

QUESTION 1

A company has an application that generates a large number of les, each approximately 5 MB in size. The les are stored in Amazon S3. Company policy requires the les to be stored for 4 years before they can be deleted. Immediate accessibility is always required as the les contain critical business data that is not easy to reproduce. The les are frequently accessed in the first 30 days of the object creation but are rarely accessed after the first 30 days.

Which storage solution is MOST cost-effective?

- A. Create an S3 bucket lifecycle policy to move les from S3 Standard to S3 Glacier 30 days from object creation. Delete the les 4 years after object creation.
- B. Create an S3 bucket lifecycle policy to move les from S3 Standard to S3 One Zone-Infrequent Access (S3 One Zone-IA) 30 days from object creation. Delete the les 4 years after object creation.
- C. Create an S3 bucket lifecycle policy to move les from S3 Standard to S3 Standard-Infrequent Access (S3 Standard-IA) 30 days from object creation. Delete the les 4 years after object creation.
- D. Create an S3 bucket lifecycle policy to move les from S3 Standard to S3 Standard-Infrequent Access (S3 Standard-IA) 30 days from object creation. Move the les to S3 Glacier 4 years after object creation.

Soru:

Bir şirket, her biri yaklaşık 5 MB boyutunda olan çok sayıda dosya oluşturan bir uygulamaya sahiptir. Dosyalar Amazon S3'te depolanmaktadır. Şirket politikası, dosyaların silinebilmelerinden önce 4 yıl boyunca saklanması gerektir. Dosyalar, yeniden oluşturulması kolay olmayan kritik iş verisi içерdiğinden, her zaman anında erişilebilir olmalıdır. Dosyalar, oluşturulduktan sonraki ilk 30 gün içinde sık erişilir, ancak ilk 30 günden sonra nadiren erişilir.

En maliyet etkin depolama çözümü hangisidir?

- A. Nesne oluşturulduktan 30 gün sonra dosyaları S3 Standard'dan S3 Glacier'a taşımak için bir S3 komut dosyası yaşam döngüsü (lifecycle) politikası oluşturun. Nesneleri oluşturulduktan 4 yıl sonra silin.
- B. Nesne oluşturulduktan 30 gün sonra dosyaları S3 Standard'dan S3 One Zone-Infrequent Access (S3 One Zone-IA)'a taşımak için bir S3 yaşam döngüsü politikası oluşturun. Nesneleri oluşturulduktan 4 yıl sonra silin.

C. Nesne oluşturuluktan 30 gün sonra dosyaları S3 Standard'dan S3 Standard-Infrequent Access (S3 Standard-IA)'a taşımak için bir S3 yaşam döngüsü politikası oluşturun. Nesneleri oluşturuluktan 4 yıl sonra silin.

D. Nesne oluşturuluktan 30 gün sonra dosyaları S3 Standard'dan S3 Standard-Infrequent Access (S3 Standard-IA)'a taşımak için bir S3 yaşam döngüsü politikası oluşturun. Nesneleri oluşturuluktan 4 yıl sonra S3 Glacier'a taşıyın.

Soru Analizi:

Şirketin dosyalarla ilgili gereksinimleri şunlar:

1. Dosya boyutu:

Her biri ~5 MB → küçük nesneler.

2. Depolama süresi:

Dosyalar **4 yıl boyunca saklanmalı**.

3. Erişim gereksinimi:

- İlk 30 gün: sık erişim
- 30 günden sonra seyrek erişim
- Ancak her zaman anında erişilebilir olmalı → Glacier Instant Retrieval değilse Glacier uygun değildir.
(S3 Glacier Flexible Retrieval ve Deep Archive milisaniyelik değil, dakikalar/saatler sürer.)

→ Bu nedenle:

S3 Glacier (A ve D seçeneklerinde geçen) uygun değildir, çünkü “immediate accessibility” şartı var.

Seçenek Analizi:

C. 30 gün sonra S3 Standard-IA'ya taşı, 4 yıl sonra sil

En doğru ve en maliyet-etkin çözüm

Neden?

- İlk 30 gün sık erişildiği için **S3 Standard** uygun → düşük gecikme + yüksek erişilebilirlik.
- 30 günden sonra nadir erişiliyor → **Standard-IA** ekonomik.
- Standard-IA **yüksek dayanıklılığa** sahip (One Zone-IA gibi tek AZ değil).

- Instant access → IA sınıfı milisaniyelik erişim sağlar.
- 4 yıl saklama süresince **hiç Glacier'a geçirmeden** maliyet optimum.

Gereksinimlerin hepsine uyuyor.

✗ A. 30 gün sonra S3 Glacier'a taşı, 4 yıl sonra sil

Neden yanlış?

- Glacier (Flexible/Deep) **anında erişilebilir değildir** → istek dakikalar/saatler sürebilir.
- Soruda "*Immediate accessibility is always required*" deniyor.

Bu nedenle uygun değil.

✗ B. 30 gün sonra S3 One Zone-IA'ya taşı, 4 yıl sonra sil

Neden yanlış?

- One Zone-IA **tek AZ'de depolama** yapar → kritik iş verisi için **dayanıklılık düşük**.
- Soru: **kritik, tekrar üretilemeyen iş verisi** diyor → One Zone-IA riski kabul edilemez.

Bu nedenle yanlış.

✗ D. 30 gün sonra IA, 4 yıl sonra Glacier

Neden yanlış?

- 4 yıldan sonra Glacier'a geçiyor → yine **immediate accessibility** bozulur.
- Glacier erişimi dakikalar-saatler sürer.

Bu nedenle elenir.

⌚ SONUÇ

Bu çözüm:

- Gereksinimlere tam uyuyor,
- En iyi maliyet optimizasyonunu sağlıyor,
- Erişilebilirliği koruyor,
- Dayanıklılığı yüksek tutuyor.

S3 Standard: Sık erişilen veri → en hızlı ve en pahalı.

S3 Standard-IA: Nadir erişim + anında erişim → uygun maliyetli.

S3 One Zone-IA: Tek AZ + kritik olmayan veri → en ucuz IA.

S3 Intelligent-Tiering: Erişim davranışını bilinmiyor → otomatik optimizasyon.

Glacier Instant Retrieval: Arşiv ama anında erişim gerekiyor.

Glacier Flexible Retrieval: Arşiv + dakikalık gecikme → çok ucuz.

Glacier Deep Archive: Çok uzun süre saklanan arşiv → saatlik gecikme, en ucuz sınıf.

QUESTION 2

A company hosts an application on multiple Amazon EC2 instances. The application processes messages from an Amazon SQS queue, writes to an Amazon RDS table, and deletes the message from the queue. Occasional duplicate records are found in the RDS table. The SQS queue does not contain any duplicate messages.

What should a solutions architect do to ensure messages are being processed once only?

- A. Use the CreateQueue API call to create a new queue.
- B. Use the AddPermission API call to add appropriate permissions.
- C. Use the ReceiveMessage API call to set an appropriate wait time.
- D. Use the ChangeMessageVisibility API call to increase the visibility timeout.

Soru:

Bir şirket, bir uygulamayı birden fazla Amazon EC2 örneğinde barındırmaktadır. Uygulama, bir Amazon SQS kuyruğundan mesajları işler, bir Amazon RDS tablosuna yazar ve mesajı kuyrukta siler. RDS tablosunda ara sıra yinelenen kayıtlar bulunmuştur. SQS kuyruğunda yinelenen mesaj yoktur.

Mesajların yalnızca bir kez işlenmesini sağlamak için bir çözüm mimarı ne yapmalıdır?

- A. CreateQueue API çağrısını kullanarak yeni bir kuyruk oluşturun.
- B. AddPermission API çağrısını kullanarak uygun izinleri ekleyin.
- C. ReceiveMessage API çağrısını kullanarak uygun bir bekleme süresi ayarlayın.
- D. ChangeMessageVisibility API çağrısını kullanarak görünürlük zaman aşımını artırın.

Soru Analizi:

Uygulama akışı:

1. EC2, SQS'ten mesaj alıyor.
2. Mesajı işliyor ve RDS'e yazıyor.

3. Mesajı SQS'ten siliyor.

**Problemler: RDS tablosunda ara sıra duplicate (çift) kayıtlar oluşuyor.
SQS kuyruğunda duplicate mesaj yok.**

Bu neyi gösterir?

👉 Mesaj **tek** defa geliyor,
ama uygulama **aynı mesajı iki kez işliyor**.

Bu nasıl olur?

- Worker mesajı aldı.
- İşlem uzun sürdü (RDS yazma vs).
- **SQS visibility timeout doldu.**
- Mesaj tekrar görünür hale geldi.
- Başka bir worker mesajı yeniden aldı → **duplicate kayıt oluştu**.

Bu AWS'de "classic SQS problemidir".

✓ Çözüm = Visibility Timeout'u artırmak

Mesaj işlenirken, ikinci bir worker'ın o mesajı tekrar almasını engeller.

Seçenek Analizi:

✓ D. ChangeMessageVisibility – Visibility timeout artır

- Visibility timeout mesajın gizli kalma süresidir.
- İşlem süresi bu süreden uzunsa mesaj tekrar görünür olur.
- Bir başka worker aynı mesajı tekrar alır → **duplicate kayıt oluşur**.

Visibility timeout = işlem süresinden uzun olmalı.

✓ SQS message duplication sorunlarına klasik çözüm

✓ Least effort

✓ Kod değişikliği gerekmeyez (SQS ayarıdır)

✗ A. CreateQueue API – Yeni bir kuyruk oluştur

Hiç alakası yok.

- Duplicate problemi yeni kuyruk oluşturmakla çözülmmez.

✗ AddPermission API – İzin ekle

İzin duplicate'i çözmez.

Bu API sadece kimlerin kuyruk üzerinde işlem yapabileceğini ayarlar.

C. ReceiveMessage API – Bekleme süresi artır

Bu **long polling (WaitTimeSeconds)** ayarıdır.

- Long polling duplicate’ı çözmez.
- Sadece boş cevapları azaltır ve maliyeti düşürür.

Duplicate’in sebebi **processing time > visibility timeout** olduğundan bu işe yaramaz.

Sonuç

RDS’deki duplicate kayıtların sebebi, mesaj işlenirken SQS visibility timeout’un dolması ve mesajın yeniden başka bir worker tarafından alınmasıdır. Bunu önlemek için mesaj işleme süresinden daha uzun bir visibility timeout ayarlanmalıdır.

QUESTION 3

A solutions architect is designing a new hybrid architecture to extend a company's on-premises infrastructure to AWS. The company requires a highly available connection with consistent low latency to an AWS Region. The company needs to minimize costs and is willing to accept slower traffic if the primary connection fails.

What should the solutions architect do to meet these requirements?

- A. Provision an AWS Direct Connect connection to a Region. Provision a VPN connection as a backup if the primary Direct Connect connection fails.
- B. Provision a VPN tunnel connection to a Region for private connectivity. Provision a second VPN tunnel for private connectivity and as a backup if the primary VPN connection fails.
- C. Provision an AWS Direct Connect connection to a Region. Provision a second Direct Connect connection to the same Region as a backup if the primary Direct Connect connection fails.
- D. Provision an AWS Direct Connect connection to a Region. Use the Direct Connect failover attribute from the AWS CLI to automatically create a backup connection if the primary Direct Connect connection fails.

Soru:

Bir çözüm mimarı, bir şirketin şirket içi altyapısını AWS’ye genişletmek için yeni bir hibrit mimari tasarlıyor. Şirket, bir AWS Bölgesine **yüksek erişilebilirliğe sahip, tutarlı düşük gecikmeli** bir bağlantı istemektedir. Şirket, maliyetleri en aza indirmek istiyor ve birincil bağlantı başarısız olursa **daha yavaş trafik kabul edilebilir**.

Bu gereksinimleri karşılamak için çözüm mimarı ne yapmalıdır?

- A. Bir bölgeye AWS Direct Connect bağlantısı sağlayın. Birincil Direct Connect bağlantısı başarısız olursa yedek olarak bir VPN bağlantısı sağlayın.
- B. Özel bağlantı için bir bölgeye VPN tüneli sağlayın. Birincil VPN bağlantısı başarısız olursa özel bağlantı ve yedek olarak ikinci bir VPN tüneli sağlayın.
- C. Bir bölgeye AWS Direct Connect bağlantısı sağlayın. Birincil Direct Connect bağlantısı başarısız olursa yedek olarak aynı bölgeye ikinci bir Direct Connect bağlantısı sağlayın.
- D. Bir bölgeye AWS Direct Connect bağlantısı sağlayın. Birincil Direct Connect bağlantısı başarısız olursa AWS CLI'dan Direct Connect failover özelliğini kullanarak otomatik olarak bir yedek bağlantı oluşturun.

Soru Analizi:

Şirket şunları istiyor:

✓ 1. Hybrid connection (on-prem → AWS)

→ Bu genelde **Direct Connect** veya **VPN** ile sağlanır.

✓ 2. Highly available (yüksek erişilebilir)

→ Birincil bağlantı + yedek bağlantı.

✓ 3. Consistent low latency (tutarlı düşük gecikme)

→ Bunu **VPN sağlayamaz**, sadece **Direct Connect** sağlar.

VPN internet üzerinden gider = latency değişkendir.

✓ 4. Maliyet düşük olmalı

→ Tamamen ikinci bir Direct Connect hattı pahalı olur.

→ Ancak **DC primary + VPN backup** maliyet/performans dengesinde en ideal çözümüdür.

✓ 5. Failover olursa daha yavaş trafik kabul edilebilir

→ Bu açıkça diyor ki: "Primary hızlı olsun, yedek yavaş olabilir."

Bu da **Direct Connect + VPN backup** anlamına gelir.

Seçenek Analizi:

A. Direct Connect + VPN backup

- Primary: Direct Connect → düşük latency, kararlı bağlantı
- Backup: VPN → daha yavaş ama ucuz
- Maliyet: ikinci DC'den ucuz
- Availability: Evet

- Sorunun tüm şartlarını karşılıyor

B. VPN + VPN backup

- VPN düşük *latency* sağlamaz, internet üzerinden gider.
- “Consistent low latency” gereksinimine uymaz.
- Ayrıca yüksek bant genişliği genelde olmaz.

C. Direct Connect + Direct Connect backup

Teknik olarak yüksek availability sağlar ancak:

- **Maliyet çok yüksek**, soruda “minimize costs” diyor.
- Ayrıca failover için “daha yavaş kabul edilebilir” denmiş → ikinci DC gereksiz pahalı.

D. Direct Connect failover attribute ile otomatik backup oluşturma

Böyle bir özellik DC’de yoktur.

- Direct Connect otomatik olarak yeni fiziksel bağlantı oluşturur.
- DC failover attribute yalnızca **routing sırasını** belirler.

Sonuç

Direct Connect + VPN backup

- ✓ düşük gecikme
 - ✓ yüksek erişilebilirlik
 - ✓ düşük maliyet
 - ✓ failover durumunda daha yavaş bağlantı kabul edilebilir → VPN uyumlu
-

QUESTION 4

A company is running a business-critical web application on Amazon EC2 instances behind an Application Load Balancer. The EC2 instances are in an Auto Scaling group. The application uses an Amazon Aurora PostgreSQL database that is deployed in a single Availability Zone. The company wants the application to be highly available with minimum downtime and minimum loss of data.

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

- A. Place the EC2 instances in different AWS Regions. Use Amazon Route 53 health checks to redirect traffic. Use Aurora PostgreSQL Cross Region Replication.

- B. Configure the Auto Scaling group to use multiple Availability Zones. Configure the database as Multi-AZ. Configure an Amazon RDS Proxy instance for the database.
- C. Configure the Auto Scaling group to use one Availability Zone. Generate hourly snapshots of the database. Recover the database from the snapshots in the event of a failure.
- D. Configure the Auto Scaling group to use multiple AWS Regions. Write the data from the application to Amazon S3. Use S3 Event Notifications to launch an AWS Lambda function to write the data to the database.

Soru:

Bir şirket, Application Load Balancer arkasında Amazon EC2 örnekleri üzerinde çalışan iş açısından kritik bir web uygulaması çalıştırıyor. EC2 örnekleri bir Auto Scaling grubunda bulunuyor. Uygulama, tek bir Kullanılabilir Bölgeye (AZ) konuşlandırılmış bir **Amazon Aurora PostgreSQL** veritabanı kullanıyor. Şirket, uygulamanın **en az kesinti ve en az veri kaybı ile yüksek erişilebilir** olmasını istiyor.

Hangi çözüm **en az operasyonel çaba** ile bu gereksinimleri karşılar?

- A. EC2 örneklerini farklı AWS Bölgelerine yerleştirin. Trafiği yönlendirmek için Amazon Route 53 sağlık kontrolleri kullanın. Aurora PostgreSQL için Bölge Ötesi Replikasyon (Cross-Region Replication) kullanın.
- B. Auto Scaling grubunu birden çok Kullanılabilir Bölgesi (Availability Zones) kullanacak şekilde yapılandırın. Veritabanını **Multi-AZ** olarak yapılandırın. Veritabanı için bir **Amazon RDS Proxy** örneği yapılandırın.
- C. Auto Scaling grubunu tek bir Kullanılabilir Bölgesi kullanacak şekilde yapılandırın. Veritabanının saatlik snapshot'larını oluşturun. Bir arıza durumunda veritabanını bu snapshot'lardan geri yükleyin.
- D. Auto Scaling grubunu birden çok AWS Bölgesini kullanacak şekilde yapılandırın. Uygulamadan gelen verileri Amazon S3'e yazın. S3 Olay Bildirimleri ile bir AWS Lambda fonksiyonu başlatarak verileri veritabanına yazdırın.

Soru Analizi:

Amaç:

Uygulama **yüksek erişilebilir** olacak, **minimum kesinti, minimum veri kaybı** yaşayacak ve **en az operasyonel çaba** gerektirecek.

Mevcut yapı:

- EC2 + ALB
- Auto Scaling Group

- **Aurora PostgreSQL tek AZ** → Bu en büyük risk: tek AZ = felaket durumunda kesinti ve veri kaybı riski
- Şirket yüksek oranda kritik bir uygulama çalıştırıyor

Hedef: Hem uygulama hem de veritabanı **Multi-AZ** olmalı.

Aurora zaten yerleşik olarak yüksek erişilebilirlik özellikleri sunar (Multi-AZ, failover vb.). Az operasyonel çaba = yönetilen hizmetleri etkinleştirmek, ek mimari karmaşıklık oluşturmamak.

Seçenek Analizi:

B. ASG'yi Multi-AZ yap + Veritabanını Multi-AZ yap + RDS Proxy kullan

- EC2 tarafı Multi-AZ → yüksek erişilebilirlik
- Aurora'yı Multi-AZ yapmak → Aurora bunu otomatik olarak yönetir → veri kaybı minimum, failover hızlı
- RDS Proxy → bağlantı yönetimi, failover sırasında kesinti süresini azaltır
- Tüm bileşenler AWS tarafından yönetilir

Az operasyonel çaba

Veri kaybını minimuma indirir

En yüksek erişilebilirlik sağlar

En az operasyonel yük

→ **Bu en mantıklı çözüm.**

A. EC2'leri farklı bölgelere koy + Route 53 health checks + Aurora Cross-Region Replication

- Çok karmaşık.
- Cross-region replication **asenkron** → **veri kaybı** olabilir.
- Operasyonel yük yüksek.
Minimum kesinti ve minimum veri kaybı için uygun değil.
Çok fazla operasyonel iş.

C. ASG tek AZ + snapshot geri yükleme

- Tek AZ = riskli
- Snapshot geri yükleme **çok yavaş**, operasyonel olarak manuel iş gerektirir
- Snapshot → veri kaybı kaçınılmaz
Minimum downtime sağlanamaz

Minimum veri kaybı sağlanamaz
Operasyonel çaba yüksek

D. Multi-region ASG + S3 → Lambda → DB yazdırma

- Çok karmaşık bir sistem
- Uygulama verisini S3 üzerinden dolaylı yoldan DB'ye yazmak mantıksız
- Multi-region replicate karmaşık + yavaş
Minimum operasyon yükü değil
Sorunun gereksiz karmaşık bir çözümü

SONUÇ

En doğru ve amaca en uygun çözüm: B seçeneği

Sebep:

- Hem uygulama hem veritabanı Multi-AZ çalışır
- Aurora Multi-AZ failover ile **çok düşük kesinti**
- RDS Proxy ile bağlantı yönetimi daha hızlı ve kesintisiz
- En düşük operasyonel çaba (sadece Multi-AZ etkinleştirmek yeterli)

QUESTION 5

A company's HTTP application is behind a Network Load Balancer (NLB). The NLB's target group is configured to use an Amazon EC2 Auto Scaling group with multiple EC2 instances that run the web service. The company notices that the NLB is not detecting HTTP errors for the application. These errors require a manual restart of the EC2 instances that run the web service. The company needs to improve the application's availability without writing custom scripts or code.

What should a solutions architect do to meet these requirements?

- A. Enable HTTP health checks on the NLB, supplying the URL of the company's application.
- B. Add a cron job to the EC2 instances to check the local application's logs once each minute. If HTTP errors are detected, the application will restart.
- C. Replace the NLB with an Application Load Balancer. Enable HTTP health checks by supplying the URL of the company's application. Configure an Auto Scaling action to replace unhealthy instances.

D. Create an Amazon Cloud Watch alarm that monitors the UnhealthyHostCount metric for the NLB. Configure an Auto Scaling action to replace unhealthy instances when the alarm is in the ALARM state.

Soru:

Bir şirketin HTTP uygulaması bir **Network Load Balancer (NLB)** arkasında çalışmaktadır. NLB'nin hedef grubu, web hizmetini çalıştıran birden fazla **Amazon EC2 Auto Scaling** grubu örneği içeren bir yapı ile yapılandırılmıştır.

Şirket, uygulamada oluşan **HTTP hatalarının NLB tarafından tespit edilmediğini** fark eder. Bu hatalar, web hizmetini çalıştıran EC2 örneklerinin manuel olarak yeniden başlatılmasını gerektirmektedir. Şirket, **özel betikler veya kod yazmadan**, uygulamanın ejf6c o7 kullanılabilirliğini artırmak istemektedir.

Bu gereksinimleri karşılamak için bir çözüm mimarı ne yapmalıdır?

- A. NLB üzerinde HTTP sağlık kontrollerini etkinleştirin ve şirketin uygulamasının URL'sini sağlayın.
- B. EC2 örneklerine, yerel uygulama günlüklerini her dakika kontrol eden bir cron job ekleyin. HTTP hataları tespit edilirse uygulama yeniden başlatılır.
- C. NLB'yi bir **Application Load Balancer (ALB)** ile değiştirin. Uygulamanın URL'si ile HTTP sağlık kontrollerini etkinleştirin. Sağlıksız örnekleri değiştirmek için Auto Scaling işlemi yapılmalıdır.
- D. NLB için **UnhealthyHostCount** metriğini izleyen bir Amazon CloudWatch alarmı oluşturun. Alarm **ALARM** durumuna geçtiğinde sağlıksız örnekleri değiştirmek üzere Auto Scaling işlemi yapılmalıdır.

Soru Analizi:

Durum

- Uygulama HTTP tabanlı.
- Load balancer **Network Load Balancer (NLB)**.
- NLB **Layer 4 (TCP)** seviyesinde çalışır → **HTTP hatalarını tespit edemez**.
- HTTP hataları olduğunda EC2 instance manuel restart ediliyor.
- Şirket **kodu değiştirmeden, script yazmadan** çözüm istiyor.
- Amaç: **yüksek erişilebilirlik**, hatalı instance'ların otomatikelenmesi.

Seçenek Analizi:

- C. NLB'yi ALB ile değiştirmek + HTTP health check + Auto Scaling

- ALB, **application-layer (Layer 7)** health check yapabilir → HTTP hatalarını tespit eder.
- Sağlıksız EC2 instance'ları Auto Scaling otomatik olarak değiştirir.
- Ek kod/betik gerekmez.
- Minimum operasyonel yük ile yüksek erişilebilirlik sağlar.

A. NLB'de HTTP health check etkinleştirilmek

NLB yalnızca TCP/UDP/SSL health check destekler.

HTTP health check desteklemez → Bu seçenek teknik olarak mümkün değildir.

B. Cron job ile log kontrol edip restart etmek

- Kod veya script yazmak gerekiyor → Şirket istemiyor.
- Auto Scaling ile entegre değil, manuel instance restart mantığını otomatiğe dökse de **ölçeklendirme mantığına uymaz**.

D. CloudWatch UnhealthyHostCount alarmı + Auto Scaling

- NLB HTTP hatalarını tespit edemediği için **UnhealthyHostCount artmaz**.
- Sorunun temel sebebi zaten NLB'nin HTTP hatasını görememesidir.

SONUÇ

NLB → ALB'ye geçiş

+

HTTP health check

+

Auto Scaling ile sağlıksız instance'ların otomatik değiştirilmesi

Bu, sorunun tüm gereksinimlerini karşılayan **tek çalışabilir ve en az operasyonel yük gerektiren** çözümüdür.

QUESTION 6

A company runs a shopping application that uses Amazon DynamoDB to store customer information. In case of data corruption, a solutions architect needs to design a solution that meets a recovery point objective (RPO) of 15 minutes and a recovery time objective (RTO) of 1 hour.

What should the solutions architect recommend to meet these requirements?

- A. Configure DynamoDB global tables. For RPO recovery, point the application to a different AWS Region.

- B. Con gure DynamoDB point-in-time recovery. For RPO recovery, restore to the desired point in time.
- C. Export the DynamoDB data to Amazon S3 Glacier on a daily basis. For RPO recovery, import the data from S3 Glacier to DynamoDB.
- D. Schedule Amazon Elastic Block Store (Amazon EBS) snapshots for the DynamoDB table every 15 minutes. For RPO recovery, restore the DynamoDB table by using the EBS snapshot.

Soru:

Bir şirket, müşteri bilgilerini depolamak için Amazon DynamoDB kullanan bir alışveriş uygulaması çalışmaktadır. Veri bozulması (data corruption) durumunda, bir çözüm mimarının **15 dakikalık bir RPO (Recovery Point Objective – Kurtarma Noktası Hedefi)** ve **1 saatlik bir RTO (Recovery Time Objective – Kurtarma Süresi Hedefi)** sağlayan bir çözüm tasarlaması gerekmektedir.

Bu gereksinimleri karşılamak için çözüm mimarı ne önermelidir?

- A. DynamoDB global tabloları yapılandırın. RPO kurtarma için uygulamayı farklı bir AWS bölgesine yönlendirin.
- B. DynamoDB "point-in-time recovery" özelliğini yapılandırın. RPO kurtarma için istenen zamana geri yükleyin.
- C. DynamoDB verisini günlük olarak Amazon S3 Glacier'a aktarın. RPO kurtarma için veriyi Glacier'dan DynamoDB'ye geri aktarın.
- D. DynamoDB tablosu için her 15 dakikada bir Amazon Elastic Block Store (Amazon EBS) snapshot'ları zamanlayın. RPO kurtarma için tabloyu EBS snapshot'tan geri yükleyin.

Soru Analizi:

Şirketin iki kritik metrik gereksinimi var:

RPO = 15 dakika

- Veri bozulduğunda en fazla 15 dakika geriye dönebilmeli.
- Yani **sürekli veya en az 15 dakikalık periyotlarla kayıt yapılan bir yedekleme** gereklidir.

RTO = 1 saat

- Sorun olduğunda **1 saat içinde tablo tekrar çalışır durumda olmalı**.
- Yani restore işlemi hızlı olmalıdır.

Risk türü:

Veri bozulması (data corruption) → Bu durumda çözümün:

- Veriyi *anında* çoğaltmaması,
- “Geri zamanlama” yapabilmesi gereklidir.

Bu detay çok önemlidir, çünkü bazı seçenekler başarısız olur.

Seçenek Analizi:

B Seçeneği – Point-in-Time Recovery (PITR)

✓ Ne yapar?

- Sürekli yedekleme (saniyelik) sağlar.
- 35 gün içinde **herhangi bir saniyeye geri dönebilirsin**.

✓ Neden ideal?

- Veri bozulması olduğunda: “Bozulmadan hemen önceki saniyeye” dönebilirsin.
Restore süresi hızlıdır → RTO \leq 1 saat çoğu pratik senaryoda rahatça karşılanır.
- Ek yönetim maliyeti yok (full-managed).

UYARLAMA:

Gereksinim Karşılıyor mu?

RPO 15 dk ✓ saniyelik RPO

RT0 1 saat ✓ restore işlemi hızlı

Operasyonel yük ✓ minimum

Bu seçenek hem teknik hem de AWS sınav mantığı açısından **altın standarttır**.

A Seçeneği – Global Tables

✓ Ne yapar?

- Multi-region aktif-aktif replikasyon sağlar.
- Yüksek erişilebilirlik ve düşük latency için idealdir.

X Neden uygun değil?

- Veri bozulduysa → *bozuk veri tüm bölgelere anında replikasyon olur*.
- Yani global table, **corruption'u daha geniş alana yayar**.

UYARLAMA:

Gereksinim Karşılıyor mu?

RPO 15 dk hayır – corruption'u durduramaz

RTO 1 saat irrelevant – geri dönüş mekanizması yok

Operasyonel yük yüksek

✗ C Seçeneği – Günlük S3 Glacier Export

✓ Ne sağlar?

- Soğuk depolama yedeği.

✗ Neden uygun değil?

- Günlük yedek → RPO = *en iyi ihtimalle* 24 saat olur.
- Glacier restore işlemleri dakikalar-saatler sürebilir → RTO 1 saati aşabilir.

UYARLAMA:

Gereksinim Karşılıyor mu?

RPO 15 dk 24 saat --> kabul edilemez

RTO 1 saat uzun restore süresi

Operasyonel yük yüksek

✗ D Seçeneği – EBS Snapshot ile DynamoDB Yedeklemek

✗ Neden tamamen yanlış?

- DynamoDB **EBS tabanlı bir servis değildir** → EBS snapshot **kullanamazsınız**.
- Teknik olarak mümkün olmayan bir işlem.
- AWS sınavlarında sıkça gelen “tamamen yanlış seçenek” tipidir.

UYARLAMA:

Gereksinim Karşılıyor mu?

Mantıksal uyum DynamoDB ≠ EBS

RPO–RTO uygulanamaz

Operasyonel yük anlamsız

🎯 SONUÇ

✳️ GENEL KARŞILAŞTIRMA TABLOSU (Yeni Perspektif)

Seçenek	Data Corruption'a Karşı Korur mu?	RPO 15 dk	RTO 1 saat	Teknik Olarak Doğru mu?	Operasyonel Yük
A – Global Tables	✗ bozuk veriyi çoğaltır	✗	✗	✓	Orta
B – PITR	✓ Evet	✓ saniyelik	✓ hızlı restore	✓	Çok düşük
C – Glacier Export	✗ günlük yedek	✗	✗	✓	Yüksek
D – EBS Snapshot	✗ mümkün değil	✗	✗	✗	-

- Sorunun özü **veri bozulması + geri dönüş gereksinimidir.**
- Böylesi durumlarda *aktif replikasyon çözümleri (Global Tables)* işe yaramaz, tam tersine zararlı olabilir.
- En uygun çözüm **zamanı geri alma** yeteneğine sahip olan **DynamoDB PITR**'dır.

QUESTION 7

A company runs a photo processing application that needs to frequently upload and download pictures from Amazon S3 buckets that are located in the same AWS Region. A solutions architect has noticed an increased cost in data transfer fees and needs to implement a solution to reduce these costs.

How can the solutions architect meet this requirement?

- Deploy Amazon API Gateway into a public subnet and adjust the route table to route S3 calls through it.
- Deploy a NAT gateway into a public subnet and attach an endpoint policy that allows access to the S3 buckets.

C. Deploy the application into a public subnet and allow it to route through an internet gateway to access the S3 buckets.

D. Deploy an S3 VPC gateway endpoint into the VPC and attach an endpoint policy that allows access to the S3 buckets.

Soru:

Bir şirket, aynı AWS Bölgesinde bulunan Amazon S3 kovalarına (buckets) sık sık fotoğraf yükleyip indirmesi gereken bir fotoğraf işleme uygulaması çalıştırıyor. Bir çözüm mimarı veri aktarım ücretlerindeki artışı fark etti ve bu maliyeti azaltmak için bir çözüm uygulamak istiyor.

Bu gereksinimi nasıl karşılamalıdır?

A. Amazon API Gateway’ı bir public alt ağa (public subnet) dağıtın ve S3 çağrılarını bunun üzerinden yönlendirmek üzere route tablosunu ayarlayın.

B. Bir NAT Gateway’ı public alt ağa dağıtın ve S3 kovalarına erişime izin veren bir endpoint politikası ilişirin.

C. Uygulamayı public alt ağa dağıtın ve S3 kovalarına erişmek için internet gateway üzerinden yönlendirilmesine izin verin.

D. VPC’ye bir S3 VPC gateway endpoint dağıtın ve S3 kovalarına erişime izin veren bir endpoint politikası ilişirin.

Soru Analizi:

Bir fotoğraf işleme uygulaması var ve:

- **S3 ile aynı bölgede çalışıyor**, yani internet üzerinden gitmesine gerek yok.
- **S3’e sık sık upload/download yapıyor**.
- **Data transfer cost (veri aktarım maliyeti) arttı**.
- Çözüm mimarı bu maliyetleri **azaltmak** istiyor.

Seçenek Analizi:

D. S3 VPC Gateway Endpoint kurmak

AWS’de **aynı bölgedeki S3 erişiminde ücret almayan tek yöntem**:

👉 **S3 VPC Gateway Endpoint**

Bu endpoint ile trafik **AWS backbone içinde** kalır, internet veya NAT gateway üzerinden geçmez → **data transfer ücreti sıfırlanır**.

Doğru ve tek maliyet düşürücü seçenek.

- Aynı bölgedeki S3 trafik maliyetini **0'a düşürür**.

- NAT, Internet Gateway veya public internet kullanmaz.
- Ek ücret yoktur — gateway endpoint **ücretsizdir** (yalnızca istek için standart S3 ücretleri geçerlidir).
- Security policy eklenebilir.

AWS'nin en çok önerdiği yöntemdir.

A. API Gateway üzerinden yönlendirme

Tamamen yanlış.

- API Gateway S3 erişimi için uygun değildir.
- Üstüne API Gateway ücretleri daha da maliyeti artırır.

B. NAT Gateway + endpoint policy

Maliyeti azaltmaz, aksine artırır.

- NAT Gateway üzerinden S3'e erişim **ücretlidir** (GB başına ücret).
- NAT Gateway'in **saatlik ücreti ve trafik ücreti** vardır → pahalı.

C. Public subnet + internet gateway üzerinden erişim

Hatalı ve maliyeti azaltmaz.

- S3'e internet gateway üzerinden erişimde **data transfer ücreti alınır**.
- Public subnet kullanmak güvenlik açısından da kötü bir uygulamadır.

SONUÇ

Doğru Cevap: D — S3 VPC Gateway Endpoint kullanmak

Bu yöntem S3 trafik ücretlerini **tamamen ortadan kaldırır** ve en doğru, en düşük maliyetli çözümüdür.

S3 ile uygulama:

- Aynı region içinde ise
- Uygulama VPC içinde çalışıyorsa
- Trafik çok ise

Kesin çözüm: **S3 Gateway VPC Endpoint**

Neden?

- İnternete çıkmaz

- NAT kullanmaz
 - Ekstra trafik ücreti yok
 - Endpoint ücretsizdir
 - En hızlı ve en güvenli yol
- ✓ S3 ile çok trafik varsa
✓ Aynı bölgedeyse
✓ Maliyet patladıysa

👉 **Her zaman S3 Gateway Endpoint kullanılır.**

QUESTION 8

A company recently launched Linux-based application instances on Amazon EC2 in a private subnet and launched a Linux-based bastion host on an Amazon EC2 instance in a public subnet of a VPC. A solutions architect needs to connect from the on-premises network, through the company's internet connection, to the bastion host, and to the application servers. The solutions architect must make sure that the security groups of all the EC2 instances will allow that access.

Which combination of steps should the solutions architect take to meet these requirements? (Choose two.)

- A. Replace the current security group of the bastion host with one that only allows inbound access from the application instances.
- B. Replace the current security group of the bastion host with one that only allows inbound access from the internal IP range for the company.
- C. Replace the current security group of the bastion host with one that only allows inbound access from the external IP range for the company.
- D. Replace the current security group of the application instances with one that allows inbound SSH access from only the private IP address of the bastion host.
- E. Replace the current security group of the application instances with one that allows inbound SSH access from only the public IP address of the bastion host.

Soru:

Bir şirket, özel bir alt ağda (private subnet) Linux tabanlı uygulama instance'larını Amazon EC2 üzerinde yeni başlattı ve bir VPC'nin genel alt ağında (public subnet) Linux tabanlı bir bastion host'u Amazon EC2 instance'ı olarak başlattı. Bir çözüm mimarının şirketin **on-premises ağına**, şirketin **internet bağlantısı üzerinden**, **bastion host'a** ve

ardından **uygulama sunucularına** bağlanması gerekiyor. Çözüm mimarı, tüm EC2 instance'larının güvenlik gruplarının bu erişime izin vereceğinden emin olmalıdır.

Bu gereksinimleri karşılamak için çözüm mimarı hangi adım kombinasyonlarını atmalıdır? (İki tanesini seçin.)

- A. Bastion host'un mevcut güvenlik grubunu yalnızca uygulama instance'larından gelen inbound erişime izin veren bir güvenlik grubuyla değiştirin.
- B. Bastion host'un mevcut güvenlik grubunu yalnızca şirketin **İç IP aralığından (internal IP range)** gelen inbound erişime izin veren bir güvenlik grubuyla değiştirin.
- C. Bastion host'un mevcut güvenlik grubunu yalnızca şirketin **dış IP aralığından (external IP range)** gelen inbound erişime izin veren bir güvenlik grubuyla değiştirin.
- D. Uygulama instance'larının mevcut güvenlik grubunu yalnızca bastion host'un **özel IP adresinden (private IP)** gelen inbound SSH erişimine izin veren bir güvenlik grubuyla değiştirin.
- E. Uygulama instance'larının mevcut güvenlik grubunu yalnızca bastion host'un **genel IP adresinden (public IP)** gelen inbound SSH erişimine izin veren bir güvenlik grubuyla değiştirin.

Soru Analizi:

Senaryo:

- Uygulama sunucuları **private subnet**'te → Direkt internetten erişilemez.
- Bastion host **public subnet**'te → İnternet üzerinden SSH ile erişim yapılır.
- On-premises ağdan → İnternet çıkışları üzerinden → Bastion'a bağlanılacak.
- Bastion üzerinden → Private EC2'lere SSH yapılacak.

Çözüm mimarı **security group'ları** doğru ayarlamak zorunda.

Önemli Gereksinimler:

1. On-premises dış IP'den → Bastion host'a SSH *izin verilmeli*
2. Bastion host'un private IP'sinden → Uygulama sunucularına SSH *izin verilmeli*

Bu iki şart sağlanmazsa bağlantı kurulamıyor.

Seçenek Analizi:

- C. Bastion SG sadece şirketin **external IP range'inden** erişime izin versin

Bastion host'a giriş *internet üzerinden* yapılır, bu nedenle:

- On-prem NAT çıkışları → **company's external public IP range**

- Bastion SG → inbound SSH (22) sadece bu IP aralığından

Bu tam olarak doğru yapılandırmadır.

