/uygun maliyetli olmayabilir.
- Ayrıca "tag'leri sorgulayarak doğrudan arşivten alma" ifadesi pratik değil: S3 object tag'leriyle uygun bir indeksleme kurmazsanız arama/filtreleme için ek mekanizma gereklidir.
- Sonuç: erişim paterni (ilk 1 yıl sık) için uygun en maliyetli yaklaşım değil.

C. S3 Standard + metadata S3 + lifecycle → Glacier Instant Retrieval

Yanlış / Daha az uygun

- Tüm dosyaları S3 Standard'ta tutmak ilk yıl için hızlı erişim sağlar ama **maliyet yüksek** olabilir (Intelligent-Tiering bu maliyeti otomatik azaltır).
- "Arama metadata'sını S3'te sakla" yaklaşımı mümkün ama:
 - Metadata'yi aramak için ek düzen gereklidir (Athena ile düzenli indeks dosyaları hazırlamak vs.).
 - Yönetim ve geliştirme çabası artar.

- 1 yıl sonra **Glacier Instant Retrieval**'a taşımak Instant Retrieval kullanımıyla maliyet/erişim dengesi karmaşık olur; Intelligent-Tiering + lifecycle daha yönetilebilir ve genelde daha uygun.

D. S3 Standard → Glacier Deep Archive + metadata in RDS

Yanlış / En maliyetli ya da ağır

- Glacier Deep Archive en ucuz arşivdir ama **geri getirme süresi çok uzun (saatler)** ve genelde arşivleme amaçlıdır. İlk 1 yıl içindeki hızlı erişim ihtiyacını karşılamaz çünkü dosyalar ilk yıl S3 Standard'da tutuluyor ama:
 - Metadata'yi **RDS**'e koymak ekstra maliyet ve işletim yükü getirir.
 - Bu tasarım, gereksiz şekilde yönetim/işletim maliyeti arttırmır; daha az maliyet-etkin.

Sonuç:

Ek notlar (teknik nüanslar)

- **Athena**, S3'teki verileri (objeler veya indekslenmiş meta dosyaları) sorgulamada çok uygundur; ilk yıl S3'te kalan dosyalar için hızlı arama sağlar.
- **Arşivlenmiş** (Glacier Flexible Retrieval vb.) nesneler genellikle önceden restore edilmeden Athena ile sorgulanamaz; Glacier Select bazı durumlarda arşiv içi sorgu sunar ama tipik senaryoda restore gereklidir — fakat soru daha eski dosyalar için gecikme kabul ettiğini söylüyor, bu yüzden restore gecikmesi sorun olmaz.
- **Intelligent-Tiering**: erişim paterni öngörülemiyorsa yönetim yükü çok az, maliyet optimizasyonu otomatik sağlar — bu senaryoda önemli avantaj.

Özet

B en uygun ve maliyet-optimum çözümdür:

- İlk 1 yıl için Intelligent-Tiering ile hızlı erişim/otomatik maliyet optimizasyonu,
- 1 yıl sonra lifecycle ile Glacier Flexible Retrieval'a taşıma (düşük maliyet),
- Athena ile S3'teki (yeni) dosyaları hızlıca sorgulama; arşivlenen dosyalar için restore/Glacier Select kullanılabilir çünkü gecikme kabul ediliyor.

QUESTION 50

A company has a production workload that runs on 1,000 Amazon EC2 Linux instances. The workload is powered by third-party software. The company needs to patch the third-

party software on all EC2 instances as quickly as possible to remediate a critical security vulnerability. What should a solutions architect do to meet these requirements?

- A. Create an AWS Lambda function to apply the patch to all EC2 instances.
- B. Configure AWS Systems Manager Patch Manager to apply the patch to all EC2 instances.
- C. Schedule an AWS Systems Manager maintenance window to apply the patch to all EC2 instances.
- D. Use AWS Systems Manager Run Command to run a custom command that applies the patch to all EC2 instances.

Soru:

Bir şirket, üretim ortamında çalışan 1.000 adet Amazon EC2 Linux instance'ına sahiptir. İş yükü üçüncü taraf bir yazılım tarafından çalıştırılmaktadır. Şirket, kritik bir güvenlik açığını gidermek için üçüncü taraf yazılımını tüm EC2 instance'larında mümkün olan en hızlı şekilde yamalamalıdır.

Bir çözüm mimarı bu gereksinimleri karşılamak için ne yapmalıdır?

- A. Yamayı tüm EC2 instance'larına uygulamak için bir AWS Lambda fonksiyonu oluşturun.
- B. Yamayı tüm EC2 instance'larına uygulamak için AWS Systems Manager Patch Manager'ı yapılandırın.
- C. Yamayı tüm EC2 instance'larına uygulamak için bir AWS Systems Manager bakım penceresi (maintenance window) planlayın.
- D. Yamayı tüm EC2 instance'larına uygulayan özel bir komutu çalıştırmak için AWS Systems Manager Run Command'ı kullanın.

Soru Analizi:

Durum:

- Şirketin **1000 adet EC2 Linux instance'**ı var.
- Üçüncü taraf (third-party) bir yazılım çalışıyor → **AWS Patch Manager bu yazılımı otomatik yamayamaz.**
- Kritik güvenlik açığı var → **yama en hızlı şekilde uygulanmalı.**
- Amaç: Tüm 1000 instance'a **aynı komutu merkezi olarak gönderip hızlıca yamalamak.**

Anahtar nokta:

- Bu bir üçüncü taraf uygulama.
- AWS Systems Manager Patch Manager **YALNIZCA OS-level (Linux/Windows)** yamalarını yönetir, üçüncü taraf yazılımları yamalayamaz.
- Bu yüzden patch'i bir komut veya script ile bizim uygulamamız gereklidir.

Bu nedenle **en hızlı, anında, manuel tetiklenen toplu çalışma yöntemi: Run Command**.

Seçenek Analizi:

✓ D. Use AWS Systems Manager Run Command to run a custom command that applies the patch to all EC2 instances.

- Run Command:
 - Komutları **thousands of instances** üzerinde paralel çalıştırılabilir.
 - **Anlık** (immediate) çalıştırılır, zaman planlaması gerekmeyez.
 - Custom third-party software patch script'i çalıştırılabilir.
- Ayrıca en hızlı yöntemdir.
 - **Doğru cevap.**

✗ A. Create an AWS Lambda function to apply the patch to all EC2 instances.

- Lambda EC2 içine direkt bağlanıp patch uygulamaz.
- 1000 instance için SSH/SSM komutu göndermek Lambda ile yönetilmez.
- Lambda bu iş için **design pattern değil**.
 - **Elinir.**

✗ Configure AWS Systems Manager Patch Manager to apply the patch to all EC2 instances.

- Patch Manager **sadece OS patch'leri içindir**.
- Üçüncü taraf software patch'leri için kullanılmaz.
 - **Elinir.**

✗ C. Schedule an AWS Systems Manager maintenance window to apply the patch to all EC2 instances.

- Maintenance Window **belirli zamanlarda** çalışır.
- “**As quickly as possible**” şartına uymaz.
 - **Elinir.**

 **Sonuç:**

Use AWS Systems Manager Run Command to run a custom command that applies the patch to all EC2 instances.

Bu seçenek hem hız, hem ölçeklenebilirlik, hem de third-party yazılıma özel patch komutu çalıştırmayı sağlar.

QUESTION 51

A company is developing an application that provides order shipping statistics for retrieval by a REST API. The company wants to extract the shipping statistics, organize the data into an easy-to-read HTML format, and send the report to several email addresses at the same time every morning.

Which combination of steps should a solutions architect take to meet these requirements? (Choose two.)

- A. Configure the application to send the data to Amazon Kinesis Data Firehose.
- B. Use Amazon Simple Email Service (Amazon SES) to format the data and to send the report by email.
- C. Create an Amazon EventBridge (Amazon CloudWatch Events) scheduled event that invokes an AWS Glue job to query the application's API for the data.
- D. Create an Amazon EventBridge (Amazon CloudWatch Events) scheduled event that invokes an AWS Lambda function to query the application's API for the data.
- E. Store the application data in Amazon S3. Create an Amazon Simple Notification Service (Amazon SNS) topic as an S3 event destination to send the report by email.

Soru:

Bir şirket, sipariş gönderim istatistiklerinin bir REST API tarafından alınmasını sağlayan bir uygulama geliştiriyor. Şirket, gönderim istatistiklerini çıkarmak, veriyi okunması kolay bir HTML formatında düzenlemek ve her sabah aynı saatte birden fazla e-posta adresine raporu göndermek istiyor.

Bir çözüm mimarı bu gereksinimleri karşılamak için hangi adım kombinasyonlarını uygulamalıdır?

(İki seçenek seçin.)

- A. Uygulamayı veriyi Amazon Kinesis Data Firehose'a gönderecek şekilde yapılandırın.
- B. Veriyi formatlamak ve raporu e-posta ile göndermek için Amazon Simple Email Service (Amazon SES) kullanın.
- C. Veriyi uygulamanın API'sinden almak için bir AWS Glue işini tetikleyecek zamanlanmış bir Amazon EventBridge (Amazon CloudWatch Events) olayı oluşturun.
- D. Veriyi uygulamanın API'sinden almak için bir AWS Lambda fonksiyonunu tetikleyecek

zamanlanmış bir Amazon EventBridge (Amazon CloudWatch Events) olayı oluşturun.
E. Uygulama verisini Amazon S3'te saklayın. Raporu e-posta ile göndermek için S3 olay hedefi olarak bir Amazon Simple Notification Service (Amazon SNS) konusu oluşturun.

Soru Analizi:

Şirket şunları yapmak istiyor:

1. Her sabah aynı saatte **uygulamanın API'sini çağrıp** “shipping statistics” verisini almak.
2. Bu veriyi **HTML formatında bir rapora dönüştürmek**.
3. Raporu **birden fazla e-posta adresine göndermek**.

Bu işlem bir **batch / scheduled job** formatında çalışıyor.

Dolayısıyla iki temel bileşen gerekiyor:

Zamanlanmış bir tetikleyici (EventBridge Scheduler)

→ API'yi her sabah çağıracak.

Veriyi işleyip HTML rapor oluşturacak ve e-posta gönderecek mekanizma

→ Genellikle **Lambda + SES** en basit ve doğru kombinasyondur.

Seçenek Analizi:

B. Use Amazon SES to format the data and to send the report by email.

- SES email gönderir, veriyi işlemez veya **HTML** oluşturmaz.
- “Formatlama” işlemini doğrudan SES yapamaz; bu işi Lambda yapmalıdır.
- Bu madde kısmen doğru: **Raporu göndermek için SES kullanılabilir**.

Doğru cevabın bir parçası (email gönderimi için).

D. Create an EventBridge scheduled event that invokes a Lambda function to query the application's API.

- Her sabah belirli saatte Lambda tetiklenebilir.
- Lambda API'den veri alıp HTML formatına dönüştürebilir.
- Aynı Lambda SES ile e-posta da gönderebilir.

En uygun seçenek.

A. Configure the application to send the data to Amazon Kinesis Data Firehose.

- Firehose sürekli veri akışı içindir.

- Burada “her sabah rapor oluşturma” var → streaming değil.
- Ayrıca HTML rapor oluşturma ihtiyacını karşılamaz.