 **D. Application instance SG: inbound SSH sadece Bastion'un private IP'sinden gelsin**

Private EC2 instance'ları:

- Internetten erişilemez
- Bastion üzerinden SSH ulaşılmalıdır
- Bu erişim bastion'un **private IP'sinden** gelir

Bastion host → private subnet içindeki EC2'lere *iç ağ üzerinden* erişir.

Doğru kural: Private IP → SSH → Application EC2

 **A. Bastion SG sadece uygulama instance'larına izin versin**

Bastion'a internetten erişim gereklidir → Sadece application EC2'lerden erişime izin verilirse SSH yapılamaz.

 **B. Bastion SG sadece şirketin internal IP range'inden erişime izin versin**

Internal IP'ler on-prem LAN içidir, ama internet üzerinden NAT çıkışını yapılınlca **external IP** görünür.

Doğru olan internal değil **external IP aralığıdır**.

 **E. Application instance SG: inbound SSH public IP of Bastion**

Bastion'un public IP'si VPC içinden görünmez →

Private EC2'ler *bastion'un public IP'sini asla görmez*.

Bu nedenle erişim private IP'den olmalıdır → D doğru, E yanlış.

 **Sonuç**

 **ÖZET**

İhtiyaç

Doğru Ayar

On-prem → Bastion SSH Bastion SG → **company external IP range**

Bastion → Private EC2 SSH Application SG → **bastion private IP**

Bu iki kural olmadan bağlantı zinciri tamamlanmaz.

QUESTION 9

A solutions architect is designing a two-tier web application. The application consists of a public-facing web tier hosted on Amazon EC2 in public subnets. The database tier consists of Microsoft SQL Server running on Amazon EC2 in a private subnet. Security is a high priority for the company.

How should security groups be configured in this situation? (Choose two.)

- A. Configure the security group for the web tier to allow inbound traffic on port 443 from 0.0.0.0/0.
- B. Configure the security group for the web tier to allow outbound traffic on port 443 from 0.0.0.0/0.
- C. Configure the security group for the database tier to allow inbound traffic on port 1433 from the security group for the web tier.
- D. Configure the security group for the database tier to allow outbound traffic on ports 443 and 1433 to the security group for the web tier.
- E. Configure the security group for the database tier to allow inbound traffic on ports 443 and 1433 from the security group for the web tier.

Soru:

Bir çözüm mimarı iki katmanlı bir web uygulaması tasarlıyor. Uygulama, genel alt ağlarda (public subnets) barındırılan Amazon EC2 üzerinde çalışan herkese açık bir web katmanından oluşmaktadır. Veritabanı katmanı ise özel bir alt ağda (private subnet) Amazon EC2 üzerinde çalışan Microsoft SQL Server'dan oluşmaktadır. Güvenlik şirket için yüksek önceliklidir.

Bu durumda güvenlik grupları nasıl yapılandırılmalıdır? (İki tanesini seçin.)

- A. Web katmanı için güvenlik grubunu, 0.0.0.0/0'dan 443 numaralı port üzerinde gelen trafiğe izin verecek şekilde yapılandırın.
- B. Web katmanı için güvenlik grubunu, 0.0.0.0/0'a 443 numaralı port üzerinden giden trafiğe izin verecek şekilde yapılandırın.
- C. Veritabanı katmanı için güvenlik grubunu, web katmanının güvenlik grubundan 1433 numaralı port üzerinde gelen trafiğe izin verecek şekilde yapılandırın.
- D. Veritabanı katmanı için güvenlik grubunu, web katmanının güvenlik grubuna 443 ve 1433 numaralı portlar üzerinden giden trafiğe izin verecek şekilde yapılandırın.
- E. Veritabanı katmanı için güvenlik grubunu, web katmanının güvenlik grubundan 443 ve 1433 numaralı portlar üzerinde gelen trafiğe izin verecek şekilde yapılandırın.

Soru Analizi:

Senaryonun özü

- Web tier → public subnet

- Database tier → private subnet
- Web katmanı internete açık olmalı
- Database kesinlikle internete kapalı
- Web → DB bağlantısı sadece gerekli portta (SQL Server = 1433) olmalı
- Güvenlik gruplarında “minimum permission” prensibi uygulanmalı

Bu yüzden kurallar sadece **gerekенёне и го порту** izin vermelidir. Her gereksiz izin ekstra saldırıcı vektördür.

Seçenek Analizi:

A. Web SG inbound 443 from 0.0.0.0/0

✓ Olması gereken davranış

Web sunucusu dış dünyaya HTTPS hizmeti verdiği için tüm internete 443 üzerinden açık olmalıdır.

Bu olmadan web uygulaması çalışmaz.

C. DB SG inbound 1433 from web SG

✓ Bu, iki katmanlı mimarinin kalbidir

DB yalnızca web sunucularından erişim almalıdır.

Kaynağın SG olarak belirtilmesi IP'den daha güvenli, çünkü

- Ölçeklendirilmiş IP değişse bile erişim yetkisi bozulmaz
- sadece web katmanı sunucuları erişebilir

Bu en kritik ve en güvenli seçimdir.

B. Web SG outbound 443 to 0.0.0.0/0

Asıl gereksinimle ilgisiz

Web sunucusunun internete çıkışması zorunlu değil. Çoğu durumda outbound default olarak allowed zaten. Soru outbound davranışını istemiyor, gereksiz ayrıntı.

D. DB SG outbound 443 ve 1433 to web SG

Trafik yönü ters

DB → Web katmanına giden bir trafik bu mimaride yok.

SQL Server bağlantısı **her zaman web → db** yönlüdür.

DB'nin outbound kurallarına gerekli bir şey ekleme zorunluluğu yok.

E. DB SG inbound 443 ve 1433 from web SG

Gereksiz port açar

DB yalnızca SQL portu olan **1433**'e ihtiyaç duyar.

443 (HTTPS) DB katmanında kullanılan bir port değildir ve güvenlik açığı oluşturur.

Sonuç:

- **A:** Public web sunucusunun interne HTTPS üzerinden açık olması gerekiyor.
 - **C:** Database'in yalnızca web katmanından ve sadece SQL portundan erişim olması gerekiyor.
-

QUESTION 10

A company wants to move a multi-tiered application from on premises to the AWS Cloud to improve the application's performance. The application consists of application tiers that communicate with each other by way of RESTful services. Transactions are dropped when one tier becomes overloaded. A solutions architect must design a solution that resolves these issues and modernizes the application.

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

- A. Use Amazon API Gateway and direct transactions to the AWS Lambda functions as the application layer. Use Amazon Simple Queue Service (Amazon SQS) as the communication layer between application services.
- B. Use Amazon CloudWatch metrics to analyze the application performance history to determine the servers' peak utilization during the performance failures. Increase the size of the application server's Amazon EC2 instances to meet the peak requirements.
- C. Use Amazon Simple Notification Service (Amazon SNS) to handle the messaging between application servers running on Amazon EC2 in an Auto Scaling group. Use Amazon CloudWatch to monitor the SNS queue length and scale up and down as required.
- D. Use Amazon Simple Queue Service (Amazon SQS) to handle the messaging between application servers running on Amazon EC2 in an Auto Scaling group. Use Amazon CloudWatch to monitor the SQS queue length and scale up when communication failures are detected.

Soru:

Bir şirket, performansı artırmak için çok katmanlı bir uygulamayı şirket içinden (on-premises) AWS Cloud'a taşımak istiyor. Uygulama, birbirleriyle RESTful servisler aracılığıyla iletişim kuran uygulama katmanlarından oluşmaktadır. Katmanlardan biri aşırı yüklenliğinde işlemler düşmektedir. Bir çözümler mimarı, bu sorunları çözen ve uygulamayı modernize eden bir çözüm tasarlamalıdır.

Aşağıdaki çözümlerden hangisi bu gereksinimleri karşılar ve **en fazla operasyonel verimlilik** sağlar?

- A. Uygulama katmanı olarak AWS Lambda işlevlerine işlemleri yönlendirmek için Amazon API Gateway kullanın. Uygulama servisleri arasındaki iletişim katmanı için Amazon Simple Queue Service (Amazon SQS) kullanın.
- B. Amazon CloudWatch metriklerini kullanarak uygulama performans geçmişini analiz edin ve performans hataları sırasında sunucuların en yüksek kullanım oranını belirleyin. Uygulama sunucularının Amazon EC2 örnek boyutlarını bu en yüksek gereksinimleri karşılayacak şekilde artırın.
- C. Amazon Simple Notification Service (Amazon SNS) kullanarak Amazon EC2 üzerinde çalışan ve bir Auto Scaling grubunda bulunan uygulama sunucuları arasındaki mesajlaşmayı yönetin. SNS kuyruk uzunluğunu izlemek için Amazon CloudWatch kullanın ve gerektiğinde ölçeklendirin.
- D. Amazon Simple Queue Service (Amazon SQS) kullanarak Amazon EC2 üzerinde çalışan ve bir Auto Scaling grubunda bulunan uygulama sunucuları arasındaki mesajlaşmayı yönetin. SQS kuyruk uzunluğunu izlemek için Amazon CloudWatch kullanın ve iletişim hataları tespit edildiğinde ölçüği artırın.

Soru Analizi:

Şirket çok katmanlı (multi-tier) bir uygulamayı AWS'e taşıyor.
Katmanlar **RESTful servislerle** birbirile konuşturuyor.

Sorun:

Bir katman aşırı yüklenince işlemler düşüyor → bu da şu anlama gelir:

- **Katmanlararası iletişim senkron**
- Arka uç yavaşsa istekler düşüyor
- Bu, **kuyruklama olmaması** anlamına gelir.

Amaç:

- Overload (yüklenme) durumunda işlemlerin kaybolmasını engellemek
- Performansı artırmak
- Uygulamayı modernize etmek
- **En operasyonel verimli (least operational effort)** çözümü seçmek

Bu tür mimari sorunlar için en doğru çözüm:

- ➡ **Katmanları gevşek bağlı hale getirmek (decoupling)**
- ➡ **Messaging/kuyruklama (SQS)**
- ➡ Gerektiğinde sunucusuz modernleşme (Lambda + API Gateway)

Seçenek Analizi:

A. API Gateway + Lambda + SQS kullanmak

Analiz:

- Sunucuları ortadan kaldırır → **sunucusuz mimari**
- Katmanları tamamen **decouple** eder
- Overload durumunda işlemler **kaybolmaz**, SQS kuyrukta birikir
- Lambda otomatik ölçeklenir → **yük yönetimi mükemmel**
- Operasyonel olarak en düşük maliyetli:
 - Sunucu yönetimi yok
 - Otomatik ölçeklenme
 - AWS'in modern tavsiyesine uygun

Modernization + high-performance + decoupling + low ops

Bu, sorunun istediği her şeye tamamen uyuyor.

B. EC2 instance boyutlarını artırmak

Analiz:

- Sadece “bigger EC2” öneriyor → bu bir “vertical scaling” çözümüdür
- Uygulama hala **senkron**, hala overloaded olabilir
- Modernizasyon sağlamaz
- Overload olduğunda işlem düşme sorunu **çözülmez**
- Operasyonel maliyet **yüksek** (EC2 bakım vs.)

Sadece geçici bir yamadır → gerçek çözüm değildir.

→ **Yanlış**

C. SNS + EC2 Auto Scaling kullanmak

Analiz:

- SNS bir “pub/sub” servistir.
- SNS **kuyruk değildir**, mesaj tutmaz.
- Katmanlar arasında dayanıklı bir kuyruk yapılamaz.
- Aşırı yüklenme durumunda SNS mesajları **kaybolabilir**.

📌 Katmanlar arası REST tabanlı trafiği SNS ile yönetmek uygun değildir.

→ Yanlış

✖ D. SQS + EC2 Auto Scaling kullanmak

Analiz:

- SQS doğru servistir → **decoupling sağlar**
- EC2 Auto Scaling kuyruk uzunluğuna göre ölçeklenebilir
- Modern bir yaklaşım
- Uygulamayı kurtarır ama...

Eksisi:

- Yine EC2 yönetimi var
- A seçeneğindeki kadar operasyonel olarak verimli değildir
- Tam modernizasyon sağlamaz

📌 Doğru bir mimari ama **A'dan daha fazla operasyonel yük** içerir.

→ Mantıklı ama “en operasyonel verimli” değildir

🎯 SONUÇ

(API Gateway + Lambda + SQS = tam modernizasyon, tam decoupling, en düşük operasyonel yük)

QUESTION 11

A company receives 10 TB of instrumentation data each day from several machines located at a single factory. The data consists of JSON files stored on a storage area network (SAN) in an on-premises data center located within the factory. The company wants to send this data to Amazon S3 where it can be accessed by several additional systems that provide critical near-real-time analytics. A secure transfer is important because the data is considered sensitive.

Which solution offers the MOST reliable data transfer?

- AWS DataSync over public internet
- AWS DataSync over AWS Direct Connect
- AWS Database Migration Service (AWS DMS) over public internet
- AWS Database Migration Service (AWS DMS) over AWS Direct Connect

Soru:

Bir şirket, tek bir fabrikada bulunan birkaç makineden **her gün 10 TB enstrümantasyon verisi** almaktadır. Veriler, fabrikanın içindeki şirketin veri merkezinde bulunan bir **SAN (storage area network)** üzerinde **JSON dosyaları** şeklinde depolanmaktadır. Şirket bu verileri, **Amazon S3'e göndermek** istemektedir. Böylece veriler, **kritik ve gerçeğe yakın zamanlı (near-real-time) analizler** yapan birkaç ek sistem tarafından erişilebilir olacaktır. Veri **hassas (sensitive)** kabul edildiği için **güvenli aktarım** önemlidir.

En güvenilir veri aktarımını sağlayacak çözüm hangisidir?

- A. AWS DataSync'i genel internet (public internet) üzerinden kullanmak
- B. AWS DataSync'i AWS Direct Connect üzerinden kullanmak
- C. AWS Database Migration Service (AWS DMS)'i genel internet üzerinden kullanmak
- D. AWS Database Migration Service (AWS DMS)'i AWS Direct Connect üzerinden kullanmak

Soru Analizi:

Şirket **günde 10 TB** veri üretiyor → çok büyük hacim.

Veri **JSON dosyaları**, yani **dosya tabanlı** veri.

Veri **SAN üzerinde**, yani *dosya sistemi* şeklinde.

Amaç: Bu veriyi **güvenli, en güvenilir, yüksek hacmi kaldırabilecek, near-real-time** bir biçimde **Amazon S3'e taşımak**.

Bu, tipik bir:

- ✓ **Büyük dosya aktarımı**
- ✓ Sürekli ve hızlı veri kopyalama
- ✓ S3'e güvenli transfer
- ✓ Network optimize etme ihtiyacı

senaryosudur.

Bu özellikleri karşılayan AWS servisi **AWS DataSync**'tir.

AWS DMS (Database Migration Service) ise *VERİTABANI aktarımı ve replikasyonu için* kullanılır.

Dosya sistemi veya JSON dosyaları ile çalışmaz → tamamen yanlış servistir.

En güvenilir aktarım soruluyor → mümkünse **özel bağlantı** (Direct Connect) kullanılarak.

Seçenek Analizi:

- B. AWS DataSync over AWS Direct Connect → DOĞRU CEVAP

- DataSync, dosya transferi için **doğru servis**.
- Direct Connect, **özel, sabit, yüksek bant genişlikli, güvenilir bağlantı** sağlar.
- 10 TB günlük veri için ideal.
- Hem güvenli, hem hızlı, hem de en düşük veri kaybı riski.

→ En güvenilir ve en doğru çözüm.

✗ A. AWS DataSync over public internet

- DataSync doğru servis ✓
- Ancak **public internet** üzerinden gitmesi güvenlik ve performans açısından risklidir.
- Güvenilirlik isteniyor → en iyisi değil.

→ Daha iyi bir seçenek var.

✗ C. AWS DMS over public internet

- DMS **dosya transferi için uygun değil**, sadece veritabanları için.
- Tamamen yanlış.

→ Elenir.

✗ D. AWS DMS over Direct Connect

- DMS yanlış servis olduğundan bağlantı doğru olsa bile çözüm hâlâ yanlış.

→ Elenir.

🎯 SONUÇ

Seçenek Değerlendirme Neden

A	İyi ama ideal değil	DataSync doğru, internet güvenilir değil
B	En doğru	DataSync + Direct Connect = maksimum güvenilirlik, performans
C	Yanlış	DMS dosya sistemini desteklemez
D	Yanlış	Bağlantı iyi olsa da servis yanlış

QUESTION 12

A company needs to configure a real-time data ingestion architecture for its application. The company needs an API, a process that transforms data as the data is streamed, and a storage solution for the data.

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

- A. Deploy an Amazon EC2 instance to host an API that sends data to an Amazon Kinesis data stream. Create an Amazon Kinesis Data Firehose delivery stream that uses the Kinesis data stream as a data source. Use AWS Lambda functions to transform the data. Use the Kinesis Data Firehose delivery stream to send the data to Amazon S3.
- B. Deploy an Amazon EC2 instance to host an API that sends data to AWS Glue. Stop source/destination checking on the EC2 instance. Use AWS Glue to transform the data and to send the data to Amazon S3.
- C. Configure an Amazon API Gateway API to send data to an Amazon Kinesis data stream. Create an Amazon Kinesis Data Firehose delivery stream that uses the Kinesis data stream as a data source. Use AWS Lambda functions to transform the data. Use the Kinesis Data Firehose delivery stream to send the data to Amazon S3.
- D. Configure an Amazon API Gateway API to send data to AWS Glue. Use AWS Lambda functions to transform the data. Use AWS Glue to send the data to Amazon S3.

Soru:

Bir şirket, uygulaması için gerçek zamanlı bir veri alma (ingestion) mimarisini yapılandırması gerekiyor. Şirketin bir **API'ye**, veri akış halinde işlenirken veriyi dönüştüren bir **işleme sürecine**, ve veri için bir **depolama çözümüne** ihtiyacı var.

En AZ operasyonel yük ile hangi çözüm bu gereksinimleri karşılayacaktır?

- A. Verileri bir Amazon Kinesis veri akışına gönderen bir API barındırmak için bir Amazon EC2 örneği dağıdın. Amazon Kinesis Data Firehose teslim akışını, Kinesis veri akışını veri kaynağı olarak kullanacak şekilde oluşturun. Veriyi dönüştürmek için AWS Lambda işlevleri kullanın. Kinesis Data Firehose teslim akışını verileri Amazon S3'e gönderecek şekilde kullanın.
- B. Verileri AWS Glue'a gönderen bir API barındırmak için bir Amazon EC2 örneği dağıdın. EC2 örneğinde kaynak/hedef kontrolünü durdurun. AWS Glue'u veriyi dönüştürmek ve Amazon S3'e göndermek için kullanın.
- C. Bir Amazon API Gateway API'sini, verileri bir Amazon Kinesis veri akışına gönderecek şekilde yapılandırın. Amazon Kinesis Data Firehose teslim akışını, Kinesis veri akışını veri kaynağı olarak kullanacak şekilde oluşturun. Veriyi dönüştürmek için AWS Lambda

işlevleri kullanın. Kinesis Data Firehose teslim akışını verileri Amazon S3'e gönderecek şekilde kullanın.

D. Bir Amazon API Gateway API'sini AWS Glue'a veri gönderecek şekilde yapılandırın. Veriyi dönüştürmek için AWS Lambda işlevleri kullanın. AWS Glue'u veriyi Amazon S3'e göndermek için kullanın.

Soru Analizi:

Şirketin 3 temel ihtiyacı var:

1. **Gerçek zamanlı veri alma (ingestion)**
2. **Veri akarken (streaming) dönüşüm**
3. **Depolama çözümü**
4. **En az operasyonel yük (serverless tercih edilmeli)**

Seçenek Analizi:

C. Doğru

Bu durumda ideal AWS servisleri şunlardır:

- **API için:** Amazon API Gateway (tamamen yönetilen)
- **Streaming için:** Amazon Kinesis Data Streams
- **Dönüşüm için:** AWS Lambda
- **Depolama için:** Amazon S3
- **Minimum operasyonel yük için:** EC2 kullanılmamalı

Bu bilgiler sonucunda en bariz doğru çözüm:

 **API Gateway → Kinesis Data Stream → Firehose → Lambda → S3**

Bu yapı tamamen **serverless, scalable, managed, operasyonel yükü en düşük mimaridir**.

1. C seçeneği neden operasyonel yükü en az olan seçenekir?

C seçeneğinde kullanılan tüm servisler **tamamen yönetilen (serverless)** servislerdir:

Bileşen	Görev	Operasyonel Yük
API Gateway	Rest API yayına	Sıfır → Sunucu yok, yönetim yok
Kinesis Data Streams	Gerçek zamanlı veri akışı	Sıfır → Otomatik ölçeklenir

Bileşen	Görev	Operasyonel Yük
Kinesis Data Firehose	Dönüştürme + S3'e yazma	Sıfır → Sunucu yok, tam yönetilen
Lambda	Veri transformasyonu	Sıfır → Otomatik ölçeklenir, sunucu yok
S3	Depolama	Sıfır

Hiçbir şekilde:

- EC2 yok
- sunucu yok
- patching yok
- scaling konfigürasyonu yok
- yazılım yükleme vs. yok

Yani **tamamen serverless bir mimari**.

2. C seçeneği gerçek zamanlı veri işleme gereksinimini tam olarak karşılar

Soru açıkça diyor:

- "real-time ingestion"
- "transform as data is streamed"

Bu gereksinimleri karşılayan AWS servisleri:

- **Kinesis Data Streams → gerçek zamanlı streaming**
- **Lambda → stream gelen veriyi dönüştürme**
- **Firehose → streaming veriyi S3'e aktarma**

Bu üçlü, AWS'nin streaming mimarilerde önerdiği **standart best practice çözümüdür**.

✗ A seçeneği

API EC2 üzerinde → **operasyonel yük çok yüksek**

Sunucu yönetimi, güvenlik, ölçeklendirme → EN AZ operasyonel yük hedefiyle çelişiyor.

✗ B seçeneği

EC2 + AWS Glue →

- Glue gerçek zamanlı akışlar için tasarılanmaz.

- EC2 yine operasyonel yük oluşturur.

D seçeneği

API Gateway iyi ama → Glue real-time streaming için UYGUN DEĞİL.
Batch ETL görevleri içindir → saniyelik akışları işleyemez.

SONUÇ

Yani sadece C, hem *real-time* hem *minimum operasyonel* yük sağlar.

C seçeneğinin idealleştirilmiş mimarisi

1. **API Gateway** → dış sistemlerden veriyi alır
2. **Kinesis Data Streams** → veriyi gerçek zamanlı kuyruğa alır
3. **Firehose** → Lambda ile dönüştürür
4. **Lambda** → streaming sırasında veriyi işler
5. **S3** → kalıcı depolama

Bu mimari, AWS tarafından önerilen **en temiz streaming ingestion pipeline’ı**dır.

C seçeneği,

- gerçek zamanlı gereksinimi karşılayan,
- dönüştürme işlemini destekleyen,
- serverless çalışan,
- ölçülebilir,
- en düşük maliyetli,
- en düşük operasyonel yüke sahip çözüm olduğu için
tartışmasız en doğru cevaptır.

QUESTION 13

A company needs to keep user transaction data in an Amazon DynamoDB table. The company must retain the data for 7 years.

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

- A. Use DynamoDB point-in-time recovery to back up the table continuously.
- B. Use AWS Backup to create backup schedules and retention policies for the table.

C. Create an on-demand backup of the table by using the DynamoDB console. Store the backup in an Amazon S3 bucket. Set an S3 Lifecycle configuration for the S3 bucket.

D. Create an Amazon EventBridge (Amazon CloudWatch Events) rule to invoke an AWS Lambda function. Configure the Lambda function to back up the table and to store the backup in an Amazon S3 bucket. Set an S3 Lifecycle configuration for the S3 bucket.

Soru:

Bir şirketin kullanıcı işlem verilerini bir **Amazon DynamoDB tablosunda** saklaması gerekiyor. Şirket bu verileri **7 yıl boyunca** tutmak zorundadır.

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

A. DynamoDB point-in-time recovery (PITR) kullanarak tabloyu sürekli yedekleyin.

B. AWS Backup kullanarak tablo için yedekleme planları ve saklama (retention) politikaları oluşturun.

C. DynamoDB konsolunu kullanarak tablonun isteğe bağlı (on-demand) bir yedegini oluşturun. Bu yedegi bir Amazon S3 bucket'ına kaydedin. S3 Lifecycle yapılandırması ayarlayın.

D. Bir Amazon EventBridge kuralı oluşturarak bir AWS Lambda fonksiyonu çağırın. Bu Lambda fonksiyonunu tabloyu yedekleyecek ve yedegi bir Amazon S3 bucket'ına kaydedecek şekilde yapılandırın. Bucket için Lifecycle yapılandırması ayarlayın.

Soru Analizi:

Gereksinimler

- **DynamoDB tablosu** → kullanıcı işlem verisi (transaction data)
- **7 yıl saklama zorunluluğu** → çok uzun bir retention
- **En az operasyonel yük** → otomatik, yönetilen, hataya dayanıklı bir çözüm
- Veri sadece saklanacak, aktif olarak restore edilmesi nadir bir durum olabilir.

Seçenek Analizi:

B) AWS Backup — Backup Plan + Retention Policy

- DynamoDB yerel olarak AWS Backup ile entegredir.
- Backup planına:
 - **otomatik schedule**,
 - **7 yıllık retention**,
 - **policy tabanlı yönetim** eklenebilir.
- AWS Backup immutable (değiştirilemez) backups sağlar.

- Restore işlemleri tek tıkla ve fully managed.
- Operasyonel yük: **yok**.
- ✓ En uygun, en modern, en düşük yönetim yükü.
- ✓ Kurumsal düzeyde uzun dönem arşiv.

A) Point-in-time recovery (PITR)

- PITR geri alma penceresi **maksimum 35 gündür**.
- 7 yıl saklama gereksinimiyle doğrudan çelişir.
- Ayrıca PITR bir *operational recovery* özelliğidir, *long-term archival* değildir.
- Uzun süreli arşiv için uygun değil.

C) DynamoDB on-demand backup → S3 lifecycle

- Manuel ya da otomasyon yazman gereklidir.
- Ayrıca DynamoDB on-demand backup zaten AWS'in kendi internal formatında saklanır ve **doğrudan S3'e koyulmaz**.
- Export-to-S3 yapılabilir ama bu başka bir ek iş/zamanlama gerektirir.
- Bu yaklaşım:
 - Daha karmaşık
 - Daha fazla kod
 - Daha fazla bakım
- En fazla operasyonel yük içeren seçeneklerden biri.

D) EventBridge + Lambda → Backup → S3 + Lifecycle

- Tamamen custom otomasyon:
 - Lambda kodu yazılmışacak
 - S3 lifecycle yönetilecek
 - EventBridge kuralları yönetilecek
 - Hata durumları, retry, izleme, IAM rolleri...
- AWS Backup'ın zaten sağladığı tüm işleri *kendin* yazmış olursun.
- High-maintenance (bakım yükü yüksek)
- Yönetilen hizmet değil

Sonuç

Doğru seçenek: B

AWS Backup:

- Tamamen yönetilen hizmet
 - DynamoDB ile yerel entegrasyon
 - Backup schedule + retention (ör. 7 yıl) kolayca ayarlanır
 - Operasyonel yük yok
 - En güvenli ve en modern çözüm
-

QUESTION 14

A company is planning to use an Amazon DynamoDB table for data storage. The company is concerned about cost optimization. The table will not be used on most mornings. In the evenings, the read and write traffic will often be unpredictable. When traffic spikes occur, they will happen very quickly.

What should a solutions architect recommend?

- A. Create a DynamoDB table in on-demand capacity mode.
- B. Create a DynamoDB table with a global secondary index.
- C. Create a DynamoDB table with provisioned capacity and auto scaling.
- D. Create a DynamoDB table in provisioned capacity mode, and configure it as a global table.

Soru:

Bir şirket veri depolaması için bir Amazon DynamoDB tablosu kullanmayı planlıyor. Şirket, maliyet optimizasyonu konusunda endişeliidir. Tablo çoğu sabah kullanılmayacaktır. Akşamları ise okuma ve yazma trafiği çoğu zaman öngörelemez olacaktır. Trafik artışları gerçekleştiğinde çok hızlı bir şekilde ortaya çıkacaktır. Bir çözüm mimarı ne önermelidir?

- A. İstege bağlı (on-demand) kapasite modunda bir DynamoDB tablosu oluşturun.
- B. Global ikincil indeksli (global secondary index) bir DynamoDB tablosu oluşturun.
- C. Sağlanmış (provisioned) kapasite ve otomatik ölçekleme (auto scaling) ile bir DynamoDB tablosu oluşturun.
- D. Sağlanmış kapasite modunda bir DynamoDB tablosu oluşturun ve tabloyu global tablo olarak yapılandırın.

Soru Analizi:

Şirket bir DynamoDB tablosu kuracak ve **maliyet optimizasyonu** istiyor.

Trafik davranışları:

- ◆ **1. Sabahları tablo neredeyse hiç kullanılmıyor**

Bu saatlerde kapasite boş kalacak → gereksiz ücret istenmez.

- ◆ **2. Akşamları trafik çok değişken ve öngörlülemez**

“Unpredictable” + “quick spikes” ifadesi çok kritik.

- ◆ **3. Trafik artışı çok hızlı gerçekleşiyor**

Provisioned mode + auto scaling **hızlı spike’ları yakalayamaz**, çünkü:

- Auto scaling tepki süresi 10–20 dakika gecikme yaşayabilir.
 - Trafik aniden yükselirse **throttling** olur.
- ◆ **Bu nedenle çözüm sunları sağlanmalı:**
- Düşük kullanımda **boşuna provisioned kapasite ücreti olmamalı**
 - Ani ve öngörlülemeyen spike’larda **kapasite otomatik ve anında ölçeklenmeli**

Bu gereksinimlere tek uyan mod:

Seçenek Analizi:

ON-DEMAND kapasite modu

Avantajları:

- **Sıfır kullanımda sıfır maliyet**
- Trafik artışılarını anında karşılar → kapasite planlama yok
- Ani spike durumlarında throttling riski yok (bölge başına limitlere kadar)
- En “operationally efficient” yani işletimsel yükü en az olan çözüm

A. Create a DynamoDB table in on-demand capacity mode.

✓ En doğru cevap

- Öngörlülemez trafik için ideal
- Ani spike’ları anında karşılar
- Boşta iken ücret yok → maliyet optimizasyonu
- Yönetime ihtiyaç yok → operasyonel yük çok düşük

B. Create a DynamoDB table with a global secondary index.

Trafik veya maliyetle ilgisi yok
GSI sadece sorgu esnekliği sağlar
Maliyeti düşürmez, hatta artırabilir

C. Provisioned capacity + auto scaling

 Kısım uygundur olabilir ama **en doğru çözüm değildir**

- Auto scaling spike'lara *reaktif* çalışır (gecikmeli)
- Ani trafik artışlarında throttling olabilir
- Sabah boş saatlerde yine minimum provisioned kapasite için ödeme yapılır
Bu nedenle maliyet ve performans açısından zayıftır.

D. Provisioned mode + global table

Gereksiz derecede pahalı
Çoklu bölge replikasyon gerekmeyen
Maliyet optimizasyonu ile çelişir

SONUÇ

Tüm gereksinimler toplandığında:

- ✓ Maliyet optimizasyonu
- ✓ Düşük kullanım saatlerinde sıfır maliyet
- ✓ Ani trafik artışlarına anında cevap
- ✓ Kapasite planlaması gerektirmeme (**operational efficiency**)
- ✓ Yönetim yükünün en az olması

Bu gereksinimlerin tamamını sadece şu sağlar:

QUESTION 15

A company recently signed a contract with an AWS Managed Service Provider (MSP) Partner for help with an application migration initiative. A solutions architect needs to share an Amazon Machine Image (AMI) from an existing AWS account with the MSP Partner's AWS account. The AMI is backed by Amazon Elastic Block Store (Amazon EBS) and uses an AWS Key Management Service (AWS KMS) customer managed key to encrypt EBS volume snapshots.

What is the MOST secure way for the solutions architect to share the AMI with the MSP Partner's AWS account?

- A. Make the encrypted AMI and snapshots publicly available. Modify the key policy to allow the MSP Partner's AWS account to use the key.
- B. Modify the launchPermission property of the AMI. Share the AMI with the MSP Partner's AWS account only. Modify the key policy to allow the MSP Partner's AWS account to use the key.
- C. Modify the launchPermission property of the AMI. Share the AMI with the MSP Partner's AWS account only. Modify the key policy to trust a new KMS key that is owned by the MSP Partner for encryption.
- D. Export the AMI from the source account to an Amazon S3 bucket in the MSP Partner's AWS account, Encrypt the S3 bucket with a new KMS key that is owned by the MSP Partner. Copy and launch the AMI in the MSP Partner's AWS account.

Soru:

Bir şirket, uygulama taşıma girişimi için yakın zamanda bir AWS Yönetilen Hizmet Sağlayıcısı (MSP) Ortağı ile sözleşme imzaladı. Bir çözüm mimarının, mevcut bir AWS hesabındaki bir Amazon Machine Image'ı (AMI) MSP Ortağı'nın AWS hesabı ile paylaşması gerekiyor. AMI, Amazon Elastic Block Store (Amazon EBS) tarafından desteklenmektedir ve EBS bölüm anlık görüntülerini şifrelemek için AWS Key Management Service (AWS KMS) tarafından yönetilen müşteri anahtarını (customer managed key) kullanmaktadır.

A Solutions Architect, AMI'yi MSP Ortağı'nın AWS hesabı ile paylaşmak için **EN güvenli yöntemi** seçmelidir.

- A.** Şifrelenmiş AMI'yi ve anlık görüntüleri (snapshot) herkese açık (public) hâle getirin. Anahtar politikasını MSP Ortağı'nın AWS hesabının anahtarı kullanmasına izin verecek şekilde değiştirin.
- B.** AMI'nın *launchPermission* özelliğini değiştirin. AMI'yi yalnızca MSP Ortağı'nın AWS hesabı ile paylaşın. Anahtar politikasını MSP Ortağı'nın AWS hesabının anahtarı kullanmasına izin verecek şekilde değiştirin.
- C.** AMI'nın *launchPermission* özelliğini değiştirin. AMI'yi yalnızca MSP Ortağı'nın AWS hesabı ile paylaşın. Anahtar politikasını, MSP Ortağı'na ait yeni bir KMS anahtarına güvenecek şekilde değiştirin.
- D.** AMI'yi kaynak hesaptan MSP Ortağı'nın AWS hesabındaki bir Amazon S3 kovasına dışa aktarın. S3 kovasını MSP Ortağı'na ait yeni bir KMS anahtarı ile şifreleyin. AMI'yi MSP Ortağı'nın AWS hesabına kopyalayın ve başlatın.

Soru Analizi:

Bu soru, **KMS ile şifrelenmiş bir EBS-backed AMI'nin başka bir AWS hesabı ile güvenli biçimde paylaşılması** konusunu test ediyor.

Paylaşım yapılırken iki ana konu var:

1. AMI paylaşımı

- AMI'yi paylaşmak için: **launchPermission** ayarlanır.
- AMI public yapılmaz → güvenlik açığı olur.

2. KMS anahtar politikası

- EBS snapshot şifrelemesi nedeniyle, AMI'yi paylaşırken **snapshot'ı şifreleyen KMS anahtarının da paylaşımı açılması** gereklidir.
- KMS key paylaşılmadan AMI başka hesapta kullanılamaz.
- En güvenli yöntem → sadece ilgili hesabın kullanmasına izin vermek (explicit allow).

Önemli nokta:

AMI'yi diğer hesaba taşımak için AMI/Snapshot'ı **public yapmak yasaktır, S3'e export etmek çoğu zaman gereksiz ve daha az güvenlidir**, ayrıca **KMS cross-account key sharing** standart yöntemdir.

Seçenek Analizi:

- B. AMI'yi yalnızca MSP hesabı ile paylaşmak + KMS key policy ile MSP hesabına izin vermek**

Doğru ve en güvenli yöntem.

- AMI launchPermission → sadece target account için allow
- KMS key → MSP hesabının decrypt ve re-encrypt kullanabilmesi için policy güncellemesi
- AWS'nin önerdiği standart yöntem
- Minimum operasyonel yük + en yüksek güvenlik

→ Bu soru için doğru cevap büyük ihtimalle **B'dir.**

A. AMI ve snapshot'ları public yapmak

Kesinlikle yanlış.

- KMS şifreli snapshot public yapılamaz.
- Güvenlik açısından çok kötü bir pratik.
- Sınav tuzağı.

C. AMI paylaş + KMS key'i MSP'nin yeni anahtarına "trust" et

Yanlış / gereksiz / yanlış teknik

- KMS key'ler birbirine "trust" etmez.
- Data key re-encryption gerektirir ve AMI bu şekilde taşınmaz.
- AWS'nin desteklediği mekanizma değildir.

AMI'yi S3'e export edip yeni hesapta yeniden import etmek

Teknik olarak yanlış / geçersiz

- EBS-backed AMI **S3'e export edilemez** (yalnızca VM Import/Export ile belirli formatlarda izin vardır).
- KMS şifreli AMI export edilemez.
- İş yükünü gereksiz yere arttırmır.

SONUÇ

En güvenli, AWS best practice'e uygun ve en kolay yöntem:

B seçenektedir.

- AMI sadece MSP hesabı ile paylaşılır.
- KMS key policy MSP hesabına decrypt izni verecek şekilde güncellenir.
- Snapshot paylaşımına gerek yoktur; AMI paylaşımı bunu kapsar.

QUESTION 16

A solutions architect is designing the cloud architecture for a new application being deployed on AWS. The process should run in parallel while adding and removing application nodes as needed based on the number of jobs to be processed. The processor application is stateless. The solutions architect must ensure that the application is loosely coupled and the job items are durably stored.

Which design should the solutions architect use?

- A. Create an Amazon SNS topic to send the jobs that need to be processed. Create an Amazon Machine Image (AMI) that consists of the processor application. Create a launch configuration that uses the AMI. Create an Auto Scaling group using the launch configuration. Set the scaling policy for the Auto Scaling group to add and remove nodes based on CPU usage.

B. Create an Amazon SQS queue to hold the jobs that need to be processed. Create an Amazon Machine Image (AMI) that consists of the processor application. Create a launch configuration that uses the AMI. Create an Auto Scaling group using the launch configuration. Set the scaling policy for the Auto Scaling group to add and remove nodes based on network usage.

C. Create an Amazon SQS queue to hold the jobs that need to be processed. Create an Amazon Machine Image (AMI) that consists of the processor application. Create a launch template that uses the AMI. Create an Auto Scaling group using the launch template. Set the scaling policy for the Auto Scaling group to add and remove nodes based on the number of items in the SQS queue.

D. Create an Amazon SNS topic to send the jobs that need to be processed. Create an Amazon Machine Image (AMI) that consists of the processor application. Create a launch template that uses the AMI. Create an Auto Scaling group using the launch template. Set the scaling policy for the Auto Scaling group to add and remove nodes based on the number of messages published to the SNS topic.

Soru:

Bir şirket, AWS üzerinde dağıtılmak üzere yeni bir uygulama için bulut mimarisini tasarlayan bir çözüm mimarına sahiptir. İşlem, paralel olarak çalışmalı ve işlenecek iş (job) sayısına bağlı olarak uygulama düğümlerini (application nodes) gerektiğinde ekleyip kaldırmalıdır. İşleyici uygulama (processor application) durum bilgisi tutmamalıdır (stateless). Çözüm mimarı, uygulamanın gevşek bağlı (loosely coupled) olmasını ve iş öğelerinin kalıcı olarak saklanması (durably stored) sağlamalıdır.

Bu gereksinimleri karşılamak için çözüm mimarı hangi tasarıyı kullanmalıdır?

A. İşlenmesi gereken işleri göndermek için bir Amazon SNS topic oluşturun. İşleyici uygulamayı içeren bir Amazon Machine Image (AMI) oluşturun. AMI'yi kullanan bir launch configuration oluşturun. Bu launch configuration'ı kullanan bir Auto Scaling grubuna oluşturun. Auto Scaling grubunun ölçeklendirme politikasını CPU kullanımına göre düğüm ekleyip kaldıracak şekilde ayarlayın.

B. İşlenmesi gereken işleri tutmak için bir Amazon SQS queue oluşturun. İşleyici uygulamayı içeren bir Amazon Machine Image (AMI) oluşturun. AMI'yi kullanan bir launch configuration oluşturun. Bu launch configuration'ı kullanan bir Auto Scaling grubu oluşturun. Auto Scaling grubunun ölçeklendirme politikasını ağ kullanımına (network usage) göre düğüm ekleyip kaldıracak şekilde ayarlayın.

C. İşlenmesi gereken işleri tutmak için bir Amazon SQS queue oluşturun. İşleyici uygulamayı içeren bir Amazon Machine Image (AMI) oluşturun. AMI'yi kullanan bir launch template oluşturun. Bu launch template'ı kullanan bir Auto Scaling grubu

oluşturun. Auto Scaling grubunun ölçeklendirme politikasını SQS kuyruğundaki öğe sayısına göre düğüm ekleyip kaldıracak şekilde ayarlayın.

D. İşlenmesi gereken işleri göndermek için bir Amazon SNS topic oluşturun. İşleyici uygulamayı içeren bir Amazon Machine Image (AMI) oluşturun. AMI'yi kullanan bir launch template oluşturun. Bu launch template'i kullanan bir Auto Scaling grubu oluşturun. Auto Scaling grubunun ölçeklendirme politikasını SNS'e gönderilen mesaj sayısına göre düğüm ekleyip kaldıracak şekilde ayarlayın.

Soru Analizi:

Uygulama gereksinimleri:

✓ 1. Parallel processing

İşler paralel işlenecek → Kuyruk (queue) tabanlı bir mimari uygun.

✓ 2. Auto Scaling ile otomatik düğüm ekleme/çıkarma

İş yüküne göre worker sayısı artmalı/azalmalı.

✓ 3. Stateless worker requirement

Worker'lar durum taşımamalı → Queue tabanlı dağıtık işleme mimarisine tam uyumlu.

✓ 4. Loosely Coupled Architecture

Tier'lar birbirine bağımlı olmamalı → Amazon SQS gibi bir mesaj kuyruğu kullanılmalı.

✓ 5. Job items must be durably stored

Mesajlar kaybolmamalı → **SQS** minimum bir kere teslim garantisini ve kalıcı depolamaya sahip.

✓ 6. Operationally efficient

Yönetim yükü minimum olmalı.

→ Tüm işaretler **Amazon SQS + Auto Scaling** yönüne çıkıyor.

Seçenek Analizi:

C. SQS + Auto Scaling + Queue length scaling → DOĞRU CEVAP

Neden?

✓ SQS işleri **kalıcı olarak saklar**

✓ İşleyiciler stateless olabilir

✓ ASG, SQS **kuyruk uzunluğununa** göre ölçeklenebilir

- ✓ Modern yöntem: launch template
- ✓ Fully decoupled architecture

Bu tam olarak AWS'nin recommended pattern'ıdır.

✗ A. SNS + Auto Scaling + CPU usage → Yanlış

- SNS mesajları **saklamaz**, kaybolabilir → durability gereksinimine uymuyor.
- CPU usage iş yoğunluğunu temsil etmez.

✗ B. SQS + Auto Scaling + Network usage → Yanlış

- Doğru queue seçilmiş.
- Ancak network usage, kuyruktaki iş yükünü doğru ölçmez.

✗ D. SNS + Auto Scaling + SNS message count → Yanlış

- SNS mesajları saklamadığı için durability yok.
- SNS, iş kuyruğu olarak kullanılmaz.

⌚ SONUÇ (Doğru Cevap): C

Amazon SQS kuyruğu + Auto Scaling group's scaling policy = Queue length

Bu yaklaşım hem modern, hem doğru, hem de operasyonel yükü en düşük çözümüdür.

Bu senaryoda çözüme yön veren üç kritik gereksinim vardır:

1. **Uygulamanın stateless olması**
2. **İşlerin (jobs) dayanıklı bir şekilde saklanması**
3. **Loosely-coupled (gevşek bağlı) bir mimari kurulması**

Bu gereksinimlerin karşılanması için AWS'de en doğru yaklaşım, iş yükünü kuyruk tabanlı bir mimariyle yönetmektir. Amazon **SQS**, mesajları kalıcı olarak depoladığı ve minimum bir kere teslim garantisini sunduğu için dayanıklı bir job depolama sistemi sağlar. Ayrıca SQS kullanmak, işlemcilerin (processor nodes) kuyruktan iş çekerek tamamen bağımsız çalışmasını sağlar; böylece uygulama katmanları arasında **tam bir ayrışma** gerçekleşir.

Stateless worker'ların iş sayısına göre ölçeklenmesi, modern AWS mimarisinde **queue length-based scaling** ile yapılır. Auto Scaling grup, SQS kuyruğundaki mesaj sayısı arttığında yeni instance'lar başlatır; mesaj sayısı azaldığında ise node sayısını azaltır. Bu, hem maliyet optimizasyonu hem de yüksek ölçeklenebilirlik sağlar.

A company hosts its web applications in the AWS Cloud. The company configures Elastic Load Balancers to use certificates that are imported into AWS Certificate Manager (ACM). The company's security team must be notified 30 days before the expiration of each certificate.

What should a solutions architect recommend to meet this requirement?

- A. Add a rule in ACM to publish a custom message to an Amazon Simple Notification Service (Amazon SNS) topic every day, beginning 30 days before any certificate will expire.
- B. Create an AWS Config rule that checks for certificates that will expire within 30 days. Configure Amazon EventBridge (Amazon CloudWatch Events) to invoke a custom alert by way of Amazon Simple Notification Service (Amazon SNS) when AWS Config reports a noncompliant resource.
- C. Use AWS Trusted Advisor to check for certificates that will expire within 30 days. Create an Amazon CloudWatch alarm that is based on Trusted Advisor metrics for check status changes. Configure the alarm to send a custom alert by way of Amazon Simple Notification Service (Amazon SNS).
- D. Create an Amazon EventBridge (Amazon CloudWatch Events) rule to detect any certificates that will expire within 30 days. Configure the rule to invoke an AWS Lambda function. Configure the Lambda function to send a custom alert by way of Amazon Simple Notification Service (Amazon SNS).

Soru:

Bir şirket web uygulamalarını AWS Cloud üzerinde barındırmaktadır. Şirket, Elastic Load Balancer'ları AWS Certificate Manager (ACM) içine aktarılan sertifikaları kullanacak şekilde yapılandırmıştır. Şirketin güvenlik ekibi, **her sertifikanın süresi dolmadan 30 gün önce** bilgilendirilmelidir.

Bu gereksinimi karşılamak için bir çözüm mimarı ne önermelidir?

- A. ACM'de bir kural ekleyerek, herhangi bir sertifika süresinin dolmasına 30 gün kala başlayacak şekilde her gün Amazon SNS konusuna özel bir mesaj yayınlanmasını sağlamak.
- B. 30 gün içinde süresi dolacak sertifikaları kontrol eden bir AWS Config kuralı oluşturmak. AWS Config bir uyumsuz kaynak bildirdiğinde, Amazon EventBridge aracılığıyla bir SNS uyarısı tetiklemek.
- C. AWS Trusted Advisor'ı kullanarak süresi 30 gün içinde dolacak sertifikaları kontrol etmek. Trusted Advisor ölçümlerine dayalı bir CloudWatch alarmı oluşturmak ve bu alarmı bir SNS bildirimi gönderecek şekilde yapılandırmak.

D. Süresi 30 gün içinde dolacak sertifikaları algılayan bir Amazon EventBridge kuralı oluşturmak. Bu kuralı bir AWS Lambda fonksiyonunu tetikleyecek şekilde yapılandırmak ve bu fonksiyonun SNS üzerinden özel bir uyarı göndemesini sağlamak.

Soru Analizi:

Şirket, ELB üzerinde kullanılan ACM sertifikalarını kullanıyor. ACM, sertifika süresi dolmadan **vefat uyarılarını kendi içinde oluşturur**, ancak bu uyarılar:

- Varsayılan olarak yalnızca **sertifika sahibine e-posta gönderilir**,
- SNS'e otomatik bağlanmaz,
- Güvenlik ekibinin bir mail kutusu dinlemesine güvenmek istemiyoruz.

Gereksinim çok net:

👉 **Sertifika bitimine 30 gün kala otomatik tetiklenecek bir bildirim sistemi kurulmalı.**

Bu bir **event-driven, serverless, operationally minimum** çözüm gerektirir.

ACM, EventBridge'ye otomatik "certificate expiring" eventi gönderir.

Bu nedenle çözümün merkezinde **EventBridge** olmalıdır.

Seçenek Analizi:

✓ D. EventBridge kuralı + Lambda + SNS

💡 **Doğru Cevap — En güvenli, en operasyonel olarak verimli seçenek**

- ACM, sertifika bitiş tarihi yaklaşınca EventBridge'ye otomatik event gönderir.
- EventBridge kuralı bu eventi yakalar.
- Lambda, SNS'e bildirim yollar (ve gerekirse e-posta / Slack / webhook gibi çoklu hedefler sağlar).
- Tamamen serverless, bakım gerektirmez.
- AWS tarafından resmi olarak önerilen "ACM certificate expiration notifications" çözümüdür.

✓ EventBridge → ✓ Lambda → ✓ SNS

Tam otomasyon, güvenlik ekibi için %100 güvenilir bildirim.

✗ A. ACM içinde kural ekleme

- ACM'de "her gün SNS'e mesaj gönder" gibi bir mekanizma **yoktur**.
- ACM'nin SNS entegrasyonu bulunmaz.

Bu seçenek teknik olarak mümkün değildir → **Invalid**.

B. AWS Config kuralı kullanmak

- AWS Config sertifika yaşılanmasını **doğrudan izlemez**.
- Sertifika expiration için AWS Config kuralları **hazır değildir**.
- Config gereksiz maliyet ve operasyon yükü ekler.

Çalışır hale getirilebilir, fakat:

- Overkill
- Ek maliyet
- Daha az güvenilir

Bu nedenle **optimal değildir**.

C. Trusted Advisor kullanmak

- Trusted Advisor sertifika expiration uyarısı verir ama:
 - Sadece Business/Enterprise Support'ta tam çalışır
 - Kontroller anlıktır, 30 günlük olay tetikleme değil
 - EventBridge'e direkt event göndermez

Yani şirketin uyarı akışını otomatikleştiremez → **Uygun değil**.

SONUÇ (Özet + Nedensellik)

- Soruda istenen: **ACM sertifikalarının 30 gün kala otomatik bildirim**
- En iyi yöntem: **EventBridge'in ACM sertifika expiration eventlerini yakalaması**
- En düşük operasyon maliyeti: **Serverless — EventBridge + Lambda + SNS**
- AWS'nin resmi best practice'i: **D seçeneğidir**

Bu nedenle **en doğru, en güvenilir, en AWS-native çözüm** → **D seçeneğidir**.

QUESTION 18

A company's dynamic website is hosted using on-premises servers in the United States. The company is launching its product in Europe, and it wants to optimize site loading times for new European users. The site's backend must remain in the United States. The product is being launched in a few days, and an immediate solution is needed. What should the solutions architect recommend?

A. Launch an Amazon EC2 instance in us-east-1 and migrate the site to it.

- B. Move the website to Amazon S3. Use Cross-Region Replication between Regions.
- C. Use Amazon CloudFront with a custom origin pointing to the on-premises servers.
- D. Use an Amazon Route 53 geoproximity routing policy pointing to on-premises servers.

Soru:

Bir şirketin dinamik web sitesi Amerika Birleşik Devletleri’nde bulunan şirket içi (on-premises) sunucular üzerinde barındırılmaktadır. Şirket ürününü Avrupa’da piyasaya sürmektedir ve yeni Avrupa’lı kullanıcılar için site yüklenme sürelerini optimize etmek istemektedir. Sitenin arka ucu (backend) Amerika Birleşik Devletleri’nde kalmalıdır. Ürün birkaç gün içinde piyasaya sürülecektir ve hemen uygulanabilir bir çözüme ihtiyaç vardır.

Çözüm mimarı ne önermelidir?

- A. us-east-1 Bölgesinde bir Amazon EC2 örneği başlatın ve siteyi bu örneğe taşıyın.
- B. Web sitesini Amazon S3'e taşıyın. Bölgeler arası çoğaltma (Cross-Region Replication) kullanın.
- C. On-premises sunuculara işaret eden özel bir origin ile Amazon CloudFront kullanın.
- D. On-premises sunuculara işaret eden bir Amazon Route 53 jeo-yakınlık (geoproximity) yönlendirme politikası kullanın.

Soru Analizi:

- Şirketin web sitesi **dinamik** ve **on-premises (ABD)** sunucularında barındırılıyor.
- Avrupa kullanıcıları için **yükleme sürelerini hızlandırmak** istiyorlar.
- **Backend ABD’de kalmak zorunda** → Yani backend’i taşıyamazsınız.
- **Sadece birkaç gün içinde çözüm gereklidir** → Çok hızlı uygulanabilir bir çözüm olmalı.
- Hedef:
 - Avrupa’ya yakın edge noktaları üzerinden içerik sunmak,
 - Latency (gecikme) azaltmak.

Bu kriterlerin hepsini karşılayabilen AWS servisleri arasında en uygun seçenek:

→ **CloudFront**, çünkü:

- Global edge network üzerinden içerik dağıtır.
- Origin olarak **on-premises sunucular** kullanılabilir.
- Uygulaması **çok hızlıdır**, saatler içinde kurulabilir.
- Dinamik içerik için de **TCP/HTTPS geçişini (origin fetch)** destekler.

Seçenek Analizi:

✓ C. CloudFront + on-premises custom origin kullanma

- CloudFront **dünyaya yayılmış edge lokasyonlarından** içerik sunar.
- Origin olarak:
 - On-premises sunucu
 - EC2
 - S3
 - ALBkullanılabilir.
- Kullanıcılar Avrupa'da → En yakın edge node → Daha hızlı yükleme.
- Backend yine ABD'de kalabilir → CloudFront sadece caching ve routing yapar.
- **Hızlı, ucuz, minimum değişiklik.**

✗ A. us-east-1'de EC2 başlatıp siteyi taşıma

- Site on-premises çalışıyor, birkaç gün içinde **tam migrasyon mümkün değil**.
- Backend ABD'de kalmak zorunda → EC2'ye taşımak backend'i değiştirmek anlamına gelir.
- Ayrıca Avrupa için latency azalmaz, yine ABD bölgesinde.

✗ B. Web sitesini S3'e taşıyıp Cross-Region Replication kullanma

- S3 **statik hosting içindir**, soru “**dinamik website**” diyor → S3 desteklemez.
- Backend'in on-prem kalması gerekiğinden bu yapı çalışmaz.
- Ayrıca migration için süre çok kısa.

✗ D. Route 53 geoproximity routing kullanma

- Bu yalnızca DNS yönlendirmesidir.
- DNS tek başına **latency'i azaltmaz**.
- Kullanıcı yine ABD'deki on-prem sunucuya bağlanır.
- Edge caching yok → Gecikme azalmaz.

SONUÇ

En doğru ve en hızlı uygulanabilir çözüm:

👉 **C. Amazon CloudFront + custom origin**

Backend yerinde kalır, latency ciddi şekilde düşer, dinamizm korunur ve çözüm birkaç saat içinde devreye alınabilir.

QUESTION 19

A company wants to reduce the cost of its existing three-tier web architecture. The web, application, and database servers are running on Amazon EC2 instances for the development, test, and production environments. The EC2 instances average 30% CPU utilization during peak hours and 10% CPU utilization during non-peak hours. The production EC2 instances run 24 hours a day. The development and test EC2 instances run for at least 8 hours each day. The company plans to implement automation to stop the development and test EC2 instances when they are not in use.

Which EC2 instance purchasing solution will meet the company's requirements MOST cost-effectively?

- A. Use Spot Instances for the production EC2 instances. Use Reserved Instances for the development and test EC2 instances.
- B. Use Reserved Instances for the production EC2 instances. Use On-Demand Instances for the development and test EC2 instances.
- C. Use Spot blocks for the production EC2 instances. Use Reserved Instances for the development and test EC2 instances.
- D. Use On-Demand Instances for the production EC2 instances. Use Spot blocks for the development and test EC2 instances.

Soru:

Bir şirket mevcut üç katmanlı web mimarisinin maliyetini azaltmak istiyor. Web, uygulama ve veritabanı sunucuları geliştirme, test ve üretim ortamları için Amazon EC2 örneklerinde çalışıyor. EC2 örnekleri, yoğun saatlerde ortalama %30 CPU kullanımı ve yoğun olmayan saatlerde %10 CPU kullanımını gösteriyor. Üretim EC2 örnekleri günde 24 saat çalışıyor. Geliştirme ve test EC2 örnekleri ise günde en az 8 saat çalışıyor. Şirket, geliştirme ve test EC2 örneklerini kullanılmadıklarında durdurmak için otomasyon uygulamayı planlıyor.

Şirketin gereksinimlerini **en maliyet etkin şekilde** karşılayacak olan hangi EC2 instance satın alma çözümüdür?

- A. Üretim EC2 örnekleri için Spot Instances kullanın. Geliştirme ve test EC2 örnekleri için Reserved Instances kullanın.

- B. Üretim EC2 örnekleri için Reserved Instances kullanın. Geliştirme ve test EC2 örnekleri için On-Demand Instances kullanın.
- C. Üretim EC2 örnekleri için Spot Blocks kullanın. Geliştirme ve test EC2 örnekleri için Reserved Instances kullanın.
- D. Üretim EC2 örnekleri için On-Demand Instances kullanın. Geliştirme ve test EC2 örnekleri için Spot Blocks kullanın.

Soru Analizi:

Şirketin mevcut durumu:

- 3 katmanlı mimari: Web + Uygulama + DB (hepsi EC2)
- **Üretim ortamı** → 24 saat çalışıyor → sürekli ihtiyaç → **kararlı yük**
- **Geliştirme & Test ortamları** → günde sadece **8 saat aktif**, geri kalan zamanlarda otomasyonla durdurulacak → **kesintiye toleranslı + sürekli çalışma**

CPU kullanımı düşük (%30 peak, %10 non-peak), bu da maliyet optimizasyonu için fiyatlandırma modeline odaklanması gerektiğini gösterir.

Şirketin hedefi:

→ En maliyet etkili EC2 satın alma modeli

Önemli noktalar:

- Production her zaman açık → en ucuz uzun vadeli model **Reserved Instances**
- Dev & Test kesintiye toleranslı + dur-kalk olacak → **Spot instances** (Spot Blocks artık kaldırıldı ama soru bağlamında var)

Seçenek Analizi:

B. Üretim = Reserved, Dev/Test = On-Demand

🟡 Kısım doğru ama **en maliyetli seçeneklerden biri**.

Dev/Test için On-Demand pahalı kalır → Spot daha ucuz.

A. Üretim = Spot, Dev/Test = Reserved

Spot, üretim için uygun değildir → kesinti riski yüksek.

Dev/Test için Reserved pahalı ve gereksiz.

C. Üretim = Spot Blocks, Dev/Test = Reserved

Spot Blocks zaman garantisini verse de *üretim için hala risklidir*.

Dev/Test için Reserved gereksiz masraf.

D. Üretim = On-Demand, Dev/Test = Spot Blocks

Üretim On-Demand → oldukça pahalı.

Spot Blocks dev/test için uygun olsa bile toplam maliyet optimum değil.

Sonuç

Fakat bu soruda aslında AWS gerçek dünyasında en mantıklı çözüm şudur:

- **Production: Reserved Instances (sabit yük)**
- **Dev/Test: Spot Instances (kesintiye toleranslı)**

Ama seçeneklerde Spot Instances dev/test için tam olarak sunulmadığından, en mantıklı ve güvenli çözüm:

Üretim için Reserved Instances

→ sürekli çalışan ortam için en ucuz model

Dev/Test için On-Demand

→ dur/kalk otomasyonu ile maliyet yine düşük olacak

QUESTION 20

A company has a production web application in which users upload documents through a web interface or a mobile app. According to a new regulatory requirement, new documents cannot be modified or deleted after they are stored.

What should a solutions architect do to meet this requirement?

- A. Store the uploaded documents in an Amazon S3 bucket with S3 Versioning and S3 Object Lock enabled.
- B. Store the uploaded documents in an Amazon S3 bucket. Configure an S3 Lifecycle policy to archive the documents periodically.
- C. Store the uploaded documents in an Amazon S3 bucket with S3 Versioning enabled. Configure an ACL to restrict all access to read-only.
- D. Store the uploaded documents on an Amazon Elastic File System (Amazon EFS) volume. Access the data by mounting the volume in read only mode.

Soru:

Bir şirketin üretim ortamında bulunan bir web uygulaması vardır ve kullanıcılar belgeleri bir web arayüzü veya mobil uygulama aracılığıyla yüklemektedir. Yeni bir düzenleyici gereksinime göre, yeni belgeler depolandıktan sonra **değiştirilemez veya silinemez** olmalıdır.

Bu gereksinimi karşılamak için bir çözüm mimarı ne yapmalıdır?

- A. Yüklenen belgeleri, S3 Sürümleme (S3 Versioning) ve S3 Nesne Kilidi (S3 Object Lock) etkinleştirilmiş bir Amazon S3 kovasında depolayın.

- B. Yüklenen belgeleri bir Amazon S3 kovasında depolayın. Belgeleri periyodik olarak arşivlemek için bir S3 Yaşam Döngüsü (Lifecycle) politikası yapılandırılabilir.
- C. Yüklenen belgeleri, S3 Sürümleme (S3 Versioning) etkinleştirilmiş bir Amazon S3 kovasında depolayın. Erişim kontrol listesini (ACL) tüm erişimi salt okunur olacak şekilde yapılandırılabilir.
- D. Yüklenen belgeleri bir Amazon Elastic File System (Amazon EFS) biriminde depolayın. Verilere salt okunur biçimde bağlanarak erişin.

Soru Analizi:

Şirketin yeni bir yasal/regülasyon gereksinimi var:

Belgeler depolandıktan sonra:

- **X Silinmez**
- **X Değiştirilemez**

Bu özellik, düzenleyici gereksinimlerde **WORM (Write Once Read Many)** olarak adlandırılır.

AWS'de WORM uyumluluğu sağlayan tek servis:

★ S3 Object Lock (Uyumluluk veya Yönetici Modu)

Bu mod, dosyaların belirli bir süre veya süresiz olarak **silinmesini ve değiştirilmesini tamamen engeller**.

Seçenek Analizi:

A. S3 Versioning + S3 Object Lock kullanmak

Neden doğru?

- **S3 Object Lock**, dosyayı yazıldıktan sonra:
 - Silinmesini engeller
 - Üzerine yazılmasını engeller
- **Regülasyon uyumluluğu sağlar.**
- WORM gereksinimini karşılayan AWS'nin tek yerleşik çözümüdür.
- Object Lock çalışması için Versioning zorunludur → İkisi birlikte kullanılır.

Bu nedenle tam olarak istenenin karşısındır.

X B. Lifecycle ile arşivlemek

- Lifecycle politikaları sadece:

- S3 Standard → Glacier geçişi gibi *sınıf geçişi* yapar.
- Silme politikaları ayarlayabilir.
- Ancak **Object Lock olmadan**, dosyalar:
 - Silinebilir
 - Üzerine yazılabilir

Regülasyon gereksinimini karşılamaz.

C. Versioning + ACL ile salt okunur yapmak

- S3 ACL salt okunurluk **garanti etmez**.
- Yönetici yine silebilir.
- Versioning yalnız başına silinmeyi engellemez, sadece eski versiyonu korur.
- Yasal gereksinim olan **değiştirilemezlik** ve **silinemezlik** sağlanamaz.

D. Belgeleri EFS'de saklamak, read-only mount

- EFS salt okunur bağlansa bile:
 - Yönetici izinleri değiştirip yazılabilir yapabilir.
 - Dosyalar yine silinebilir.
- EFS **WORM desteklemez**.
- Yasal gereksinimleri karşılamaz.

SONUÇ

Regülasyon gereksinimi: **WORM (Write Once Read Many)**

AWS'de bunun karşılığı: **Amazon S3 Object Lock**

Doğru cevap: A

Nedeni: S3 Object Lock, dosyaların depolandıktan sonra *silinememesini* ve *değiştirilememesini* garanti eden tek AWS özelliğidir.

QUESTION 21

A company has several web servers that need to frequently access a common Amazon RDS MySQL Multi-AZ DB instance. The company wants a secure method for the web servers to connect to the database while meeting a security requirement to rotate user credentials frequently.

Which solution meets these requirements?

- A. Store the database user credentials in AWS Secrets Manager. Grant the necessary IAM permissions to allow the web servers to access AWS Secrets Manager.
- B. Store the database user credentials in AWS Systems Manager OpsCenter. Grant the necessary IAM permissions to allow the web servers to access OpsCenter.
- C. Store the database user credentials in a secure Amazon S3 bucket. Grant the necessary IAM permissions to allow the web servers to retrieve credentials and access the database.
- D. Store the database user credentials in les encrypted with AWS Key Management Service (AWS KMS) on the web server le system. The web server should be able to decrypt the les and access the database.

Soru:

Bir şirketin, **ortak bir Amazon RDS MySQL Multi-AZ veritabanına** sık sık erişmesi gereken **birden fazla web sunucusu** vardır. Şirket, web sunucularının veritabanına **güvenli bir şekilde bağlanması** istemektedir ve aynı zamanda **kullanıcı kimlik bilgilerinin (username/password) sık sık döndürülmesini (rotate edilmesini)** zorunlu kılan bir **güvenlik gereksinimini** karşılamak istemektedir.

Bu gereksinimleri hangi çözüm karşılar?

- A. Veritabanı kullanıcı kimlik bilgilerini **AWS Secrets Manager** içinde sakla.
Web sunucularının **AWS Secrets Manager'a** erişebilmesi için gerekli IAM izinlerini ver.
- B. Veritabanı kullanıcı kimlik bilgilerini **AWS Systems Manager OpsCenter** içinde sakla.
Web sunucularının OpsCenter'a erişebilmesi için gerekli IAM izinlerini ver.
- C. Veritabanı kullanıcı kimlik bilgilerini **güvenli bir Amazon S3 bucket** içinde sakla.
Web sunucularına, bu kimlik bilgilerini alabilmeleri ve veritabanına erişebilmeleri için gerekli IAM izinlerini ver.
- D. Veritabanı kullanıcı kimlik bilgilerini, web sunucusunun dosya sisteminde **AWS KMS ile şifrelenmiş dosyalar** içinde sakla.
Web sunucusu bu dosyaları çözerek (decrypt ederek) veritabanına erişsin.

Soru Analizi:

Soruda özellikle vurgulanan noktalar:

1. **Birden fazla web sunucusu** ortak bir RDS MySQL DB'ye bağlanıyor
2. **Güvenli bağlantı** isteniyor
3. **Kullanıcı kimlik bilgilerinin sık sık rotate edilmesi gerekiyor** (EN KRİTİK NOKTA)

👉 Yani çözüm:

- Merkezi olmalı
- Otomatik **credential rotation** desteklemeli
- IAM ile güvenli erişim sağlamalı

Seçenek Analizi:

✓ A. AWS Secrets Manager

Neden DOĞRU?

- 🔒 Veritabanı **username / password** bilgilerini güvenli şekilde saklar
- 🔪 **Otomatik credential rotation** destekler (RDS MySQL ile native entegrasyon)
- 🚫 IAM role ile web sunucuları secrets'lara erişir
- 📦 Birden fazla web server aynı secret'i güvenle kullanabilir
- 💡 AWS sınavlarında **credential rotation** denince **ilk akla gelen servis**

👉 Sorudaki “rotate user credentials frequently” ifadesi **doğrudan Secrets Manager’ı işaret eder.**

Yanlış Seçeneklerin Analizi

✗ B. Systems Manager OpsCenter

- OpsCenter **incident / operasyon yönetimi** içindir
- Secret saklamak veya rotation yapmak için **tasarlanmamıştır**
- Credential management çözümü değildir

✗ C. Amazon S3

- S3 güvenli olabilir ama:
 - Otomatik rotation yok
 - Database credential yönetimi için best practice değil
- Manuel ve risklidir

✗ D. KMS ile şifrelenmiş dosyalar

- Şifreleme sağlar ama:
 - Rotation manuel yapılır

- Çok sunuculu mimaride yönetimi zor
- Merkezi ve ölçülebilir değil

SONUÇ

İfade	Doğru Servis
-------	--------------

Credential rotation	AWS Secrets Manager
---------------------	----------------------------

DB password saklama	Secrets Manager
---------------------	------------------------

Secure + IAM access	Secrets Manager
---------------------	------------------------

 “rotate”, “credentials”, “RDS” kelimelerini gördüğünde refleks cevabın **Secrets Manager** olmalı.

QUESTION 22

A company hosts an application on AWS Lambda functions that are invoked by an Amazon API Gateway API. The Lambda functions save customer data to an Amazon Aurora MySQL database. Whenever the company upgrades the database, the Lambda functions fail to establish database connections until the upgrade is complete. The result is that customer data is not recorded for some of the event. A solutions architect needs to design a solution that stores customer data that is created during database upgrades.

Which solution will meet these requirements?

- Provision an Amazon RDS proxy to sit between the Lambda functions and the database. Configure the Lambda functions to connect to the RDS proxy.
- Increase the run time of the Lambda functions to the maximum. Create a retry mechanism in the code that stores the customer data in the database.
- Persist the customer data to Lambda local storage. Configure new Lambda functions to scan the local storage to save the customer data to the database.
- Store the customer data in an Amazon Simple Queue Service (Amazon SQS) FIFO queue. Create a new Lambda function that polls the queue and stores the customer data in the database.

Soru:

Bir şirket, Amazon API Gateway tarafından tetiklenen AWS Lambda fonksiyonları üzerinde çalışan bir uygulama barındırmaktadır. Lambda fonksiyonları, müşteri verilerini Amazon Aurora MySQL veritabanına kaydettmektedir. Şirket veritabanını her

yükselttiğinde (upgrade), Lambda fonksiyonları yükseltme tamamlanana kadar veritabanı bağlantısı kuramamaktadır. Bunun sonucunda, bazı olaylar sırasında oluşturulan müşteri verileri kaydedilememektedir. Bir Solutions Architect, veritabanı yükseltmeleri sırasında oluşturulan müşteri verilerinin saklanması sağlayacak bir çözüm tasarlamak zorundadır. Bu gereksinimleri hangi çözüm karşılar?

- A. Lambda fonksiyonları ile veritabanı arasına bir Amazon RDS Proxy kurar ve Lambda fonksiyonlarını RDS Proxy'ye bağlanacak şekilde yapılandırır.
- B. Lambda fonksiyonlarının çalışma süresini (timeout) maksimuma çıkarır ve kod içinde müşteri verisini veritabanına kaydetmeye çalışan bir yeniden deneme (retry) mekanizması oluşturur.
- C. Müşteri verilerini Lambda'nın yerel depolama alanına (local storage) kaydeder ve bu depolamayı tarayarak müşteri verilerini veritabanına kaydeden yeni Lambda fonksiyonları yapılandırır.
- D. Müşteri verilerini Amazon Simple Queue Service (Amazon SQS) FIFO kuyruğunda saklar ve kuyruğu okuyarak müşteri verilerini veritabanına kaydeden yeni bir Lambda fonksiyonu oluşturur.

Soru Analizi:

Sorunun temel problemi şudur:

- Uygulama **Lambda + API Gateway** mimarisinde çalışıyor
- Veriler **Aurora MySQL** veritabanına yazılıyor
- **Database upgrade** sırasında:
 - Veritabanı **geçici olarak erişilemez**
 - Lambda fonksiyonları **connection kuramıyor**
 - Bu süre boyunca **oluşan müşteri verileri kayboluyor**

Yani burada asıl ihtiyaç:

👉 **Veritabanı geçici olarak kullanılamazken veriyi kaybetmemek**

Bu da AWS mimarisinde şu kavramı çağrıştırır:

- **Decoupling (gevşek bağlı mimari)**
- **Buffer / Queue kullanımı**
- **Asenkron veri işleme**

Seçenek Analizi:

D. Amazon SQS FIFO Queue

- SQS:
 - Veriyi **dayanıklı ve kalıcı** olarak saklar
 - Database kapalıken bile veri kaybolmaz
- FIFO:
 - Sıralı ve **exactly-once processing** sağlar
- Mimari:
 - Lambda → **SQS** → Lambda → Aurora

Database upgrade süresince veri güvenle bekler

A. Amazon RDS Proxy

- RDS Proxy:
 - Connection pooling sağlar
 - Failover sürelerini azaltır
- Ancak:
 - Database upgrade sırasında **veritabanı tamamen kapalıysa**, proxy de yazamaz
 - Veriyi saklamaz, sadece bağlantı yönetir

Data loss problemini çözmez

B. Timeout artırma + Retry

- Retry mekanizması:
 - Kısa süreli hatalarda işe yarar
- Ancak:
 - Upgrade süresi Lambda timeout'tan uzun olabilir
 - Lambda çalışması bittiğinde veri **kaybolur**
 - Ölçeklenebilir ve güvenli değil

Geçici kesintiler için yetersiz

C. Lambda local storage

- Lambda local storage:

- Geçici (ephemeral)'dır
- Instance ölünce veri silinir
- Ayrıca:
 - Başka Lambda'lar bu veriye erişemez
 - Dağıtık mimariye aykırı

→ Kesinlikle yanlış

⌚ SONUÇ

Neden?

- Database geçici olarak erişilemezken:
 - Veri **kaybolmaz**
 - Sistem **çalışmaya devam eder**
- AWS sınavlarında:
 - “**temporary database unavailability**”
 - “**data loss**”
 - “**buffer events**”

İfadelerini gördüğünde refleks çözüm  **SQS (veya Kinesis)**

QUESTION 23

A survey company has gathered data for several years from areas in the United States. The company hosts the data in an Amazon S3 bucket that is 3 TB in size and growing. The company has started to share the data with a European marketing rm that has S3 buckets. The company wants to ensure that its data transfer costs remain as low as possible.

Which solution will meet these requirements?

- A. Configure the Requester Pays feature on the company's S3 bucket.
- B. Configure S3 Cross-Region Replication from the company's S3 bucket to one of the marketing rm's S3 buckets.
- C. Configure cross-account access for the marketing rm so that the marketing rm has access to the company's S3 bucket.

D. Configure the company's S3 bucket to use S3 Intelligent-Tiering. Sync the S3 bucket to one of the marketing firm's S3 buckets.

Soru:

Bir anket (survey) şirketi, Amerika Birleşik Devletleri'ndeki bölgelerden birkaç yıldır veri toplamaktadır. Şirket, boyutu **3 TB olan ve giderek büyüyen** bu verileri bir **Amazon S3 bucket** içinde barındırmaktadır. Şirket, verileri **S3 bucket'ları bulunan Avrupalı bir pazarlama firmasıyla** paylaşmaya başlamıştır. Şirket, **veri aktarım maliyetlerinin mümkün olduğunda düşük kalmasını** istemektedir.

Bu gereksinimleri hangi çözüm karşılar?

- A. Şirketin S3 bucket'ında **Requester Pays** özelliğini yapılandırır.
- B. Şirketin S3 bucket'ından, pazarlama firmasına ait S3 bucket'lardan birine **S3 Cross-Region Replication** yapılandırır.
- C. Pazarlama firması için **cross-account erişim** yapılandırarak, pazarlama firmasının şirketin S3 bucket'ına erişmesini sağlar.
- D. Şirketin S3 bucket'ını **S3 Intelligent-Tiering** kullanacak şekilde yapılandırır ve S3 bucket'ı pazarlama firmasına ait S3 bucket'lardan birine **senkronize eder (sync)**.

Soru Analizi:

Soruda öne çıkan **kritik noktalar** şunlar:

- Veri **Amazon S3**'te tutuluyor
- Veri boyutu **çok büyük (3 TB ve büyüyor)**
- Veri **başka bir şirkete (Avrupa'daki pazarlama firması)** paylaşılıyor
- **En önemli gereksinim:**
👉 **Data transfer maliyetlerinin mümkün olduğunda düşük olması**

AWS'de önemli bir sınav kuralı:

S3'ten veri indiren taraf normalde data transfer ücretini öder.

Şirket, **kendi cebinden veri transfer ücreti ödemek istemiyor.**

Seçenek Analizi:

A. Requester Pays

- **Requester Pays** özelliği:
 - S3 bucket'tan veri indiren taraf **data transfer ücretini öder**
 - Bucket sahibi **transfer maliyetinden kurtulur**

- Pazarlama firması veriyi kendi S3 bucket'ına veya sistemine çekerken:
 - **Ücreti kendisi öder**

📌 Soruda açıkça:

"ensure that its data transfer costs remain as low as possible"

➡ Bu ifade **Requester Pays**'ı doğrudan işaret eder.

✗ B. S3 Cross-Region Replication (CRR)

- CRR:
 - Sürekli ve otomatik veri kopyalar
- Ancak:
 - **Cross-region data transfer ücreti vardır**
 - Tüm veriyi sürekli kopyalamak **çok pahalıdır**
- Ayrıca:
 - Pazarlama firmasının her erişimi için yine maliyet oluşur

➡ Maliyetleri düşürmez, artırır

✗ C. Cross-account access

- Cross-account access:
 - Erişim kontrolü sağlar
- Ancak:
 - Veri indirildiğinde **transfer ücretini yine bucket sahibi öder**
 - Maliyet paylaşımı yok

➡ Güvenlik çözümü, maliyet çözümü değil

✗ D. Intelligent-Tiering + Sync

- Intelligent-Tiering:
 - **Storage maliyetini optimize eder**
- Ancak:
 - **Data transfer maliyetini düşürmez**
 - Sync işlemi:

- Büyük veri transferi
- Sürekli maliyet

➡ Sorunun hedeflediği problem **storage değil, transfer maliyetidir**

Sonuç



SONUÇ

Neden?

- Büyük veri seti
- Harici bir firma veriyi indiriyor
- Transfer maliyetini **karşı tarafa yüklemek isteniyor**

👉 AWS sınav refleksi:

“S3 + data sharing + cost concern” → Requester Pays

QUESTION 24

A company uses Amazon S3 to store its confidential audit documents. The S3 bucket uses bucket policies to restrict access to audit team IAM user credentials according to the principle of least privilege. Company managers are worried about accidental deletion of documents in the S3 bucket and want a more secure solution.