➡ Uygun değil.

✗ C. Create an EventBridge scheduled event that invokes an AWS Glue job to query the API.

- Glue ETL sağlar; API çağrılmak için gereksiz derecede ağırdır.
- API polling + HTML raporlama için Glue fazla maliyetli ve uygunsuz.

➡ Uygun değil.

✗ E. Store the data in S3 and trigger SNS to email report when new objects are added.

- Burada raporlama S3 yükleme olayıyla tetikleniyor, ama soru:
 - “Her sabah API’den veriyi çek ve HTML rapor gönder.” diyor.
- Ayrıca SNS **HTML email** için uygun değildir; SES gerekir.

➡ Uygun değil.

⌚ Sonuç:

✓ D → EventBridge her sabah Lambda'yı tetikler, Lambda API'den veriyi çeker ve HTML raporu oluşturur.

✓ B → Lambda raporu HTML olarak SES ile e-posta adreslerine gönderir.

📌 Kısa Özet

Gereksinim

Zamanlanmış çalışma

API'den veri çekme + HTML oluşturma

E-posta gönderme

En iyi AWS hizmeti

EventBridge Scheduler

Lambda

SES

QUESTION 52

A company wants to migrate its on-premises application to AWS. The application produces output files that vary in size from tens of gigabytes to hundreds of terabytes. The application data must be stored in a standard file system structure. The company wants

a solution that scales automatically, is highly available, and requires minimum operational overhead.

Which solution will meet these requirements?

- A. Migrate the application to run as containers on Amazon Elastic Container Service (Amazon ECS). Use Amazon S3 for storage.
- B. Migrate the application to run as containers on Amazon Elastic Kubernetes Service (Amazon EKS). Use Amazon Elastic Block Store (Amazon EBS) for storage.
- C. Migrate the application to Amazon EC2 instances in a Multi-AZ Auto Scaling group. Use Amazon Elastic File System (Amazon EFS) for storage.
- D. Migrate the application to Amazon EC2 instances in a Multi-AZ Auto Scaling group. Use Amazon Elastic Block Store (Amazon EBS) for storage.

Soru:

Bir şirket, şirket içindeki (on-premises) uygulamasını AWS'ye taşımak istiyor. Uygulama, boyutları onlarca gigabayttan yüzlerce terabayta kadar değişen çıktı dosyaları üretiyor. Uygulama verileri standart bir dosya sistemi yapısında depolanmalıdır. Şirket, otomatik olarak ölçülen, yüksek oranda kullanılabilir (highly available) ve minimum operasyonel iş yükü gerektiren bir çözüm istiyor.

Bu gereksinimleri hangi çözüm karşılar?

- A. Uygulamayı Amazon Elastic Container Service (Amazon ECS) üzerinde container olarak çalıştırın. Depolama için Amazon S3 kullanın.
- B. Uygulamayı Amazon Elastic Kubernetes Service (Amazon EKS) üzerinde container olarak çalıştırın. Depolama için Amazon Elastic Block Store (Amazon EBS) kullanın.
- C. Uygulamayı Çoklu AZ (Multi-AZ) Auto Scaling grubu içinde Amazon EC2 instance'larına taşıyın. Depolama için Amazon Elastic File System (Amazon EFS) kullanın.
- D. Uygulamayı Çoklu AZ (Multi-AZ) Auto Scaling grubu içinde Amazon EC2 instance'larına taşıyın. Depolama için Amazon Elastic Block Store (Amazon EBS) kullanın.

Soru Analizi:

Uygulamanın gereksinimleri:

✓ Çok büyük dosyalar üretiyor

- GB → yüzlerce TB.

✓ Standart bir dosya sistemi (POSIX) gerekiyor

- Yani uygulama **S3 gibi object storage** değil, **NFS tarzı bir file system** istiyor.

✓ Otomatik ölçeklenme (auto-scaling)

✓ Yüksek erişilebilirlik (high availability)

✓ Minimum operasyonel yük

Bu gereksinimler birlikte değerlendirildiğinde AWS'de **tek bir servis tam uyuyor**:

Seçenek Analizi:

✓ C. EC2 Auto Scaling + EFS

 **Amazon EFS (Elastic File System)**

- Tamamen yönetilen
- Multi-AZ, yüksek erişilebilir
- Otomatik olarak petabaytlara kadar ölçeklenir
- Standart bir dosya sistemi (POSIX NFS)
- Birden çok EC2 instance tarafından aynı anda mount edilebilir
- Operasyonel yük minimum

Dolayısıyla doğru cevabın depolama kısmı kesinlikle **EFS** olmalı.

- EFS:
 - Multi-AZ
 - Otomatik ölçeklenir (petabyte seviyesine kadar)
 - Tam bir POSIX uyumlu dosya sistemi
 - Minimum yönetim gerektirir
 - EC2 Auto Scaling → uygulama autoscale'lenebilir
-  **Gereksinimleri mükemmel karşılar. ✓**

✗ A. ECS + S3

- S3 bir **dosya sistemi değil**, object storage'dır.
 - POSIX, NFS, file system tree yapısı gereksinimini karşılamaz.
-  **Elinir.**

✗ B. EKS + EBS

- EBS:
 - Bir **block storage**'dır.

- Tek bir availability zone'a bağlıdır → Multi-AZ **değil**.
- Tek instance'a bağlanabilir (multi-attach kısıtlı ve küçük dosya sistemleri için).
- Petabyte ölçüğinde otomatik ölçeklenme sağlamaz.
→ **Elinir.**

D. EC2 Auto Scaling + EBS

- EBS:
 - Multi-AZ değildir
 - Otomatik ölçeklenmez
 - Yüzlerce TB veriyi desteklemek zordur
 - File system'ı birden çok instance'ın paylaşması **imkânsız**
→ **Elinir.**

Sonuç:

**Migrate the application to Amazon EC2 instances in a Multi-AZ Auto Scaling group.
Use Amazon Elastic File System (Amazon EFS) for storage.**

Bu çözüm:

- POSIX dosya sistemi
- Büyük dosya desteği
- Multi-AZ HA
- Otomatik ölçekleme
- Sıfıra yakın işletme yükü

gibi tüm gereksinimleri karşılar.

QUESTION 53

A company needs to store its accounting records in Amazon S3. The records must be immediately accessible for 1 year and then must be archived for an additional 9 years. No one at the company, including administrative users and root users, can be able to delete the records during the entire 10-year period. The records must be stored with maximum resiliency.

Which solution will meet these requirements?

- A. Store the records in S3 Glacier for the entire 10-year period. Use an access control policy to deny deletion of the records for a period of 10 years.
- B. Store the records by using S3 Intelligent-Tiering. Use an IAM policy to deny deletion of the records. After 10 years, change the IAM policy to allow deletion.
- C. Use an S3 Lifecycle policy to transition the records from S3 Standard to S3 Glacier Deep Archive after 1 year. Use S3 Object Lock in compliance mode for a period of 10 years.
- D. Use an S3 Lifecycle policy to transition the records from S3 Standard to S3 One Zone-Infrequent Access (S3 One Zone-IA) after 1 year. Use S3 Object Lock in governance mode for a period of 10 years.

Soru:

Bir şirket, muhasebe kayıtlarını Amazon S3'te depolamak zorundadır. Kayıtlara **1 yıl boyunca anında erişilebilir** olması gereklidir ve daha sonra **ek 9 yıl boyunca arşivlenmelidir**.

Şirket içindeki hiç kimse — yönetici kullanıcılar ve root kullanıcılar da dahil — **10 yıl boyunca bu kayıtları silememelidir**.

Kayıtlar **maksimum dayanıklılık (resiliency)** ile saklanmalıdır.

Bu gereksinimleri hangi çözüm karşıları?

- A. Kayıtları 10 yıl boyunca S3 Glacier'da saklayın. Kayıtların 10 yıl boyunca silinmesini engellemek için bir erişim kontrol politikası kullanın.
- B. Kayıtları S3 Intelligent-Tiering kullanarak saklayın. Silmeyi engellemek için bir IAM politikası kullanın. 10 yıl sonra IAM politikasını değiştirerek silmeye izin verin.
- C. Bir S3 Lifecycle policy kullanarak kayıtları 1 yıl sonra S3 Standard'dan S3 Glacier Deep Archive'a geçirin. 10 yıl boyunca S3 Object Lock'u **compliance mode** ile kullanın.
- D. Bir S3 Lifecycle policy kullanarak kayıtları 1 yıl sonra S3 Standard'dan S3 One Zone-Infrequent Access'e (S3 One Zone-IA) geçirin. 10 yıl boyunca S3 Object Lock'u **governance mode** ile kullanın.

Soru Analizi:

Şirketin gereksinimleri:

✓ **1 yıl boyunca anında erişilebilir olmalı**

✓ **Sonraki 9 yıl arşivlenmeli**

✓ **10 yıl boyunca kimse silememeli**

→ IAM politikası yetmez. Root bile silemeyecek

✓ **Maksimum dayanıklılık isteniyor**

- One Zone türleri elenir.
- Çoklu AZ dayanıklılığı

Seçenek Analizi:

✓ C. S3 Standard → 1 yıl sonra → S3 Glacier Deep Archive

S3 Object Lock Compliance Mode = 10 yıl

- 1 yıl sıcak erişim ihtiyacı: **S3 Standard** uygun.
- 9 yıl arşiv ihtiyacı: **Deep Archive** en düşük maliyetli.
- S3 Object Lock **Compliance Mode** = hiç kimse (root dahil) silemez.
- Çoklu AZ dayanıklılığı = maximum resiliency.
→ Tüm gereksinimleri karşılayan tek seçenek.

✗ A. Kayıtları 10 yıl boyunca S3 Glacier'da saklayın. Erişim kontrolü ile silmeyi engelleyin.

- Glacier “immediately accessible” değildir (erişim dakikalar saatler).
- IAM/Access policy ile silme engellemesi YETERSİZ → root override eder.
→ Yanlış.

✗ S3 Intelligent-Tiering + IAM ile silmeyi engelleme

- IAM policy root kullanıcısını engelleyemez → root override eder.
- WORM (Write Once Read Many) gerekir → Object Lock compliance gerekir.
→ Yanlış.

✗ D. S3 One Zone-IA + Object Lock Governance Mode

- One Zone-IA **multi-AZ değildir** → maximum resiliency gereksinimini karşılamaz.
- Governance mode, yetkili kişiler tarafından override edilebilir → root silebilir.
→ Yanlış.

⌚ Sonuç

S3 Lifecycle → S3 Standard → S3 Glacier Deep Archive

- S3 Object Lock (Compliance Mode, 10 yıl)

Bu çözüm:

- ✓ Sıcak erişim
- ✓ Çok yıllık arşiv

- ✓ WORM uyumluluğu
- ✓ Root bile silemez
- ✓ Maksimum dayanıklılık

gibi tüm şartları karşılayan tek mimaridir.

QUESTION 54

A company runs multiple Windows workloads on AWS. The company's employees use Windows file shares that are hosted on two Amazon EC2 instances. The file shares synchronize data between themselves and maintain duplicate copies. The company wants a highly available and durable storage solution that preserves how users currently access the files. What should a solutions architect do to meet these requirements?