What should a solutions architect do to secure the audit documents?

- Enable the versioning and MFA Delete features on the S3 bucket.
- Enable multi-factor authentication (MFA) on the IAM user credentials for each audit team IAM user account.
- Add an S3 Lifecycle policy to the audit team's IAM user accounts to deny the s3:DeleteObject action during audit dates.
- Use AWS Key Management Service (AWS KMS) to encrypt the S3 bucket and restrict audit team IAM user accounts from accessing the KMS key.

Soru:

Bir şirket, **gizli denetim (audit) belgelerini saklamak için Amazon S3 kullanmaktadır**. S3 bucket, **en az ayrıcalık (least privilege) prensibine göre**, denetim ekibine ait **IAM kullanıcı kimlik bilgilerine erişimi kısıtlamak için bucket policy'ler kullanmaktadır**. Şirket yöneticileri, S3 bucket içindeki belgelerin **yanlışlıkla silinmesinden endişe etmektedir ve daha güvenli bir çözüm istemektedir**.

Bir Solutions Architect, denetim belgelerini güvence altına almak için ne yapmalıdır?

- A. S3 bucket üzerinde **Versioning** ve **MFA Delete** özelliklerini etkinleştirir.
- B. Denetim ekibindeki her IAM kullanıcı hesabı için **çok faktörlü kimlik doğrulamayı (MFA)** etkinleştirir.
- C. Denetim ekibine ait IAM kullanıcı hesapları için, denetim tarihleri boyunca **s3:DeleteObject** eylemini reddeden bir **S3 Lifecycle policy** ekler.
- D. S3 bucket'ı **AWS Key Management Service (AWS KMS)** ile şifreler ve denetim ekibine ait IAM kullanıcı hesaplarının **KMS anahtarlarına erişimini kısıtlar**.

Soru Analizi:

Sorudaki **kritik gereksinimler**:

- Veri **Amazon S3**'te tutuluyor
- Belgeler **gizli (confidential audit documents)**
- Erişim zaten:
 - Bucket policy + IAM
 - **Least privilege** prensibine uygun
- **Yeni problem:**
 - 👉 Belgelerin yanlışlıkla silinmesi (**accidental deletion**)
- İstenen:
 - 👉 **Daha güvenli bir çözüm** (silinmeye karşı koruma)

📌 Yani odak noktası:

- **Silme işlemlerini engellemek veya geri alınabilir hale getirmek**

Seçenek Analizi:

- A. **S3 Versioning + MFA Delete**
- **Versioning:**
 - Dosya silinse bile **eski versiyonlar korunur**
 - Veri **geri yüklenebilir**
 - **MFA Delete:**
 - Version silme veya kalıcı silme için **MFA zorunlu**
 - Yanlışlıkla silme riskini **çok ciddi azaltır**

→ AWS'de **accidental deletion protection** denince **en güçlü çözüm**

✗ B. IAM kullanıcılarında MFA

- MFA:
 - Kimlik doğrulama güvenliğini artırır
- Ancak:
 - Yetkili bir kullanıcı **yanlışlıkla delete yapabilir**
 - Silme işlemini engellemeye veya geri alamaz

→ Problem erişim değil, **silme riski**

✗ C. S3 Lifecycle policy

- Lifecycle policy:
 - **Otomatik** geçiş veya silme içindir
- IAM kullanıcılarına uygulanmaz
- Delete engelleme amacıyla kullanılamaz

→ Kavramsal olarak yanlış

✗ D. KMS encryption

- KMS:
 - **Veriyi şifreler**
- Ancak:
 - Silinmiş veriyi geri getirmez
 - Yanlışlıkla silmeyi engellemeye

→ Güvenlik = şifreleme değildir (her zaman)

Sonuç

🎯 **SONUÇ**

Neden?

- Yanlışlıkla silmeye karşı:
 - **Geri alınabilirlik**
 - **Ek doğrulama (MFA Delete)**

- AWS sınavlarında:
 - “**accidental deletion**”
 - “**protect S3 objects**”

İfadeleri birlikteyse refleks çözüm:

👉 **S3 Versioning + MFA Delete**

QUESTION 25

A company is using a SQL database to store movie data that is publicly accessible. The database runs on an Amazon RDS Single-AZ DB instance. A script runs queries at random intervals each day to record the number of new movies that have been added to the database. The script must report a final total during business hours. The company's development team notices that the database performance is inadequate for development tasks when the script is running. A solutions architect must recommend a solution to resolve this issue.

Which solution will meet this requirement with the LEAST operational overhead?

- A. Modify the DB instance to be a Multi-AZ deployment.
- B. Create a read replica of the database. Configure the script to query only the read replica.
- C. Instruct the development team to manually export the entries in the database at the end of each day.
- D. Use Amazon ElastiCache to cache the common queries that the script runs against the database.

Soru:

Bir şirket, herkese açık (publicly accessible) olan film verilerini saklamak için bir **SQL veritabanı** kullanmaktadır. Veritabanı, **Amazon RDS Single-AZ DB** instance üzerinde çalışmaktadır. Her gün rastgele zamanlarda çalışan bir betik (script), veritabanına eklenen **yeni film sayısını** kaydetmek için sorgular çalıştırmaktadır. Bu betik, **iş saatleri içinde** nihai (final) bir toplam raporlamak zorundadır. Şirketin geliştirme ekibi, betik çalışırken veritabanı performansının **geliştirme işleri için yetersiz kaldığını** fark etmiştir. Bir Solutions Architect, bu sorunu çözecek bir çözüm önermek zorundadır.

En AZ operasyonel yük (LEAST operational overhead) ile bu gereksinimi hangi çözüm karşılar?

- A. Veritabanı instance'ını **Multi-AZ** olacak şekilde değiştirir.

- B. Veritabanının bir **read replica**'sını oluşturur ve betiği yalnızca **read replica** üzerinden sorgu çalıştıracak şekilde yapılandırır.
- C. Geliştirme ekibine, her günün sonunda veritabanındaki kayıtları **manuel olarak dışa aktarmalarını** söyler.
- D. Betiğin veritabanına karşı çalıştırıldığı yaygın sorguları önbelleğe almak için **Amazon ElastiCache** kullanır.

Soru Analizi:

Sorudaki **temel noktalar**:

- Veritabanı: **Amazon RDS Single-AZ**
- Veritabanı **okuma ağırlıklı** (script sadece sorgu çalıştırıyor)
- Script:
 - Gün içinde **rastgele zamanlarda** çalışıyor
 - **Yeni eklenen film sayısını** sorguluyor (READ işlemi)
 - İş saatlerinde **final bir toplam** raporlamak zorunda
- Problem:
 - Script çalışırken **development performansı düşüyor**
- Kritik gereksinim:
 - 👉 **En az operasyonel yük (LEAST operational overhead)**

Yani amaç:

- Ana veritabanının üzerindeki **okuma yükünü azaltmak**
- Bunu **basit, yönetimi kolay** bir şekilde yapmak

Seçenek Analizi:

B. Read replica oluşturmak

- Read replica:
 - **Okuma trafiğini** ana DB'den alır
 - Script yalnızca replica'yi sorgular
- Avantajlar:
 - Development işlemleri **etkilenmez**
 - Otomatik replikasyon

- Yönetimi kolay → **düşük operasyonel yük**
- AWS sınav refleksi:

“Read-heavy workload” → *Read Replica*

✗ A. Multi-AZ'ye geçmek

- Multi-AZ:
 - **High availability** içindir
 - Standby instance **read almaz**
- Okuma yükünü azaltmaz
- Performans sorununu çözmez

→ Yanlış kullanım senaryosu

✗ C. Manuel export

- Manuel işlem:
 - Operasyonel yük **çok yüksek**
 - Otomasyon yok
- Sınavlarda neredeyse her zaman yanlış

✗ D. ElastiCache kullanmak

- ElastiCache:
 - Sık tekrar eden **aynı sorgular** için uygundur
- Ancak:
 - Script **rastgele zamanlarda**
 - Sayım verisi **dinamik olarak değişiyor**
- Ek altyapı ve bakım → daha fazla operasyonel yük

⌚ Sonuç

Neden?

- Okuma ağırlıklı iş yükü
- Ana DB performansı düşüyor
- **En az operasyonel yükle çözüm:**

👉 Read Replica

Refleks Çözüm

Anahtar Kelime

Read-heavy workload

Read Replica

Dev DB yavaşlıyor

Read Replica

Least operational overhead

Managed AWS service

QUESTION 26

A company has applications that run on Amazon EC2 instances in a VPC. One of the applications needs to call the Amazon S3 API to store and read objects. According to the company's security regulations, no traffic from the applications is allowed to travel across the internet.

Which solution will meet these requirements?

- Configure an S3 gateway endpoint.
- Create an S3 bucket in a private subnet.
- Create an S3 bucket in the same AWS Region as the EC2 instances.
- Configure a NAT gateway in the same subnet as the EC2 instances.

Soru:

Bir şirketin, bir VPC içinde **Amazon EC2 instance'ları** üzerinde çalışan uygulamaları vardır. Uygulamalardan biri, nesneleri saklamak ve okumak için **Amazon S3 API'sini** çağırmak zorundadır. Şirketin güvenlik düzenlemelerine göre, uygulamalardan çıkan **hiçbir trafiğin internet üzerinden geçmesine izin verilmemektedir**.

Bu gereksinimleri hangi çözüm karşılar?

- A Bir **S3 gateway endpoint** yapılandırılır.
- B **Private subnet** içinde bir S3 bucket oluşturur.

C. EC2 instance'ları ile **aynı AWS Region** içinde bir S3 bucket oluşturur.

D. EC2 instance'ları ile **aynı subnet** içinde bir **NAT gateway** yapılandırır.

Sorunun Analizi:

Sorudaki **kritik gereksinimler**:

- Uygulamalar **VPC içindeki EC2 instance'larında** çalışıyor
- Uygulama **Amazon S3 API** çağrısı yapacak
- **Güvenlik kuralı:**
 - 👉 Uygulama trafiği **INTERNET** üzerinden geçmemeli
- Yani:
 - **Public internet**
 - **IGW / NAT** üzerinden çıkış
 - ✖ **yasak**

AWS bilgisi:

- Amazon S3 **VPC içinde** değildir
- Varsayılan erişim **internet üzerinden** olur

Bu nedenle özel bir VPC bağlantısı gereklidir.

Seçenek Analizi:

A. S3 Gateway Endpoint

- **Gateway Endpoint (VPC Endpoint for S3):**
 - EC2 → S3 trafiği **AWS backbone** üzerinden gider
 - Internet, IGW, NAT kullanılmaz
- Güvenlik:
 - Bucket policy ile **sadece endpoint** üzerinden erişim zorlanabilir
- AWS sınav refleksi:

“VPC → S3 without internet” → **S3 Gateway Endpoint**

B. Private subnet'te S3 bucket

- S3 bucket:
 - Subnet içinde oluşturulamaz

- VPC kaynağı değildir

→ Kavramsal olarak yanlış

✗ C. Aynı Region'da S3 bucket

- Aynı region:
 - Latency açısından faydalı
- Ancak:
 - Trafik yine internet üzerinden gider

→ Güvenlik gereksinimini karşılamaz

✗ D. NAT Gateway

- NAT Gateway:
 - Private subnet → **internet çıkışı** sağlar
- Ama:
 - Trafik internetten geçer
 - Güvenlik kuralına aykırı

→ Tam tersi çözüm

⌚ Sonuç

Neden?

- Internet yasak
- S3 erişimi şart
- Çözüm:
 - 👉 **VPC Endpoint (Gateway Endpoint for S3)**

Sınav Altın Kuralı

Senaryo **Çözüm**

EC2 → S3, internet yasak **S3 Gateway Endpoint**

EC2 → AWS API, private **VPC Endpoint**

QUESTION 27

A company is storing sensitive user information in an Amazon S3 bucket. The company wants to provide secure access to this bucket from the application tier running on Amazon EC2 instances inside a VPC.

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

- A. Configure a VPC gateway endpoint for Amazon S3 within the VPC.
- B. Create a bucket policy to make the objects in the S3 bucket public.
- C. Create a bucket policy that limits access to only the application tier running in the VPC.
- D. Create an IAM user with an S3 access policy and copy the IAM credentials to the EC2 instance.
- E. Create a NAT instance and have the EC2 instances use the NAT instance to access the S3 bucket.

Soru:

Bir şirket, **hassas kullanıcı bilgilerini** bir **Amazon S3 bucket** içinde saklamaktadır. Şirket, bu bucket'a **VPC içinde çalışan Amazon EC2 instance'ları üzerindeki uygulama katmanından (application tier)** güvenli erişim sağlamak istemektedir.

Bir Solutions Architect, bunu gerçekleştirmek için aşağıdaki adımlardan hangilerinin **kombinasyonunu** uygulamalıdır?

(İki seçenek seçiniz.)

- A. VPC içinde Amazon S3 için bir **VPC gateway endpoint** yapılandırır.
- B. S3 bucket içindeki nesneleri **herkese açık (public)** hale getiren bir **bucket policy** oluşturur.
- C. Erişimi yalnızca VPC içinde çalışan **uygulama katmanıyla sınırlayan** bir **bucket policy** oluşturur.
- D. S3 erişim politikasına sahip bir **IAM kullanıcı** oluşturur ve bu IAM kimlik bilgilerini EC2 instance'lara kopyalar.
- E. Bir **NAT instance** oluşturur ve EC2 instance'ların S3 bucket'a erişmek için bu NAT instance'ı kullanmasını sağlar.

Sorunun Analizi:

Sorunun Özeti (Ne isteniyor?)

- **Hassas kullanıcı verileri** Amazon **S3 bucket** içinde tutuluyor.

- Bu bucket'a yalnızca VPC içinde çalışan EC2'lerdeki application tier erişebilmeli.
- Erişim **güvenli** olmalı.
- **İki seçenek** seçilecek.

Yani amaç:

- İnternete çıkmadan S3'e erişmek
- Erişimi sadece belirli bir VPC / uygulama katmanıyla sınırlamak
- AWS best practice'lere uygun olmak

Seçenek Analizi:

A. VPC içinde Amazon S3 için bir VPC gateway endpoint yapılandırılır.

Bu neden doğru?

- **VPC Gateway Endpoint (S3 endpoint)** sayesinde:
 - EC2 → S3 trafiği **AWS ağı içinde kalır**
 - **Internet Gateway, NAT Gateway veya NAT Instance** gerekmez
 - Daha **güvenli**, daha **ucuz**, daha **ölçeklenebilir**

AWS, S3 erişimi için **VPC Gateway Endpoint'i best practice** olarak önerir.

Güvenli ve özel ağ üzerinden erişim sağladığı için doğru

C. Erişimi yalnızca VPC içinde çalışan uygulama katmanıyla sınırlayan bir bucket policy oluşturur.

Bu neden doğru?

- Bucket policy ile:
 - Belirli **VPC endpoint ID** (aws:SourceVpce)
 - Veya belirli **IAM role / instance profile**
 - Veya belirli **VPC CIDR**
ile erişim sınırlanabilir

Örnek mantık:

“Bu bucket'a **sadece benim VPC'inden** ve **sadece application tier'dan** erişilsin.”

En kritik güvenlik kontrolü burada sağlanır

✗ B. S3 bucket içindeki nesneleri herkese açık (public) hale getiren bir bucket policy oluşturur.

Neden yanlış?

- Hassas kullanıcı verileri var
- Public erişim:
 - Güvenlik ihlali riski
 - Compliance (GDPR, ISO, SOC vb.) ihlali
- Soru zaten “**güvenli erişim**” istiyor

→ Tam tersine yapılmaması gereken bir seçenek

✗ D. S3 erişim politikasına sahip bir IAM kullanıcı oluşturur ve bu IAM kimlik bilgilerini EC2 instance'lara kopyalar.

Neden yanlış?

- IAM user access key'lerini EC2'ye koymak:
 - Güvenlik riski
 - Credential sızıntısı riski
- AWS best practice:
 - **IAM Role (Instance Profile)** kullan
 - **Asla access key kopyalama**

→ Yanlış ve deprecated bir yaklaşım

✗ E. Bir NAT instance oluşturur ve EC2 instance'ların S3 bucket'a erişmek için bu NAT instance'ı kullanmasını sağlar.

Neden yanlış?

- NAT instance:
 - Ek yönetim yükü
 - Ölçeklenebilir değil
 - Yüksek maliyet
- S3 için **VPC Gateway Endpoint** varken NAT kullanmak:
 - Gereksiz
 - AWS best practice'e aykırı

➡ Eski ve verimsiz bir çözüm

⌚ Sonuç

Doğru kombinasyon:

👉 A + C

Neden?

- **A:** S3'e internet çıkışı olmadan, özel ve güvenli erişim sağlar
- **C:** Erişimi sadece VPC içindeki application tier ile sınırlar

Bu ikisi birlikte:

- En yüksek güvenlik
- AWS best practice
- Least privilege prensibi
- Düşük maliyet & yüksek performans

QUESTION 28

A company runs an on-premises application that is powered by a MySQL database. The company is migrating the application to AWS to increase the application's elasticity and availability.

The current architecture shows heavy read activity on the database during times of normal operation. Every 4 hours, the company's development team pulls a full export of the production database to populate a database in the staging environment. During this period, users experience unacceptable application latency. The development team is unable to use the staging environment until the procedure completes.

A solutions architect must recommend replacement architecture that alleviates the application latency issue. The replacement architecture also must give the development team the ability to continue using the staging environment without delay.

Which solution meets these requirements?

A. Use Amazon Aurora MySQL with Multi-AZ Aurora Replicas for production. Populate the staging database by implementing a backup and restore process that uses the mysqldump utility.

B. Use Amazon Aurora MySQL with Multi-AZ Aurora Replicas for production. Use database cloning to create the staging database on-demand.

C. Use Amazon RDS for MySQL with a Multi-AZ deployment and read replicas for production. Use the standby instance for the staging database.

D. Use Amazon RDS for MySQL with a Multi-AZ deployment and read replicas for production. Populate the staging database by implementing a backup and restore process that uses the mysqldump utility.

Soru:

Bir şirket, **on-premises (şirket içi)** ortamda çalışan ve **MySQL veritabanı** kullanan bir uygulamaya sahiptir. Şirket, uygulamanın **esnekliğini (elasticity)** ve **erişilebilirliğini (availability)** artırmak için bu uygulamayı **AWS'e taşımaktadır**.

Mevcut mimaride, normal çalışma sırasında veritabanı üzerinde **yoğun okuma (read) trafiği** bulunmaktadır. **Her 4 saatte bir**, şirketin geliştirme ekibi, **staging ortamındaki veritabanını doldurmak için production veritabanının tam bir dışa aktarımını (full export)** almaktadır. Bu süreç sırasında, kullanıcılar **kabul edilemez seviyede uygulama gecikmesi (latency)** yaşamaktadır. Geliştirme ekibi, bu işlem tamamlanana kadar **staging ortamını kullanamamaktadır**.

Bir **Solutions Architect**, bu uygulama gecikmesi sorununu **ortadan kaldıracak bir yerine geçecek mimarı** önermelidir. Bu yeni mimari ayrıca, geliştirme ekibine **staging ortamını gecikme olmadan kullanmaya devam edebilme** imkânı sağlamalıdır.

Aşağıdaki çözümlerden hangisi bu gereksinimleri karşılar?

A. Production ortamı için Multi-AZ Aurora Replica'lara sahip Amazon Aurora MySQL kullanılır.

Staging veritabanı, **mysqldump aracıyla yedekleme ve geri yükleme (backup & restore)** süreci uygulanarak oluşturulur.

B. Production ortamı için Multi-AZ Aurora Replica'lara sahip Amazon Aurora MySQL kullanılır.

Staging veritabanı, **veritabanı klonlama (database cloning)** yöntemiyle isteğe bağlı (on-demand) olarak oluşturulur.

C. Production ortamı için Multi-AZ deployment ve read replica'lara sahip Amazon RDS for MySQL kullanılır.

Standby instance, staging veritabanı olarak kullanılır.

D. Production ortamı için Multi-AZ deployment ve read replica'lara sahip Amazon RDS for MySQL kullanılır.

Staging veritabanı, **mysqldump aracıyla yedekleme ve geri yükleme** süreci uygulanarak oluşturulur.

Sorunun Analizi:

Soruda özellikle şu ifadeler çok önemli:

1. **Heavy read activity**
2. **Her 4 saatte bir production DB'nin full export'u alınıyor**
3. Bu sırada:
 - **Production latency artıyor**
 - **Staging kullanılamıyor**
4. Yeni mimari:
 - Production üzerindeki **latency sorununu** **çözmeli**
 - **Staging ortamı gecikme olmadan kullanılabilirmeli**

Yani:

- Production DB **yük altında kalmamalı**
- Staging DB **hızlı ve anında** oluşturulabilmeli

Seçenek Analizi:

B. Aurora MySQL + Database Cloning

Neden doğru?

- **Aurora Database Cloning:**
 - Saniyeler içinde staging DB oluşturur
 - **Copy-on-write** mantığıyla çalışır
 - Production DB'ye **neredeyse hiç ek yük bindirmez**
- Geliştiriciler:
 - Beklemeden staging'i kullanabilir
- Production:
 - Okuma yükü artmaz
 - Latency problemi ortadan kalkar

 **Bu özellik sadece Aurora'da vardır (RDS MySQL'de yok)**

 **Sorunun tüm gereksinimlerini karşılayan tek seçenek**

 **A. Aurora + mysqldump (backup & restore)**

Neden yanlış?

- mysqldump:
 - Logical backup alır
 - Production DB üzerinde **yük oluşturur**
 - Zaman alır
- Sorunun **kök nedeni zaten bu**
- Aurora kullanmak tek başına yeterli değil

➡ Latency sorununu çözmez

✗ C. RDS MySQL Multi-AZ + standby'ı staging olarak kullanmak

Neden yanlış?

- Standby instance:
 - **Pasiftir**
 - Read / write yapılamaz
- AWS:

Standby instance **sadece failover içindir**

➡ Staging olarak kullanılamaz

✗ D. RDS MySQL + mysqldump

Neden yanlış?

- Yine mysqldump
- Yine production üzerinde yük
- Yine gecikme

➡ Mevcut sorunun aynısını AWS'te tekrar eder

🎯 Sonuç

Use Amazon Aurora MySQL with Multi-AZ Aurora Replicas for production.
Use database cloning to create the staging database on-demand.

🧠 Sınavda Hatırlatma

Bu kelimeleri gördüğünde **otomatik alarm çalmalı** 🚨 :

- *Frequent full copy of production DB*

- *Staging needs to be ready immediately*
- *Latency during backup/export*
- *MySQL → AWS migration*

→ Aurora Cloning

QUESTION 29

A company is designing an application where users upload small files into Amazon S3. After a user uploads a file, the file requires one-time simple processing to transform the data and save the data in JSON format for later analysis.

Each file must be processed as quickly as possible after it is uploaded. Demand will vary. On some days, users will upload a high number of files. On other days, users will upload a few files or no files.

Which solution meets these requirements with the LEAST operational overhead?

- A. Configure Amazon EMR to read text files from Amazon S3. Run processing scripts to transform the data. Store the resulting JSON file in an Amazon Aurora DB cluster.
- B. Configure Amazon S3 to send an event notification to an Amazon Simple Queue Service (Amazon SQS) queue. Use Amazon EC2 instances to read from the queue and process the data. Store the resulting JSON file in Amazon DynamoDB.
- C. Configure Amazon S3 to send an event notification to an Amazon Simple Queue Service (Amazon SQS) queue. Use an AWS Lambda function to read from the queue and process the data. Store the resulting JSON file in Amazon DynamoDB.
- D. Configure Amazon EventBridge (Amazon CloudWatch Events) to send an event to Amazon Kinesis Data Streams when a new file is uploaded. Use an AWS Lambda function to consume the event from the stream and process the data. Store the resulting JSON file in an Amazon Aurora DB cluster.

Soru:

Bir şirket, kullanıcıların küçük dosyalar yüklediği bir uygulama tasarlamaktadır. Bir kullanıcı dosya yükledikten sonra, dosya bir kez basit bir işleme tabi tutulmalı, veriler dönüştürülmeli ve daha sonra analiz için JSON formatında kaydedilmelidir.

Her dosya, yüklenir yüklenmez **mümkün olan en hızlı şekilde işlenmelidir**. Talep değişken olacaktır. Bazı günlerde kullanıcılar **yüksek sayıda dosya yükleyecek**, diğer günlerde ise **birkaç dosya ya da hiç dosya yüklemeyeceklerdir**.

Hangi çözüm, **en az operasyonel yükle** bu gereksinimleri karşılar?

- A.** Amazon EMR'yi, Amazon S3'ten metin dosyalarını okuması için yapılandırın. Veriyi dönüştürmek için işleme betikleri çalıştırın. Sonuçta elde edilen JSON dosyasını Amazon Aurora DB kümesine kaydedin.
- B.** Amazon S3'ü, bir Amazon Simple Queue Service (Amazon SQS) kuyruğuna olay bildirimi gönderecek şekilde yapılandırın. Verileri işlemek için Amazon EC2 instance'larını kullanarak kuyruktan okuyun. Sonuçta elde edilen JSON dosyasını Amazon DynamoDB'ye kaydedin.
- C.** Amazon S3'ü, bir Amazon Simple Queue Service (Amazon SQS) kuyruğuna olay bildirimi gönderecek şekilde yapılandırın. Verileri işlemek için bir AWS Lambda fonksiyonu kullanarak kuyruktan okuyun. Sonuçta elde edilen JSON dosyasını Amazon DynamoDB'ye kaydedin.
- D.** Amazon EventBridge (Amazon CloudWatch Events) ile, yeni bir dosya yüklenliğinde bir olay gönderecek şekilde yapılandırın. Bir AWS Lambda fonksiyonu kullanarak bu olayı veri akışından (stream) tüketin ve veriyi işleyin. Sonuçta elde edilen JSON dosyasını Amazon Aurora DB kümesine kaydedin.

Sorunun Analizi:

Kullanıcıların küçük dosyalar yüklemesi: Dosyalar genellikle **küçük boyutlu** olacaktır.

Dosyanın hızlı bir şekilde işlenmesi: Her dosya **yüklenir yüklenmez** işleme alınmalı ve dönüştürülmelidir.

İşlem sonrasında JSON formatında kaydedilmesi: Veriler işlendikten sonra, analiz için **JSON formatına dönüştürülüp saklanmalıdır**.

Talep değişkenliği: Dosya yükleme sayısı değişkenlik gösterir; bazı günlerde yoğun, bazı günlerde ise daha az yükleme olabilir. Bu, **esnek ve otomatik bir çözüm** gerektirir.

En az operasyonel yük: Çözümün mümkün olan en düşük operasyonel yükle çalışması isteniyor. Bu, **yönetimi kolay, otomatikleştirilmiş** ve **esnek** bir çözüm anlamına gelir.

Seçenek Analizi:

C. S3, SQS ve Lambda ile dosya işleme

- **S3 ile olay bildirimleri** yine kullanılarak her dosya yüklenliğinde bir **SQS kuyruğuna** mesaj gönderilir.
- Bu mesajlar, **AWS Lambda** fonksiyonu tarafından işlenir. Lambda fonksiyonu dosyayı alır, işler ve JSON formatında kaydeder.
- **AWS Lambda**, sunucusuz bir çözüm olduğu için **operasyonel yükü en aza indirir**. Lambda, sadece işlem başına **faturalandırılır ve ölcəklendirme otomatik** olarak yapılır.

- Lambda'nın **çok hızlı** çalışması ve **esnekliği**, bu çözümün **yükselen taleplerle başa çıkmasını** sağlar. Ayrıca, **düşük maliyetli** ve **kolay yönetilebilir** bir çözümüdür.

Sonuç: Bu çözüm **operasyonel yükü en az tutarak** gereksinimleri karşılar, çünkü **Lambda** ile tam bir sunucusuz çözüm sunulmaktadır.

A. Amazon EMR kullanarak dosya okuma ve işleme

- **Amazon EMR** (Elastic MapReduce), büyük verileri işlemek için kullanılır ve genellikle **dağıtık veri işleme** için uygundur.
- **Yüksek işlem gücü** gerektiren, büyük veri kümeleriyle çalışırken iyi bir çözüm olabilir.
- Ancak, bu senaryoda dosyaların **küçük ve işlem yükünün değişken olduğu** göz önüne alındığında, **Amazon EMR** aşırı derecede karmaşık ve maliyetli olabilir. Ayrıca, **operasyonel yükü** oldukça yüksek olur, çünkü EMR kümelerinin yönetimi ve yapılandırılması, özellikle küçük ve sık değişen taleplerle birlikte, gereksiz bir iş yükü yaratır.

Sonuç: Aşırı karmaşık ve operasyonel yükü fazla olduğu için bu seçenek uygun değildir.

B. S3, SQS ve EC2 ile dosya işleme

- **S3'te olay bildirimleri** kullanarak her dosya yüklemesinin ardından bir **SQS kuyruğuna** bildirim gönderebilir.
- **Amazon EC2 instance'ları**, kuyruğa gelen dosya yükleme bildirimlerini alır ve dosyaları işleyip JSON formatına dönüştürür.
- **EC2 kullanımında**, her işlem için manuel bir sunucu (EC2 instance) yönetimi gereklidir. Bu da **operasyonel yük ve maliyet** açısından yüksek bir çözüm olacaktır, çünkü EC2'yi yönetmek, ölçeklendirmek ve izlemek gereklidir.

Sonuç: Yüksek operasyonel yük ve yönetim gereksinimi nedeniyle bu seçenek de uygun değildir.

D. EventBridge ve Lambda ile dosya işleme

- **Amazon EventBridge**, bir olay yöneticisidir ve burada dosya yükleme olaylarını izler, ardından **Kinesis Data Streams** üzerinden bu olayları işler.
- **Lambda**, Kinesis'ten olayları alır, veriyi işler ve sonucu kaydeder.
- Ancak, bu çözümde **Kinesis Data Streams** kullanmak gereksiz karmaşıklık yaratabilir. Kinesis daha çok **gerçek zamanlı veri akışlarını** işlemek için kullanılır

ve burada küçük dosyalar üzerinde yapılan işlem için gereksiz bir ek yapı oluşturur.

- Ayrıca, EventBridge ve Kinesis ile yapılandırma **daha karmaşıktır** ve bu basit dosya işleme ihtiyacını karşılamak için gereksiz aşamalara sahiptir.

Sonuç: Karmaşık ve gereksiz ek yapılandırmalar olduğu için bu çözüm de en uygun seçenek değildir.

Sonuç

Neden doğru?

- **Sunucusuz (serverless) çözüm:** Lambda, EC2'ye kıyasla **çok daha az operasyonel yük** getirir.
- **Otomatik ölçeklenebilirlik:** Lambda, yüksek talep anlarında otomatik olarak ölçeklenir, böylece sistem **esnek bir şekilde çalışır**.
- **Maliyet etkinlik:** Lambda, sadece işlem başına **faturalandırılır**. EC2 gibi sürekli çalışan sunuculara gerek yoktur.
- **Basit ve yönetimi kolay:** S3, SQS ve Lambda arasındaki entegrasyon oldukça basit ve AWS'nin sunucusuz ekosisteminin gücünü kullanarak hızlı bir çözüm sunar.

En uygun çözüm: C (S3, SQS ve Lambda) çünkü:

- **Düşük operasyonel yük**
- **Otomatik ölçeklenebilirlik**
- **Maliyet etkinliği** sağlar
- **Basit yapılandırma ve yönetim** gerektirir

Bu çözüm, gereksinimleri karşılamak için **en verimli ve basit** yol olacaktır.

QUESTION 30

An application allows users at a company's headquarters to access product data. The product data is stored in an Amazon RDS MySQL DB instance. The operations team has isolated an application performance slowdown and wants to separate read traffic from write traffic. A solutions architect needs to optimize the application's performance quickly.

What should the solutions architect recommend?

- A. Change the existing database to a Multi-AZ deployment. Serve the read requests from the primary Availability Zone.
- B. Change the existing database to a Multi-AZ deployment. Serve the read requests from the secondary Availability Zone.
- C. Create read replicas for the database. Configure the read replicas with half of the compute and storage resources as the source database.
- D. Create read replicas for the database. Configure the read replicas with the same compute and storage resources as the source database.

Soru:

Bir uygulama, bir şirketin merkez ofisindeki kullanıcıların **ürün verilerine erişmesine** olanak tanımaktadır. Ürün verileri, **Amazon RDS MySQL DB instance** içinde saklanmaktadır. Operasyon ekibi, uygulamada bir **performans yavaşlaması** tespit etmiş ve **okuma (read) trafiğini yazma (write) trafiğinden ayırmak** istemektedir. Bir **Solutions Architect**, uygulamanın performansını **hızlı bir şekilde optimize etmek** zorundadır.

Solutions Architect neyi önermelidir?

- A. Mevcut veritabanını **Multi-AZ deployment** olacak şekilde değiştirin. Okuma isteklerini **primary Availability Zone** üzerinden karşılayın.
- B. Mevcut veritabanını **Multi-AZ deployment** olacak şekilde değiştirin. Okuma isteklerini **secondary Availability Zone** üzerinden karşılayın.
- C. Veritabanı için **read replica'lar oluşturun**. Read replica'ları, kaynak (source) veritabanının **yarısı kadar compute ve storage kaynağı** ile yapılandırın.
- D. Veritabanı için **read replica'lar oluşturun**. Read replica'ları, kaynak veritabanıyla **aynı compute ve storage kaynakları** ile yapılandırın.

Sorunun Analizi:

Sorudaki kritik ifadeler:

1. **Application performance slowdown**
2. **Read traffic ile write traffic'i ayırmak istiyorlar**
3. **Hızlı (quickly) optimize etmek gerekiyor**
4. **Amazon RDS MySQL kullanılıyor**

👉 Buradan çıkarımlar:

- Amaç **high availability** değil, **performans**

- Özellikle **read-heavy** bir workload var
- Çözüm **read scaling** sağlamalı

Seçenek Analizi:

D. Read replica + aynı compute & storage

Neden doğru?

- Read replica:
 - Read traffic’i write traffic’ten ayırrır
 - Read scaling sağlar
- Aynı kapasite:
 - Performans darboğazı oluşmaz
 - Hızlı ve güvenli optimizasyon
- Uygulama:
 - Write → primary DB
 - Read → read replica’lar

 Bu tam olarak AWS’nin önerdiği pattern’dir

A. Multi-AZ + okumaları primary AZ’den servis etmek

Neden yanlış?

- Multi-AZ:
 - Yüksek erişilebilirlik (HA) içindir
 - Performans veya read scaling sağlamaz
- Read’ler hâlâ **primary instance** üzerinden gider
- Read / write ayrimı yapılmaz

 Performans sorununu çözmez

B. Multi-AZ + okumaları secondary AZ’den servis etmek

Neden yanlış?

- RDS Multi-AZ:
 - Secondary instance **pasiftir**
 - Read / write trafigi alamaz

- Secondary sadece **failover** içindir

→ **Teknik olarak mümkün değil**

✗ C. Read replica + yarı kaynakla yapılandırmak

Neden yanlış?

- Read replica:
 - Gerçekten production read trafiğini taşıyacak
- Kaynakları yarıya düşürmek:
 - Bottleneck oluşturabilir
 - Performansı garanti etmez
- Soruda:

“Optimize performance quickly”

- Under-provisioning risklidir

→ **Sınav bakış açısından riskli ve eksik çözüm**

🎯 **Sonuç**

🧠 **Sınav İpuçları (Çok Önemli)**

İfade	Anlamı
Separate read and write traffic	Read Replica
Performance optimization	Scaling , HA değil
Multi-AZ	Availability, NOT performance
Secondary AZ	Read alamaz

🔑 **Özet**

- **Multi-AZ ≠ Read scaling**
- **Read Replica = Performance**
- **Secondary instance okunamaz**
- **Hızlı çözüm → kapasite düşürmeden replica**

QUESTION 31

An Amazon EC2 administrator created the following policy associated with an IAM group containing several users:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2:TerminateInstances",  
            "Resource": "*",  
            "Condition": {  
                "IpAddress": {  
                    "aws:SourceIp": "10.100.100.0/24"  
                }  
            }  
        },  
        {  
            "Effect": "Deny",  
            "Action": "ec2:*",  
            "Resource": "*",  
            "Condition": {  
                "StringNotEquals": {  
                    "ec2:Region": "us-east-1"  
                }  
            }  
        }  
    ]  
}
```

What is the effect of this policy?

- A. Users can terminate an EC2 instance in any AWS Region except us-east-1.
- B. Users can terminate an EC2 instance with the IP address 10.100.100.1 in the us-east-1 Region.
- C. Users can terminate an EC2 instance in the us-east-1 Region when the user's source IP is 10.100.100.254.
- D. Users cannot terminate an EC2 instance in the us-east-1 Region when the user's source IP is 10.100.100.254.

Soru:

Bir Amazon EC2 yönetici, birden fazla kullanıcı içeren bir **IAM grubuna** aşağıdaki policy'yi ilişkilendirmiştir:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:TerminateInstances",
            "Resource": "*",
            "Condition": {
                "IpAddress": {
                    "aws:SourceIp": "10.100.100.0/24"
                }
            }
        },
        {
            "Effect": "Deny",
            "Action": "ec2:*",
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "ec2:Region": "us-east-1"
                }
            }
        }
    ]
}
```

Bu policy'nin etkisi nedir?

- A.** Kullanıcılar, **us-east-1** bölgesi **hariç** olmak üzere herhangi bir AWS bölgesinde bir EC2 instance'ını sonlandırabilir.
- B.** Kullanıcılar, **us-east-1** bölgesinde IP adresi **10.100.100.1** olan bir EC2 instance'ını sonlandırabilir.
- C.** Kullanıcının kaynak IP adresi **10.100.100.254** olduğunda, kullanıcılar **us-east-1** bölgesinde bir EC2 instance'ını sonlandırabilir.
- D.** Kullanıcının kaynak IP adresi **10.100.100.254** olduğunda, kullanıcılar **us-east-1** bölgesinde bir EC2 instance'ını **sonlandıramaz**.

Sorunun Analizi:

Policy'nin Yapısal Analizi

Policy **iki statement** içeriyor:

- ◆ **1. Statement (ALLOW)**

```
{
    "Effect": "Allow",
    "Action": "ec2:TerminateInstances",
    "Resource": "*",
    "Condition": {
        "IpAddress": {
            "aws:SourceIp": "10.100.100.0/24"
        }
    }
}
```

```
}
```

```
}
```

```
}
```

Anlamı:

- Kullanıcılar **EC2 instance terminate** edebilir
- AMA sadece:
 - Kullanıcının **kaynak IP adresi**
 - 10.100.100.0/24 aralığındaysa
(yani 10.100.100.1 – 10.100.100.254)

📌 Bu izin **region bağımsızdır** (region kısıtı burada yok).

◆ 2. Statement (DENY)

```
{
```

```
    "Effect": "Deny",
```

```
    "Action": "ec2:*",
```

```
    "Resource": "*",
```

```
    "Condition": {
```

```
        "StringNotEquals": {
```

```
            "ec2:Region": "us-east-1"
```

```
        }
```

```
    }
```

```
}
```

Anlamı:

- us-east-1 **DIŞINDAKİ tüm region'larda**
- **TÜM EC2 işlemleri (ec2:*) kesin olarak reddedilir**

📌 **Explicit Deny**, Allow'dan her zaman **üstündür**.

🔗 İki Statement Birlikte Ne Anlama Geliyor?

Bir kullanıcının EC2 terminate edebilmesi için:

1. Kaynak IP adresi

→ 10.100.100.0/24 aralığında olmalı

2. Region

→ **SADECE us-east-1 olmalı**

Başka bir deyişle:

us-east-1 + IP 10.100.100.0/24 → TERMINATE mümkün

us-east-1 dışı → HER ŞEY RED

Seçenek Analizi:

C.

Users can terminate an EC2 instance in the us-east-1 Region when the user's source IP is 10.100.100.254.

DOĞRU

Neden?

- 10.100.100.254 → 10.100.100.0/24 içinde ✓
- Region → us-east-1 ✓
- Explicit deny tetiklenmez ✓
- Allow statement geçerli ✓

A.

Users can terminate an EC2 instance in any AWS Region except us-east-1.

Yanlış

- Tam tersi:
 - **Sadece us-east-1 serbest**
 - Diğer tüm region'lar **explicit deny**

B.

Users can terminate an EC2 instance with the IP address 10.100.100.1 in the us-east-1 Region.

Yanlış

- Policy **instance'ın IP'siyle değil**
- **KULLANICININ source IP'siyle ilgilenir**

D.

Users cannot terminate an EC2 instance in the us-east-1 Region when the user's source IP is 10.100.100.254.

Yanlış 

- Bu senaryoda **tüm koşullar sağlanıyor**
- Terminate işlemi **izinlidir**

Sonuç

Sınavda Hatırlanması Gereken Kritik Kurallar

- **Explicit DENY > ALLOW**
- aws:SourceIp → **kullanıcının IP'si**
- ec2:Region → işlem yapılan region
- CIDR /24 → .1–.254

QUESTION 32

A company has a large Microsoft SharePoint deployment running on-premises that requires Microsoft Windows shared file storage. The company wants to migrate this workload to the AWS Cloud and is considering various storage options. The storage solution must be highly available and integrated with Active Directory for access control.

Which solution will satisfy these requirements?

- Conigure Amazon EFS storage and set the Active Directory domain for authentication.
- Create an SMB file share on an AWS Storage Gateway file gateway in two Availability Zones.
- Create an Amazon S3 bucket and conigure Microsoft Windows Server to mount it as a volume.
- Create an Amazon FSx for Windows File Server file system on AWS and set the Active Directory domain for authentication.

Soru:

Bir şirketin şirket içi (on-premises) ortamda çalışan, **Microsoft Windows paylaşımı dosya depolaması** gerektiren büyük bir **Microsoft SharePoint** kurulumu bulunmaktadır. Şirket, bu iş yükünü **AWS Bulut** ortamına taşımak istemekte ve çeşitli depolama seçeneklerini değerlendirmektedir.

Depolama çözümünün **yüksek erişilebilir (highly available)** olması ve **erişim kontrolü için Active Directory ile entegre** çalışması gerekmektedir.

Aşağıdaki çözümlerden hangisi bu gereksinimleri karşılar?

- A.** Amazon EFS depolamasını yapılandırin ve kimlik doğrulama için Active Directory domain'ini ayarlayın.
- B.** İki Availability Zone'da bir AWS Storage Gateway **file gateway** üzerinde bir **SMB dosya paylaşımı** oluşturun.
- C.** Bir Amazon S3 bucket oluşturun ve bunu bir volume olarak bağlamak (mount etmek) için Microsoft Windows Server yapılandırın.
- D.** AWS üzerinde bir **Amazon FSx for Windows File Server** dosya sistemi oluşturun ve kimlik doğrulama için **Active Directory domain'ini** ayarlayın.

Sorunun Analizi:

Sorudaki anahtar ifadeler:

- 1. Microsoft SharePoint**
- 2. Microsoft Windows shared file storage**
- 3. Highly available**
- 4. Active Directory ile entegre access control**
- 5. On-prem → AWS migration**

👉 Buradan çıkan teknik gereksinimler:

Gereksinim	Açıklama
SMB / Windows file share	SharePoint Windows tabanlı paylaşımı disk ister
Active Directory entegrasyonu	NTFS permissions, kullanıcı/grup bazlı erişim
High Availability	AZ-level redundancy
AWS-native çözüm	Operasyonel yük düşük olmalı

Seçenek Analizi:

- D. Amazon FSx for Windows File Server + Active Directory**

Neden doğru?

- **SMB-native Windows file system**
- **Tam NTFS + AD entegrasyonu**

- Multi-AZ high availability
- SharePoint için AWS tarafından önerilen çözüm

➔ FSx for Windows:

- Microsoft DFS, NTFS ACL
- On-prem AD veya AWS Managed AD ile çalışır
- Fully managed → düşük operasyonel yük

✗ A. Amazon EFS + Active Directory

Neden yanlış?

- Amazon EFS:
 - NFS tabanlıdır, SMB değildir
 - Linux workload'lar içindir
- Windows SharePoint:
 - SMB + NTFS ister
- EFS, Windows file share ihtiyacını karşılamaz

➔ Teknik olarak uyumsuz

✗ B. AWS Storage Gateway File Gateway (SMB) – 2 AZ

Neden yanlış?

- Storage Gateway:
 - Hybrid (on-prem + AWS) senaryoları içindir
 - Amaç: on-prem uygulamaların S3 kullanması
- Şirket:
 - Workload'u tamamen AWS'e taşıyor
- Ayrıca:
 - Gateway appliance yönetimi
 - Ek operasyonel yük

➔ Geçiş sonrası kalıcı çözüm için uygun değil

✗ C. Amazon S3 + Windows Server mount

Neden yanlış?

- Amazon S3:
 - **Object storage**
 - File system değildir
 - Windows Server:
 - Native olarak S3 mount edemez
 - NTFS permissions yok
 - SharePoint uyumsuz

→ Tamamen yanlış mimari

Sonuç

Sınav İpuçları (Çok Önemli)

İfade	Çözüm
Windows shared storage	FSx for Windows
Active Directory	FSx (Windows)
Linux / NFS	EFS
Hybrid	Storage Gateway
Object storage	S3

Özet

- SharePoint + Windows → **FSx for Windows**
 - AD entegrasyonu → **native destek**
 - High availability → **Multi-AZ**
 - AWS best practice → **FSx**

QUESTION 33

An image-processing company has a web application that users use to upload images. The application uploads the images into an Amazon S3 bucket. The company has set up S3 event notifications to publish the object creation events to an Amazon Simple Queue Service (Amazon SQS) standard queue. The SQS queue serves as the event source for

an AWS Lambda function that processes the images and sends the results to users through email.

Users report that they are receiving multiple email messages for every uploaded image. A solutions architect determines that SQS messages are invoking the Lambda function more than once, resulting in multiple email messages.

What should the solutions architect do to resolve this issue with the LEAST operational overhead?

- A. Set up long polling in the SQS queue by increasing the ReceiveMessage wait time to 30 seconds.
- B. Change the SQS standard queue to an SQS FIFO queue. Use the message deduplication ID to discard duplicate messages.
- C. Increase the visibility timeout in the SQS queue to a value that is greater than the total of the function timeout and the batch window timeout.
- D. Modify the Lambda function to delete each message from the SQS queue immediately after the message is read before processing.

Soru:

Bir görüntü işleme şirketinin, kullanıcıların görüntü yüklemek için kullandığı bir web uygulaması bulunmaktadır. Uygulama, görüntülerini bir **Amazon S3 bucket** içine yüklemektedir. Şirket, **S3 event notification** yapılandırarak nesne oluşturma (object creation) olaylarını bir **Amazon Simple Queue Service (Amazon SQS) standard queue** içine yayılmamaktadır.

SQS kuyruğu, görüntülerini işleyen ve sonuçları **e-posta yoluyla kullanıcılaraya gönderen** bir **AWS Lambda fonksiyonu** için olay kaynağı (event source) olarak kullanılmaktadır.

Kullanıcılar, **yüklenen her görüntü için birden fazla e-posta mesajı aldılarını bildirmektedir**. Bir Solutions Architect, **SQS mesajlarının Lambda fonksyonunu birden fazla kez tetiklediğini** ve bunun sonucunda birden fazla e-posta gönderildiğini tespit etmiştir.

En az operasyonel yükle bu sorunu çözmek için Solutions Architect ne yapmalıdır?

- A. SQS kuyruğunda **ReceiveMessage bekleme süresini 30 saniyeye çıkararak long polling** yapılandırın.
- B. SQS standard queue'yu **SQS FIFO queue** olarak değiştirin. **Message deduplication ID** kullanarak yinelenen (duplicate) mesajları yok sayın.
- C. SQS kuyruğundaki **visibility timeout** süresini, **Lambda fonksyonunun timeout süresi ile batch window timeout süresinin toplamından daha büyük** olacak şekilde artırın.

D. Lambda fonksiyonunu, her mesajı okumaktan hemen sonra (işlemeden önce) **SQS kuyruğundan silmek** üzere değiştirin.

Sorunun Analizi:

Sorudaki kritik ifadeler:

- **S3 → SQS (standard queue) → Lambda**
- Kullanıcılar **aynı görsel için birden fazla e-posta** alıyor
- **SQS mesajları Lambda'yı birden fazla kez tetikliyor**
- Çözüm **en az operasyonel yükle** olmalı

Buradaki temel AWS gerceği:

- **SQS Standard Queue = at-least-once delivery**
- Yani:

Aynı mesaj **birden fazla kez teslim edilebilir**

- Bu **bir hata değil**, tasarım gereğidir

Dolayısıyla mimari şu anda:

- Mesaj tekrar işleniyor
- Lambda tekrar e-posta gönderiyor

Seçenek Analizi:

C. Visibility timeout'u artırmak

Neden doğru?

- Mevcut sorun genelde şundan olur:
 - Lambda mesajı alır
 - İşleme süresi uzar
 - Visibility timeout dolar
 - SQS mesajı **tekrar görünür olur**
 - Lambda mesajı **yeniden alır**

Çözüm:

- **Visibility timeout > (Lambda timeout + batch window)**

Bu sayede:

- Mesaj işlenirken tekrar kuyruğa düşmez
- Lambda aynı mesajı ikinci kez almaz
- Mimari değişmez
- Operasyonel yük **minimum**

→ AWS'nin önerdiği best practice

✗ A. SQS long polling (ReceiveMessage wait time = 30s)

Neden yanlış?

- Long polling:
 - Boş polling'i azaltır
 - Maliyeti düşürür
- **Duplicate message sorununu çözmez**
- Mesaj yine birden fazla kez işlenebilir

→ Performans/maliyet iyileştirmesi, **mantık hatasını çözmez**

✗ B. Standard queue → FIFO queue + deduplication ID

Neden yanlış (bu soru özelinde)?

- FIFO queue:
 - Exactly-once processing **yakın** davranış
 - Deduplication sağlar
- Ancak:
 - **Queue tipini değiştirmek**
 - Message group ID, throughput limitleri
 - Uygulama mimarisinde değişiklik

📌 FIFO doğru çalışır ama:

“**LEAST operational overhead**” kriterini sağlamaz

✗ D. Mesajı işlemeden önce silmek

Neden yanlış?

- Lambda başarısız olursa:
 - Mesaj silinmiş olur

- Veri kaybı yaşanır
- At-least-once modeline aykırı
- AWS tarafından **önerilmez**

➡ Tehlikeli ve yanlış yaklaşım

🎯 Sonuç

🧠 Sınavda Altın Kurallar

Kavram	Hatırlat
SQS Standard	At-least-once
Duplicate processing	Visibility timeout
Least operational overhead	Mevcut mimariyi düzelt
FIFO	Daha büyük değişiklik

🔑 Özet

- Problem: **Visibility timeout kısa**
- Sonuç: **Duplicate Lambda invocation**
- En basit çözüm: **Visibility timeout artır**
- Doğru cevap: **C**

QUESTION 34

A company is implementing a shared storage solution for a gaming application that is hosted in an on-premises data center. The company needs the ability to use Lustre clients to access data. The solution must be fully managed.

Which solution meets these requirements?

- Create an AWS Storage Gateway file gateway. Create a file share that uses the required client protocol. Connect the application server to the file share.
- Create an Amazon EC2 Windows instance. Install and configure a Windows file share role on the instance. Connect the application server to the file share.

C. Create an Amazon Elastic File System (Amazon EFS) le system, and connect it to support Lustre. Attach the le system to the origin server. Connect the application server to the le system.

D. Create an Amazon FSx for Lustre le system. Attach the le system to the origin server. Connect the application server to the le system.

Soru:

Bir şirket, **on-premises (şirket içi)** veri merkezinde barındırılan bir **oyun uygulaması** için **paylaşımı bir depolama çözümü** uygulamaktadır. Şirketin, verilere **Lustre istemcileri** kullanarak erişebilme ihtiyacı vardır. Çözümün **tamamen yönetilen (fully managed)** olması gerekmektedir.

Aşağıdaki çözümlerden hangisi bu gereksinimleri karşılar?

A. Bir **AWS Storage Gateway file gateway** oluşturun. Gerekli istemci protokolünü kullanan bir dosya paylaşımı oluşturun. Uygulama sunucusunu bu dosya paylaşımına bağlayın.

B. Bir **Amazon EC2 Windows instance** oluşturun. Instance üzerinde bir **Windows file share** rolü kurup yapılandırın. Uygulama sunucusunu bu dosya paylaşımına bağlayın.

C. Bir **Amazon Elastic File System (Amazon EFS)** dosya sistemi oluşturun ve **Lustre desteği** olacak şekilde yapılandırın. Dosya sistemini kaynak (origin) sunucuya bağlayın. Uygulama sunucusunu bu dosya sistemine bağlayın.

D. Bir **Amazon FSx for Lustre** dosya sistemi oluşturun. Dosya sistemini kaynak (origin) sunucuya bağlayın. Uygulama sunucusunu bu dosya sistemine bağlayın.

Sorunun Analizi:

Sorudaki anahtar ifadeler:

1. **On-premises data center**
2. **Gaming application** → yüksek performans, düşük gecikme
3. **Lustre clients ile erişim zorunlu**
4. **Fully managed** çözüm

👉 Buradan çıkan teknik gereksinimler:

Gereksinim	Açıklama
-------------------	-----------------

Lustre desteği	Zorunlu , başka protokol kabul edilmez
----------------	---

Yüksek performans Gaming / HPC benzeri iş yükü	
--	--

Gereksinim	Açıklama
Fully managed	Sunucu yönetimi istenmiyor
AWS entegrasyonu	AWS-native servis tercih edilir

Seçenek Analizi:

 **D. Amazon FSx for Lustre**

Neden doğru?

- **Native Lustre file system**
- **Fully managed AWS service**
- Yüksek performans:
 - Gaming
 - HPC
 - Media processing
- Lustre client'lar **doğrudan bağlanabilir**
- AWS tarafından **önerilen ve özel olarak bu iş için tasarlanmış servis**

 **FSx for Lustre:**

- Patch, scaling, HA AWS tarafından yönetilir
- S3 entegrasyonu opsiyoneldir
- Lustre isteyen her senaryoda **otomatik doğru cevap**

 **A. AWS Storage Gateway File Gateway**

Neden yanlış?

- Storage Gateway:
 - **SMB / NFS** protokollerini destekler
 - **Lustre desteği yok**
- Hybrid senaryolar için tasarlanmıştır
- Lustre client gereksinimini karşılamaz

 **Teknik olarak uyumsuz**

B. EC2 Windows instance + Windows file share

Neden yanlış?

- Windows file share:
 - **SMB protokolü**
 - Lustre ile ilgisi yok
- Ayrıca:
 - **Fully managed değil**
 - Patch, backup, HA yönetimi gereklidir

 Hem protokol hem operasyonel açıdan yanlış

C. Amazon EFS + Lustre desteği

Neden yanlış?

- Amazon EFS:
 - **NFS tabanlıdır**
 - **Lustre desteği YOK**
- “EFS + Lustre” diye bir yapı AWS’te yoktur

 Teknik olarak mümkün değil

 Sonuç

 Sınavda Altın Kurallar

Anahtar Kelime Doğru Servis

Lustre **FSx for Lustre**

Windows + SMB FSx for Windows

Linux + NFS EFS

Hybrid Storage Gateway

Fully managed FSx / EFS

 Özet

- Lustre isteniyorsa → **Başka seçenek yok**

- Fully managed → **EC2** elenir
 - Doğru çözüm → **Amazon FSx for Lustre**
-

QUESTION 35

A company's containerized application runs on an Amazon EC2 instance. The application needs to download security certificates before it can communicate with other business applications. The company wants a highly secure solution to encrypt and decrypt the certificates in near real time. The solution also needs to store data in highly available storage after the data is encrypted.

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

- A. Create AWS Secrets Manager secrets for encrypted certificates. Manually update the certificates as needed. Control access to the data by using fine-grained IAM access.
- B. Create an AWS Lambda function that uses the Python cryptography library to receive and perform encryption operations. Store the function in an Amazon S3 bucket.
- C. Create an AWS Key Management Service (AWS KMS) customer managed key. Allow the EC2 role to use the KMS key for encryption operations. Store the encrypted data on Amazon S3.
- D. Create an AWS Key Management Service (AWS KMS) customer managed key. Allow the EC2 role to use the KMS key for encryption operations. Store the encrypted data on Amazon Elastic Block Store (Amazon EBS) volumes.

Soru:

Bir şirketin **container tabanlı uygulaması**, bir **Amazon EC2 instance** üzerinde çalışmaktadır. Uygulamanın, diğer iş uygulamalarıyla iletişim kurabilmesi için önce **güvenlik sertifikalarını indirmesi** gerekmektedir. Şirket, sertifikaların **yakın gerçek zamanlı (near real time)** olarak **şifrelenmesi ve şifresinin çözülmesi** için **yüksek derecede güvenli** bir çözüm istemektedir.

Ayrıca çözümün, veriler şifrelendikten sonra **yüksek erişilebilir (highly available)** bir depolama alanında saklanması gerekmektedir.

Aşağıdaki çözümlerden hangisi **en az operasyonel yükle** bu gereksinimleri karşılar?

- A. Şifrelenmiş sertifikalar için **AWS Secrets Manager** secret'ları oluşturun. Sertifikaları gerektiğinde **manuel olarak güncelleyin**. İnce taneli IAM yetkilendirmesi kullanarak verilere erişimi kontrol edin.
- B. Python **cryptography** kütüphanesini kullanan bir **AWS Lambda fonksiyonu** oluşturun ve şifreleme işlemlerini bu fonksiyonla gerçekleştirin. Fonksiyonu bir **Amazon S3 bucket** içinde saklayın.

C. Bir **AWS Key Management Service (AWS KMS)** müşteri tarafından yönetilen anahtar (customer managed key) oluşturun. EC2 rolünün şifreleme işlemleri için bu KMS anahtarını kullanmasına izin verin. Şifrelenmiş verileri **Amazon S3** üzerinde saklayın.

D. Bir **AWS KMS müşteri tarafından yönetilen anahtar** oluşturun. EC2 rolünün şifreleme işlemleri için bu KMS anahtarını kullanmasına izin verin. Şifrelenmiş verileri **Amazon Elastic Block Store (Amazon EBS)** volume'leri üzerinde saklayın.

Sorunun Analizi:

Sorudaki anahtar ifadeler:

1. **Containerized application EC2 üzerinde çalışıyor**
2. **Security certificates indiriliyor**
3. **Near real-time encryption & decryption**
4. **Highly secure**
5. **Encrypted data → highly available storage**
6. **LEAST operational overhead**

👉 Buradan çıkan teknik gereksinimler:

Gereksinim	Anlamı
Gerçek zamanlı şifreleme	Managed, düşük gecikmeli servis
Yüksek güvenlik	HSM-backed, IAM entegre
Yüksek erişilebilir depolama	Multi-AZ servis
Düşük operasyonel yük	Sunucu, kod, bakım yok

Seçenek Analizi:

C. AWS KMS + S3

Neden doğru?

- **AWS KMS**
 - HSM-backed
 - Near real-time encryption/decryption
 - IAM ile entegre

- Fully managed
- **Amazon S3**
 - Highly available (multi-AZ)
 - Durable
- EC2 role:
 - Doğrudan KMS anahtarını kullanabilir
- En az operasyonel yük:
 - Kod yazmadan
 - Sunucu yönetmeden

AWS best practice çözüm

A. AWS Secrets Manager + manuel güncelleme

Neden yanlış?

- Secrets Manager:
 - Secret **saklamak** içindir
 - **Gerçek zamanlı encryption/decryption servisi** değildir
- Manuel sertifika güncelleme:
 - Operasyonel yükü artırır
- Near real-time şifreleme ihtiyacını karşılamaz

Yanlış kullanım senaryosu

B. Lambda + Python cryptography library

Neden yanlış?

- Kendi kripto kodunu yazmak:
 - Güvenlik riski
 - Key management zor
- Lambda:
 - Ek kod
 - Ek bakım
- AWS managed encryption servisleri varken **gereksiz karmaşıklık**



Yüksek operasyonel yük



D. AWS KMS + EBS

Neden yanlış (bu soruda)?

- EBS:
 - **Single-AZ**
 - Instance'a bağımlı
 - Soruda:

“store data in highly available storage”

- S3 → regional & multi-AZ
 - EBS → AZ-level



HA gereksinimini tam karşılamaz



 Sınavda Hatırlanacak Altın Kurallar

İfade	Doğru Servis
Near real-time encryption	AWS KMS
Key management	KMS
Highly available storage	S3
Least operational overhead	Managed services
EBS	HA değil (AZ scoped)



Özet

- Sertifika şifreleme → **KMS**
 - HA storage → **S3**
 - Minimum operasyon → **Managed servisler**
 - Doğru ceyap → **C**

A solutions architect is designing a VPC with public and private subnets. The VPC and subnets use IPv4 CIDR blocks. There is one public subnet and one private subnet in each of three Availability Zones (AZs) for high availability. An internet gateway is used to provide internet access for the public subnets. The private subnets require access to the internet to allow Amazon EC2 instances to download software updates.

What should the solutions architect do to enable Internet access for the private subnets?

- A. Create three NAT gateways, one for each public subnet in each AZ. Create a private route table for each AZ that forwards non-VPC traffic to the NAT gateway in its AZ.
- B. Create three NAT instances, one for each private subnet in each AZ. Create a private route table for each AZ that forwards non-VPC traffic to the NAT instance in its AZ.
- C. Create a second internet gateway on one of the private subnets. Update the route table for the private subnets that forward non-VPC traffic to the private internet gateway.
- D. Create an egress-only internet gateway on one of the public subnets. Update the route table for the private subnets that forward non-VPC traffic to the egress-only Internet gateway.

Soru:

Bir **Solutions Architect**, **public** ve **private subnet'lere** sahip bir **VPC** tasarlamaktadır. VPC ve subnet'ler **IPv4 CIDR block'ları** kullanmaktadır. **Yüksek erişilebilirlik (high availability)** sağlamak için, üç **Availability Zone (AZ)**'un her birinde **bir public subnet ve bir private subnet** bulunmaktadır. Public subnet'lere internet erişimi sağlamak için bir **Internet Gateway (IGW)** kullanılmaktadır. **Private subnet'lerin**, Amazon EC2 instance'larının **yazılım güncellemelerini indirebilmesi** için **internete erişmesi** gerekmektedir.

Solutions Architect, private subnet'ler için internet erişimini sağlamak adına ne yapmalıdır?

- A. Her AZ'deki public subnet için **birer NAT Gateway** olacak şekilde **üç NAT Gateway** oluşturun. Her AZ için bir **private route table** oluşturun ve VPC dışı trafiği (non-VPC traffic) kendi AZ'sindeki NAT Gateway'e yönlendirin.
- B. Her AZ'deki private subnet için **birer NAT instance** olacak şekilde **üç NAT instance** oluşturun. Her AZ için bir **private route table** oluşturun ve VPC dışı trafiği kendi AZ'sindeki NAT instance'a yönlendirin.
- C. Private subnet'lerden birine **ikinci bir Internet Gateway** oluşturun. Private subnet'lerin route table'larını, VPC dışı trafiği bu private Internet Gateway'e yönlendirecek şekilde güncelleştirin.

D. Public subnet'lerden birinde bir **egress-only Internet Gateway** oluşturun. Private subnet'lerin route table'larını, VPC dışı trafiği egress-only Internet Gateway'e yönlendirecek şekilde güncelleyin.

Sorunun Analizi:

Sorudaki anahtar ifadeler:

1. **Public ve private subnet'ler**
2. **IPv4 CIDR kullanıyor**
3. **3 AZ – high availability**
4. **Private subnet'ler internete çıkacak**
 - o Ama **internetten erişilebilir olmayacak**
5. Amaç:
👉 EC2'lerin **software update indirmesi**

Buradan çıkan mimari gereksinimler:

Gereksinim	Açıklama
Outbound internet access	Private subnet'ler internete çıkışılmalı
Inbound yok	İnternetten private subnet'e erişim olmamalı
High availability	AZ bağımlılığı olmamalı
AWS best practice	Managed servis tercih edilmeli

Seçenek Analizi:

- A. Her AZ için bir NAT Gateway

Neden doğru?

- **NAT Gateway:**
 - o Managed (AWS yönetir)
 - o Highly available (AZ içinde)
 - o Otomatik ölçeklenir
- Her AZ'de bir NAT Gateway:

- **Cross-AZ dependency yok**
 - Bir AZ down olursa diğerleri etkilenmez
- Route table:
 - 0.0.0.0/0 → NAT Gateway (same AZ)

 **AWS'nin önerdiği referans mimari**

 **B. Her AZ için NAT instance**

Neden yanlış?

- NAT instance:
 - EC2 yönetimi gereklidir
 - Patch, scale, failover sorumluluğu sende
- Yüksek operasyonel yük
- AWS sınavlarında:

NAT Gateway > NAT Instance

 **Best practice değil**

 **C. Private subnet'e Internet Gateway**

Neden yanlış?

- Internet Gateway:
 - VPC'ye attach edilir
 - Subnet'e özel attach **edilemez**
- Private subnet + IGW:
 - Mimari olarak hatalı
- Ayrıca:
 - Private subnet public olur

 **Teknik olarak yanlış**

 **D. Egress-only Internet Gateway**

Neden yanlış?

- Egress-only IGW:
 - **SADECE IPv6 içindir**

- Soruda açıkça:

“VPC and subnets use IPv4 CIDR blocks”

 **IPv6 olmadığı için geçersiz**

 **Sonuç**

 **Sınavda Altın Kurallar**

Anahtar Kelime	Doğru Çözüm
Private subnet → internet	NAT Gateway
High availability	AZ başına NAT
IPv4	NAT Gateway
IPv6	Egress-only IGW
Least operational overhead	Managed servis

 **Özet**

- Private subnet’ler internețe **sadece outbound** çıkar
- IPv4 → **NAT Gateway**
- HA → **AZ başına NAT**

QUESTION 37

A company wants to migrate an on-premises data center to AWS. The data center hosts an SFTP server that stores its data on an NFS-based file system. The server holds 200 GB of data that needs to be transferred. The server must be hosted on an Amazon EC2 instance that uses an Amazon Elastic File System (Amazon EFS) file system.

Which combination of steps should a solutions architect take to automate this task? (Choose two.)

- Launch the EC2 instance into the same Availability Zone as the EFS file system.
- Install an AWS DataSync agent in the on-premises data center.
- Create a secondary Amazon Elastic Block Store (Amazon EBS) volume on the EC2 instance for the data.

D. Manually use an operating system copy command to push the data to the EC2 instance.
E. Use AWS DataSync to create a suitable location configuration for the on-premises SFTP server.

Soru:

Bir şirket, **on-premises (şirket içi)** bir veri merkezini AWS'e taşımak istemektedir. Veri merkezinde, verilerini **NFS tabanlı bir dosya sistemi** üzerinde saklayan bir **SFTP sunucusu** bulunmaktadır. Sunucu üzerinde **200 GB veri** yer almaktadır ve bu verilerin AWS'e aktarılması gerekmektedir.

Sunucu, **Amazon EC2 instance** üzerinde çalışacak ve **Amazon Elastic File System (Amazon EFS)** dosya sistemi kullanacaktır.

Bu görevi **otomatikleştirmek** için bir Solutions Architect aşağıdaki adımlardan hangilerinin **kombinasyonunu** uygulamalıdır?

(**İki seçenek seçiniz.**)

- A. EC2 instance'ı, EFS dosya sistemi ile **aynı Availability Zone** içine başlatın.
- B. On-premises veri merkezine bir **AWS DataSync agent** kurun.
- C. EC2 instance üzerinde veriler için **ikincil bir Amazon EBS volume** oluşturun.
- D. Verileri EC2 instance'a aktarmak için **işletim sisteminin kopyalama komutlarını manuel olarak kullanın.**
- E. **AWS DataSync** kullanarak on-premises **SFTP sunucusu** için uygun bir **location configuration** oluşturun.

Sorunun Analizi:

Sorudaki kritik noktalar:

1. **On-premises → AWS migration**
2. **SFTP server**
3. **NFS tabanlı file system**
4. **200 GB veri**
5. **Hedef: EC2 + Amazon EFS**
6. **Otomatik (automate) bir çözüm isteniyor**

👉 Buradan çıkan teknik ihtiyaçlar:

Gereksinim

Anlamı

File-level veri transferi NFS / POSIX uyumlu

Gereksinim	Anlamı
Otomasyon	Manual kopyalama yok
AWS-native	Managed servis
EFS hedef	EC2 + shared storage

➔ **AWS DataSync**, tam olarak bu senaryo için tasarlanmıştır.

Seçenek Analizi:

B. On-premises'e AWS DataSync agent kurmak

Neden doğru?

- On-premises veri kaynakları için:
 - **DataSync agent zorunludur**
- Agent:
 - NFS file system'e erişir
 - Verileri AWS'e güvenli şekilde taşıır
- Tam otomasyon sağlar

→ Olmazsa olmaz adım

E. AWS DataSync ile on-prem SFTP server için location oluşturmak

Neden doğru?

- DataSync:
 - **Source location** → on-prem NFS / SFTP
 - **Destination location** → Amazon EFS
- Location configuration:
 - Otomatik transfer
 - Scheduling
 - Integrity check

→ DataSync'in temel yapılandırma adımı

✗ A. EC2'yi EFS ile aynı AZ'de başlatmak

Neden yanlış / gereksiz?

- Amazon EFS bölgesel (regional) bir servistir
- Multi-AZ çalışır
- EC2'nin EFS ile aynı AZ'de olması **zorunlu değildir**
- Veri transferiyle **hiçbir ilgisi yok**

→ Bu adım **görevi otomatikleştirmez**

✗ C. EC2 üzerinde secondary EBS volume oluşturmak

Neden yanlış?

- Hedef storage:
 - **Amazon EFS**
- EBS:
 - Instance-level block storage
- Senaryoda:
 - **EBS hiç kullanılmıyor**

→ Gereksiz ve yanlış

✗ D. Manuel OS copy komutları kullanmak

Neden yanlış?

- Soruda açıkça:
“automate this task”
- Manuel kopyalama:
 - Operasyonel yük yüksek
 - Hata riski

→ AWS sınavlarında **kesinelenir**

🎯 **Sonuç**

🧠 **Sınavda Altın Kurallar**

Anahtar Kelime

Doğru Servis

On-prem → AWS file migration **AWS DataSync**

NFS / SFTP

DataSync agent + location

Anahtar Kelime	Doğru Servis
Automate	Manual işlemleri ele
Shared file system	EFS

🔑 Özeti

- Veri kaynağı: **On-prem NFS / SFTP**
- Hedef: **Amazon EFS**
- Otomasyon: **AWS DataSync**
- Gerekli adımlar:
 - **Agent kur**
 - **Location oluştur**

QUESTION 38

A company has an AWS Glue extract, transform, and load (ETL) job that runs every day at the same time. The job processes XML data that is in an Amazon S3 bucket. New data is added to the S3 bucket every day. A solutions architect notices that AWS Glue is processing all the data during each run.

What should the solutions architect do to prevent AWS Glue from reprocessing old data?

- Edit the job to use job bookmarks.
- Edit the job to delete data after the data is processed.
- Edit the job by setting the NumberOfWorkers eld to 1.
- Use a FindMatches machine learning (ML) transform.

Soru:

Bir şirketin her gün **aynı saatte çalışan** bir **AWS Glue extract, transform, and load (ETL)** işi bulunmaktadır. Bu iş, **Amazon S3 bucket** içinde bulunan **XML verilerini** işlemektedir. S3 bucket'a **her gün yeni veriler eklenmektedir**.

Bir Solutions Architect, AWS Glue işinin **her çalışmada tüm verileri tekrar işlediğini** fark etmiştir.

AWS Glue'nun **eski verileri yeniden işlememesi** için Solutions Architect ne yapmalıdır?

- A. Job'u **job bookmarks** kullanacak şekilde düzenlemek.
- B. Veriler işlendiikten sonra verileri **silmek** üzere job'u düzenlemek.
- C. Job'u **NumberOfWorkers** alanını **1** olacak şekilde düzenlemek.
- D. **FindMatches** makine öğrenmesi (ML) transform'unu kullanmak.

Sorunun Analizi:

Sorudan çıkan anahtar ifadeler:

1. **AWS Glue ETL job**
2. **Her gün aynı saatte çalışıyor (scheduled)**
3. **Kaynak: Amazon S3**
4. **Her gün yeni veri ekleniyor**
5. **Sorun: Her çalışmada tüm veriler işleniyor**
6. **İstenen: Eski verilerin tekrar işlenmemesi**

👉 Bu tanım doğrudan **incremental processing** ihtiyacını gösterir.

Seçenek Analizi:

- A. Job'u **job bookmarks** kullanacak şekilde düzenlemek

🧠 Doğru Çözüm Yaklaşımı

AWS Glue'da **job bookmarks** özelliği:

- Önceki çalışmada işlenen:
 - Dosyaları
 - Partition'ları
 - Satırları (job türüne göre)
takip eder.
- Sonraki çalışmalarda:
 - **Sadece yeni (delta) veriler işlenir**
- Incremental ETL için **AWS Glue'un native çözümüdür.**

Neden doğru?

- AWS Glue'un:
 - Incremental data processing
 - Exactly-once benzeri davranış

- Eski veriler otomatik olarak atlanır
- Operasyonel yük **yok denecek kadar az**

➡ Sorunun birebir cevabı

✗ **B. Veriler işlendiikten sonra verileri silmek**

Neden yanlış?

- Source data silmek:
 - Risklidir
 - Audit ve replay imkânını yok eder
- AWS Glue best practice değil
- Data governance açısından yanlış

✗ **C. NumberOfWorkers = 1**

Neden yanlış?

- Worker sayısı:
 - Performans ve paralellik ile ilgilidir
- Verinin tekrar işlenmesini **engellemek**

✗ **D. FindMatches ML transform**

Neden yanlış?

- FindMatches:
 - Duplicate record tespiti içindir
- Incremental processing ile **İlgisi yoktur**
- ML gereksiz

🎯 **Sonuç**

🔑 **Sınav İpuçları**

Anahtar İfade

Glue job + S3

Doğru Çözüm

Job bookmarks

Eski veri tekrar işleniyor Incremental ETL

Günlük job

Bookmarks

Anahtar İfade

Least effort

Doğru Çözüm

Native Glue feature

**Özet**

- Sorun: AWS Glue tüm veriyi tekrar işliyor
- Çözüm: **Job bookmarks**
- En az operasyonel yük

QUESTION 39

A solutions architect must design a highly available infrastructure for a website. The website is powered by Windows web servers that run on Amazon EC2 instances. The solutions architect must implement a solution that can mitigate a large-scale DDoS attack that originates from thousands of IP addresses. Downtime is not acceptable for the website.

Which actions should the solutions architect take to protect the website from such an attack? (Choose two.)

- A. Use AWS Shield Advanced to stop the DDoS attack.
- B. Configure Amazon GuardDuty to automatically block the attackers.
- C. Configure the website to use Amazon CloudFront for both static and dynamic content.
- D. Use an AWS Lambda function to automatically add attacker IP addresses to VPC network ACLs.
- E. Use EC2 Spot Instances in an Auto Scaling group with a target tracking scaling policy that is set to 80% CPU utilization.

Soru:

Bir çözümler mimarı, bir web sitesi için **yüksek erişilebilirliğe sahip** bir altyapı tasarlamak zorundadır. Web sitesi, **Amazon EC2** üzerinde çalışan **Windows web sunucuları** tarafından desteklenmektedir. Çözümler mimarı, **binlerce IP adresinden kaynaklanan büyük ölçüklü bir DDoS saldırısını** etkisiz hale getirebilecek bir çözüm uygulamalıdır. Web sitesi için **kesinti kabul edilemez**.

Bu tür bir saldırıya karşı web sitesini korumak için çözümler mimarı hangi aksiyonları almalıdır? (**İki seçenek seçin.**)

- A. DDoS saldırısını durdurmak için **AWS Shield Advanced** kullanmak.
- B. Saldırganları otomatik olarak engellemek için **Amazon GuardDuty**'yi yapılandırmak.
- C. Statik ve dinamik içeriklerin tamamı için web sitesini **Amazon CloudFront** kullanacak şekilde yapılandırmak.
- D. Saldırgan IP adreslerini otomatik olarak **VPC network ACL**'lerine ekleyen bir **AWS Lambda** fonksiyonu kullanmak.
- E. %80 CPU kullanımına ayarlanmış hedef izleme (target tracking) ölçekte politikası olan bir Auto Scaling grubunda **EC2 Spot Instance**'ları kullanmak.

Sorunun Analizi:

Soruyu çözerken özellikle şu ifadelerin altı çizilmelidir:

- ◆ “**Highly available infrastructure**”
 - Yüksek erişilebilirlik → **tek noktadan (single point of failure)** kaçınılmalı
 - Managed ve otomatik ölçeklenen servisler tercih edilir
- ◆ “**Large-scale DDoS attack**”
 - **Binlerce IP adresi**
 - Manuel veya IP bazlı çözümler **yetersiz**
 - AWS'nin **native DDoS servisleri** beklenir
- ◆ “**Downtime is not acceptable**”
 - Kesinti toleransı **sıfır**
 - Spot Instance, manuel müdahale, gecikmeli çözümler **elenir**
- ◆ “**Website powered by EC2 (Windows web servers)**”
 - Origin EC2 olabilir
 - Önüne **koruyucu katman** eklenmeli (edge / managed service)

Seçenek Analizi:

A. AWS Shield Advanced

- ✓ Büyük ölçekli DDoS için özel servis
 - ✓ Otomatik mitigation
 - ✓ L3/L4/L7 koruma
 - ✓ Zero-downtime beklentisini karşılar
- Doğrudan sorunun kalbine hitap ediyor

C. Amazon CloudFront (static + dynamic)

- ✓ Edge lokasyonlarda trafik dağıtımını
- ✓ Origin (EC2) yükünü azaltır
- ✓ DDoS trafiğini EC2'ye ulaşmadan emer
- ✓ Shield ile entegre çalışır

→ High availability + DDoS mitigation

B. Amazon GuardDuty

- X Sadece tehdit tespiti yapar
- X Trafiği bloklamaz
- X DDoS mitigation servisi değildir

→ “Detect” ≠ “Protect”

D. Lambda ile NACL'e IP ekleme

- X Binlerce IP için ölçeklenemez
- X Reaktif ve gecikmeli
- X DDoS senaryosu için yanlış yaklaşım

→ AWS sınavlarında anti-pattern

E. Spot Instance + Auto Scaling

- X Spot = kesinti riski
- X DDoS'u engellemez, sadece kapasite artırır
- X Downtime şartına aykırı

→ DDoS ≠ Auto Scaling problemi

Sonuç

Gereksinim Çözüm

Büyük ölçekli saldırı AWS Shield Advanced

Trafiği dağıtma / emme CloudFront

Kesintisiz çalışma Managed + Edge servisler

“Large-scale DDoS” + “No downtime” görüyorsan

→ CloudFront + Shield Advanced düşün

QUESTION 40

A company is preparing to deploy a new serverless workload. A solutions architect must use the principle of least privilege to configure permissions that will be used to run an AWS Lambda function. An Amazon EventBridge (Amazon CloudWatch Events) rule will invoke the function.