- A. Migrate all the data to Amazon S3. Set up IAM authentication for users to access files.
- B. Set up an Amazon S3 File Gateway. Mount the S3 File Gateway on the existing EC2 instances.
- C. Extend the file share environment to Amazon FSx for Windows File Server with a Multi-AZ configuration. Migrate all the data to FSx for Windows File Server.
- D. Extend the file share environment to Amazon Elastic File System (Amazon EFS) with a Multi-AZ configuration. Migrate all the data to Amazon EFS.

Soru:

Bir şirket AWS üzerinde birçok Windows iş yükü çalıştırıyor. Şirketin çalışanları, iki Amazon EC2 instance'ında barındırılan Windows dosya paylaşımlarını kullanıyor. Dosya paylaşımı birbirleriyle veri senkronize ediyor ve yedek kopyalar tutuyor. Şirket, kullanıcıların şu anda dosyalara erişim şeklini koruyan, yüksek kullanılabilirlikte ve dayanıklı bir depolama çözümü istiyor.

Bu gereksinimleri karşılamak için bir çözüm mimarı ne yapmalıdır?

- A. Tüm veriyi Amazon S3'e taşıyın. Kullanıcıların dosyalara erişebilmesi için IAM kimlik doğrulaması kurun.
- B. Bir Amazon S3 File Gateway kurun. S3 File Gateway'i mevcut EC2 instance'larına bağlayın (mount edin).
- C. Dosya paylaşım ortamını Multi-AZ yapılandırmasıyla Amazon FSx for Windows File Server'a genişletin. Tüm veriyi FSx for Windows File Server'a taşıyın.
- D. Dosya paylaşım ortamını Multi-AZ yapılandırmasıyla Amazon Elastic File System (Amazon EFS)'a genişletin. Tüm veriyi Amazon EFS'ye taşıyın.

Soru Analizi:

Şirketin mevcut yapısı:

- Çalışanlar **Windows file share** kullanıyor.
- Paylaşımlar **iki EC2 Windows instance** üzerinde.
- Aralarında **senkronizasyon ve yedekli kopyalama** yapılıyor.
- Şirket:
 - **Yüksek erişilebilirlik (High Availability)**
 - **Dayanıklılık**
 - **Kullanıcıların dosyalara erişme yönteminin aynen korunması (SMB protokolü, Windows ACL)**
istiyor.

Burada kritik nokta:

Kullanıcılar Windows SMB dosya paylaşımı kullanıyor → **Windows-native bir file system** gerekiyor.

Seçenek Analizi:

 **C. FSx for Windows File Server (Multi-AZ), tüm veriyi FSx'e taşı**

AWS'de Windows tabanlı, SMB destekleyen ve Multi-AZ HA sağlayan tek servis:

 **Amazon FSx for Windows File Server (Multi-AZ edition)**

Bu, Windows file share ortamı için fully managed, HA, durable ve domain-joined bir çözüm sağlar.

- **Windows SMB** tamamen desteklenir.
- **Multi-AZ High Availability** sağlar.
- **Yüksek dayanıklılık.**
- **Windows ACL, AD integration, DFS, locking** gibi tam Windows dosya özellikleri.
- Minimum operasyonel yük (managed service).
- Kullanıcı erişimi **hic deşmez** (SMB share olarak görürler).

 **Tüm gereksinimleri karşılayan tek doğru seçenek. ✓**

 **A. Veriyi S3'e taşıyıp IAM ile erişim sağlamak**

- S3 **SMB** değil → Windows file share davranışını koruyamaz.
- Windows ACL'ler çalışmaz.

- “Aynı erişim yöntemi” şartını karşılamaz.

S3 File Gateway kullanıp EC2 üzerinde mount etmek

- S3 File Gateway SMB sağlar ama:
 - Windows özelliklerinin tamamını (NTFS ACL, AD, DFS, locking) tam vermez.
 - “Maksimum dayanıklılık + HA + Windows-native experience” gereksinimini tam karşılamaz.
 - Ayrıca hâlâ S3 backend kullanır → Windows file server davranışını bire bir aynısı değildir.

D. Amazon EFS kullanmak

- **EFS Linux NFS** tabanlıdır.
- Windows SMB ile uyumlu değildir.
- Windows ACL ve Windows davranışını sağlayamaz.

Sonuç:

“Amazon FSx for Windows File Server (Multi-AZ) kullanmak ve tüm veriyi FSx’e taşımak.”

Bu çözüm:

- Windows file share ile %100 uyumlu
- Multi-AZ high availability
- Fully managed
- Dayanıklı
- Kullanıcılar hiçbir değişiklik hissetmez

gibi tüm şartları karşılayan tek çözümdür.

QUESTION 55

A solutions architect is developing a VPC architecture that includes multiple subnets. The architecture will host applications that use Amazon EC2 instances and Amazon RDS DB instances. The architecture consists of six subnets in two Availability Zones. Each Availability Zone includes a public subnet, a private subnet, and a dedicated subnet for databases. Only EC2 instances that run in the private subnets can have access to the RDS databases.

Which solution will meet these requirements?

- A. Create a new route table that excludes the route to the public subnets' CIDR blocks. Associate the route table with the database subnets.
- B. Create a security group that denies inbound traffic from the security group that is assigned to instances in the public subnets. Attach the security group to the DB instances.
- C. Create a security group that allows inbound traffic from the security group that is assigned to instances in the private subnets. Attach the security group to the DB instances.
- D. Create a new peering connection between the public subnets and the private subnets. Create a different peering connection between the private subnets and the database subnets.

Soru:

Bir solutions architect, birden fazla alt ağ (subnet) içeren bir VPC mimarisi geliştirmeye çalışıyor. Bu mimarı, Amazon EC2 instance'larını ve Amazon RDS veritabanı instance'larını kullanan uygulamalara ev sahipliği yapacak. Mimari, iki Availability Zone'da altı subnet'ten oluşuyor. Her Availability Zone'da bir public subnet, bir private subnet ve veritabanları için özel bir subnet bulunuyor. **Yalnızca private subnet'lerde çalışan EC2 instance'ları RDS veritabanlarına erişebilmelidir.**

Bu gereksinimleri hangi çözüm karşılar?

- A. Public subnet'lerin CIDR bloklarına giden route'u içermeyen yeni bir route table oluşturun. Bu route table'ı database subnet'lerine ilişştirin.
- B. Public subnet'lerdeki instance'lara atanmış security group'tan gelen inbound trafiği reddeden bir security group oluşturun. Bu security group'u DB instance'larına bağlayın.
- C. Private subnet'lerdeki instance'lara atanmış security group'tan gelen inbound trafiğe izin veren bir security group oluşturun. Bu security group'u DB instance'larına bağlayın.
- D. Public subnet'ler ile private subnet'ler arasında yeni bir VPC peering bağlantısı oluşturun. Private subnet'ler ile database subnet'leri arasında farklı bir peering bağlantısı oluşturun.

Soru Analizi:

Senaryo:

- Bir VPC var.
- 2 Availability Zone → Her AZ'de 3 subnet:
 - 1 public subnet

- 1 private subnet
 - 1 database subnet (DB subnet)
- Amazon EC2 (uygulama katmanı) → private subnet'lerde çalışıyor.
- Amazon RDS → database subnet'lerinde çalışıyor.
- **Amaç:**
Sadece private subnet'lerdeki EC2 instance'larının RDS instance'larına erişebilmesini sağlamak.
 Public subnet'lerdeki EC2'ler veya başka kaynaklar DB'ye erişememeli.

Bu, AWS mimarisinde **security group bazlı izin kontrolü** ile yapılır.

Seçenek Analizi:

C Seçeneği

“Private subnet SG’sinden gelen trafiğe izin veren bir security group oluşturun.”

Doğru — En temiz, en güvenli, AWS best practice

Yapılacak doğru işlem:

1. Private EC2'lere ait security group → **sg-app**
2. RDS'e ait security group → **sg-db**
3. sg-db inbound rule:
 - Type: DB portu (ör. 3306)
 - Source: **sg-app (security group ID)**

Bu sayede:

- Sadece private subnet EC2'leri RDS'e bağlanabilir.
- Public subnet EC2'leri bağlanamaz.
- Ek firewall/NACL/route gerektirmez.

A Seçeneği

“Public subnet’lerin CIDR bloklarına giden route'u içermeyen bir route table oluşturun.”

Route table trafiği engellemez.

VPC içindeki subnet'ler arasında trafik default lokal route ile gelir → route table bunu engelleyemez.

Neden çalışmaz?

- Route table'lar sadece *hangi gateway*'e yönlendireceğini belirler.
- “Bir subnet DB'ye erişemesin” türü kurallar route table ile konamaz.

B Seçeneği

“Public subnet SG'sinden gelen trafiği reddeden bir security group oluşturun.”

Security group'lar:

- **DENY (reddetme)** kuralı içermez → sadece “ALLOW” vardır.
- “Şu SG'den gelen trafik reddedilsin” mantığı SG'lerde yoktur.

Dolayısıyla bu seçenek **mümkün değil**.

D Seçeneği

“Subnetler arasında peering bağlantısı kurun.”

- VPC peering **VPC–VPC arasındaki** bir bağlantıdır.
- Aynı VPC içindeki subnetler arasında peering olmaz.
- Zaten aynı VPC'de tüm subnetler otomatik olarak birbirine bağlıdır.

Bu seçenek **teknik olarak anlamsızdır**.

SONUÇ

“Private subnet EC2'lerinin security group'unu kaynak olarak kabul eden bir DB security group'u oluşturup bunu RDS'e uygulamak.”

AWS'in önerdiği, doğru, güvenli ve tek geçerli çözümüdür.

- En güvenli ✓
- En az yönetim gerektirir ✓
- AWS mimari prensiplerine tamamen uygundur ✓
- Modern AWS çözümlerinin çalıştığı yöntem budur ✓

QUESTION 56

A company has registered its domain name with Amazon Route 53. The company uses Amazon API Gateway in the ca-central-1 Region as a public interface for its backend microservice APIs. Third-party services consume the APIs securely. The company wants to design its API Gateway URL with the company's domain name and corresponding certificate so that the third-party services can use HTTPS.

Which solution will meet these requirements?

- A. Create stage variables in API Gateway with Name="Endpoint-URL" and Value="Company Domain Name" to overwrite the default URL. Import the public certificate associated with the company's domain name into AWS Certificate Manager (ACM).
- B. Create Route 53 DNS records with the company's domain name. Point the alias record to the Regional API Gateway stage endpoint. Import the public certificate associated with the company's domain name into AWS Certificate Manager (ACM) in the us-east-1 Region.
- C. Create a Regional API Gateway endpoint. Associate the API Gateway endpoint with the company's domain name. Import the public certificate associated with the company's domain name into AWS Certificate Manager (ACM) in the same Region. Attach the certificate to the API Gateway endpoint. Configure Route 53 to route traffic to the API Gateway endpoint.
- D. Create a Regional API Gateway endpoint. Associate the API Gateway endpoint with the company's domain name. Import the public certificate associated with the company's domain name into AWS Certificate Manager (ACM) in the us-east-1 Region. Attach the certificate to the API Gateway APIs. Create Route 53 DNS records with the company's domain name. Point an A record to the company's domain name.