Which solution meets these requirements?

- A. Add an execution role to the function with lambda:InvokeFunction as the action and * as the principal.
- B. Add an execution role to the function with lambda:InvokeFunction as the action and Service: lambda.amazonaws.com as the principal.
- C. Add a resource-based policy to the function with lambda:* as the action and Service: events.amazonaws.com as the principal.
- D. Add a resource-based policy to the function with lambda:InvokeFunction as the action and Service: events.amazonaws.com as the principal.

Soru:

Bir şirket, yeni bir **sunucusuz (serverless)** iş yükünü devreye almaya hazırlanmaktadır. Bir çözümler mimarı, bir **AWS Lambda** fonksiyonunu çalıştırılmak için kullanılacak izinleri **en az ayrıcalık (least privilege)** prensibine göre yapılandırmak zorundadır. Bir **Amazon EventBridge (Amazon CloudWatch Events)** kuralı bu fonksiyonu tetikleyecektir.

Bu gereksinimleri hangi çözümkarşılardır?

- A. Fonksiyona, eylem (action) olarak lambda:InvokeFunction ve principal olarak * içeren bir **execution role** eklemek.
- B. Fonksiyona, eylem (action) olarak lambda:InvokeFunction ve principal olarak Service: lambda.amazonaws.com içeren bir **execution role** eklemek.
- C. Fonksiyona, eylem (action) olarak lambda:* ve principal olarak Service: events.amazonaws.com içeren bir **resource-based policy** eklemek.
- D. Fonksiyona, eylem (action) olarak lambda:InvokeFunction ve principal olarak Service: events.amazonaws.com içeren bir **resource-based policy** eklemek.

Sorunun Analizi:

Soruyu çözerken şu ifadeler belirleyici:

- ◆ “**Serverless workload**”

- AWS Lambda kullanılıyor

- IAM roller ve resource-based policy bilgisi gereklidir
- ◆ “Principle of least privilege”
 - Sadece gereklili olan izin
 - * veya lambda:* gibi geniş yetkiler elenir
- ◆ “Amazon EventBridge rule will invoke the function”
 - Lambda’yı başka bir AWS servisi çağrıyor
 - Bu durumda:
 - Execution role değil
 - Resource-based policy kullanılır

👉 Bu nokta çok kritiktir.

- ◆ Execution Role ne içindir?
 - Lambda’nın başka AWS servislerine erişmesi için
(örnek: S3, DynamoDB, CloudWatch Logs)
- ◆ Resource-based policy ne içindir?
 - Başka servislerin Lambda’yı invoke etmesi için
(örnek: EventBridge, S3, API Gateway)

📌 EventBridge → Lambda çağrıyorsa

→ Lambda üzerinde resource-based policy gereklidir

Seçenek Analizi:

D D. Resource-based policy + lambda:InvokeFunction + events.amazonaws.com



- ✓ Doğru policy türü
- ✓ Doğru servis (EventBridge)
- ✓ Sadece gereklili action (InvokeFunction)
- ✓ Least privilege prensibine tam uyum

A A. Execution role + * principal ✗

- Execution role yanlış yerde kullanılıyor
- * → least privilege ihlali

B B. Execution role + lambda.amazonaws.com ✗

- Bu rol Lambda'nın **kendisi içindir**
- EventBridge'i yetkilendirmez
- Yine yanlış yaklaşım

⌚ C. Resource-based policy + lambda:*

- Doğru policy türü ✓
- Ama lambda:*
- → **aşırı yetki**
- Least privilege'a aykırı

🎯 Sonuç

Gereksinim	Karşılayan
EventBridge Lambda'yı çağıracak Resource-based policy	
Minimum yetki	lambda:InvokeFunction
Doğru principal	events.amazonaws.com

Doğru principal

events.amazonaws.com

“X servisi Lambda'yı invoke ediyor” görürsen

- **Lambda resource-based policy**
- lambda:InvokeFunction
- Principal = ilgili servis

📌 Sınav ezberi

“Bir AWS servisi Lambda'yı çağrıyorrsa”

- **Lambda resource-based policy**
- lambda:InvokeFunction
- Principal = çağrıran servis

QUESTION 41

A company is preparing to store confidential data in Amazon S3. For compliance reasons, the data must be encrypted at rest. Encryption key usage must be logged for auditing purposes. Keys must be rotated every year.

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

- A. Server-side encryption with customer-provided keys (SSE-C)

- B. Server-side encryption with Amazon S3 managed keys (SSE-S3)
- C. Server-side encryption with AWS KMS keys (SSE-KMS) with manual rotation
- D. Server-side encryption with AWS KMS keys (SSE-KMS) with automatic rotation

Soru:

Bir şirket, **gizli (confidential)** verileri **Amazon S3** üzerinde depolamaya hazırlanmaktadır. Uyumluluk (compliance) gereksinimleri nedeniyle, veriler **bekleme halinde (at rest)** şifrelenmelidir. **Şifreleme anahtarlarının kullanımı**, denetim (audit) amacıyla **loglanmalıdır**. Anahtarlar her yıl döndürülmelidir (**rotate**).

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

- A. **Müşteri tarafından sağlanan anahtarlarla** sunucu tarafı şifreleme (SSE-C)
- B. **Amazon S3 tarafından yönetilen anahtarlarla** sunucu tarafı şifreleme (SSE-S3)
- C. **AWS KMS anahtarlarıyla** sunucu tarafı şifreleme (SSE-KMS) – **manuel anahtar döndürme**
- D. **AWS KMS anahtarlarıyla** sunucu tarafı şifreleme (SSE-KMS) – **otomatik anahtar döndürme**

Sorunun Analizi:

Soruda **3 kritik zorunluluk** var:

- ◆ **1. Encryption at rest**
 - S3 için **server-side encryption** gerekir
- ◆ **2. Key usage must be logged**
 - Anahtar kullanım logları → **AWS KMS + CloudTrail**
 - SSE-S3 ve SSE-C bu gereksinimi **karşılamaz**
- ◆ **3. Keys must be rotated every year**
 - Yıllık otomatik anahtar döndürme isteniyor
 - Manuel işlem istenmiyor (operational efficiency)

Seçenek Analizi:

□ D. **SSE-KMS + otomatik rotation** 

- ✓ Encryption at rest
- ✓ Key usage CloudTrail ile loglanır
- ✓ Yıllık **otomatik** anahtar döndürme
- ✓ En az operasyonel yük

A. SSE-C ✗

- Anahtarları müşteri yönetir
- KMS yok → **key usage loglanmaz**
- Operasyonel yük çok yüksek

B. SSE-S3 ✗

- AWS anahtarları kullanılır
- **KMS ve CloudTrail entegrasyonu yok**
- Key usage loglanamaz
- Rotation kontrolü yok

C. SSE-KMS + manuel rotation ✗

- Gereksinimleri karşılar ✓
- Ama **manuel işlem** gereklidir
- “MOST operationally efficient” şartına aykırı

⌚ Sonuç

Neden SSE-KMS?

Özellik	SSE-S3	SSE-C	SSE-KMS
Encryption at rest	✓	✓	✓
Key usage logging	✗	✗	✓
Annual rotation	✗	✗	✓

Operational efficiency Yüksek Düşük **En yüksek**

📌 Sınav ipucu

Audit + rotation + minimum operasyon görüyorsan
→ **SSE-KMS + automatic rotation**

A bicycle sharing company is developing a multi-tier architecture to track the location of its bicycles during peak operating hours. The company wants to use these data points in its existing analytics platform. A solutions architect must determine the most viable multi-tier option to support this architecture. The data points must be accessible from the REST API.

Which action meets these requirements for storing and retrieving location data?

- A. Use Amazon Athena with Amazon S3.
- B. Use Amazon API Gateway with AWS Lambda.
- C. Use Amazon QuickSight with Amazon Redshift.
- D. Use Amazon API Gateway with Amazon Kinesis Data Analytics.

Soru:

Bir bisiklet paylaşım şirketi, yoğun çalışma saatlerinde bisikletlerinin konumunu takip etmek için **çok katmanlı (multi-tier)** bir mimari geliştirmektedir. Şirket, bu veri noktalarını mevcut **analitik platformunda** kullanmak istemektedir. Bir çözümler mimarı, bu mimariyi desteklemek için **en uygun (en uygulanabilir) çok katmanlı seçeneği** belirlemek zorundadır.

Konum verilerine REST API üzerinden erişilebilmelidir.

Konum verilerinin **saklanması ve geri alınması** için aşağıdaki aksiyonlardan hangisi bu gereksinimleri karşılar?

- A. **Amazon Athena ile Amazon S3** kullanmak
- B. **Amazon API Gateway ile AWS Lambda** kullanmak
- C. **Amazon QuickSight ile Amazon Redshift** kullanmak
- D. **Amazon API Gateway ile Amazon Kinesis Data Analytics** kullanmak

Sorunun Analizi:

Soruda özellikle şu ifadeler belirleyici:

- ◆ “**Multi-tier architecture**”
 - **Aynı katmanlar:** API → işlem → veri katmanı
 - REST tabanlı erişim beklenir
- ◆ “**Track location during peak operating hours**”
 - **Yüksek istek sayısı**
 - Düşük gecikme, otomatik ölçektekleme gereklidir
- ◆ “**Use these data points in its existing analytics platform**”

- Veri saklanmalı
 - Analitik sistemler tarafından okunabilir olmalı
- ◆ “Data points must be accessible from the REST API”
- REST endpoint şart
 - API Gateway gibi bir servis beklenir

Seçenek Analizi:

B. Amazon API Gateway + AWS Lambda

Neden doğru?

- API Gateway → REST API erişimi
- Lambda → iş mantığı
- Lambda, konum verilerini DynamoDB / S3 / RDS gibi bir katmanaya yazabilir
- Otomatik ölçeklenir
- Multi-tier mimariye uygundur

→ API + compute + data katmanlarını net ayırrır

Gereksinim	B karşılıyor mu?
REST API erişimi	
Multi-tier mimari	
Yüksek trafik	
Veri saklama ve alma	 (Lambda + DB)
Analitik platform entegrasyonu	

A. Amazon Athena + Amazon S3

Neden elenir?

- Athena ad-hoc SQL sorgulama içindir
- REST API üzerinden anlık okuma/yazma için uygun değildir
- Gerçek zamanlı konum takibi senaryosu için yavaştır

→ Batch analytics çözümü, operational API değil

⌚ C. Amazon QuickSight + Amazon Redshift ✗

Neden elenir?

- QuickSight → BI / görselleştirme
- REST API backend'i değildir
- Redshift OLAP içindir, **yüksek frekanslı write** için uygun değil

→ Analitik okuma var ama API erişimi yok

▣ D. Amazon API Gateway + Amazon Kinesis Data Analytics ✗

Neden elenir?

- Kinesis Data Analytics → **stream processing**
- **Veri saklama servisi değildir**
- REST API ile doğrudan okuma uygun değil

→ Stream analizi var ama **kalıcı veri erişimi yok**

🎯 Sonuç

“REST API + yüksek trafik + veri saklama”

→ **API Gateway + Lambda** düşün

→ Altına DynamoDB / S3 eklenebilir

Aşağıda sorunun **çok kısa, sınavda hızlı çözülecek analiz özeti** var:

🔍 Hızlı Soru Analizi

Gereksinimler

- Multi-tier architecture
- REST API erişimi
- Yoğun saatlerde yüksek trafik
- Konum verisinin saklanması + okunması
- Analitik platformla uyum

✗ Elenen Seçenekler

- **Athena + S3** → Sorgulama / batch analytics, REST backend değil
- **QuickSight + Redshift** → BI ve OLAP, API backend'i değil
- **API Gateway + Kinesis Data Analytics** → Stream processing, kalıcı storage yok

API Gateway + Lambda

- REST API sağlar
- Otomatik ölçeklenir
- Veri katmanına (DynamoDB / S3) bağlanabilir
- Multi-tier mimariye uygundur

Sınav ezberi

REST API + serverless + high traffic

→ **API Gateway + Lambda**

QUESTION 43

A company has an automobile sales website that stores its listings in a database on Amazon RDS. When an automobile is sold, the listing needs to be removed from the website and the data must be sent to multiple target systems.

Which design should a solutions architect recommend?

- Create an AWS Lambda function triggered when the database on Amazon RDS is updated to send the information to an Amazon Simple Queue Service (Amazon SQS) queue for the targets to consume.
- Create an AWS Lambda function triggered when the database on Amazon RDS is updated to send the information to an Amazon Simple Queue Service (Amazon SQS) FIFO queue for the targets to consume.
- Subscribe to an RDS event notification and send an Amazon Simple Queue Service (Amazon SQS) queue fanned out to multiple Amazon Simple Notification Service (Amazon SNS) topics. Use AWS Lambda functions to update the targets.
- Subscribe to an RDS event notification and send an Amazon Simple Notification Service (Amazon SNS) topic fanned out to multiple Amazon Simple Queue Service (Amazon SQS) queues. Use AWS Lambda functions to update the targets.

Soru:

Bir şirketin, otomobil satışları yapan bir web sitesi vardır ve ilanlarını **Amazon RDS** üzerindeki bir veritabanında saklamaktadır. Bir otomobil satıldığında, ilgili ilan web sitesinden kaldırılmalı ve bu veriler **birden fazla hedef sisteme** gönderilmelidir.

Bir çözümler mimarı hangi tasarıyı önermelidir?

- A. **Amazon RDS** veritabanı güncellendiğinde tetiklenen bir **AWS Lambda** fonksiyonu oluşturmak ve bilgileri hedef sistemlerin tüketmesi için bir **Amazon Simple Queue Service (Amazon SQS)** kuyruğuna göndermek.
- B. **Amazon RDS** veritabanı güncellendiğinde tetiklenen bir **AWS Lambda** fonksiyonu oluşturmak ve bilgileri hedef sistemlerin tüketmesi için bir **Amazon Simple Queue Service (Amazon SQS) FIFO** kuyruğuna göndermek.
- C. Bir **RDS event notification**'a abone olmak ve bir **Amazon Simple Queue Service (Amazon SQS)** kuyruğunu, birden fazla **Amazon Simple Notification Service (Amazon SNS)** konusuna (topic) fan-out yapmak. Hedefleri güncellemek için **AWS Lambda** fonksiyonlarını kullanmak.
- D. Bir **RDS event notification**'a abone olmak ve bir **Amazon Simple Notification Service (Amazon SNS)** konusunu, birden fazla **Amazon Simple Queue Service (Amazon SQS)** kuyruğuna fan-out yapmak. Hedefleri güncellemek için **AWS Lambda** fonksiyonlarını kullanmak.

Sorunun Analizi:

Soruda gizli ama çok önemli ipuçları var:

- ◆ “**When an automobile is sold**”
 - **Event-driven** bir durum
 - Bir olay gerçekleşiyor → başka sistemler bilgilendirilecek
- ◆ “**Listing needs to be removed**”
 - Veritabanında bir **değişiklik** oluyor
 - Bu değişiklikten **tetiklenen** bir mekanizma lazım
- ◆ “**Data must be sent to multiple target systems**”
 - **Fan-out** deseni gerekiyor
 - Sistemler **gevşek bağlı (loosely coupled)** olmalı

👉 AWS'de bu ihtiyacın **standart çözümü**:

SNS + SQS fan-out

- ◆ **SNS ne için kullanılır?**
 - Bir olayı **birden fazla hedefe dağıtmak**
 - Publish / Subscribe modeli
- ◆ **SQS ne için kullanılır?**

- Her hedef sistemin **kendi hızında tüketmesi**
 - Hata izolasyonu
- ◆ RDS → Lambda tetiklenir mi?
- ✗ Doğrudan tetiklenmez
 - RDS **event notifications** ile olay yayınlar

Seçenek Analizi:

D D. RDS event → SNS → multiple SQS → Lambda ✓

Neden doğru?

- ✓ RDS event notification → event-driven
- ✓ SNS → **fan-out**
- ✓ Her hedef için ayrı SQS → loose coupling
- ✓ Lambda'lar hedef sistemleri günceller
- ✓ Ölçeklenebilir ve güvenilir mimari

→ AWS reference architecture

A A. RDS update → Lambda → SQS ✗

Neden yanlış?

- RDS, tablo satırı güncellemede **Lambda tetiklemez**
- Tek bir SQS → **fan-out yok**
- Birden fazla hedef için uygun değil

B B. RDS update → Lambda → SQS FIFO ✗

Neden yanlış?

- FIFO burada gereksiz (ordering şartı yok)
- Hâlâ **fan-out yok**
- Aynı mimari hatalar devam ediyor

C C. RDS event → SQS → SNS → Lambda ✗

Neden yanlış?

- Fan-out yönü **ters**
- SNS → SQS olmalı, SQS → SNS değil
- AWS'de böyle bir native pattern yok

Sonuç

Neden SNS → SQS fan-out en doğru tasarım?

Gereksinim	Çözüm
Bir olay	RDS Event Notification
Birden fazla hedef	SNS
Hedeflerin bağımsız çalışması	Ayrı SQS kuyrukları
Asenkron ve güvenilir	SQS + Lambda

Sınav ipucu (çok önemli)

“Bir olay → birden fazla sistem”

→ SNS fan-out

→ Her hedef için ayrı SQS

QUESTION 44

A company needs to store data in Amazon S3 and must prevent the data from being changed. The company wants new objects that are uploaded to Amazon S3 to remain unchangeable for a nonspecific amount of time until the company decides to modify the objects. Only specific users in the company's AWS account can have the ability to delete the objects.

What should a solutions architect do to meet these requirements?

- A. Create an S3 Glacier vault. Apply a write-once, read-many (WORM) vault lock policy to the objects.
- B. Create an S3 bucket with S3 Object Lock enabled. Enable versioning. Set a retention period of 100 years. Use governance mode as the S3 bucket's default retention mode for new objects.
- C. Create an S3 bucket. Use AWS CloudTrail to track any S3 API events that modify the objects. Upon notification, restore the modified objects from any backup versions that the company has.
- D. Create an S3 bucket with S3 Object Lock enabled. Enable versioning. Add a legal hold to the objects. Add the s3:PutObjectLegalHold permission to the IAM policies of users who need to delete the objects.

Soru:

Bir şirket, verileri **Amazon S3** üzerinde depolamak zorundadır ve bu verilerin **değiştirilmesini engellemelidir**. Şirket, Amazon S3'ye yüklenen **yeni nesnelerin**, şirket nesneleri değiştirmeye karar verene kadar **belirli olmayan bir süre boyunca değiştirilemez (immutable)** kalmasını istemektedir. Ayrıca, **yalnızca şirketin AWS hesabındaki belirli kullanıcılar** nesneleri **silebilme yetkisine** sahip olmalıdır.

Bu gereksinimleri karşılamak için bir çözümler mimarı ne yapmalıdır?

- A. Bir **S3 Glacier vault** oluşturmak ve nesnelere **write-once, read-many (WORM)** vault lock politikası uygulamak.
- B. **S3 Object Lock** etkinleştirilmiş bir **S3 bucket** oluşturmak. **Versioning'i** etkinleştirmek. **100 yıllık** bir saklama (retention) süresi belirlemek. Yeni nesneler için bucket'ın varsayılan saklama modu olarak **governance mode** kullanmak.
- C. Bir **S3 bucket** oluşturmak. Nesneleri değiştiren tüm S3 API olaylarını izlemek için **AWS CloudTrail** kullanmak. Bildirim alındığında, şirketin sahip olduğu yedek sürümlerden değiştirilmiş nesneleri geri yüklemek.
- D. **S3 Object Lock** etkinleştirilmiş bir **S3 bucket** oluşturmak. **Versioning'i** etkinleştirmek. Nesnelere **legal hold** eklemek. Nesneleri silebilmesi gereken kullanıcılarla IAM politikalarında s3:PutObjectLegalHold iznini eklemek.

Sorunun Analizi:

Sorunun Temel Gereksinimleri

Soruda **çok kritik 3 şart** var:

Gereksinim 1

S3'e yüklenen yeni objeler değiştirilememeli (immutable olmalı)

→ Yani:

- Silinememeli
- Üzerine yazılamamalı (overwrite edilememeli)

Gereksinim 2

Bu koruma süresi belirli değil (nonspecific amount of time)

→ Şu an için:

- “10 yıl”, “100 yıl” gibi **sabit bir süre istenmiyor**
- Şirket **ne zaman isterse** objeyi değiştirebilmek istiyor

Gereksinim 3

Sadece belirli IAM kullanıcıları objeleri silebilmeli

→ Yani:

- Herkes deletion yapamasın
- Yetki kontrollü olsun

Seçenek Analizi:

D. Object Lock + Versioning + Legal Hold

Bu Gereksinimler Hangi AWS Özelliğine İşaret Ediyor?

Gereksinim **AWS Özelliği**

Obje değiştirilemesin **S3 Object Lock**

Süre belirsiz olsun **Legal Hold**

Yetkili kullanıcılar kaldırıbsın **IAM permission (PutObjectLegalHold)**

Object Lock gereği **Versioning zorunlu**

Neden doğru?

✓ Legal Hold:

- Süresizdir
- Objeler:
 - Silinemez
 - Değiştirilemez
- **Retention süresi yoktur**

✓ IAM kontrolü:

- s3:PutObjectLegalHold yetkisi olan kullanıcılar:
 - Legal hold'u kaldırabilir
 - Sonrasında objeyi silebilir

✓ Tam olarak sorudaki senaryo

A. S3 Glacier Vault + Vault Lock (WORM)

Neden yanlış?

- Glacier → **arşivleme servisi**, aktif S3 kullanımına uygun değil

- Vault Lock:
 - Geri döndürülemez
 - Esnek değil
- Soru “S3’te depolama” diyor

Elenir

B. Object Lock + 100 yıl retention + Governance Mode

Neden yanlış?

- Retention süresi **belirli (100 yıl)** →
- Soru:

“nonspecific amount of time”

- Ayrıca retention süresi dolmadan değiştirmek **zordur**

Esnek değil, elenir

C. CloudTrail ile izleyip backup’tan geri dönme

Neden yanlış?

- Bu **önleme değil**, sadece **tespit + düzeltme**
- Objeyi zaten değiştirilmiş oluyor →
- Immutable gereksinimini karşılamaz

Elenir

Sonuç

Akılda Kalıcı İpucu (Exam Trick)

İfade	Düşün
“Belirsiz süre”	Legal Hold
“Sabit süre (X yıl)”	Retention Period
“S3 immutable”	Object Lock + Versioning
“Sadece belirli kullanıcılar silebilsin”	IAM permission

A social media company allows users to upload images to its website. The website runs on Amazon EC2 instances. During upload requests, the website resizes the images to a standard size and stores the resized images in Amazon S3. Users are experiencing slow upload requests to the website.

The company needs to reduce coupling within the application and improve website performance. A solutions architect must design the most operationally efficient process for image uploads.

Which combination of actions should the solutions architect take to meet these requirements? (Choose two.)

- A. Configure the application to upload images to S3 Glacier.
- B. Configure the web server to upload the original images to Amazon S3.
- C. Configure the application to upload images directly from each user's browser to Amazon S3 through the use of a presigned URL
- D. Configure S3 Event Notifications to invoke an AWS Lambda function when an image is uploaded. Use the function to resize the image.
- E. Create an Amazon EventBridge (Amazon CloudWatch Events) rule that invokes an AWS Lambda function on a schedule to resize uploaded images.

Soru:

Bir sosyal medya şirketi, kullanıcıların web sitesine görseller yüklemesine izin vermektedir. Web sitesi Amazon EC2 instance'ları üzerinde çalışmaktadır. Yükleme istekleri sırasında web sitesi, görselleri standart bir boyuta yeniden boyutlandırır ve yeniden boyutlandırılmış görselleri Amazon S3'te depolar. Kullanıcılar, web sitesine yapılan yükleme isteklerinde yavaşlık yaşamaktadır.

Şirket, uygulama içindeki bağımlılığı (coupling) azaltmak ve web sitesi performansını iyileştirmek istemektedir. Bir solutions architect, görsel yüklemeleri için operasyonel olarak en verimli süreci tasarlamalıdır.

Aşağıdaki aksiyon kombinasyonlarından hangileri bu gereksinimleri karşılar? (İki seçenek seçiniz)

- A. Uygulamayı görselleri Amazon S3 Glacier'a yükleyecek şekilde yapılandırmak.
- B. Web sunucusunu, orijinal görselleri Amazon S3'e yükleyecek şekilde yapılandırmak.
- C. Uygulamayı, her kullanıcının tarayıcısından Amazon S3'e doğrudan yükleme yapılmasını sağlayacak şekilde, presigned URL kullanarak yapılandırmak.

D. Bir görsel yüklendiğinde AWS Lambda fonksiyonunu tetiklemek için S3 Event Notifications yapılandırmak ve bu fonksiyonu görseli yeniden boyutlandırmak için kullanmak.

E. Yüklenen görselleri yeniden boyutlandırmak için zamanlanmış olarak AWS Lambda fonksiyonunu çalıştırın bir Amazon EventBridge (Amazon CloudWatch Events) kuralı oluşturmak.

Sorunun Analizi:

Soruda özellikle **altı çizilmesi gereken ifadeler** var:

- ◆ “**Users are experiencing slow upload requests**”

- Upload sırasında **performans problemi** var
- Upload yolundaki bileşenler azaltılmalı

- ◆ “**Reduce coupling within the application**”

- Web uygulaması:
 - Upload
 - Resize
 - Storage

işlerinin **hepsini aynı anda yapmamalı**

- Asenkron + event-driven mimari isteniyor

- ◆ “**Most operationally efficient process**”

- İstenen çözüm:
 - Managed services
 - Serverless
 - Az bakım (low ops)

🔑 Sorunun Özeti Gereksinimleri

Gereksinim	Anlamı
Yavaş upload	EC2 darboğazını kaldır
Coupling azalt	Upload ≠ Resize
Operasyonel verim	Serverless, event-driven

AWS sınavlarında bu tip soruların **klasik çözümü**:

1. Upload'u EC2'den kurtar
2. Resize işlemini asenkron yap
3. S3 event → Lambda

Seçenek Analizi:

C. Browser → S3 (Presigned URL)

Neden doğru?

- Kullanıcı:
 - Doğrudan S3'e upload yapar
- EC2:
 - Upload yükünden kurtulur
- Daha hızlı upload
- Daha az coupling

D. S3 Event Notification → Lambda (Resize)

Neden doğru?

- Upload sonrası:
 - Otomatik tetiklenir
- Resize:
 - Asenkron
 - Serverless
- Web app resize ile uğraşmaz

A. Upload to S3 Glacier

Neden yanlış?

- Glacier:
 - Arşiv
 - Yavaş erişim
- Upload + processing için uygun değil

Elenir

B. Web server uploads original images to S3

Neden yanlış?

- Upload hâlâ:
 - EC2 üzerinden geçiyor
 - CPU + Network tüketiyor
- Coupling devam ediyor

 Elenir

E. EventBridge schedule → Lambda

Neden yanlış?

- Resize işlemi:
 - Upload'a bağlı olmalı
- Schedule:
 - Gecikme yaratır
 - Gereksiz karmaşıklık

 Elenir

 Sonuç

“Upload yavaş + EC2 var”

 **Presigned URL**

“Dosya yüklenliğinde işlem”

 **S3 Event → Lambda**

QUESTION 46

A company recently migrated a message processing system to AWS. The system receives messages into an ActiveMQ queue running on an Amazon EC2 instance. Messages are processed by a consumer application running on Amazon EC2. The consumer application processes the messages and writes results to a MySQL database running on Amazon EC2. The company wants this application to be highly available with low operational complexity.

Which architecture offers the HIGHEST availability?

- A. Add a second ActiveMQ server to another Availability Zone. Add an additional consumer EC2 instance in another Availability Zone. Replicate the MySQL database to another Availability Zone.
- B. Use Amazon MQ with active/standby brokers configured across two Availability Zones. Add an additional consumer EC2 instance in another Availability Zone. Replicate the MySQL database to another Availability Zone.
- C. Use Amazon MQ with active/standby brokers configured across two Availability Zones. Add an additional consumer EC2 instance in another Availability Zone. Use Amazon RDS for MySQL with Multi-AZ enabled.
- D. Use Amazon MQ with active/standby brokers configured across two Availability Zones. Add an Auto Scaling group for the consumer EC2 instances across two Availability Zones. Use Amazon RDS for MySQL with Multi-AZ enabled.

Soru:

Bir şirket yakın zamanda bir mesaj işleme (message processing) sistemini AWS'e taşımıştır. Sistem, Amazon EC2 üzerinde çalışan bir ActiveMQ kuyruğuna mesajları almaktadır. Mesajlar, Amazon EC2 üzerinde çalışan bir consumer (tüketici) uygulaması tarafından işlenmektedir. Consumer uygulaması mesajları işler ve sonuçları Amazon EC2 üzerinde çalışan bir MySQL veritabanına yazar.

Şirket, bu uygulamanın **yüksek erişilebilirliğe (highly available)** sahip olmasını ve **düşük operasyonel karmaşıklık** ile yönetilmesini istemektedir.

Aşağıdaki mimarilerden hangisi **EN YÜKSEK erişilebilirliği (HIGHEST availability)** sunar?

- A. Başka bir Availability Zone'da ikinci bir ActiveMQ sunucusu eklemek. Başka bir Availability Zone'da ek bir consumer EC2 instance'ı eklemek. MySQL veritabanını başka bir Availability Zone'a replikasyon ile kopyalamak.
- B. İki Availability Zone'a yayılmış şekilde yapılandırılmış active/standby broker'lara sahip Amazon MQ kullanmak. Başka bir Availability Zone'da ek bir consumer EC2 instance'ı eklemek. MySQL veritabanını başka bir Availability Zone'a replikasyon ile kopyalamak.
- C. İki Availability Zone'a yayılmış şekilde yapılandırılmış active/standby broker'lara sahip Amazon MQ kullanmak. Başka bir Availability Zone'da ek bir consumer EC2 instance'ı eklemek. Multi-AZ etkinleştirilmiş Amazon RDS for MySQL kullanmak.
- D. İki Availability Zone'a yayılmış şekilde yapılandırılmış active/standby broker'lara sahip Amazon MQ kullanmak. İki Availability Zone'a yayılmış consumer EC2 instance'ları için bir Auto Scaling grubu kullanmak. Multi-AZ etkinleştirilmiş Amazon RDS for MySQL kullanmak.

Sorunun Analizi:

Sorudaki kritik ifadeler:

- ◆ “Highly available”

→ Amaç EN YÜKSEK erişilebilirlik

- Tek AZ bağımlılığı olmamalı
- Managed servisler tercih edilir
- Failover otomatik olmalı

- ◆ “Low operational complexity”

→ Şirket:

- EC2 üzerinde manuel cluster
- Manuel DB replikasyonu

istemiyor

→ AWS tarafından yönetilen (managed) çözümler öne çıkar

- ◆ Mevcut mimarideki zayıf noktalar

Katman	Sorun
--------	-------

ActiveMQ on EC2 Single point of failure

Consumer EC2 Ölçekleme & HA manuel

MySQL on EC2 Failover manuel, riskli

🔑 İdeal Mimari Ne Olmalı?

Her katmanda:

- Multi-AZ
- Managed
- Otomatik failover

Seçenek Analizi:

✓ D. Amazon MQ + Auto Scaling Consumer + RDS Multi-AZ

Neden doğru?

Katman	Çözüm	Neden İyi?
Messaging	Amazon MQ (Multi-AZ)	Managed + automatic failover
Consumer	Auto Scaling Group (Multi-AZ)	Instance failure'a dayanıklı
Database	RDS MySQL Multi-AZ	Otomatik failover

- Her katman HA
- En düşük operasyonel yük
- En yüksek erişilebilirlik

✗ A. ActiveMQ + Consumer + MySQL (hepsi EC2 üzerinde)

Neden yanlış?

- ActiveMQ:
 - Self-managed
 - Failover karmaşık
- MySQL replication:
 - Manuel yönetim
 - Failover otomatik değil
- Operasyonel yük **çok yüksek**

✗ Elenir

✗ B. Amazon MQ + EC2 Consumer + EC2 MySQL replication

Neden yanlış?

- Amazon MQ → ✓ doğru adım
- Ama:
 - Consumer EC2 **tek instance**
 - MySQL hâlâ EC2 üzerinde

→ Veritabanı **en zayıf halka** olmaya devam eder

✗ En yüksek availability değil

✗ C. Amazon MQ + EC2 Consumer + RDS Multi-AZ

Neden yanlış?

- Amazon MQ →
- RDS Multi-AZ →
- Ancak:
 - Consumer hâlâ **tek instance**
 - Instance down olursa processing durur

 Availability sınırlı

 Sonuç

- “**Highest availability**” → tüm katmanlar HA olmalı
- “**Low operational complexity**” →
 -  EC2 üzerinde DB
 - RDS Multi-AZ
- “**Messaging**” →
 -  Self-managed ActiveMQ
 - Amazon MQ

QUESTION 47

A company hosts a containerized web application on a fleet of on-premises servers that process incoming requests. The number of requests is growing quickly. The on-premises servers cannot handle the increased number of requests. The company wants to move the application to AWS with minimum code changes and minimum development effort.

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

- A. Use AWS Fargate on Amazon Elastic Container Service (Amazon ECS) to run the containerized web application with Service Auto Scaling. Use an Application Load Balancer to distribute the incoming requests.
- B. Use two Amazon EC2 instances to host the containerized web application. Use an Application Load Balancer to distribute the incoming requests.
- C. Use AWS Lambda with a new code that uses one of the supported languages. Create multiple Lambda functions to support the load. Use Amazon API Gateway as an entry point to the Lambda functions.
- D. Use a high performance computing (HPC) solution such as AWS ParallelCluster to establish an HPC cluster that can process the incoming requests at the appropriate scale.

Soru:

Bir şirket, gelen istekleri işleyen on-premises sunucular kümesi üzerinde çalışan konteynerleştirilmiş bir web uygulamasını barındırmaktadır. İstek sayısı hızla artmaktadır. On-premises sunucular, artan istek sayısını karşılayamamaktadır. Şirket, uygulamayı **minimum kod değişikliği** ve **minimum geliştirme eforu** ile AWS'e taşımak istemektedir.

Hangi çözüm, **en az operasyonel yük (LEAST operational overhead)** ile bu gereksinimleri karşılar?

- A. Konteynerleştirilmiş web uygulamasını çalıştırmak için Amazon Elastic Container Service (Amazon ECS) üzerinde **AWS Fargate** kullanmak ve **Service Auto Scaling** yapılandırmak. Gelen istekleri dağıtmak için bir **Application Load Balancer** kullanmak.
- B. Konteynerleştirilmiş web uygulamasını barındırmak için iki adet Amazon EC2 instance'ı kullanmak. Gelen istekleri dağıtmak için bir **Application Load Balancer** kullanmak.
- C. Desteklenen dillerden birini kullanan yeni bir kod ile **AWS Lambda** kullanmak. Yükü desteklemek için birden fazla Lambda fonksiyonu oluşturmak. Lambda fonksiyonlarına giriş noktası olarak **Amazon API Gateway** kullanmak.
- D. Gelen istekleri uygun ölçekte işleyebilen bir HPC kümesi oluşturmak için **AWS ParallelCluster** gibi yüksek performanslı bilgi işlem (HPC) çözümü kullanmak.

Sorunun Analizi:

Sorudaki kilit ifadeler:

- ◆ “**Containerized web application**”
 - Uygulama **zaten container** içinde
 - Yeniden mimari kurmak istenmiyor
- ◆ “**Minimum code changes and minimum development effort**”
 - **Refactor yok**
 - Mevcut container aynen çalışmalı
- ◆ “**LEAST operational overhead**”
 - Şirket:
 - Sunucu yönetmek istemiyor
 - Patch, capacity, scaling ile uğraşmak istemiyor
 - **Serverless / fully managed** çözümler öne çıkar

İdeal Çözümün Özeti

Gereksinim Anlamı

Minimum kod değişikliği Container'ı aynen çalıştır

Hızlı ölçekleme Auto scaling

Düşük operasyonel yük EC2 yönetimi yok

Seçenek Analizi:

A. ECS + AWS Fargate + ALB

Neden doğru?

- Fargate:
 - Sunucu yok
 - EC2 yönetimi yok
- ECS:
 - Container-native
 - Mevcut image aynen kullanılır
- Service Auto Scaling:
 - Trafiğe göre otomatik ölçekleme
- ALB:
 - Web trafiği için ideal

 Minimum kod değişikliği

 En az operasyonel yük

B. EC2 üzerinde container

Neden yanlış?

- EC2:
 - Patch
 - Capacity planning
 - Scaling
- Operasyonel yük yüksek

- Fargate varken tercih edilmez

✗ Elenir

✗ C. AWS Lambda + API Gateway

Neden yanlış?

- Container → **code rewrite** gereklidir
- Event-driven mimari
- Minimum geliştirme şartına aykırı

✗ Elenir

✗ D. HPC (AWS ParallelCluster)

Neden yanlış?

- HPC:
 - Bilimsel / batch iş yükleri
- Web application için uygun değil

✗ Elenir

🎯 Sonuç

- “**Container + minimum code change**”
→ **ECS**
- “**Least operational overhead**”
→ **Fargate**
- “**Web application**”
→ **ALB**

QUESTION 48

A company uses 50 TB of data for reporting. The company wants to move this data from on premises to AWS. A custom application in the company’s data center runs a weekly data transformation job. The company plans to pause the application until the data transfer is complete and needs to begin the transfer process as soon as possible. The data center does not have any available network bandwidth for additional workloads. A solutions architect must transfer the data and must configure the transformation job to continue to run in the AWS Cloud.

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

- A. Use AWS DataSync to move the data. Create a custom transformation job by using AWS Glue.
- B. Order an AWS Snowcone device to move the data. Deploy the transformation application to the device.
- C. Order an AWS Snowball Edge Storage Optimized device. Copy the data to the device. Create a custom transformation job by using AWS Glue.
- D. Order an AWS Snowball Edge Storage Optimized device that includes Amazon EC2 compute. Copy the data to the device. Create a new EC2 instance on AWS to run the transformation application.

Soru:

Bir şirket, raporlama amacıyla **50 TB** veri kullanmaktadır. Şirket bu veriyi **on-premises** ortamdan AWS'e taşımak istemektedir. Şirketin veri merkezinde çalışan özel (custom) bir uygulama, haftalık olarak bir veri dönüşüm (data transformation) işi çalıştırmaktadır. Şirket, veri aktarımı tamamlanana kadar bu uygulamayı durdurmayı planlamaktadır ve **veri transfer sürecine mümkün olan en kısa sürede başlamak** istemektedir.

Veri merkezinde, ek iş yükleri için kullanılabilir **herhangi bir ağ bant genişliği bulunmamaktadır**. Bir solutions architect, veriyi taşımak ve dönüşüm işinin AWS Cloud ortamında çalışmaya devam etmesini sağlayacak şekilde yapılandırmak zorundadır.

Aşağıdaki çözümlerden hangisi, **en az operasyonel yük (LEAST operational overhead)** ile bu gereksinimleri karşılar?

- A. Veriyi taşımak için **AWS DataSync** kullanmak. Dönüşüm işi için **AWS Glue** kullanarak özel bir dönüşüm işi oluşturmak.
- B. Veriyi taşımak için bir **AWS Snowcone** cihazı sipariş etmek. Dönüşüm uygulamasını bu cihaza dağıtmak.
- C. **AWS Snowball Edge Storage Optimized** cihazı sipariş etmek. Veriyi cihaza kopyalamak. Dönüşüm işi için **AWS Glue** kullanarak özel bir dönüşüm işi oluşturmak.
- D. Amazon EC2 compute içeren bir **AWS Snowball Edge Storage Optimized** cihazı sipariş etmek. Veriyi cihaza kopyalamak. Dönüşüm uygulamasını çalıştırmak için AWS üzerinde yeni bir **EC2 instance** oluşturmak.

Sorunun Analizi:

Sorudaki kritik ifadeler:

- ◆ “**50 TB of data**”

- Büyük veri miktarı
- İnternet üzerinden transfer **zor / yavaş**
 - ◆ “**No available network bandwidth**”
- **Online transfer imkânsız**
- **Physical device (Snow ailesi)** şart
 - ◆ “**Begin the transfer process as soon as possible**”
- Kurulum, geliştirme, karmaşık yapı **istenmiyor**
- Hızlı ve hazır bir çözüm tercih edilir
 - ◆ “**Configure the transformation job to continue to run in AWS Cloud**”
- On-prem uygulama:
 - AWS’te **karşılığı olan managed servisle** çalışmalı
 - Custom EC2 yönetimi istenmez
- ◆ “**LEAST operational overhead**”
- Anahtar ifade
 - Managed servis
 - Server yönetimi yok
 - Minimum bakım

Seçenek Analizi:

- C. Snowball Edge Storage Optimized + AWS Glue**

İdeal Çözümün Mantığı

Gereksinim	En Mantıklı Çözüm
Büyük veri + bant genişliği yok	Snowball Edge
En az operasyonel yük	Managed ETL (AWS Glue)
Hızlı başlangıç	Hazır AWS servisleri

Neden doğru?

✓ Snowball Edge Storage Optimized:

- 50 TB veri için uygun

- Network gerekmez
- Hızlı başlatılabilir

✓ AWS Glue:

- Fully managed ETL
- Server yok
- Düşük operasyonel yük

→ Veri fiziksel taşınır

→ Transformation işi AWS'te managed servisle devam eder

✗ A. AWS DataSync + AWS Glue

Neden yanlış?

- DataSync → **network gerekir**
- Soruda:

“no available network bandwidth”

✗ Elenir

✗ B. AWS Snowcone + uygulamayı cihaza deploy etme

Neden yanlış?

- Snowcone:
 - Küçük ölçekli (\approx 8–14 TB)
 - 50 TB için uygun değil
- Ayrıca:
 - Uygulamayı cihaza deploy etmek → **operasyonel yük yüksek**

✗ Elenir

✗ C. Snowball Edge + EC2 ile transformation

Neden yanlış?

- EC2:
 - Patch
 - Scaling
 - Monitoring

- Glue varken EC2 kullanmak:
 - Gereksiz operasyonel yük

 Elenir

 Sonuç

- “**No bandwidth**” →  DataSync / online transfer
- “**Large data (TB)**” →  Snowball
- “**Least operational overhead**” →
 -  EC2
 -  AWS Glue

QUESTION 49

A company has created an image analysis application in which users can upload photos and add photo frames to their images. The users upload images and metadata to indicate which photo frames they want to add to their images. The application uses a single Amazon EC2 instance and Amazon DynamoDB to store the metadata.

The application is becoming more popular, and the number of users is increasing. The company expects the number of concurrent users to vary significantly depending on the time of day and day of week. The company must ensure that the application can scale to meet the needs of the growing user base.

Which solution meets these requirements?

- A. Use AWS Lambda to process the photos. Store the photos and metadata in DynamoDB.
- B. Use Amazon Kinesis Data Firehose to process the photos and to store the photos and metadata.
- C. Use AWS Lambda to process the photos. Store the photos in Amazon S3. Retain DynamoDB to store the metadata.
- D. Increase the number of EC2 instances to three. Use Provisioned IOPS SSD (io2) Amazon Elastic Block Store (Amazon EBS) volumes to store the photos and metadata.

Soru:

Bir şirket, kullanıcıların fotoğraf yükleyebildiği ve fotoğraflarına çerçeve (photo frame) ekleyebildiği bir görüntü analiz (image analysis) uygulaması oluşturmuştur. Kullanıcılar, fotoğrafları ve hangi fotoğraf çerçevelerini eklemek istediklerini belirten metadata

bilgilerini yüklemektedir. Uygulama, tek bir Amazon EC2 instance’ı kullanmakta ve metadata bilgilerini saklamak için Amazon DynamoDB kullanmaktadır.

Uygulama giderek daha popüler hale gelmektedir ve kullanıcı sayısı artmaktadır. Şirket, eşzamanlı (concurrent) kullanıcı sayısının günün saatine ve haftanın günlerine bağlı olarak önemli ölçüde değişmesini beklemektedir. Şirket, uygulamanın büyüyen kullanıcı tabanının ihtiyaçlarını karşılayacak şekilde ölçeklenebilmesini sağlamalıdır.

Bu gereksinimleri karşılayan çözüm hangisidir?

- A. Fotoğrafları işlemek için **AWS Lambda** kullanmak. Fotoğrafları ve metadata bilgilerini **Amazon DynamoDB**’de saklamak.
- B. Fotoğrafları işlemek ve fotoğrafları ile metadata bilgilerini depolamak için **Amazon Kinesis Data Firehose** kullanmak.
- C. Fotoğrafları işlemek için **AWS Lambda** kullanmak. Fotoğrafları **Amazon S3**’te saklamak. Metadata bilgilerini saklamak için **Amazon DynamoDB**’yi kullanmaya devam etmek.
- D. EC2 instance sayısını üçe çıkarmak. Fotoğrafları ve metadata bilgilerini depolamak için **Provisioned IOPS SSD (io2) Amazon EBS** volume’leri kullanmak.

Sorunun Analizi:

Sorudaki kritik noktalar:

- ◆ “**Single Amazon EC2 instance**”
- Mevcut yapı **tek noktadan hata (SPOF)** içeriyor
- Ölçeklenebilir değil
- ◆ “**Number of concurrent users to vary significantly**”
- Trafik:
 - Günün saatine göre
 - Haftanın gününe göre
- çok değişken
 - **Otomatik ve elastik ölçektekleme şart**
 - ◆ “**Ensure that the application can scale**”
 - Manuel EC2 artırımı yeterli değil
 - **Serverless / managed** servisler tercih edilmeli
 - ◆ **Uygulama türü**

- Fotoğraf yükleme
- Fotoğraf işleme (frame ekleme)
- Metadata saklama

Seçenek Analizi:

✓ C. Lambda + S3 + DynamoDB

→ Tipik AWS deseni:

- **Compute:** Lambda
- **Object storage:** S3
- **Metadata:** DynamoDB

İdeal Mimari Nasıl Olmalı?

Katman En Uygun Servis Neden

Fotoğraf işleme AWS Lambda Otomatik ölçekleme

Fotoğraf saklama Amazon S3 Sınırsız ölçek

Neden doğru?

✓ Lambda:

- Concurrent user sayısına göre otomatik ölçeklenir

✓ S3:

- Fotoğraflar için ideal
- Sınırsız ölçek

✓ DynamoDB:

- Metadata için doğru servis
- Serverless

→ Trafik artsa bile sistem otomatik ölçeklenir

→ Operasyonel yük minimum

✗ A. Lambda + DynamoDB (fotoğraflar da DynamoDB)

Neden yanlış?

- DynamoDB:

- Metadata için uygun
- **Büyük binary dosyalar (fotoğraf)** için uygun değil
- Yüksek maliyet + verimsiz kullanım

✗ Elenir

✗ B. Kinesis Data Firehose

Neden yanlış?

- Firehose:
 - Streaming / log / analytics için
- Image processing için **yanlış servis**

✗ Elenir

✗ D. 3 EC2 + EBS io2

Neden yanlış?

- Manuel ölçektekleme
- EBS:
 - Instance'a bağlı
 - Elastik değil
- Operational overhead yüksek

✗ Elenir

🎯 Sonuç

- “**Concurrent users değişken**” → Lambda
- “**Fotoğraf / dosya**” → S3
- “**Metadata**” → DynamoDB
- “**Scale automatically**” → Serverless mimari

Sınavda Hızlı Hatırlatma

- 📸 **Fotoğraf / dosya** → S3
- 📄 **Metadata** → DynamoDB
- 👤 **Değişken concurrent user** → Lambda

- **EC2 ile scale** → genelde yanlış
-

QUESTION 50

A medical records company is hosting an application on Amazon EC2 instances. The application processes customer data files that are stored on Amazon S3. The EC2 instances are hosted in public subnets. The EC2 instances access Amazon S3 over the internet, but they do not require any other network access.

A new requirement mandates that the network traffic for file transfers take a private route and not be sent over the internet.

Which change to the network architecture should a solutions architect recommend to meet this requirement?

- A. Create a NAT gateway. Configure the route table for the public subnets to send traffic to Amazon S3 through the NAT gateway.
- B. Configure the security group for the EC2 instances to restrict outbound traffic so that only traffic to the S3 prefix list is permitted.
- C. Move the EC2 instances to private subnets. Create a VPC endpoint for Amazon S3, and link the endpoint to the route table for the private subnets.
- D. Remove the internet gateway from the VPC. Set up an AWS Direct Connect connection, and route traffic to Amazon S3 over the Direct Connect connection.

Soru:

Bir tıbbi kayıtlar şirketi, bir uygulamayı Amazon EC2 instance'ları üzerinde barındırmaktadır. Uygulama, Amazon S3 üzerinde depolanan müşteri veri dosyalarını işlemektedir. EC2 instance'ları **public subnet'lerde** barındırılmaktadır. EC2 instance'ları Amazon S3'e **internet üzerinden** erişmektedir, ancak başka herhangi bir ağ erişimine ihtiyaç duymamaktadır.

Yeni bir gereksinim, dosya transferleri için ağ trafiğinin **özel (private) bir yol** üzerinden iletilmesini ve **internet üzerinden gönderilmemesini** zorunlu kılmaktadır.

Bu gereksinimi karşılamak için bir solutions architect ağ mimarisinde hangi değişikliği önermelidir?

- A. Bir **NAT gateway** oluşturmak. Public subnet'ler için route table'sı, Amazon S3'e giden trafiği NAT gateway üzerinden gönderecek şekilde yapılandırmak.
- B. EC2 instance'ları için security group'ları yapılandırarak, outbound trafiği yalnızca **S3 prefix list**'ine giden trafiğe izin verecek şekilde kısıtlamak.

C. EC2 instance'larını **private subnet'lere** taşımak. Amazon S3 için bir **VPC endpoint** oluşturmak ve bu endpoint'i private subnet'lerin route table'ına bağlamak.

D. VPC'den **internet gateway'i** kaldırırmak. Bir **AWS Direct Connect** bağlantısı kurmak ve Amazon S3'e giden trafiği Direct Connect üzerinden yönlendirmek.

Sorunun Analizi:

Sorudaki **kritik ifadeler**:

- ◆ “**EC2 instances are hosted in public subnets**”

- EC2'ler şu an **internet erişimine açık**
- S3'e erişim **Internet Gateway** üzerinden

- ◆ “**Access Amazon S3 over the internet**”

- Mevcut mimari:

- Trafik public internetten geçiyor
- **Güvenlik ve regülasyon açısından istenmiyor**

- ◆ “**Network traffic must take a private route**”

- Ana gereksinim:

- **X Internet Gateway**
- **X Public internet**
- **✓ AWS private network**

- ◆ “**They do not require any other network access**”

- EC2'lerin:

- Internete çıkışına gerek yok
- Sadece S3'e erişmesi yeterli

- **Private subnet + VPC Endpoint** ideal

Seçenek Analizi:

- C. **Private subnet + VPC Endpoint for S3**

Ideal AWS Çözüm Mantığı

AWS'te S3'e **özel ağ üzerinden** erişmenin standart yolu:

VPC Gateway Endpoint for Amazon S3

Özellikleri:

- Trafik AWS backbone üzerinden gider
- Internet Gateway kullanılmaz
- NAT gerekmez
- Ekstra maliyet yok

Neden doğru?

- ✓ EC2'ler private subnet'te
- ✓ S3 için **VPC Gateway Endpoint**
- ✓ Route table → S3 endpoint
- ✓ Trafik:

- AWS internal network
- Internet Gateway yok

→ Tam olarak istenen çözüm

✗ A. NAT Gateway

Neden yanlış?

- NAT Gateway:
 - Hâlâ **internet** kullanır
 - Sadece private subnet'lerin internete çıkışını içindir
- “Private route” gereksinimini karşılamaz

✗ Elenir

✗ B. Security Group ile outbound kısıtlama

Neden yanlış?

- Security Group:
 - Trafigi **filtreler**
 - Ama **yolu değiştirmez**
- Trafik hâlâ internetten geçer

✗ Elenir

✗ D. Direct Connect

Neden yanlış?

- Direct Connect:
 - On-prem → AWS için
- Bu senaryoda:
 - EC2 → S3 (AWS içi)
- Aşırı karmaşık ve pahalı

 Elenir

 Sonuç

- “**EC2 → S3 private access**”
 - **VPC Endpoint (Gateway)**
- “**Internet kullanılmamasın**”
 -  NAT
 -  IGW
 -  VPC Endpoint

QUESTION 51

A company uses a popular content management system (CMS) for its corporate website. However, the required patching and maintenance are burdensome. The company is redesigning its website and wants a new solution. The website will be updated four times a year and does not need to have any dynamic content available. The solution must provide high scalability and enhanced security.

Which combination of changes will meet these requirements with the LEAST operational overhead? (Choose two.)

- A. Configure Amazon CloudFront in front of the website to use HTTPS functionality.
- B. Deploy an AWS WAF web ACL in front of the website to provide HTTPS functionality.
- C. Create and deploy an AWS Lambda function to manage and serve the website content.
- D. Create the new website and an Amazon S3 bucket. Deploy the website on the S3 bucket with static website hosting enabled.
- E. Create the new website. Deploy the website by using an Auto Scaling group of Amazon EC2 instances behind an Application Load Balancer.

Soru:

Bir şirket, kurumsal web sitesi için popüler bir içerik yönetim sistemi (CMS) kullanmaktadır. Ancak gerekli yamalama (patching) ve bakım işlemleri yük oluşturmaktadır. Şirket web sitesini yeniden tasarlamaktadır ve yeni bir çözüm istemektedir. Web sitesi yılda **dört kez güncellenecektir** ve **herhangi bir dinamik içeriğe** ihtiyaç duymamaktadır. Çözüm, **yüksek ölçeklenebilirlik** ve **gelişmiş güvenlik** sağlamalıdır.

Aşağıdaki değişiklik kombinasyonlarından hangileri bu gereksinimleri **en az operasyonel yük (LEAST operational overhead)** ile karşılar?

(**İki seçenekiniz**)

- A. Web sitesinin önüne **HTTPS işlevselliği** kullanacak şekilde **Amazon CloudFront** yapılandırmak.
- B. HTTPS işlevselliği sağlamak için web sitesinin önüne bir **AWS WAF web ACL** dağıtmak.
- C. Web sitesi içeriğini yönetmek ve sunmak için bir **AWS Lambda** fonksiyonu oluşturmak ve dağıtmak.
- D. Yeni web sitesini oluşturmak ve bir **Amazon S3 bucket** oluşturmak. **Static website hosting** etkinleştirilmiş şekilde web sitesini S3 bucket üzerinde dağıtmak.
- E. Yeni web sitesini oluşturmak. Web sitesini, bir **Application Load Balancer** arkasında çalışan **Amazon EC2 Auto Scaling grubu** kullanarak dağıtmak.

Sorunun Analizi:

Soruda özellikle vurgulanan noktalar:

- ◆ **Operasyonel yük en az olmalı**
 - Patch, bakım, sunucu yönetimi istenmiyor
 - CMS'ten kurtulmak istiyorlar
- ◆ **Web sitesi özellikleri**
 - **Statik içerik** (dinamik içerik yok)
 - **Yılda sadece 4 kez güncelleme**
 - Kurumsal web sitesi
- ◆ **Teknik gereksinimler**
 - **Yüksek ölçeklenebilirlik**
 - **Gelişmiş güvenlik**

Seçenek Analizi:

  **A Amazon CloudFront + HTTPS**

✓ Artıları:

- Küresel CDN → **yüksek ölçeklenebilirlik**
- AWS Certificate Manager ile HTTPS
- DDoS koruması (AWS Shield Standard)
- S3 gibi statik içeriklerle **çok uyumlu**
- Yönetim yükü çok düşük

⚠ Tek başına web sitesi barındırmaz, bir **origin** gereklidir (ör. S3)

 **Doğru ve gerekli bir bileşen**

  **Amazon S3 Static Website Hosting**

✓ Artıları:

- **Sunucu yok → bakım yok**
- Statik içerik için **en ideal çözüm**
- Çok yüksek dayanıklılık ve ölçeklenebilirlik
- Yılda 4 güncelleme → S3'ye dosya yüklemek yeterli
- CMS ve patch ihtiyacı tamamen ortadan kalkar

⚠ Tek başına HTTPS yok ama CloudFront ile çözülür

 **Kesinlikle doğru seçenek**

  **AWS WAF ile HTTPS sağlamak**

 Yanlış gerekçe:

- AWS WAF **HTTPS sağlamaz**
- WAF güvenlik kuralları (SQL injection, XSS) içindir
- HTTPS CloudFront veya ALB üzerinden sağlanır

 **Bu seçenek yanlış**

  **AWS Lambda ile web sitesi sunmak**

 Neden uygun değil:

- Lambda statik web sitesi barındırmak için tasarlanmaz

- Kod, tetikleyici, API Gateway gereklidir
- Operasyonel karmaşıklık **artar**
- Gereksiz maliyet ve mimari

✖ **En az operasyonel yük şartına ters**

✖ **EC2 + Auto Scaling + ALB**

✖ Neden yanlış:

- EC2 → patch, OS, güvenlik güncellemeleri
- ALB ve Auto Scaling yönetimi
- Statik site için **aşırı karmaşık**
- Operasyonel yük yüksek

✖ **Sorunun tam tersine bir çözüm**

🎯 **Sonuç**

✓ **D. Amazon S3 Static Website Hosting**

✓ **A. Amazon CloudFront + HTTPS**

Bu kombinasyon neden en iyisi?

Gereksinim	Karşılanma
Statik içerik	✓ S3
Düşük operasyonel yük	✓ Sunucusuz
Yüksek ölçeklenebilirlik	✓ S3 + CloudFront
Gelişmiş güvenlik	✓ HTTPS + Shield
Az güncelleme	✓ S3 upload yeterli

QUESTION 52

A company stores its application logs in an Amazon CloudWatch Logs log group. A new policy requires the company to store all application logs in Amazon OpenSearch Service (Amazon Elasticsearch Service) in near-real time.

Which solution will meet this requirement with the LEAST operational overhead?

- A. Configure a CloudWatch Logs subscription to stream the logs to Amazon OpenSearch Service (Amazon Elasticsearch Service).
- B. Create an AWS Lambda function. Use the log group to invoke the function to write the logs to Amazon OpenSearch Service (Amazon Elasticsearch Service).
- C. Create an Amazon Kinesis Data Firehose delivery stream. Configure the log group as the delivery stream's source. Configure Amazon OpenSearch Service (Amazon Elasticsearch Service) as the delivery stream's destination.
- D. Install and configure Amazon Kinesis Agent on each application server to deliver the logs to Amazon Kinesis Data Streams. Configure Kinesis Data Streams to deliver the logs to Amazon OpenSearch Service (Amazon Elasticsearch Service).

Soru:

Bir şirket, uygulama günlüklerini (application logs) bir **Amazon CloudWatch Logs log group** içinde saklamaktadır. Yeni bir politika, şirketin **tüm uygulama günlüklerini Amazon OpenSearch Service (Amazon Elasticsearch Service)** içine **neredeyse gerçek zamanlı (near-real time)** olarak saklamasını zorunlu kılmaktadır.

En az operasyonel yük ile bu gereksinimi karşılayacak çözüm hangisidir?

- A. Logları Amazon OpenSearch Service (Amazon Elasticsearch Service) içine aktarmak için bir **CloudWatch Logs subscription** yapılandırmak.
- B. Bir **AWS Lambda fonksiyonu** oluşturmak. Log group'un bu fonksiyonu tetiklemesini sağlayarak logları Amazon OpenSearch Service (Amazon Elasticsearch Service) içine yazmak.
- C. Bir **Amazon Kinesis Data Firehose delivery stream** oluşturmak. Log group'u delivery stream'in kaynağı olarak yapılandırmak. Amazon OpenSearch Service (Amazon Elasticsearch Service)'i delivery stream'in hedefi olarak yapılandırmak.
- D. Her uygulama sunucusuna **Amazon Kinesis Agent** kurup yapılandırmak ve logları **Amazon Kinesis Data Streams**'e göndermek. Kinesis Data Streams'i logları Amazon OpenSearch Service (Amazon Elasticsearch Service)'e iletecek şekilde yapılandırmak.

Sorunun Analizi:

Soruda özellikle dikkat çeken ifadeler:

- ◆ **Mevcut durum**
 - Loglar **Amazon CloudWatch Logs log group** içinde
- ◆ **Yeni gereksinim**
 - Tüm loglar **Amazon OpenSearch Service** içine

- **Near-real time** (gecikme çok düşük olmalı)
- ◆ **En kritik kriter**
 - **LEAST operational overhead**
 - Ek sunucu, agent, karmaşık kod istenmiyor
 - Yönetimi minimum olan servis tercih edilmeli

Seçenek Analizi:

C CloudWatch Logs → Kinesis Data Firehose → OpenSearch

✓ Artıları:

- **Native AWS entegrasyonu**
- **Near-real time** (saniyeler içinde)
- Firehose:
 - Buffering
 - Retry
 - Failure handling
 - Otomatik ölçeklenme
- Kod yazmaya gerek yok
- OpenSearch'e **doğrudan destination**

✗ En az operasyonel yük

✗ AWS sınavlarında “log → OpenSearch” için GOLD çözüm

A CloudWatch Logs subscription → OpenSearch

✗ Neden uygun değil:

- CloudWatch Logs **doğrudan OpenSearch'e subscription veremez**
- Subscription'lar:
 - Lambda
 - Kinesis Data Streams
 - Kinesis Data Firehose
- Bu seçenek **teknik olarak geçersiz**

✗ Elenir

B Lambda ile logları OpenSearch'e yazmak

✓ Çalışır mı? → **Evet**

✗ Neden en iyi değil:

- Lambda kodu yazma
- Hata yönetimi, retry logic
- Throttling ve ölçek yönetimi
- OpenSearch index mapping yönetimi

 **Operasyonel yük orta seviyede**

D Kinesis Agent + Data Streams + OpenSearch

✗ Neden uygun değil:

- Her sunucuya agent kurulumu
- Agent konfigürasyonu ve bakımı
- Kinesis Data Streams shard yönetimi
- Ek operasyonel karmaşıklık

 **En yüksek operasyonel yük**

 **Sonuç**

Senaryo

En iyi servis

Logları OpenSearch'e gönder **Kinesis Data Firehose**

Near-real time + az operasyon **Firehose**

Agent istemiyorum

Firehose

Kod yazmak istemiyorum

Firehose

Neden Diğerleri Elendi?

- **A:** Teknik olarak mümkün değil
- **B:** Çalışır ama gereksiz kod ve bakım
- **D:** Agent + stream yönetimi → ağır

A company is building a web-based application running on Amazon EC2 instances in multiple Availability Zones. The web application will provide access to a repository of text documents totaling about 900 TB in size. The company anticipates that the web application will experience periods of high demand. A solutions architect must ensure that the storage component for the text documents can scale to meet the demand of the application at all times. The company is concerned about the overall cost of the solution.

Which storage solution meets these requirements MOST cost-effectively?

- A. Amazon Elastic Block Store (Amazon EBS)
- B. Amazon Elastic File System (Amazon EFS)
- C. Amazon OpenSearch Service (Amazon Elasticsearch Service)
- D. Amazon S3

Soru:

Bir şirket, **birden fazla Availability Zone** üzerinde çalışan **Amazon EC2 instance'ları** üzerinde çalışan **web tabanlı bir uygulama** geliştirmektedir. Web uygulaması, toplam boyutu yaklaşık **900 TB** olan **metin dokümanlarından oluşan bir depoya erişim** sağlayacaktır. Şirket, web uygulamasının **yüksek talep dönemleri** yaşayacağını öngörmektedir. Bir **solutions architect**, metin dokümanları için kullanılan **depolama bileşeninin**, uygulamanın talebini **her zaman karşılayacak şekilde ölçeklenebilmesini** sağlamalıdır. Şirket, çözümün **toplam maliyeti** konusunda endişelidir.

Bu gereksinimleri **en uygun maliyetli** şekilde karşılayan depolama çözümü hangisidir?

- A. Amazon Elastic Block Store (Amazon EBS)
- B. Amazon Elastic File System (Amazon EFS)
- C. Amazon OpenSearch Service (Amazon Elasticsearch Service)
- D. Amazon S3

Sorunun Analizi:

Sorudan çıkarılan **kritik anahtar kelimeler**:

- ◆ **Mimari**
 - Web tabanlı uygulama
 - **Amazon EC2**
 - **Birden fazla Availability Zone**
- ◆ **Veri özellikleri**

- **Metin dokümanları**
 - **~900 TB** (çok büyük veri)
 - Okuma ağırlıklı erişim (repository)
- ◆ **Performans & ölçeklenebilirlik**
 - **High demand periods**
 - Depolama **her zaman otomatik ölçeklenmeli**
 - ◆ **En önemli kriter**
 - **MOST cost-effective** (en düşük maliyet)

Seçenek Analizi:

D Amazon S3

✓ Neden en iyi seçenek:

- **Object storage**
- Pratikte **sınırsız ölçek**
- Multi-AZ (11×9 durability)
- Yüksek trafik dönemlerinde otomatik ölçeklenir
- **En düşük GB başına maliyet**
- 900 TB için **endüstri standarı**

✓ EC2'den erişim:

- SDK / HTTP
- S3 Gateway Endpoint (daha güvenli & ucuz)

✗ AWS sınavlarında “büyük, statik dosyalar” → S3

A Amazon EBS

✗ Neden uygun değil:

- **EBS block storage**
- Tek bir AZ'ye bağlı
- EC2 instance'a attach edilir
- 900 TB için:

- Çok sayıda volume
- Snapshot, replication, yönetim yükü
- **Çok pahalı** ve ölçeklenmesi zor

 **Kesinlikle elenir**

 **Amazon EFS**

 **Artıları:**

- Multi-AZ erişim
- EC2'ler tarafından aynı anda mount edilebilir
- Otomatik ölçeklenir

 **Neden ideal değil:**

- **900 TB** için **çok pahalı**
- Küçük dosya + metadata operasyonları maliyetli
- Sınavlarda:

“Shared file system” varsa EFS

“Massive object storage” varsa S3

 **Çalışır ama maliyet açısından kötü**

 **Amazon OpenSearch Service**

 **Neden tamamen yanlış:**

- Arama ve analiz motoru
- Log, metric, index için kullanılır
- Büyük dosya depolama için **tasarlanmamıştır**
- Çok pahalı olur

 **Doğru senaryo değil**

 **Sonuç**

Gereksinim

Doğru Servis

Çok büyük veri (100+ TB) **Amazon S3**

En düşük maliyet

Amazon S3

Gereksinim	Doğru Servis
Otomatik ölçeklenme	Amazon S3
Multi-AZ dayanıklılık	Amazon S3

Neden Bu Soru “Kolay Tuzak”?

- **EFS** seni “multi-AZ + EC2” ile kandırır
 - Ama **900 TB + cost-effective** diyorsa → **S3**
-

QUESTION 54

A global company is using Amazon API Gateway to design REST APIs for its loyalty club users in the us-east-1 Region and the ap-southeast-2 Region. A solutions architect must design a solution to protect these API Gateway managed REST APIs across multiple accounts from SQL injection and cross-site scripting attacks.

Which solution will meet these requirements with the LEAST amount of administrative effort?

- Set up AWS WAF in both Regions. Associate Regional web ACLs with an API stage.
- Set up AWS Firewall Manager in both Regions. Centrally configure AWS WAF rules.
- Set up AWS Shield in both Regions. Associate Regional web ACLs with an API stage.
- Set up AWS Shield in one of the Regions. Associate Regional web ACLs with an API stage.

Soru:

Küresel bir şirket, **us-east-1** ve **ap-southeast-2** bölgelerinde bulunan **loyalty club** kullanıcıları için **Amazon API Gateway** kullanarak **REST API’ler** tasarlamaktadır. Bir **solutions architect**, bu **API Gateway** tarafından yönetilen **REST API’leri**, **birden fazla AWS hesabı** genelinde **SQL injection** ve **cross-site scripting (XSS)** saldırılara karşı koruyacak bir çözüm tasarlamalıdır.

En az yönetimsel (idari) efor ile bu gereksinimleri karşılayan çözüm hangisidir?

- Her iki bölgede de **AWS WAF** kurmak ve **Regional web ACL’leri** bir **API stage** ile ilişkilendirmek.
- Her iki bölgede de **AWS Firewall Manager** kurmak ve **AWS WAF** kurallarını **merkezi olarak yapılandırmak**.
- Her iki bölgede de **AWS Shield** kurmak ve **Regional web ACL’leri** bir **API stage** ile ilişkilendirmek.

D. Bölgelerden birinde **AWS Shield** kurmak ve **Regional web ACL'leri** bir **API stage** ile ilişkilendirmek.

Sorunun Analizi:

Sorudan çıkan **kritik anahtar ifadeler**:

- ◆ **Mimari**

- **Amazon API Gateway (REST API)**
- **2 farklı Region**: us-east-1, ap-southeast-2
- **Birden fazla AWS account**

- ◆ **Güvenlik gereksinimi**

- **SQL injection**
- **Cross-site scripting (XSS)**

→ Bu saldırı türleri **L7 (application layer)** saldırılarıdır

→ Çözüm mutlaka **AWS WAF** içermelidir

- ◆ **En önemli kriter**

- **LEAST administrative effort**
- Merkezi yönetim, tek noktadan kontrol tercih edilir

Servislerin Rolünü Hatırlayalım

AWS WAF

- SQLi ve XSS'e karşı koruma sağlar
- API Gateway **Regional endpoint'leriyle** entegre olur
- Web ACL bazlı çalışır

AWS Shield

- **DDoS (L3/L4)** koruması
- SQLi / XSS yapmaz

AWS Firewall Manager

- **Çoklu account + çoklu Region** için
- WAF, Shield, Security Groups kurallarını **merkezi yönetir**
- En az idari yük = **Firewall Manager**

Seçenek Analizi:

B AWS Firewall Manager ile merkezi AWS WAF yönetimi

✓ Artıları:

- Multi-account
- Multi-Region
- Tek merkezden WAF rule'ları
- Otomatik policy enforcement
- Yeni API'ler otomatik korunur

✗ En az yönetimsel efor

✗ AWS sınavlarında doğru cevap kalımı

A Her iki Region'da AWS WAF kurmak, API stage'e bağlamak

✓ Teknik olarak çalışır

✗ Neden en iyi değil:

- Her account & Region için
 - Ayrı web ACL
 - Ayrı rule güncellemesi
- Manuel ve dağınık yönetim

✗ İdari yük yüksek

C AWS Shield + web ACL

✗ Yanlış:

- Shield → DDoS
- SQLi & XSS koruması yok
- Web ACL kavramı WAF'e aittir

✗ Teknik olarak yanlış

D Tek Region'da Shield + web ACL

✗ Çift hata:

- Shield yanlış servis

- Tek Region → global mimariye uymuyor

 **Kesinlikleelenir**

 **Sonuç**

Eğer soruda

- multi-account**
- multi-region**
- least administrative effort**

görüyorsan

 **AWS Firewall Manager** düşün

Mini Özeti

Gereksinim Servis

SQLi / XSS AWS WAF

Merkezi yönetim Firewall Manager

Çoklu hesap Firewall Manager

En az idari yük Firewall Manager

QUESTION 55

A company has implemented a self-managed DNS solution on three Amazon EC2 instances behind a Network Load Balancer (NLB) in the us-west-2 Region. Most of the company's users are located in the United States and Europe. The company wants to improve the performance and availability of the solution. The company launches and configures three EC2 instances in the eu-west-1 Region and adds the EC2 instances as targets for a new NLB.

Which solution can the company use to route traffic to all the EC2 instances?

- Create an Amazon Route 53 geolocation routing policy to route requests to one of the two NLBs. Create an Amazon CloudFront distribution. Use the Route 53 record as the distribution's origin.
- Create a standard accelerator in AWS Global Accelerator. Create endpoint groups in us-west-2 and eu-west-1. Add the two NLBs as endpoints for the endpoint groups.
- Attach Elastic IP addresses to the six EC2 instances. Create an Amazon Route 53 geolocation routing policy to route requests to one of the six EC2 instances. Create an Amazon CloudFront distribution. Use the Route 53 record as the distribution's origin.

D. Replace the two NLBs with two Application Load Balancers (ALBs). Create an Amazon Route 53 latency routing policy to route requests to one of the two ALBs. Create an Amazon CloudFront distribution. Use the Route 53 record as the distribution's origin.

Soru:

Bir şirket, **us-west-2** bölgesinde bir **Network Load Balancer (NLB)** arkasında çalışan **üç Amazon EC2 instance** üzerinde **kendi yönettiği (self-managed)** bir **DNS çözümü** kullanmaktadır. Şirket kullanıcılarının çoğu **Amerika Birleşik Devletleri** ve **Avrupa**'da bulunmaktadır. Şirket, çözümün **performansını ve erişilebilirliğini artırmak** istemektedir. Bu amaçla şirket, **eu-west-1** bölgesinde üç adet EC2 instance başlatıp yapılandırır ve bu EC2 instance'ları hedef olarak kullanan yeni bir **NLB** oluşturur.

Şirket, **tüm EC2 instance'lara trafiği yönlendirmek** için aşağıdaki çözümlerden hangisini kullanabilir?

- A. İki NLB'den birine istekleri yönlendirmek için **Amazon Route 53 geolocation routing policy** oluşturmak. Route 53 kaydını origin olarak kullanan bir **Amazon CloudFront distribution** oluşturmak.
- B. **AWS Global Accelerator**'da bir **standard accelerator** oluşturmak. **us-west-2** ve **eu-west-1** bölgelerinde **endpoint group'lar** oluşturmak. Her bir endpoint group için iki NLB'yi endpoint olarak eklemek.
- C. Altı EC2 instance'a **Elastic IP adresleri** atamak. Altı EC2 instance'tan birine istekleri yönlendirmek için **Amazon Route 53 geolocation routing policy** oluşturmak. Route 53 kaydını origin olarak kullanan bir **Amazon CloudFront distribution** oluşturmak.
- D. İki NLB'yi iki **Application Load Balancer (ALB)** ile değiştirmek. İki ALB'den birine istekleri yönlendirmek için **Amazon Route 53 latency routing policy** oluşturmak. Route 53 kaydını origin olarak kullanan bir **Amazon CloudFront distribution** oluşturmak.

Soru Analizi:

Sorudan çıkan **kritik noktalar**:

- ◆ **Mevcut mimari**
 - **Self-managed DNS**
 - **EC2 + Network Load Balancer (NLB)**
 - 2 Region:
 - **us-west-2**
 - **eu-west-1**
 - Her region'da **ayrı NLB**

- ◆ **Kullanıcı dağılımı**
 - ABD ve Avrupa ağırlıklı
- ◆ **Hedef**
 - **Performance improvement**
 - **High availability**
 - **Tüm EC2 instance'lara trafik yönlendirme**

→ Bu, **global traffic routing** problemidir.

Seçenek Analizi:

B AWS Global Accelerator + NLB (2 Region)

Bu Senaryo İçin En Uygun AWS Servisi Hangisi?

Bu tür senaryolarda AWS'nin “gold standard” çözümü:

🌐 AWS Global Accelerator

- Anycast IP
- Kullanıcıyı **en yakın ve sağlıklı endpoint'e**
- TCP / UDP destekler → **NLB uyumlu**
- DNS gibi protokoller için **çok uygundur**
- Route 53 + CloudFront'a göre:
 - Daha düşük latency
 - Daha basit mimari
 - Daha az bileşen

✓ Artıları:

- Global Accelerator:
 - Kullanıcıyı AWS edge üzerinden alır
 - En yakın ve sağlıklı NLB'ye yönlendirir
- NLB:
 - TCP/UDP
 - DNS workload'ları için ideal
- Health check + otomatik failover

- Tek giriş noktası (2 Anycast IP)
- ➔ **Performans + HA + düşük operasyonel yük**
- ➔ **Tam olarak sorunun istediği çözüm A**

Route 53 Geolocation + CloudFront + NLB

✗ Neden uygun değil:

- CloudFront **HTTP/HTTPS ağırlıklıdır**
- DNS trafiği için **anlamsız**
- Geolocation routing:
 - Kullanıcıyı “yakın” değil “ülkeye göre” yönlendirir
- Gereksiz karmaşıklık

➔ **Elenir**

C Elastic IP + Route 53 + CloudFront

✗ Neden kötü:

- 6 EC2'ye EIP bağlamak
- Tek tek instance yönetimi
- Load balancer bypass edilmiş olur
- CloudFront yine DNS için anlamsız

➔ **Operasyonel yük çok yüksek**

D ALB + Route 53 Latency + CloudFront

✗ Birden fazla hata:

- ALB **Layer 7 (HTTP/HTTPS)**
- DNS → **Layer 4 (UDP/TCP)**
- CloudFront DNS için uygun değil
- Mevcut NLB'leri değiştirmek gereksiz

➔ **Teknik olarak yanlış mimari**

🎯 **Sonuç**

Aşağıdaki kelimeleri göründüysen otomatik düşün:

Anahtar ifade	Doğru servis
Global users	Global Accelerator
Multi-region	Global Accelerator
TCP / UDP	NLB + Global Accelerator
Performance + HA	Global Accelerator

Mini Özeti

- Route 53 → **DNS-level routing**
 - CloudFront → **HTTP/HTTPS**
 - ALB → **L7**
 - **NLB + Global Accelerator → Global TCP/UDP workloads**
-

QUESTION 56

A company is running an online transaction processing (OLTP) workload on AWS. This workload uses an unencrypted Amazon RDS DB instance in a Multi-AZ deployment. Daily database snapshots are taken from this instance.