Soru:

Bir şirket, etki alanı (domain) adını Amazon Route 53 ile kaydetmiştir. Şirket, arka uç mikro servis API'leri için **genel (public) bir arayüz olarak** ca-central-1 Bölgesi'nde Amazon API Gateway kullanmaktadır. Üçüncü taraf hizmetler bu API'leri **güvenli bir şekilde** tüketmektedir. Şirket, API Gateway URL'sinin şirketin **kendi domain adı ve buna karşılık gelen sertifika** ile çalışmasını istemektedir. Böylece üçüncü taraf hizmetler API'lere **HTTPS üzerinden** erişebilecektir.

Hangi çözüm bu gereksinimleri karşılar?

- A.** API Gateway içinde Name="Endpoint-URL", Value="Company Domain Name" olan stage variable'lar oluşturun ve varsayılan URL'yi bunlarla değiştirin. Şirketin domain adıyla ilişkili public sertifikayı AWS Certificate Manager'a (ACM) içe aktarın.
- B.** Route 53'te şirketin domain adıyla DNS kayıtları oluşturun. Alias kaydını *Regional API Gateway stage endpoint*'ine yönlendirin. Şirketin domain adıyla ilgili public sertifikayı *us-east-1* Bölgesi'ndeki ACM'e içe aktarın.
- C.** Regional API Gateway endpoint'i oluşturun. API Gateway endpoint'ini şirketin domain adıyla ilişkilendirin. Şirketin domain adıyla ilgili public sertifikayı **API Gateway ile aynı bölgede** ACM'e içe aktarın. Sertifikayı API Gateway endpoint'ine ekleyin. Route 53'ü, trafiği API Gateway endpoint'ine yönlendirecek şekilde yapılandırın.

D. Regional API Gateway endpoint’ı oluşturun. API Gateway endpoint’ini şirketin domain adıyla ilişkilendirin. Şirketin domain adıyla ilgili public sertifikayı *us-east-1* Bölgesi’ndeki ACM içine aktarın. Sertifikayı API Gateway API’lerine ekleyin. Route 53’te şirketin domain adına DNS kayıtları oluşturun. Bir A kaydını şirketin domain adına yönlendirin.

Soru Analizi:

Şirketin gereksinimleri:

- ✓ Domain Route 53’te
- ✓ API Gateway Regional endpoint (ca-central-1 bölgesi)
- ✓ Üçüncü taraflar API’ya HTTPS üzerinden erişecek
- ✓ API Gateway URL’si şirketin kendi domain adı ile çalışmalı
- ✓ API Gateway için ACM sertifikası gerekli
- ✓ Sertifika API Gateway hangi bölgede ise orada olmalı

Bu durumda çözüm şu adımlardan oluşur:

1. API Gateway’i **Regional endpoint** olarak yapılandırmak
2. Kendi domain adını API Gateway’e **Custom Domain Name** olarak eklemek
3. Bu domain için **ACM sertifikasını API Gateway ile aynı region'a yüklemek** → **ca-central-1**
4. Route 53’te bir alias record oluşturup API Gateway custom domain’e yönlendirmek

Seçenek Analizi:

C Seçeneği

“Regional endpoint
Aynı bölgede ACM sertifikası
Custom Domain Name yapılandırması
Route 53 yönlendirmesi”

Bu seçenekte:

- API Gateway **Regional endpoint** → ✓
- Domain API Gateway’e bağlanıyor → ✓
- ACM sertifikası **aynı bölgeye** (ca-central-1) yükleniyor → ✓
- Sertifika API Gateway endpoint’ine atanıyor → ✓

- Route 53 alias → API Gateway custom domain → ✓

Bu, AWS dokümantasyonunun önerdiği **eksiksiz doğru çözüm**dur.

A Seçeneği – Yanlış

“Stage variables ile endpoint URL değiştirme”

- Stage variables **URL değiştirmez**, sadece uygulama parametreleri için kullanılır.
- API Gateway domain’ini değiştirmek için **Custom Domain Name** kullanılmalıdır.
- Ayrıca sertifikanın hangi bölgede olması gerektiği belirtilmemiş.

B Seçeneği – Yanlış

“Alias record → Regional stage endpoint

ACM sertifikası us-east-1’da”

- **Regional API Gateway** kullanıldığındaysertifika **API Gateway’ın olduğu region’da** olmalıdır → burada **ca-central-1**
- Sertifikanın us-east-1’de olması **yalnızca CloudFront dağıtımını ile** kullanıldığındaysuperlidir.
- CloudFront yok → us-east-1 yanlış bölge.

D Seçeneği – Yanlış

“Regional API + ACM sertifikası us-east-1’da”

- Yine **sertifika yanlış bölgede**.
- API Gateway için CloudFront kullanılmadığı sürece ACM’in **aynı bölgede** olması zorunludur.
- Ayrıca A kaydını domain adına yönlendirmek gibi anlamsız ve eksik bir ifade vardır.

SONUÇ

Çünkü tek doğru çözüm:

- Custom domain
- Regional API Gateway
- Aynı region ACM sertifikası
- Route 53 alias

ile birlikte çalışır.

QUESTION 57

A company is running a popular social media website. The website gives users the ability to upload images to share with other users. The company wants to make sure that the images do not contain inappropriate content. The company needs a solution that minimizes development effort.

What should a solutions architect do to meet these requirements?

- A. Use Amazon Comprehend to detect inappropriate content. Use human review for low-confidence predictions.
- B. Use Amazon Rekognition to detect inappropriate content. Use human review for low-confidence predictions.
- C. Use Amazon SageMaker to detect inappropriate content. Use ground truth to label low-confidence predictions.
- D. Use AWS Fargate to deploy a custom machine learning model to detect inappropriate content. Use ground truth to label low-confidence predictions.

Soru:

Bir şirket popüler bir sosyal medya sitesi işletmektedir. Web sitesi, kullanıcılarla diğer kullanıcılarla paylaşmak üzere görüntü (resim) yükleme imkanı sunmaktadır. Şirket, yüklenen görüntülerin uygunsuz içerik içermemişinden emin olmak istemektedir. Şirket, geliştirme çabasını en aza indiren bir çözüm aramaktadır.

Bu gereksinimleri karşılamak için bir çözüm mimarı ne yapmalıdır?

- A. Amazon Comprehend kullanarak uygunsuz içeriği tespit edin. Düşük güvenli tahminler için insan değerlendirmesi kullanın.
- B. Amazon Rekognition kullanarak uygunsuz içeriği tespit edin. Düşük güvenli tahminler için insan değerlendirmesi kullanın.
- C. Amazon SageMaker kullanarak uygunsuz içeriği tespit edin. Düşük güvenli tahminler için Ground Truth kullanarak düşük güvenli tahminleri etiketleyin.
- D. Uygunsuz içeriği tespit etmek için özel bir makine öğrenimi modeli dağıtmak amacıyla AWS Fargate kullanın. Düşük güvenli tahminleri etiketlemek için Ground Truth kullanın.

Soru Analizi:

Şirket, **kullanıcılar tarafından yüklenen görüntüleri uygunsuz içerik açısından otomatik olarak denetlemek** istiyor.

Ana kriter: **Geliştirme çabasını en aza indirmek.**

Bu tür medya içeriği (resim) analizi için AWS hizmetleri içinde en uygun olan servis **Amazon Rekognition**'dır. Rekognition, hazır bir "**inappropriate / unsafe content moderation (icerik denetleme)**" API'si sunar. Yani ek model eğitimi veya altyapı gerektirmez.

İkinci istek:

Düşük güvenli tahminlerde insan değerlendirmesi.

Bu da **AWS Human Review / Amazon Augmented AI (A2I)** ile Rekognition'ın doğal entegrasyonuyla kolayca yapılır.

Bu nedenle minimum geliştirme gerektiren çözüm budur.

Seçenek Analizi:

- B. Amazon Rekognition kullanarak uygunsuz içeriği tespit edin. Düşük güvenli tahminler için insan değerlendirmesi kullanın.**

Bu seçenek en doğru olanıdır.

- Rekognition zaten *hazır bir moderasyon API'si* sunuyor.
- Ek model geliştirme yok → **minimum geliştirme çabası**.
- Düşük güven değerlerinde A2I ile insan doğrulaması destekleniyor.

X A. Amazon Comprehend

Comprehend metin analizi içindir, *görüntü analizi yapamaz*.

Bu yüzden tamamen yanlış.

X C. Amazon SageMaker + Ground Truth

Bu yol:

- Kendi modelinizi geliştirmenizi gerektirir
 - Veri etiketleme maliyetlidir
 - Geliştirme çabası çok yüksektir
- Dolayısıyla sorunun ana kriteri olan **minimum geliştirme çabasına** uymaz.

X D. AWS Fargate + custom model

Bu en çok geliştirme gerektiren yol:

- Modeli kendiniz eğitmelişiniz
 - Fargate ile dağıtım yapmalısınız
 - Moderasyon için hazır bir çözüm değildir
- Bu da sorunun istemediği bir senaryo.

SONUC

Geliştirme çabasını en aza indiren tek hazır çözüm Rekognition Moderation API + İnsan Doğrulama kombinasyonudur.

QUESTION 58

A company wants to run its critical applications in containers to meet requirements for scalability and availability. The company prefers to focus on maintenance of the critical applications. The company does not want to be responsible for provisioning and managing the underlying infrastructure that runs the containerized workload.

What should a solutions architect do to meet these requirements?

- A. Use Amazon EC2 instances, and install Docker on the instances.
- B. Use Amazon Elastic Container Service (Amazon ECS) on Amazon EC2 worker nodes.
- C. Use Amazon Elastic Container Service (Amazon ECS) on AWS Fargate.
- D. Use Amazon EC2 instances from an Amazon Elastic Container Service (Amazon ECS)-optimized Amazon Machine Image (AMI).

Soru:

Bir şirket, ölçeklenebilirlik ve kullanılabilirlik gereksinimlerini karşılamak için kritik uygulamalarını konteynerlerde çalıştırırmak istemektedir. Şirket, kritik uygulamaların bakımına odaklanmayı tercih etmektedir. Şirket, konteynerleştirilmiş iş yükünü çalıştırılan altyapısının sağlanması ve yönetilmesinden sorumlu olmak istememektedir.

Bu gereksinimleri karşılamak için bir çözüm mimarı ne yapmalıdır?

- A. Amazon EC2 örneklerini kullanın ve örneklerde Docker kurun.
- B. Amazon EC2 çalışan düğümlerinde Amazon Elastic Container Service (Amazon ECS) kullanın.
- C. AWS Fargate üzerinde Amazon Elastic Container Service (Amazon ECS) kullanın.
- D. Amazon Elastic Container Service (Amazon ECS) için optimize edilmiş Amazon Machine Image (AMI) kullanan Amazon EC2 örneklerini kullanın.

Soru Analizi:

Şirket, **kritik uygulamalarını konteynerlerde çalıştırırmak** istiyor.

Gereksinimler:

- ✓ Scalability (ölçeklenebilirlik)
- ✓ Availability (kullanılabilirlik)

- ✓ Şirket sadece uygulamalara odaklanmak istiyor
- ✓ Altyapıyı sağlamak veya yönetmek istemiyor (EN ÖNEMLİ MADDE)

Seçenek Analizi:

✓ C. ECS on AWS Fargate — DOĞRU CEVAP

Bu gereksinim, AWS'in tamamen yönetilen konteyner çalışma servisi olan **AWS Fargate**'i işaret eder.

Fargate sayesinde:

- Sunucuları/EC2'leri **sağlamanız gerekmez**
- Küme yönetimi yok
- Yama, kapasite planlaması vs. AWS tarafından yönetilir
- Şirket sadece uygulamalara odaklanır

Bu nedenle en uygun çözüm **ECS on Fargate**'tir.

- Sunucu yok → **serverless container compute**
- EC2 provisioning, scaling, patching yok
- Sadece container tanımlanır, gerisini AWS halleter
- Minimum operasyonel yük
- Sorunun tüm şartlarını karşılar

✗ A. Use Amazon EC2 instances, and install Docker on the instances.

- Altyapı yönetimi tamamen şirketin sorumluluğunda olur.
- Kapasite planlaması gereklidir.
- Manuel kurulum ve bakım gereklidir.

Sorunun koşullarına tamamen aykırıdır.

✗ B. ECS on EC2 worker nodes

- ECS kümesi için EC2 instance'ları şirket yönetir.
- EC2'lerin sağlanması, ölçeklenmesi ve patch edilmesi gereklidir.
- Şirket altyapı yönetmek istemiyor.

Bu nedenle uygun değildir.

✗ D. ECS-optimized AMI on EC2

- Yine EC2 instance yönetimi gereklidir.
- Altyapı bakım sorumluluğu şirkettidir.

Sorunun şartı olan “altyapı yönetmek istemiyoruz” koşuluna uymaz.

SONUÇ



C — Use Amazon ECS on AWS Fargate

Bu seçenek en az yönetim yükü, en yüksek soyutlama ve şirketin isteğine en uygun çözümüdür.

QUESTION 59

A company hosts more than 300 global websites and applications. The company requires a platform to analyze more than 30 TB of clickstream data each day.

What should a solutions architect do to transmit and process the clickstream data?

- A. Design an AWS Data Pipeline to archive the data to an Amazon S3 bucket and run an Amazon EMR cluster with the data to generate analytics.
- B. Create an Auto Scaling group of Amazon EC2 instances to process the data and send it to an Amazon S3 data lake for Amazon Redshift to use for analysis.
- C. Cache the data to Amazon CloudFront. Store the data in an Amazon S3 bucket. When an object is added to the S3 bucket, run an AWS Lambda function to process the data for analysis.
- D. Collect the data from Amazon Kinesis Data Streams. Use Amazon Kinesis Data Firehose to transmit the data to an Amazon S3 data lake. Load the data in Amazon Redshift for analysis.

Soru:

Bir şirket, dünya çapında 300'den fazla web sitesi ve uygulama barındırmaktadır. Şirketin her gün 30 TB'tan fazla tıklama akışı (clickstream) verisini analiz edebilecek bir platforma ihtiyacı vardır.

Tıklama akışı verisini iletmek ve işlemek için bir çözüm mimarı ne yapmalıdır?

- A. Veriyi bir Amazon S3 kovasına arşivlemek için bir AWS Data Pipeline tasarlayın ve analistik oluşturmak için bu verileri kullanarak bir Amazon EMRkümesi çalıştırın.
- B. Veriyi işlemek için bir Amazon EC2 Auto Scaling grubu oluşturun ve veriyi analiz için Amazon Redshift'in kullanacağı bir Amazon S3 veri gölüğe gönderin.
- C. Veriyi Amazon CloudFront'a önbelleğe alın. Veriyi bir Amazon S3 kovasında depolayın.

S3 kovasına bir nesne eklendiğinde veriyi analiz için işlemek üzere bir AWS Lambda fonksiyonu çalıştırın.

D. Veriyi Amazon Kinesis Data Streams ile toplayın. Amazon Kinesis Data Firehose kullanarak veriyi bir Amazon S3 veri gölüğe iletin. Analiz için veriyi Amazon Redshift'e yükleyin.

Soru Analizi:

Saha Gerçekleri: Clickstream Verisinin Özellikleri

Clickstream verisinin ortak özellikleri:

- Sürekli akar → “Unbounded streaming data”
- Yüksek hacimlidir → Burada **30 TB/gün**
- Parçalara bölünerek çok sık yazılır → micro-batches
- Düşük gecikme ile analiz edilmesi gereklidir
- Veri kaybına tolerans genelde **çok düşüktür**

Yani çözüm şu gereksinimleri karşılamalı:

- ✓ **High-throughput ingestion (çok yüksek veri alımı)**
- ✓ **Low latency or near real-time**
- ✓ **Auto-scaling**
- ✓ **Fault-tolerant streaming architecture**
- ✓ **Minimum operational overhead (soru bunu özellikle vurguluyor)**

Seçenek Analizi:

D) Kinesis Data Streams + Kinesis Firehose + S3 + Redshift

Bu özellikleri en iyi karşılayan AWS hizmet ailesi:

Kinesis Data Streams + Kinesis Firehose

✓ Neden en doğru çözüm?

Bu kombinasyon tam olarak şunlar için tasarlanmıştır:

✓ High-throughput stream ingestion

Bir Kinesis shard'ı 1 MB/sn ingest alabilir.

Kolayca **100+ shard** ile yatay ölçeklenebilir → 30 TB/gün rahat kaldırılır.

✓ Tamamen yönetilen veri aktarımı

- Firehose otomatik ölçeklenir
- Firehose buffering + batch yazma + sıkıştırma yapar
- Firehose → S3 veya Redshift'e sorunsuz aktarır
- Firehose veri kaybı durumunda **retry** + **backup** yapar

✓ Minimum operasyonel yük

- EC2 yok
- pipeline yönetimi yok
- scaling yok (Kinesis manual + Firehose auto-scaling)

✓ Modern clickstream data architecture

Bu mimari AWS'nin clickstream çözüm makalelerinde **resmi önerilen standardıdır**.

✗ A) AWS Data Pipeline + S3 + EMR

! Neden uygun değil?

- Data Pipeline artık modern streaming sistemleri için önerilmiyor.
- 30 TB/gün → Data Pipeline batch işleyicidir, *continuous ingestion* için uygun değildir.
- EMR ile analiz yapılabilir ama ingestion tarafı **streaming değil**.

Teknik sorun:

- Data Pipeline max throughput, bu büyüklükteki clickstream için yetersiz kalır.
- Modern event ingestion yerine batch ETL kullanmak **latency'i çok artırır**.

Sonuç:

Ölçeklenmez + modern değil + streaming ihtiyacını karşılamaz.

✗ B) Auto Scaling EC2 → S3 → Redshift

! Sorunlar:

- Clickstream işlemek için EC2 üzerinde kendi ingestion sistemini yazmak gerek.
- EC2 tabanlı ingestion → **çok fazla operasyonel yük**:
 - EC2 kapasite planlama
 - İşletim sistemi patchları
 - Dağıtım/monitoring/sorun giderme

Ek problem:

- 30 TB/gün trafik → yüzlerce EC2 instance ortaya çıkar.
- Bu tam olarak **şirketin kaçınmak istediği** şey:

"Şirket altyapıyı yönetmek istemiyor."

Sonuç:

Mimari olarak çalışır, ama operasyon maliyeti çok yüksektir.

✗ C) CloudFront Cache → S3 → S3 Event → Lambda

! Bu seçenek teknik olarak hatalı

1. **CloudFront cache** read optimizasyonu için vardır → write-heavy clickstream için değil.
2. 30 TB/gün → S3'e yazılan her nesne için Lambda tetiklenir:
 - Lambda concurrency limitlerini aşarsın
 - Maliyet patlar
 - Lambda 15 dakikadan uzun işleyemez
3. S3 event-based processing → **streaming** değildir, **event-driven**'dır.

Sonuç:

Yanlış servisler, yanlış mimari amaçlar için kullanılıyor.

🎯 Sonuç

En doğru ve modern mimari: D seçeneği

- ✓ Streaming için doğrudan tasarlanmış
- ✓ Yüksek throughput destekler
- ✓ Otomatik ölçeklenir
- ✓ En düşük operasyonel yük
- ✓ En esnek, dayanıklı, genişletilebilir yapı
- ✓ AWS tarafından önerilen best practice

QUESTION 60

A company has a website hosted on AWS. The website is behind an Application Load Balancer (ALB) that is configured to handle HTTP and HTTPS separately. The company wants to forward all requests to the website so that the requests will use HTTPS.

What should a solutions architect do to meet this requirement?

- A. Update the ALB's network ACL to accept only HTTPS traffic.
- B. Create a rule that replaces the HTTP in the URL with HTTPS.
- C. Create a listener rule on the ALB to redirect HTTP traffic to HTTPS.
- D. Replace the ALB with a Network Load Balancer configured to use Server Name Indication (SNI).

Soru:

Bir şirketin AWS üzerinde barındırılan bir web sitesi vardır. Web sitesi, HTTP ve HTTPS trafiğini ayrı ayrı işleyecek şekilde yapılandırılmış bir Application Load Balancer'ın (ALB) arkasındadır. Şirket, web sitesine gelen tüm isteklerin HTTPS kullanacak şekilde yönlendirilmesini istemektedir.

Bu gereksinimi karşılamak için bir çözüm mimarı ne yapmalıdır?

- A. ALB'nin ağ ACL'sini (network ACL) yalnızca HTTPS trafiğini kabul edecek şekilde güncelleyin.
- B. URL'deki HTTP'yi HTTPS ile değiştiren bir kural oluşturun.
- C. ALB üzerinde HTTP trafiğini HTTPS'ye yönlendirecek (redirect) bir dinleyici kuralı oluşturun.
- D. ALB'yi, Server Name Indication (SNI) kullanacak şekilde yapılandırılmış bir Network Load Balancer ile değiştirin.

Soru Analizi:

Senaryo:

- Web sitesi bir **Application Load Balancer (ALB)** arkasında.
- ALB **hem HTTP (80) hem HTTPS (443)** listener'larıyla yapılandırılmış.
- Amaç:
Tüm gelen istekler otomatik olarak HTTPS'e yönlendirilsin.

Bu AWS'nin en klasik HTTPS yönlendirme senaryosudur.

AWS ALB **redirect actions** ile bunu *yerleşik olarak* sağlar.

ALB üzerindeki **HTTP listener (port 80)** için bir kural eklenir:

IF request arrives on HTTP:80

THEN redirect to HTTPS:443

STATUS CODE = 301 (Moved Permanently)

Bu kadar.

Bu tamamen **ALB düzeyinde** yapılır; NACL, NLB, URL rewrite yapılmaz.

Seçenek Analizi:

 **DOĞRU CEVAP: C**

Create a listener rule on the ALB to redirect HTTP traffic to HTTPS.

Bu, AWS'nin resmi best practice çözümüdür.

 **A. Update the ALB's network ACL to accept only HTTPS traffic.**

Neden yanlış:

- NACL katmanı **L4**, yönlendirme (redirect) yapamaz.
- Sadece trafiği engeller/izin verir.
- HTTP'yi tamamen engellersen:
 - Kullanıcı HTTP'yi yazınca siteye hiç ulaşamaz.
 - Redirect gerçekleşmez.

Yani bu çözüm redirect gerçekleştirmez; sadece bloke eder.

 **B. Create a rule that replaces the HTTP in the URL with HTTPS.**

Neden yanlış:

- ALB URL rewrite (string replace) yapmaz.
- ALB'nin yaptığı şey **redirect**, URL metnini değiştirmek değildir.
- ALB'de böyle bir özellik yok → tamamen uydurma bir seçenek.

Yani AWS ALB'de “replace HTTP with HTTPS” diye bir action yoktur.

 **D. Replace the ALB with an NLB using SNI**

Neden yanlış:

- Network Load Balancer **L4 Load Balancer**'dır (TCP/UDP/SNI).
- NLB **HTTP redirect** desteklemez.
- Redirect sadece **ALB** üzerinde yapılabilir.
- Ayrıca ALB zaten HTTPS + redirect için idealdir, NLB gereksiz.

Doğru araç zaten ALB'dir — değiştirmek mantıksızdır.

 **SONUÇ**

 **ÖZET**

Seçenek Durum Açıklama

- A  Yanlış NACL redirect yapmaz, HTTP tamamen engellenir
- B  Yanlış ALB URL rewrite yapmaz
- C  Doğru ALB listener rule ile HTTP → HTTPS redirect
- D  Yanlış NLB redirect desteklemez

Doğru cevap: C

ALB üzerinde HTTP'den HTTPS'ye yönlendirme yapan listener kuralı oluşturmak en doğru çözümüdür.

QUESTION 61

A company is developing a two-tier web application on AWS. The company's developers have deployed the application on an Amazon EC2 instance that connects directly to a backend Amazon RDS database. The company must not hardcode database credentials in the application. The company must also implement a solution to automatically rotate the database credentials on a regular basis.

Which solution will meet these requirements with the LEAST operational overhead?

- A. Store the database credentials in the instance metadata. Use Amazon EventBridge (Amazon CloudWatch Events) rules to run a scheduled AWS Lambda function that updates the RDS credentials and instance metadata at the same time.
- B. Store the database credentials in a configuration file in an encrypted Amazon S3 bucket. Use Amazon EventBridge (Amazon CloudWatch Events) rules to run a scheduled AWS Lambda function that updates the RDS credentials and the credentials in the configuration file at the same time. Use S3 Versioning to ensure the ability to fall back to previous values.
- C. Store the database credentials as a secret in AWS Secrets Manager. Turn on automatic rotation for the secret. Attach the required permission to the EC2 role to grant access to the secret.
- D. Store the database credentials as encrypted parameters in AWS Systems Manager Parameter Store. Turn on automatic rotation for the encrypted parameters. Attach the required permission to the EC2 role to grant access to the encrypted parameters.

Soru:

Bir şirket, AWS üzerinde iki katmanlı bir web uygulaması geliştirmektedir. Şirketin geliştiricileri, uygulamayı arka ucta bir Amazon RDS veritabanına doğrudan bağlanan bir Amazon EC2 örneğine dağıttı. Şirket, veritabanı kimlik bilgilerinin uygulamaya sabit olarak (hardcode) yazılmamasını sağlamalıdır. Ayrıca şirket, veritabanı kimlik bilgilerini düzenli olarak otomatik şekilde döndürecek (rotate) bir çözüm uygulamalıdır.

En az operasyonel yük ile bu gereksinimleri hangi çözüm karşılar?

- A. Veritabanı kimlik bilgilerini instance metadata içinde saklayın. Amazon EventBridge (Amazon CloudWatch Events) kurallarını kullanarak zamanlanmış bir AWS Lambda fonksiyonu çalıştırın. Bu fonksiyon RDS kimlik bilgilerini ve instance metadata'yi aynı anda güncellesin.
- B. Veritabanı kimlik bilgilerini şifrelenmiş bir Amazon S3 kovasındaki bir yapılandırma dosyasında saklayın. Amazon EventBridge (Amazon CloudWatch Events) kurallarını kullanarak zamanlanmış bir AWS Lambda fonksiyonu çalıştırın. Bu fonksiyon RDS kimlik bilgilerini ve yapılandırma dosyasındaki kimlik bilgilerini aynı anda güncellesin. Geri dönülebilirlik için S3 Versioning'i kullanın.
- C. Veritabanı kimlik bilgilerini AWS Secrets Manager içinde bir gizli bilgi (secret) olarak saklayın. Secret için otomatik döndürmeyi (automatic rotation) etkinleştirin. EC2 rolüne, bu secret'a erişim için gereken izni ekleyin.
- D. Veritabanı kimlik bilgilerini AWS Systems Manager Parameter Store içinde şifreli parametreler olarak saklayın. Parametreler için otomatik döndürmeyi etkinleştirin. EC2 rolüne, bu şifreli parametrelere erişim için gereken izni ekleyin.

Soru Analizi:

Gereksinimler:

- ✓ Veritabanı kimlik bilgileri (username/password) uygulamada hardcoded edilmeyecek
- ✓ Kimlik bilgileri otomatik olarak düzenli aralıklarla döndürülecek (rotation)
- ✓ Çözüm en az operasyonel yük (least operational overhead) gerektirmeli
- ✓ Sistem: EC2 → RDS

Seçenek Analizi:

C. Secrets Manager + Automatic Rotation

Bu gereksinimleri AWS üzerinde **en doğal, en otomatik**, en az bakım gerektiren şekilde karşılayan servis:

AWS Secrets Manager

Secrets Manager:

- RDS için **built-in automatic rotation** sağlar.
- Rotation işlemi için gerekli Lambda fonksiyonlarını **kendisi oluşturur**.
- Uygulama sadece secret'ı **API çağrıları ile çeker**.
- Sürümleme, erişim kontrolü, audit, encryption hepsi default gelir.

Dolayısıyla sorunun istediği “least operations” = **Secrets Manager**.

Neden doğru?

- ✓ RDS ile **doğrudan entegredir** (MySQL, PostgreSQL, Aurora vb.)
 - ✓ Rotation için gerekli Lambda **otomatik oluşturulur**
 - ✓ Uygulama, IAM role + Secrets Manager SDK/API ile sadece secret'ı okur
 - ✓ Zero-downtime credential rotation
 - ✓ Encryption, versioning, audit, failover hepsi default gelir
 - ✓ **En az operasyonel yük** tam olarak bu çözümüdür
- AWS'in önerdiği modern, “best practice” çözümüdür.

✗ A. Credentials in instance metadata + Lambda rotation

Neden yanlış?

- Instance metadata **gizli bilgi saklamak için tasarlanmış değildir**.
- Metadata **şifreli değildir**.
- Credentials rotation için **manuel Lambda kodu + event yönetimi** gerektirir.
- Breakage riski çok yüksek: metadata güncellenirken instance yeniden başlatılmalıdır.
- Operation yükü **çok büyük**.

→ **Güvensiz + yönetimi zor + best practice değil**.

✗ B. Credentials in encrypted S3 file + Lambda rotation

Neden yanlış?

- Bir dosya içinde credential saklamak **çok kötü bir mimari pratik**.
- Rotation için:
 - Lambda fonksiyonunun S3 içeriğini güncellemesi gereklidir.
 - S3 Versioning geri dönüş sağlar ama **devasa operasyonel karmaşıklık** yaratır.
- Credentials'ın S3 gibi bir object store'da saklanması **tavsiye edilmez**.

→ Çok fazla operasyon yükü + güvenlik riskleri.

✗ D. SSM Parameter Store + automatic rotation

Neden yanlış? (Kısmen doğru gibi görünür ama doğru değildir)

- SSM Parameter Store **RDS automatic rotation için built-in destek sunmaz.**
- Rotation işlemi için **manuel Lambda yazmanız** gereklidir.
- Parameter Store yalnızca “AWS KMS ile encrypted parameter” sunar.
- Rotation süreci tamamen **siz tarafından yönetilir.**

Ayrıca:

- Secrets Manager RDS rotation için official/managed çözüm sunarken **Parameter Store bunu doğrudan yapmaz.**

→ Otomatik rotation çözümü yok → operasyon yükü daha yüksek.

⌚ SONUÇ

En doğru, en güvenli ve en az operasyon gerektiren çözüm: C

✳️ ÖZET TABLO

Seçenek Uygunluk Sebep

- | | | |
|---|---|--|
| A | ✗ | Metadata güvenli değil, rotation manuel, çok operasyon |
| B | ✗ | S3 credential storage yanlış, rotation karmaşık |
| C | ✓ | Built-in automatic RDS rotation, en az operasyon yükü |
| D | ✗ | Parameter Store otomatik RDS rotation desteklemez |

QUESTION 62

A company is deploying a new public web application to AWS. The application will run behind an Application Load Balancer (ALB). The application needs to be encrypted at the edge with an SSL/TLS certificate that is issued by an external certificate authority (CA). The certificate must be rotated each year before the certificate expires.

What should a solutions architect do to meet these requirements?

- A. Use AWS Certificate Manager (ACM) to issue an SSL/TLS certificate. Apply the certificate to the ALB. Use the managed renewal feature to automatically rotate the certificate.

- B. Use AWS Certificate Manager (ACM) to issue an SSL/TLS certificate. Import the key material from the certificate. Apply the certificate to the ALB. Use the managed renewal feature to automatically rotate the certificate.
- C. Use AWS Certificate Manager (ACM) Private Certificate Authority to issue an SSL/TLS certificate from the root CA. Apply the certificate to the ALB. Use the managed renewal feature to automatically rotate the certificate.
- D. Use AWS Certificate Manager (ACM) to import an SSL/TLS certificate. Apply the certificate to the ALB. Use Amazon EventBridge (Amazon CloudWatch Events) to send a notification when the certificate is nearing expiration. Rotate the certificate manually.

Soru:

Bir şirket yeni bir **genel (public) web uygulamasını** AWS'ye dağıtmaktadır. Uygulama bir **Application Load Balancer (ALB)** arkasında çalışacaktır. Uygulamanın, **harici bir sertifika otoritesi (CA)** tarafından verilen bir **SSL/TLS sertifikası** ile ucta (edge) şifrelenmesi gerekmektedir. Sertifikanın süresi dolmadan **yılda bir kez döndürülmesi (rotate)** gereklidir.

Bu gereksinimleri karşılamak için bir çözüm mimarı ne yapmalıdır?

- A. AWS Certificate Manager (ACM) kullanarak bir SSL/TLS sertifikası oluşturun. Sertifikayı ALB'ye uygulayın. Sertifikayı otomatik döndürmek için yönetilen yenileme özelliğini kullanın.
- B. AWS Certificate Manager (ACM) kullanarak bir SSL/TLS sertifikası oluşturun. Sertifikadaki anahtar materyalini içe aktarın. Sertifikayı ALB'ye uygulayın. Sertifikayı otomatik döndürmek için yönetilen yenileme özelliğini kullanın.
- C. AWS Certificate Manager (ACM) Private Certificate Authority kullanarak kök CA'dan bir SSL/TLS sertifikası oluşturun. Sertifikayı ALB'ye uygulayın. Sertifikayı otomatik döndürmek için yönetilen yenileme özelliğini kullanın.
- D. AWS Certificate Manager (ACM) içine bir SSL/TLS sertifikası **ithal (import)** edin. Sertifikayı ALB'ye uygulayın. Sertifika süresi yaklaşırken bir bildirim göndermek için Amazon EventBridge (Amazon CloudWatch Events) kullanın. Sertifikayı **manuel** olarak döndürün.

Soru Analizi:

Şirket:

- Bir **public web uygulaması** dağıtıyor.
- Uygulama **ALB** arkasında.
- SSL/TLS sertifikası **harici bir Certificate Authority (CA)** tarafından verilecek.

- Sertifika **ALB üzerinde kullanılacak**.
- Sertifika **yılda bir kez döndürülecek (rotate)**.
- Sertifika ACM tarafından otomatik yenilenmez çünkü:
 - ACM sadece **kendi verdiği** public sertifikaları otomatik yeniler.
 - Harici CA'dan alınmış sertifikalar **ACM'e import edilir ve otomatik yenilenmez**.

Dolayısıyla:

- ✓ Harici CA → ACM otomatik yenileme yok
- ✓ Harici CA sertifikası → import edilmelidir
- ✓ Bu durumda çözüm → kullanıcı sertifikayı manuel yenileyebilir ama AWS uyarı gönderebilir

Seçenek Analizi:

✓ D. ACM'e harici CA'dan alınmış sertifika import edin. EventBridge ile expiration uyarısı gönderin. Manuel olarak yenileyin.

✓ Doğru – Tek doğru seçenek

- Harici CA sertifikası → ACM'e *import* edilir.
- ACM import edilmiş sertifikaları **otomatik yenileyemez**.
- EventBridge sertifika expiration yaklaşınca **bildirim gönderebilir**.
- Yenileme manuel yapılır → soru bunun dışında özel bir otomasyon şartı vermiyor.

Bu nedenle **en uygun ve AWS tarafından önerilen çözüm D'dır**.

✗ A. ACM sertifika versin, ALB'ye takın, otomatik yenilesin.

ACM'in verdiği public sertifikalar **harici CA** değildir.

Soru açıkça **harici sertifika otoritesi** istiyor.

Ayrıca ACM sertifikasını ALB'ye bağlamak mümkün olsa da **şirket kendi CA'sını kullanmak istiyor**.

✗ B. ACM sertifika versin, key import edin, otomatik yenileyin.

- ACM'den sertifika oluşturup, ayrıca key import etmek diye bir süreç yok.
- Ayrıca ACM'e **import edilmiş sertifikalar otomatik yenilenmez**.

Bu seçenek teknik olarak da anlamsız.

✗ C. ACM Private CA kullanarak sertifika üretin, otomatik yenileyin.

- ACM Private CA **harici CA değildir**, şirketin kendi internal CA'sıdır.

- Public web uygulamasında kullanılmaz (tarayıcılar güvenmez).
- Public facing ALB üzerinde kullanılmaz.

 **SONUÇ:**

 **Önemli Nokta: Harici CA sertifikasını ACM otomatik yenilemez**

AWS Certificate Manager (ACM):

Sertifika Kaynağı	ALB Üzerinde Kullanılabilir	Otomatik Yenileme	Not
ACM Public Sertifika	Evet	Evet	AWS yönetir
ACM Private CA Sertifikası	Evet	Evet	Kendi Private CA ise
Harici CA Sertifikası (import)	Evet	Hayır	Manuel yenileme gereklidir

QUESTION 63

A company runs its infrastructure on AWS and has a registered base of 700,000 users for its document management application. The company intends to create a product that converts large .pdf files to .jpg image files. The .pdf files average 5 MB in size. The company needs to store the original files and the converted files. A solutions architect must design a scalable solution to accommodate demand that will grow rapidly over time. Which solution meets these requirements MOST cost-effectively?

- Save the .pdf files to Amazon S3. Configure an S3 PUT event to invoke an AWS Lambda function to convert the files to .jpg format and store them back in Amazon S3.
- Save the .pdf files to Amazon DynamoDB. Use the DynamoDB Streams feature to invoke an AWS Lambda function to convert the files to .jpg format and store them back in DynamoDB.
- Upload the .pdf files to an AWS Elastic Beanstalk application that includes Amazon EC2 instances, Amazon Elastic Block Store (Amazon EBS) storage, and an Auto Scaling group. Use a program in the EC2 instances to convert the files to .jpg format. Save the .pdf files and the .jpg files in the EBS store.
- Upload the .pdf files to an AWS Elastic Beanstalk application that includes Amazon EC2 instances, Amazon Elastic File System (Amazon EFS) storage, and an Auto Scaling group. Use a program in the EC2 instances to convert the files to .jpg format. Save the .pdf files and the .jpg files in the EFS store.

Soru:

Bir şirket altyapısını AWS üzerinde çalıştırmaktadır ve belge yönetimi uygulaması için **700.000 kayıtlı kullanıcısı** vardır. Şirket, büyük **.pdf dosyalarını .jpg görüntü dosyalarına dönüştüren** bir ürün oluşturmak istemektedir.

.pdf dosyalarının ortalama boyutu **5 MB**'dır. Şirket hem **orijinal dosyaları** hem de **dönüştürülmüş dosyaları** saklamaya ihtiyaç duymaktadır. Bir solutions architect, zaman içinde hızla artacak talebi karşılayacak **ölçeklenebilir ve maliyet açısından en verimli çözümü** tasarlamalıdır.

Bu gereksinimleri **en uygun maliyetli şekilde** karşılayan çözüm hangisidir?

A. .pdf dosyalarını Amazon S3'e kaydedin. S3 PUT eventi oluşturarak bir AWS Lambda fonksiyonunu tetikleyin. Dosyaları .jpg formatına dönüştürüp tekrar Amazon S3'e kaydedin.

B. .pdf dosyalarını Amazon DynamoDB'ye kaydedin. DynamoDB Streams özelliğini kullanarak bir AWS Lambda fonksiyonunu tetikleyin. Dosyaları .jpg formatına dönüştürüp tekrar DynamoDB'ye kaydedin.

C. .pdf dosyalarını bir AWS Elastic Beanstalk uygulamasına yükleyin. Uygulama, Amazon EC2 instance'ları, Amazon EBS depolaması ve bir Auto Scaling grubu içerir. EC2'deki bir programı kullanarak dosyaları .jpg formatına dönüştürün. Hem .pdf hem .jpg dosyalarını EBS'de saklayın.

D. .pdf dosyalarını bir AWS Elastic Beanstalk uygulamasına yükleyin. Uygulama Amazon EC2 instance'ları, Amazon EFS depolaması ve bir Auto Scaling grubu içerir. EC2'deki bir programı kullanarak dosyaları .jpg formatına dönüştürün. Hem .pdf hem .jpg dosyalarını EFS'de saklayın.

Soru Analizi:

Şirket:

- 700.000 kullanıcı → yüksek trafik potansiyeli
- Her PDF \approx 5 MB → orta boyutlu
- Hem orijinal PDF hem de dönüştürülmüş JPG saklanacak
- Talep zamanla hızla artacak → **yüksek ölçeklenebilirlik**
- Çözüm **maliyet açısından en verimli (cost-effective)** olmalı
- İşlem: **PDF → JPG dönüşümü**

Bu bir *dosya odaklı workload*, *event-driven işlem* ve *yüksek ölçeklenebilirlik* gerektirir.

Bu iş için AWS'nin doğru yaklaşımı:

→ **S3 + Event + Lambda + Serverless + pay-as-you-go**

Seçenek Analizi:

● **A. S3'e kaydet → S3 PUT event → Lambda → tekrar S3'e kaydet**

Bu doğru cevaptır.

Neden?

- Amazon S3 büyük dosya depolamada en ucuz (S3 Standard / Infrequent Access).
- S3 sonsuz ölçeklenebilir.
- Lambda event-driven olarak otomatik tetiklenir.
- Lambda dakikalar içinde yüzbinlerce işlemi ölçekleyebilir.
- Sunucu yönetimi yok → en düşük operasyonel maliyet.
- En düşük altyapı maliyeti → pay-as-you-go.
- Dosyayı tekrar S3'e kaydetmek çok ucuzdur.

Bu, AWS'nin *file-processing pattern* için önerdiği **resmi en iyi mimaridir**.

● **B. DynamoDB'ye kaydet + Streams + Lambda**

Bu seçenek **tamamen hatalı**.

Neden yanlış?

- DynamoDB bir *key-value / NoSQL* veritabanıdır, **dosya depolamaz**.
- 5 MB *sınırıdır*, bu PDF'ler zaten 5 MB olduğu için verimli olamaz.
- Büyük dosyalar için DynamoDB **yanlış servistir**.
- Maliyet olarak aşırı pahalı olur.

● **C. Elastic Beanstalk + EC2 + EBS + Auto Scaling**

Bu seçenek **çok pahalı ve gereksiz karmaşık**.

Neden yanlış?

- EC2 instance'ları sürekli açık kalır → yüksek maliyet.
- EBS *blok depolamadır*, büyük dosyalar için S3 kadar ucuz ve elastik değildir.
- EC2 üzerinde dönüştürme işlemi yönetim yükü getirir.
- Scalability için EC2 autoscaling daha pahalıdır, Lambda kadar hızlı ölçeklenmez.

- Dosya saklama EBS'de = **çok maliyetli ve sınırlı**.

D. Elastic Beanstalk + EC2 + EFS + Auto Scaling

Bu da **pahalı ve gereksiz** bir çözüm.

Neden yanlış?

- EFS yüksek maliyetli (S3'den 10x–20x daha pahalı).
- EFS büyük dosya arşivlemek için uygun değildir.
- EC2 yönetimi yine operasyonel yük gerektirir.
- Lambda ve S3 gibi “serverless” yaklaşım yok → capacity planning gereklidir.

Bu seçenek teknik olarak çalışır ama **en pahalı ve en az uygun çözüm**dur.

SONUÇ:

A. Save the .pdf files to S3 → S3 triggers Lambda → Lambda converts to JPG → S3'e geri kaydedilir

Bu çözüm:

- ✓ En ucuz depolama (S3)
- ✓ En düşük işlem maliyeti (Lambda)
- ✓ Sonsuz ölçeklenebilir
- ✓ Tamamen serverless
- ✓ Minimum yönetim
- ✓ Yerleşik event-driven mimari

Bu yüzden **en cost-effective ve en scalable çözüm** budur.

QUESTION 64

A company has more than 5 TB of file data on Windows file servers that run on premises. Users and applications interact with the data each day. The company is moving its Windows workloads to AWS. As the company continues this process, the company requires access to AWS and on-premises file storage with minimum latency. The company needs a solution that minimizes operational overhead and requires no significant changes to the existing file access patterns. The company uses an AWS Site-to-Site VPN connection for connectivity to AWS.

What should a solutions architect do to meet these requirements?

- A. Deploy and configure Amazon FSx for Windows File Server on AWS. Move the on-premises file data to FSx for Windows File Server. Reconfigure the workloads to use FSx for Windows File Server on AWS.
- B. Deploy and configure an Amazon S3 File Gateway on premises. Move the on-premises file data to the S3 File Gateway. Reconfigure the on-premises workloads and the cloud workloads to use the S3 File Gateway.
- C. Deploy and configure an Amazon S3 File Gateway on premises. Move the on-premises file data to Amazon S3. Reconfigure the workloads to use either Amazon S3 directly or the S3 File Gateway, depending on each workload's location.
- D. Deploy and configure Amazon FSx for Windows File Server on AWS. Deploy and configure an Amazon FSx File Gateway on premises. Move the on-premises file data to the FSx File Gateway. Configure the cloud workloads to use FSx for Windows File Server on AWS. Configure the on-premises workloads to use the FSx File Gateway.

Soru:

Bir şirketin şirket içi çalışan Windows dosya sunucularında 5 TB'tan fazla dosya verisi bulunmaktadır. Kullanıcılar ve uygulamalar her gün bu verilerle etkileşim kurmaktadır. Şirket Windows iş yüklerini AWS'ye taşımaktadır. Şirket bu süreci sürdürürken, hem AWS'deki hem de şirket içindeki dosya depolamaya minimum gecikmeyle erişim gerekmektedir. Şirket, operasyonel yükün en az olduğu ve mevcut dosya erişim yöntemlerinde önemli bir değişiklik gerektirmeyen bir çözüm istemektedir. Şirket, AWS'ye bağlanmak için bir AWS Site-to-Site VPN bağlantısı kullanmaktadır.

Bu gereksinimleri karşılamak için bir solutions architect ne yapmalıdır?

- A. AWS üzerinde Amazon FSx for Windows File Server dağıtan ve yapılandırın. Şirket içi dosya verilerini FSx for Windows File Server'a taşıyın. İş yüklerini FSx for Windows File Server'ı kullanacak şekilde yeniden yapılandırın.
- B. Şirket içinde bir Amazon S3 File Gateway dağıtan ve yapılandırın. Şirket içi dosya verilerini S3 File Gateway'e taşıyın. Şirket içi iş yüklerini ve bulut iş yüklerini S3 File Gateway'i kullanacak şekilde yeniden yapılandırın.
- C. Şirket içinde bir Amazon S3 File Gateway dağıtan ve yapılandırın. Şirket içi dosya verilerini Amazon S3'e taşıyın. İş yüklerini, konumlarına bağlı olarak doğrudan Amazon S3'ü veya S3 File Gateway'i kullanacak şekilde yeniden yapılandırın.
- D. AWS üzerinde Amazon FSx for Windows File Server dağıtan ve yapılandırın. Şirket içinde bir Amazon FSx File Gateway dağıtan ve yapılandırın. Şirket içi dosya verilerini FSx File Gateway'e taşıyın. Bulut iş yüklerini AWS'deki FSx for Windows File Server'ı kullanacak şekilde yapılandırın. Şirket içi iş yüklerini FSx File Gateway'i kullanacak şekilde yapılandırın.

Soru Analizi:

Şirketin mevcut durumu:

- On-premises Windows file server → **5 TB+ veri**
- Windows uygulamaları bu dosyalara **SMB protokolü** ile erişiyor.
- Veri **her gün kullanıcılar tarafından aktif olarak kullanılıyor.**
- Şirket Windows iş yüklerini AWS'ye taşıyor → hem AWS hem on-prem ortamları **aynı dosyalara erişmek zorunda.**
- **Minimum latency** istiyor → On-prem kullanıcıların dosyalara LAN hızında erişmesi gereklidir.
- **Minimum operasyonel yük** (managed service tercih edilir).
- **Dosya erişim modelini değiştirmek istemiyor** → SMB paylaşımalar kullanılmaya devam etmeli.
- Bağlantı: **Site-to-Site VPN** (yani AWS Direct Connect yok → AWS'deki FSx'e erişim gecikmesi yüksek olabilir)

Seçenek Analizi:

● D. FSx for Windows File Server + FSx File Gateway

Bu gibi senaryolarda AWS'nin **resmi desteklediği en ideal çözüm:**

- **AWS tarafında FSx for Windows File Server (tam yönetilen Windows SMB paylaşımı)**
- **On-prem tarafında FSx File Gateway (düşük gecikmeli cache + SMB erişimi)**

Bu yaklaşım:

- On-prem kullanıcıların düşük gecikmeli (cache'lenmiş) dosya erişimi sağlar.
- AWS'deki iş yükleri FSx'e native erişir.
- Dosya erişim modeli değişmez → SMB devam eder.
- Minimum yönetim, full managed.
- VPN üzerinden büyük dosyaların tekrar tekrar çekilmesini engeller → FSx File Gateway cache sayesinde.

Neden?

- ✓ **Windows File Server için native çözüm (SMB, NTFS ACL, locking)**
- ✓ On-prem tarafında **FSx File Gateway** düşük gecikmeli **cache** sağlar

- ✓ On-prem kullanıcılar LAN üzerinden hızlı erişir
- ✓ Aynı veri AWS'de FSx üzerinde bulunur → cloud iş yükleri native erişir
- ✓ Yönetim yükü minimum → AWS full-managed Windows file system
- ✓ File access pattern değişmez → kullanıcılar yine SMB paylaşımı kullanır
- ✓ VPN üzerinden veri sadece gerektiğinde çekilir → maliyet & latency düşer
- ✓ AWS'nin bu kullanım senaryosu için özel tasarladığı çözüm

✗ A. FSx for Windows File Server'a taşı, tüm iş yüklerini FSx'e yönlendir

Neden yanlış?

- On-prem kullanıcılar FSx'e **VPN üzerinden yüksek gecikmeyle** erişir.
- "Minimum latency" gereksinimini karşılamaz.
- FSx'e erişim SMB üzerinden yapılır ama VPN'de performans çok düşer.
- On-prem caching yok → 5 TB kullanımında çok yavaşlar.

✗ B. S3 File Gateway → S3 kullanımı

Neden yanlış?

- S3 File Gateway **SMB değil, NFS/HTTPS benzeri bir erişim modeli sağlar.**
- Windows SMB ACL yapısıyla uyumlu değildir.
- Uygulamalar ve kullanıcılar dosyaya "Windows File Share" olarak erişmek istiyor → S3 bu modeli desteklemez.
- "Erişim modelini değiştirmeme" şartını karşılamaz.

✗ C. S3 + S3 File Gateway kombinasyonu

Neden yanlış?

- Veri artık S3'te olur → **Windows FS özellikleri kaybolur (SMB ACL, locking, NTFS özellikleri vb).**
- S3 Windows file server yerine geçmez.
- On-prem ve AWS uygulamaları farklı erişim yolları kullanır → karmaşık, uyumsuz.
- Minimum değişiklik ve minimum operasyonel yük gereksinimleri karşılanmaz.

⌚ SONUÇ:

En doğru, en hızlı, en düşük gecikmeli, en az operasyonel yük taşıyan çözüm → D

D. FSx for Windows File Server + FSx File Gateway

Bu, AWS Certified Solutions Architect sınavında da benzer şekilde çıkan soruların standart doğru cevabıdır.

QUESTION 65

A hospital recently deployed a RESTful API with Amazon API Gateway and AWS Lambda. The hospital uses API Gateway and Lambda to upload reports that are in PDF format and JPEG format. The hospital needs to modify the Lambda code to identify protected health information (PHI) in the reports.

Which solution will meet these requirements with the LEAST operational overhead?

- A. Use existing Python libraries to extract the text from the reports and to identify the PHI from the extracted text.
- B. Use Amazon Textract to extract the text from the reports. Use Amazon SageMaker to identify the PHI from the extracted text.
- C. Use Amazon Textract to extract the text from the reports. Use Amazon Comprehend Medical to identify the PHI from the extracted text.
- D. Use Amazon Rekognition to extract the text from the reports. Use Amazon Comprehend Medical to identify the PHI from the extracted text.

Soru:

Bir hastane yakın zamanda Amazon API Gateway ve AWS Lambda ile bir RESTful API dağıttı. Hastane, PDF formatında ve JPEG formatında olan raporları yüklemek için API Gateway ve Lambda kullanıyor. Hastanenin, raporlardaki korunan sağlık bilgilerini (PHI) tespit etmek için Lambda kodunu değiştirmesi gerekiyor.

Aşağıdaki çözümlerden hangisi EN AZ operasyonel yük ile bu gereksinimleri karşılar?

- A. Raporlardan metin çıkarmak ve çıkarılan metinden PHI'yi tespit etmek için mevcut Python kütüphanelerini kullanın.
- B. Raporlardan metni çıkarmak için Amazon Textract kullanın. Çıkarılan metinden PHI'yi tespit etmek için Amazon SageMaker kullanın.
- C. Raporlardan metni çıkarmak için Amazon Textract kullanın. Çıkarılan metinden PHI'yi tespit etmek için Amazon Comprehend Medical kullanın.
- D. Raporlardan metni çıkarmak için Amazon Rekognition kullanın. Çıkarılan metinden PHI'yi tespit etmek için Amazon Comprehend Medical kullanın.

Soru Analizi:

Hastane API Gateway + Lambda kullanıyor → Yani **sunucusuz (serverless)** ve **operasyonel yükün düşük olması** isteniyor.

Hastane PDF ve JPEG raporlarını yükliyor.

Amaç: **PHI (Protected Health Information)** tespiti.

Bu süreçte iki aşamalı bir işlem gerekir:

1. **PDF/JPEG içinden metin çıkarma**
2. **Metin içinden sağlık bilgilerinin (PHI) tespiti**

Minimum operasyonel yük isteyen sorularda AWS'nin **tam yönetilen (fully managed)** hizmetleri tercih edilir.

Seçenek Analizi:

C. Textract + Comprehend Medical

PHI tespiti için AWS'nin özel hizmeti:

Amazon Comprehend Medical

Bu, sağlık metinlerinden PHI, tıbbi terimler, ilaçlar, bulgular, teşhisler... gibi bilgileri otomatik çıkararak tam yönetilen bir AI hizmetidir.

PDF/JPEG içinden metin çıkarmak için en doğru AWS hizmeti:

Amazon Textract

Özellikle dokümanlardan OCR ve yapılandırılmış veri çıkarma için optimize edilmiştir.

Dolayısıyla ideal akış: Textract → Comprehend Medical

En az operasyonel yük

Tam yönetilen

En yüksek doğruluk

En düşük bakım maliyeti

- Textract → dokümandan metin çıkarır
- Comprehend Medical → PHI'yı otomatik olarak bulur

Hastane gibi sağlık kurumları için zaten önerilen mimaridir.

A. Python kütüphaneleriyle metin çıkarma ve PHI tespiti

En fazla operasyonel yük

- Tesseract OCR ya da benzeri kütüphaneleri Lambda içinde çalıştırmak zor.
- PHI tespiti için özel kod yazmak gereklidir, doğru sonuçlar garanti değil.
- Periyodik güncellemeler, hata ayıklama, model eğitimi kullanıcıya kalır.

Operasyonel yük yüksek → Elenir.

B. Textract + SageMaker

Gereksiz şekilde fazla karmaşık.

- Textract doğru seçimdir ✓
- Ama PHI tespiti için bir SageMaker modeli oluşturmak:
 - model eğitimi,
 - hyperparameter tuning,
 - versiyon yönetimi,
 - bakım...

çok yüksek operasyonel yük oluşturur.

Oysa AWS bu iş için zaten Comprehend Medical'i sunuyor.

✗ Rekognition + Comprehend Medical

Rekognition OCR yapabilir, ama:

- JPEG için çalışır; PDF desteklemez.
- Textract kadar doğru değildir.
- Rekognition genelde yüz tanıma, görüntü sınıflandırma içindir.

PDF işlenemediği için gereksinim karşılanmaz.

🎯 SONUÇ

En az operasyonel yük, en doğru AWS hizmetleriyle:

👉 Cıkkı: Amazon Textract + Amazon Comprehend Medical