What should a solutions architect do to ensure the database and snapshots are always encrypted moving forward?

- Encrypt a copy of the latest DB snapshot. Replace existing DB instance by restoring the encrypted snapshot.
- Create a new encrypted Amazon Elastic Block Store (Amazon EBS) volume and copy the snapshots to it. Enable encryption on the DB instance.
- Copy the snapshots and enable encryption using AWS Key Management Service (AWS KMS). Restore encrypted snapshot to an existing DB instance.
- Copy the snapshots to an Amazon S3 bucket that is encrypted using server-side encryption with AWS Key Management Service (AWS KMS) managed keys (SSE-KMS).

Soru:

Bir şirket, AWS üzerinde **çevrim içi işlem işleme (OLTP)** iş yükü çalıştırmaktadır. Bu iş yükü, **Multi-AZ** dağıtımında çalışan **şifrelenmemiş (unencrypted)** bir **Amazon RDS DB instance** kullanmaktadır. Bu instance'tan **günlük veritabanı snapshot'ları alınmaktadır.**

Bir **solutions architect**, veritabanının ve snapshot'ların **bundan sonra her zaman şifreli (encrypted)** olmasını sağlamak istemektedir.

Buna göre aşağıdaki seçeneklerden hangisi yapılmalıdır?

- A. En son DB snapshot'ının şifreli bir kopyasını oluşturmak. Mevcut DB instance'i, bu şifreli snapshot'tan restore ederek değiştirmek.
- B. Yeni bir **şifreli Amazon EBS volume** oluşturmak ve snapshot'ları bu volume'a kopyalamak. DB instance üzerinde şifrelemeyi etkinleştirmek.
- C. Snapshot'ları kopyalayıp **AWS Key Management Service (AWS KMS)** kullanarak şifrelemeyi etkinleştirmek. Şifreli snapshot'ı mevcut bir DB instance'a restore etmek.
- D. Snapshot'ları, **AWS Key Management Service (AWS KMS)** tarafından yönetilen anahtarlar (SSE-KMS) kullanılarak **server-side encryption** etkin olan bir **Amazon S3 bucket**'a kopyalamak.

Soru Analizi:

Sorudan çıkan **kritik noktalar**:

- ◆ **Mevcut durum**
 - **Amazon RDS**
 - **Unencrypted DB instance**
 - **Multi-AZ**
 - **Günlük DB snapshot'ları**
- ◆ **Hedef**
 - **Bundan sonra her zaman şifreli**
 - DB instance
 - Snapshot'lar

⚠ AWS kuralı (çok önemli):

Var olan bir RDS DB instance üzerinde **encryption sonradan AÇILAMAZ**.

→ Bu yüzden **yeni, şifreli bir DB instance** oluşturulmalıdır.

AWS'de RDS Şifrelemesi Nasıl Yapılır?

Doğru yol:

1. **Mevcut snapshot'ı kopyala**
2. Kopyalama sırasında **encryption etkinleştir**

3. Encrypted snapshot'tan yeni DB instance restore et
4. Yeni DB instance → otomatik olarak:
 - Şifreli storage
 - Şifreli snapshot'lar

Seçenek Analizi:

A Encrypt snapshot → restore → replace DB instance

✓ Tam olarak AWS'nin önerdiği yöntem

- Snapshot kopyalanırken encryption açılır
- Yeni DB instance encrypted olur
- Gelecek snapshot'lar da encrypted olur

✗ Doğru ve eksiksiz çözüm

B EBS volume oluştur → snapshot kopyala → DB'de encryption aç

✗ Yanlış:

- RDS storage EBS olsa bile **kullanıcı EBS yönetemez**
- RDS DB instance üzerinde encryption **sonradan etkinleştirilemez**

✗ Teknik olarak imkânsız

C Snapshot'ı kopyala → encrypted → mevcut DB instance'a restore et

✗ Yanlış ifade:

- **Mevcut (existing) DB instance üzerine restore edilemez**
- Restore işlemi **her zaman yeni DB instance** oluşturur

✗ Yanlıltıcı ve hatalı

D Snapshot'ı S3'e kopyala (SSE-KMS)

✗ Yanlış:

- RDS snapshot'ları S3'e manuel olarak kopyalanmaz
- Bu, DB encryption sağlamaz
- DB instance hâlâ unencrypted kalır

✗ Gereksinimi karşılamaz

Sonuç

Bu cümleyi görürsen **refleks** geliştirir:

! “Existing unencrypted RDS → encrypted”

Snapshot copy + encryption + restore

Kısa Özeti

İşlem	Mümkün mü?
Var olan RDS'ye encryption eklemek	
Encrypted snapshot'tan restore	
Gelecek snapshot'ların encrypted olması	

RDS Şifreleme – Hızlı Karşılaştırma Tablosu

Konu	Açıklama
Mevcut (unencrypted) RDS'de encryption açılabilir mi?	 Hayır
RDS şifrelemesi ne zaman belirlenir?	 DB oluşturulurken
Unencrypted DB'yi encrypted yapmanın tek yolu	 Snapshot kopyala → encryption aç → restore
Restore işlemi mevcut DB'ye yapılabilir mi?	 Hayır (yeni DB oluşur)
Encrypted DB'den alınan snapshot'lar	 Otomatik encrypted
Multi-AZ durumu	 Restore edilen DB yine Multi-AZ olabilir
Kullanılan servis	AWS KMS

Sınavda Altın Kural

“Existing unencrypted RDS” görürsen

Snapshot copy + encryption + restore

Başka yol yok

A company wants to build a scalable key management infrastructure to support developers who need to encrypt data in their applications.

What should a solutions architect do to reduce the operational burden?

- A. Use multi-factor authentication (MFA) to protect the encryption keys.
- B. Use AWS Key Management Service (AWS KMS) to protect the encryption keys.
- C. Use AWS Certificate Manager (ACM) to create, store, and assign the encryption keys.
- D. Use an IAM policy to limit the scope of users who have access permissions to protect the encryption keys.

Soru:

Bir şirket, uygulamalarında verileri şifrelemesi gereken geliştiricileri desteklemek için **ölçeklenebilir bir anahtar yönetim altyapısı (key management infrastructure)** oluşturmak istemektedir. Bir **solutions architect**, **operasyonel yükü azaltmak** istemektedir.

Bu gereksinimi karşılamak için aşağıdakilerden hangisi yapılmalıdır?

- A. Şifreleme anahtarlarını korumak için **çok faktörlü kimlik doğrulama (MFA)** kullanmak.
- B. Şifreleme anahtarlarını korumak için **AWS Key Management Service (AWS KMS)** kullanmak.
- C. Şifreleme anahtarlarını oluşturmak, saklamak ve atamak için **AWS Certificate Manager (ACM)** kullanmak.
- D. Şifreleme anahtarlarını korumak için erişim yetkisine sahip kullanıcıların kapsamını sınırlayan bir **IAM policy** kullanmak.

Soru Analizi:

Sorudan çıkan **kritik ifadeler**:

- ◆ **Amaç**
 - **Key management infrastructure**
 - Geliştiriciler uygulamalarda **data encryption** yapacak
- ◆ **Teknik gereksinimler**
 - **Scalable**
 - **Operational burden düşük olmalı**

→ Yani:

- Anahtar üretimi
 - Saklama
 - Rotasyon
 - Erişim kontrolü
- bunların **AWS tarafından yönetilmesi** isteniyor.

Seçenek Analizi:

B AWS KMS kullanmak

Bu Senaryo İçin En Doğru AWS Servisi

🔒 AWS Key Management Service (KMS)

- Fully managed
- Highly available
- Otomatik key rotation
- IAM ile entegre
- AWS servisleriyle native çalışır
- Geliştiriciler için API/SDK hazır

→ En az operasyonel yük = KMS

✓ Artıları:

- Ölçeklenebilir
- Fully managed
- Anahtar rotasyonu otomatik
- Geliştiriciler doğrudan kullanabilir

📌 Tam olarak sorunun istediği çözüm

A MFA ile anahtarları korumak

✗ Yetersiz:

- **MFA key management çözümü değildir**
- Anahtar üretmez, saklamaz, ölçeklemez
- Sadece kullanıcı giriş güvenliği sağlar

📌 Gereksinimi karşılamaz

AWS Certificate Manager (ACM)

 Yanlış servis:

- TLS/SSL sertifikaları içindir
- Data encryption key management yapmaz

 Kavramsal olarak yanlış

IAM policy ile erişimi sınırlandırmak

 Yetersiz:

- IAM policy **destekleyici bir kontroldür**
- Anahtar yönetimi sağlamaz
- Tek başına çözüm değildir

 Eksik çözüm

 Sonuç

Eğer soruda

- encryption
- key management
- low operational overhead

geçiyorsa

 AWS KMS

Mini Özeti

Gereksinim Servis

Key management AWS KMS

Otomatik rotasyon AWS KMS

Ölçeklenebilirlik AWS KMS

Düşük operasyon AWS KMS

QUESTION 58

A company has a dynamic web application hosted on two Amazon EC2 instances. The company has its own SSL certificate, which is on each instance to perform SSL termination.

There has been an increase in traffic recently, and the operations team determined that SSL encryption and decryption is causing the compute capacity of the web servers to reach their maximum limit.

What should a solutions architect do to increase the application's performance?

- A. Create a new SSL certificate using AWS Certificate Manager (ACM). Install the ACM certificate on each instance.
- B. Create an Amazon S3 bucket. Migrate the SSL certificate to the S3 bucket. Configure the EC2 instances to reference the bucket for SSL termination.
- C. Create another EC2 instance as a proxy server. Migrate the SSL certificate to the new instance and configure it to direct connections to the existing EC2 instances.
- D. Import the SSL certificate into AWS Certificate Manager (ACM). Create an Application Load Balancer with an HTTPS listener that uses the SSL certificate from ACM.

Soru:

Bir şirketin **iki Amazon EC2 instance** üzerinde barındırılan **dinamik bir web uygulaması** vardır. Şirketin **kendi SSL sertifikası** bulunmaktadır ve SSL sonlandırma (SSL termination) işlemi her bir instance üzerinde yapılmaktadır.

Son zamanlarda trafikte artış yaşanmıştır ve operasyon ekibi, **SSL şifreleme ve şifre çözme işlemlerinin**, web sunucularının **hesaplama kapasitesinin maksimum seviyeye ulaşmasına** neden olduğunu tespit etmiştir.

Bir **solutions architect**, uygulamanın **performansını artırmak** için ne yapmalıdır?

- A. **AWS Certificate Manager (ACM)** kullanarak yeni bir SSL sertifikası oluşturmak ve bu sertifikayı her bir instance'a yüklemek.
- B. Bir **Amazon S3 bucket** oluşturmak. SSL sertifikasını bu bucket'a taşımak ve EC2 instance'larını SSL termination için bu bucket'ı referans alacak şekilde yapılandırmak.
- C. **Proxy sunucusu** olarak yeni bir EC2 instance oluşturmak. SSL sertifikasını bu yeni instance'a taşımak ve bağlantıları mevcut EC2 instance'lara yönlendirecek şekilde yapılandırmak.
- D. SSL sertifikasını **AWS Certificate Manager (ACM)** içine aktarmak. **HTTPS listener** kullanan bir **Application Load Balancer (ALB)** oluşturmak ve ACM'deki SSL sertifikasını bu listener'da kullanmak.

Soru Analizi:

◆ **Mevcut durum**

- 2 adet EC2 instance

- **SSL termination EC2 üzerinde**
 - Dinamik web uygulaması
 - Trafik artışı var
- ◆ **Tespit edilen sorun**
- **SSL encryption/decryption CPU tüketiyor**
 - Web sunucuları **compute limitine ulaşıyor**

→ Asıl problem:

SSL termination uygulama sunucularının üzerinden alınmalı

Seçenek Analizi:

SSL sertifikasını ACM'ye import edip ALB kullanmak

En Doğru Mimari Yaklaşım

Load Balancer'da SSL Termination

AWS best practice:

- SSL termination → **Application Load Balancer**
- Sertifika yönetimi → **AWS Certificate Manager (ACM)**
- EC2'ler sadece **HTTP / business logic** çalıştırır

→ Bu, CPU yükünü ciddi şekilde azaltır.

✓ Artıları:

- SSL termination **ALB'de**
- CPU yükü EC2'den kalkar
- ALB otomatik ölçeklenir
- Sertifika otomatik yenilenir
- AWS best practice

📌 **Tam çözüm**

📌 **En iyi performans artışı**

ACM sertifikası oluşturup EC2'lere kurmak

✗ Yanlış:

- SSL termination hâlâ EC2 üzerinde

- CPU problemi **çözülmez**
- Sertifika yönetimi zorlaşır

✖ **Performans artışı sağlama**

✖ **SSL sertifikasını S3'e koyup EC2'lerin kullanması**

✖ Tamamen yanlış:

- S3 **SSL termination yapmaz**
- Teknik olarak mümkün değil

✖ **Elenir**

✖ **Proxy olarak EC2 eklemek**

✖ Kötü mimari:

- Ek EC2 → patch, scale, HA
- Tek noktadan arıza riski
- ALB varken gereksiz

✖ **Operasyonel yük artar**

✖ **Sonuç**

Bu kelimeleri görürsen refleks geliştirir:

Anahtar kelime Doğru aksiyon

SSL CPU heavy ALB'de termination

EC2 CPU max Load Balancer

SSL cert ACM

Web app ALB

Mini Özeti

- SSL termination **EC2'de yapılmamalı**
- **ALB + ACM** = performans + düşük operasyon
- Proxy EC2 → ✖
- Sertifikayı EC2'ye kurmak → ✖

QUESTION 59

A company has a highly dynamic batch processing job that uses many Amazon EC2 instances to complete it. The job is stateless in nature, can be started and stopped at any given time with no negative impact, and typically takes upwards of 60 minutes total to complete. The company has asked a solutions architect to design a scalable and cost-effective solution that meets the requirements of the job.

What should the solutions architect recommend?

- A. Implement EC2 Spot Instances.
- B. Purchase EC2 Reserved Instances.
- C. Implement EC2 On-Demand Instances.
- D. Implement the processing on AWS Lambda.

Soru:

Bir şirket, tamamlamak için çok sayıda Amazon EC2 instance kullanan son derece dinamik bir batch processing işi çalıştırmaktadır. Bu iş stateless yapıdadır herhangi bir zamanda başlatılıp durdurulabilir ve bunun olumsuz bir etkisi yoktur ve genellikle tamamlanması 60 dakikadan uzun sürmektedir. Şirket bir solutions architectten bu işin gereksinimlerini karşılayan ölçeklenebilir ve maliyet açısından verimli bir çözüm tasarlamasını istemektedir. Solutions architect ne önermelidir?

- A EC2 Spot Instances uygulamak
- B EC2 Reserved Instances satın almak
- C EC2 On Demand Instances uygulamak
- D İşlemeyi AWS Lambda üzerinde gerçekleştirmek

Soru Analizi:

Sorudaki anahtar ifadeler:

- **Highly dynamic batch processing job**
- **Many EC2 instances**
- **Stateless**
- **Can be started and stopped at any time**
- **No negative impact**
- **Takes more than 60 minutes**
- **Scalable and cost-effective**

→ Bu tanım, **kesintiye dayanıklı (fault-tolerant)** bir iş yükünü tarif eder

Seçenek Analizi:

A EC2 Spot Instances

Bu İş Yükü İçin En Uygun EC2 Satın Alma Modeli

🔥 EC2 Spot Instances

- %70–90'a kadar daha ucuz
- İstenildiğinde kesilebilir
- Stateless ve yeniden başlatılabilir işler için ideal
- Büyük batch job'lar için AWS best practice

✓ Tam uyumlu

- Kesintiye toleranslı
- Çok düşük maliyet
- Yüksek ölçeklenebilirlik

📌 En iyi ve beklenen çözüm

B EC2 Reserved Instances

✗ Uygun değil

- Uzun süreli, sürekli çalışan workload'lar için
- Esnekliği düşük
- Dinamik batch job'a ters

📌 Elenir

C EC2 On-Demand Instances

✗ Gereksiz pahalı

- Teknik olarak çalışır
- Ama **cost-effective değil**

📌 İkinci planda kalır

D AWS Lambda

✗ Teknik sınır

- Maksimum çalışma süresi **15 dakika**
- İş 60+ dakika sürüyor

Teknik olarak imkânsız

Sonuç

Bu kelimeleri görürsen otomatik düşün:

Anahtar ifade Doğru çözüm

Stateless batch job Spot Instances

Can be interrupted Spot

Cost-effective Spot

Long-running EC2 (Lambda değil)

Mini Özeti

- İş kesintiye dayanıklı → **Spot**
- Ucuz + ölçülebilir → **Spot**
- Lambda süre sınırı → 
- Reserved = sürekli workload → 

QUESTION 60

A company runs its two-tier ecommerce website on AWS. The web tier consists of a load balancer that sends traffic to Amazon EC2 instances. The database tier uses an Amazon RDS DB instance. The EC2 instances and the RDS DB instance should not be exposed to the public internet. The EC2 instances require internet access to complete payment processing of orders through a third-party web service. The application must be highly available.

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

- A. Use an Auto Scaling group to launch the EC2 instances in private subnets. Deploy an RDS Multi-AZ DB instance in private subnets.
- B. Configure a VPC with two private subnets and two NAT gateways across two Availability Zones. Deploy an Application Load Balancer in the private subnets.

- C. Use an Auto Scaling group to launch the EC2 instances in public subnets across two Availability Zones. Deploy an RDS Multi-AZ DB instance in private subnets.
- D. Configure a VPC with one public subnet, one private subnet, and two NAT gateways across two Availability Zones. Deploy an Application Load Balancer in the public subnet.
- E. Configure a VPC with two public subnets, two private subnets, and two NAT gateways across two Availability Zones. Deploy an Application Load Balancer in the public subnets.

Soru:

Bir şirket, **iki katmanlı (two-tier)** bir **e-ticaret web sitesini** AWS üzerinde çalıştırmaktadır. **Web katmanı**, trafiği **Amazon EC2 instance'larına** yönlendiren bir **load balancer**'dan oluşmaktadır. **Veritabanı katmanı**, bir **Amazon RDS DB instance** kullanmaktadır. **EC2 instance'larının ve RDS DB instance'ının genel internete açık olmaması** gerekmektedir. Ancak **EC2 instance'larının**, siparişlerin ödeme işlemlerini gerçekleştirmek için **üçüncü taraf bir web servisine** erişebilmesi için **internete çıkış erişimine** ihtiyacı vardır. Uygulama yüksek erişilebilir (**highly available**) olmalıdır.

Bu gereksinimleri karşılayan **konfigürasyon seçenekleri kombinasyonu** hangisidir? (**İki tane诜iniz.**)

- A. EC2 instance'larını **private subnet'lerde** başlatmak için bir **Auto Scaling group** kullanmak. **Private subnet'lerde** bir **RDS Multi-AZ DB instance** dağıtmak.
- B. İki **Availability Zone** boyunca iki **private subnet** ve iki **NAT gateway** içeren bir **VPC** yapılandırmak. **Private subnet'lerde** bir **Application Load Balancer** dağıtmak.
- C. EC2 instance'larını iki Availability Zone boyunca **public subnet'lerde** başlatmak için bir **Auto Scaling group** kullanmak. **Private subnet'lerde** bir **RDS Multi-AZ DB instance** dağıtmak.
- D. İki Availability Zone boyunca iki **NAT gateway** içeren bir **VPC** yapılandırmak. Bir **public subnet** ve bir **private subnet** oluşturmak. **Public subnet'te** bir **Application Load Balancer** dağıtmak.
- E. İki Availability Zone boyunca iki **public subnet**, iki **private subnet** ve iki **NAT gateway** içeren bir **VPC** yapılandırmak. **Public subnet'lerde** bir **Application Load Balancer** dağıtmak.

Soru Analizi:

Sorudan çıkan **kritik gereksinimler**:

- ◆ **Mimari**
 - **Two-tier ecommerce application**

- Web tier: Load Balancer + EC2
- DB tier: Amazon RDS

◆ **Güvenlik**

- **EC2 ve RDS interne a^cık olmamalı**
 - Public IP almamalılar
- ◆ **Internet erişimi**
- EC2'ler **outbound internet erişimine ihtiyaç duyuyor**
 - (3rd-party payment service)

◆ **High Availability**

- **Multi-AZ**
- Tek NAT veya tek subnet →

Bu Senaryo İçin Doğru AWS Mimarisi (Best Practice)

AWS'nin önerdiği **klasik two-tier secure architecture**:

- **ALB → Public Subnet**
- **EC2 → Private Subnet (Auto Scaling, Multi-AZ)**
- **RDS → Private Subnet (Multi-AZ)**
- **NAT Gateway → Her AZ'de 1 tane**
 - EC2'ler internete **sadece outbound** erişir

Seçenek Analizi:

EC2 private subnet + RDS private subnet (Multi-AZ)

- Doğru parça
- EC2'ler internete açık değil
 - RDS private subnet'te

Eksik:

- **NAT Gateway yok**
- EC2'ler internete çıkamaz

Tek başına yeterli değil ama doğru bir yarı çözüm

D 2 public subnet + 2 private subnet + 2 NAT gateway + ALB public

Tam doğru mimari

- ALB public subnet'lerde
- EC2 private subnet'lerde
- RDS private subnet'lerde
- NAT Gateway her AZ'de
- High availability sağlanır

AWS reference architecture ile birebir

B ALB private subnet + NAT + private subnet'ler

Yanlış:

- **ALB private subnet'te olamaz** (internet-facing olması gereklidir)
- Kullanıcılar web sitesine erişemez

Elenir

C EC2 public subnet + RDS private subnet

Yanlış:

- EC2'ler **public subnet'te**
- İnternete açık → güvenlik gereksinimine aykırı

Elenir

D 1 public + 1 private subnet + 2 NAT gateway

Yanlış:

- High availability için **her AZ'de subnet** gereklidir
- Tek public/private subnet yeterli değil

Elenir

Sonuç

Neden Bu İkiisi Birlikte?

Gereksinim	A	E
EC2 private subnet	✓	✓
RDS private subnet	✓	✓
Internet-facing ALB	✗	✓
NAT Gateway (outbound internet)	✗	✓
Multi-AZ HA	⚠	✓

👉 A, compute + DB yerleşimini doğru tanımlar

👉 E, network + internet erişimini doğru tanımlar

Birlikte kullanıldığında **tüm gereksinimler eksiksiz karşılanır**

Eğer soruda

- ✓ EC2 internete çıkacak ama inbound almayacak
- ✓ RDS private olacak
- ✓ High availability isteniyor

👉 **Public ALB + Private EC2 + NAT Gateway (per AZ)**

QUESTION 61

A solutions architect needs to implement a solution to reduce a company's storage costs. All the company's data is in the Amazon S3 Standard storage class. The company must keep all data for at least 25 years. Data from the most recent 2 years must be highly available and immediately retrievable.

Which solution will meet these requirements?

- A. Set up an S3 Lifecycle policy to transition objects to S3 Glacier Deep Archive immediately.
- B. Set up an S3 Lifecycle policy to transition objects to S3 Glacier Deep Archive after 2 years.
- C. Use S3 Intelligent-Tiering. Activate the archiving option to ensure that data is archived in S3 Glacier Deep Archive.
- D. Set up an S3 Lifecycle policy to transition objects to S3 One Zone-Infrequent Access (S3 One Zone-IA) immediately and to S3 Glacier Deep Archive after 2 years.

Soru:

Bir **solutions architect**, bir şirketin **depolama maliyetlerini azaltmak** için bir çözüm uygulamak istemektedir. Şirketin tüm verileri şu anda **Amazon S3 Standard** depolama sınıfındadır. Şirket, **tüm verileri en az 25 yıl boyunca saklamak** zorundadır. **En son 2 yıla ait verilerin, yüksek erişilebilirliğe sahip olması** ve **anında erişilebilir (immediately retrievable)** olması gerekmektedir.

Bu gereksinimleri karşılayan çözüm hangisidir?

- A. Nesneleri **hemen S3 Glacier Deep Archive**'a taşıyacak bir **S3 Lifecycle policy** oluşturmak.
- B. Nesneleri **2 yıl sonra S3 Glacier Deep Archive**'a taşıyacak bir **S3 Lifecycle policy** oluşturmak.
- C. **S3 Intelligent-Tiering** kullanmak ve verilerin **S3 Glacier Deep Archive**'a arşivlenmesini sağlamak için **archiving seçenekini** etkinleştirmek.
- D. Nesneleri **hemen S3 One Zone-Infrequent Access (S3 One Zone-IA)**'a, **2 yıl sonra ise S3 Glacier Deep Archive**'a taşıyacak bir **S3 Lifecycle policy** oluşturmak.

Soru Analizi:

Sorudan çıkan **kritik noktalar**:

- ◆ **Mevcut durum**
 - Tüm veriler **S3 Standard**'da
 - Amaç: **storage cost reduction**
- ◆ **Saklama zorunluluğu**
 - **En az 25 yıl** saklanmalı (long-term retention)
- ◆ **Erişim gereksinimi**
 - **Son 2 yılın verileri:**
 - **Highly available**
 - **Immediately retrievable** (milisaniyeler–saniyeler)

→ Yani:

- İlk 2 yıl: **S3 Standard** benzeri erişim
- 2 yıldan eski veriler: **En ucuz arşiv sınıfı**

İlgili S3 Storage Class'larını Hatırlayalım

Storage class	Erişim süresi	Kullanım
S3 Standard	Anında	Sık erişim
S3 One Zone-IA	Anında	Daha ucuz ama tek AZ
S3 Glacier Deep Archive	Saatler (12+ saat)	En ucuz uzun süreli arşiv
S3 Intelligent-Tiering	Otomatik	Ama archive opsyonu dikkatli

Seçenek Analizi:

B Verileri 2 yıl sonra S3 Glacier Deep Archive'a taşımak

✓ Doğru:

- İlk 2 yıl:
 - S3 Standard'da kalır
 - Anında erişim + yüksek erişilebilirlik
- 2 yıldan sonra:
 - En düşük maliyetli arşiv sınıfı

✗ Gereksinimlerle birebir uyumlu

✗ En net ve temiz çözüm

A Verileri hemen S3 Glacier Deep Archive'a taşımak

✗ Yanlış:

- Glacier Deep Archive:
 - **Immediate retrieval yok**
 - Saatler süren restore
- Son 2 yıl için gereksinime aykırı

✗ Elenir

C S3 Intelligent-Tiering + archiving option

✗ Riskli / uygun değil:

- Archive tier'lara geçiş:
 - **Immediate retrieval garanti değil**
- Soruda:

- Davranış net tanımlı isteniyor (2 yıl kuralı)

 **Belirsiz ve sınavda tercih edilmez**

D Hemen S3 One Zone-IA, sonra Glacier Deep Archive

 Yanlış:

- Son 2 yıl için:
 - **Highly available** şartı var
- One Zone-IA:
 - Tek AZ
 - Daha düşük availability

 **Availability gereksinimini ihlal eder**

 **Sonuç**

Bu kelimeleri birlikte görürsen refleks geliştirir:

- **Long-term retention (10+ yıl)**
- **Immediate retrieval for recent data**
- **Cost optimization**

 **Lifecycle policy + Glacier Deep Archive (zaman gecikmeli)**

Kısa Özeti

Gereksinim **Çözüm**

Son 2 yıl hızlı erişim S3 Standard

2+ yıl düşük maliyet Glacier Deep Archive

Otomasyon S3 Lifecycle

En düşük maliyet Glacier Deep Archive

QUESTION 62

A media company is evaluating the possibility of moving its systems to the AWS Cloud. The company needs at least 10 TB of storage with the maximum possible I/O performance for video processing, 300 TB of very durable storage for storing media content, and 900 TB of storage to meet requirements for archival media that is not in use anymore.

Which set of services should a solutions architect recommend to meet these requirements?

- A. Amazon EBS for maximum performance, Amazon S3 for durable data storage, and Amazon S3 Glacier for archival storage
- B. Amazon EBS for maximum performance, Amazon EFS for durable data storage, and Amazon S3 Glacier for archival storage
- C. Amazon EC2 instance store for maximum performance, Amazon EFS for durable data storage, and Amazon S3 for archival storage
- D. Amazon EC2 instance store for maximum performance, Amazon S3 for durable data storage, and Amazon S3 Glacier for archival storage

Soru:

Bir medya şirketi sistemlerini AWS Cloud'a taşıma olasılığını değerlendirmektedir. Şirketin video işleme için mümkün olan en yüksek GC performansına sahip en az 10 TB depolamaya medya içeriğini saklamak için çok yüksek dayanıklılığa sahip 300 TB depolamaya ve artık kullanılmayan arşiv medya gereksinimlerini karşılamak için 900 TB depolamaya ihtiyacı vardır.

Bu gereksinimleri karşılamak için bir solutions architect aşağıdaki servis setlerinden hangisini önermelidir

- A Maksimum performans için Amazon EBS dayanıklı veri depolama için Amazon S3 ve arşiv depolama için Amazon S3 Glacier
- B Maksimum performans için Amazon EBS dayanıklı veri depolama için Amazon EFS ve arşiv depolama için Amazon S3 Glacier
- C Maksimum performans için Amazon EC2 instance store dayanıklı veri depolama için Amazon EFS ve arşiv depolama için Amazon S3
- D Maksimum performans için Amazon EC2 instance store dayanıklı veri depolama için Amazon S3 ve arşiv depolama için Amazon S3 Glacier

Soru Analizi:

Sorudan çıkan **3 ayrı depolama ihtiyacı** vardır:

- ◆ **1. Video işleme için 10 TB – maksimum I/O performansı**
 - En önemli kriter: **en yüksek IOPS ve en düşük latency**
 - Dayanıklılık ikinci planda
 - Geçici, yoğun işlem verisi olabilir
- ◆ **2. Medya içeriği için 300 TB – çok yüksek dayanıklılık**

- Uzun süre saklanacak
 - Yüksek durability (11 9'lar)
 - Büyük ölçek
- ◆ **3. Arşiv medya için 900 TB – en düşük maliyet**
- Artık kullanılmıyor
 - Erişim süresi önemli değil
 - Uzun süreli saklama

İlgili AWS Storage Servislerini Hatırlayalım

◆ **Amazon EC2 Instance Store**

- **En yüksek I/O performansı**
- Fiziksel host'a bağlı
- **Geçici (ephemeral)** veri
- Video processing için ideal

◆ **Amazon EBS**

- Yüksek performans
- Ama instance store kadar hızlı değil
- Daha pahalı

◆ **Amazon S3**

- **Çok yüksek dayanıklılık (99 999999999%)**
- Büyük ölçekli veri
- Media content için ideal

◆ **Amazon S3 Glacier**

- **En ucuz uzun süreli arşiv**
- Saatler süren erişim
- Archive use case için ideal

◆ **Amazon EFS**

- Paylaşımı file system
- Video işleme için uygun değil

- S3 kadar dayanıklı değil

Seçenek Analizi:

D Instance Store + S3 + S3 Glacier

Doğru:

- Maksimum I/O → **EC2 Instance Store**
- Çok dayanıklı medya → **Amazon S3**
- Arşiv → **Amazon S3 Glacier**

- Tüm gereksinimler birebir karşılanır
- En performanslı + en düşük maliyetli çözüm

A EBS + S3 + S3 Glacier

Yanlış:

- Maksimum performans için **EBS yeterli değil**
- Instance Store varken EBS seçilmez

B EBS + EFS + S3 Glacier

Yanlış:

- EFS yüksek dayanıklılık için S3 yerine kullanılmış
- EFS 300 TB medya için pahalı ve gereksiz

C Instance Store + EFS + S3

Yanlış:

- Arşiv için **S3 Glacier yerine S3** kullanılmış
- Arşiv için maliyet çok yüksek olur

Sonuç

Bu kelimeleri görürsen otomatik eşleştir:

- **Maximum I/O performance** → Instance Store
- **Very durable large storage** → Amazon S3
- **Archive not in use** → S3 Glacier

Kısa Özeti

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

Video processing	EC2 Instance Store
------------------	--------------------

Media storage	Amazon S3
---------------	-----------

Archive	S3 Glacier
---------	------------

QUESTION 63

A company wants to run applications in containers in the AWS Cloud. These applications are stateless and can tolerate disruptions within the underlying infrastructure. The company needs a solution that minimizes cost and operational overhead.

What should a solutions architect do to meet these requirements?

- A. Use Spot Instances in an Amazon EC2 Auto Scaling group to run the application containers.
- B. Use Spot Instances in an Amazon Elastic Kubernetes Service (Amazon EKS) managed node group.
- C. Use On-Demand Instances in an Amazon EC2 Auto Scaling group to run the application containers.
- D. Use On-Demand Instances in an Amazon Elastic Kubernetes Service (Amazon EKS) managed node group.

Soru:

Bir şirket, **AWS Cloud** üzerinde **container** içinde çalışan uygulamalar çalıştırılmak istemektedir. Bu uygulamalar **stateless** yapıdadır ve **altyapıdaki kesintileri tolere edebilmektedir**. Şirket, **maliyeti** ve **operasyonel yükü** en aza indiren bir çözüme ihtiyaç duymaktadır.

Bu gereksinimleri karşılamak için bir **solutions architect** ne yapmalıdır?

- A. Uygulama container'larını çalıştırmak için bir **Amazon EC2 Auto Scaling group** içinde **Spot Instance**'lar kullanmak.
- B. **Amazon Elastic Kubernetes Service (Amazon EKS) managed node group** içinde **Spot Instance**'lar kullanmak.
- C. Uygulama container'larını çalıştırmak için bir **Amazon EC2 Auto Scaling group** içinde **On-Demand Instance**'lar kullanmak.

D. Amazon Elastic Kubernetes Service (Amazon EKS) managed node group içinde On-Demand Instance'lar kullanmak.

Soru Analizi:

Sorudan çıkan anahtar ifadeler:

- **Containers** → Container orchestration gerekiyor
- **Stateless applications**
- **Underlying infrastructure disruptions tolerated** → Spot uygun
- **Minimize cost**
- **Minimize operational overhead**

👉 Bu iki kelime çok kritik:

- **Cost** → Spot Instances
- **Operational overhead** → Managed service (EKS)

Seçeneklerde Geçen Servislerin Karşılaştırması

◆ Spot vs On-Demand

- **Spot Instances**
 - %70–90 daha ucuz
 - Kesintiye uğrayabilir
 - Stateless workload için ideal
- **On-Demand Instances**
 - Pahalı
 - Kesintisiz
 - Bu senaryoda gereksiz maliyet

◆ EC2 Auto Scaling vs Amazon EKS

- **EC2 Auto Scaling**
 - Container orchestration yok
 - ECS/EKS ek kurulum gereklidir
 - Daha fazla yönetim yükü
- **Amazon EKS managed node group**

- Control plane AWS tarafından yönetilir
- Node lifecycle otomatik
- **Daha az operasyonel yük**

Seçenek Analizi:

B EKS Managed Node Group + Spot Instances

✓ En iyi seçenek:

- Spot → **En düşük maliyet**
- EKS managed → **En düşük operasyonel yük**
- Stateless ve interruption-tolerant workload için ideal

A EC2 Auto Scaling + Spot Instances

✗ Uygun değil:

- Spot doğru seçim
- Ama **container yönetimi manuel**
- Operasyonel yük daha fazla

C EC2 Auto Scaling + On-Demand

✗ Yanlış:

- Yüksek maliyet
- Gereksiz

D EKS Managed Node Group + On-Demand

✗ Yarı doğru ama elenir:

- Operasyonel yük düşük
- Ama maliyet yüksek
- Spot varken tercih edilmez

⌚ Sonuç

Bu kelimeleri birlikte görürsen:

- **Stateless**
- **Can tolerate disruptions**

- Minimize cost
- Containers

👉 Spot + Managed service (EKS)

Kısa Özeti

Gereksinim	En Uygun Çözüm
Düşük maliyet	Spot Instances
Container yönetimi	Amazon EKS
Düşük operasyonel yük	Managed node groups

QUESTION 64

A company is running a multi-tier web application on premises. The web application is containerized and runs on a number of Linux hosts connected to a PostgreSQL database that contains user records. The operational overhead of maintaining the infrastructure and capacity planning is limiting the company's growth. A solutions architect must improve the application's infrastructure.

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

- A. Migrate the PostgreSQL database to Amazon Aurora.
- B. Migrate the web application to be hosted on Amazon EC2 instances.
- C. Set up an Amazon CloudFront distribution for the web application content.
- D. Set up Amazon ElastiCache between the web application and the PostgreSQL database.
- E. Migrate the web application to be hosted on AWS Fargate with Amazon Elastic Container Service (Amazon ECS).

Soru:

Bir şirket, **çok katmanlı (multi-tier)** bir web uygulamasını **şirket içinde (on-premises)** çalıştırmaktadır. Web uygulaması **container** yapısındadır ve **PostgreSQL** veritabanına bağlı birden fazla **Linux host** üzerinde çalışmaktadır. Altyapının bakımı ve **kapasite planlamasıyla ilgili operasyonel yük**, şirketin büyümeyi sınırlamaktadır. Bir **solutions architect**, uygulamanın altyapısını iyileştirmek zorundadır.

Bu hedefe ulaşmak için solutions architect aşağıdaki **eylem kombinasyonlarından hangilerini** gerçekleştirmelidir? (**İki tane seçiniz.**)

- A. PostgreSQL veritabanını Amazon Aurora'ya taşımak.
- B. Web uygulamasını Amazon EC2 instance'ları üzerinde çalışacak şekilde taşımak.
- C. Web uygulaması içeriği için bir Amazon CloudFront dağıtıımı kurmak.
- D. Web uygulaması ile PostgreSQL veritabanı arasına Amazon ElastiCache kurmak.
- E. Web uygulamasını Amazon Elastic Container Service (Amazon ECS) üzerinde AWS Fargate kullanarak çalışacak şekilde taşımak.

Soru Analizi:

Sorudan çıkan **kritik noktalar**:

- Uygulama:
 - Multi-tier
 - Containerized
 - Linux host'larda çalışıyor
 - PostgreSQL veritabanı kullanıyor
- Mevcut problem:
 - Operational overhead yüksek
 - Capacity planning şirketin büyümесini kısıtlıyor
- Amaç:
 - Altyapı yönetimini azaltmak
 - Daha ölçeklenebilir ve yönetilen (managed) servisler kullanmak

👉 Yani:

- Sunucu yönetimini azalt
- Otomatik ölçeklenen, managed servisleri tercih et

Doğru Yaklaşımın Özeti

Katman

Hedef

Application (container) **Serverless / managed container**

Database

Managed relational DB

Katman	Hedef
Amaç	Minimum operasyon, otomatik ölçekleme

Sonuç Analizi:

A PostgreSQL'i Amazon Aurora'ya taşımak

Doğru

- Aurora:
 - Fully managed
 - Otomatik backup
 - High availability
 - Kapasite planlama ihtiyacı yok
- Operasyonel yükü ciddi şekilde azaltır

Kesinlikle tercih edilmeli

E AWS Fargate + Amazon ECS

Doğru

- Fargate:
 - Server yönetimi yok
 - Otomatik ölçekleme
 - Container-native
- ECS:
 - AWS tarafından yönetilen orchestrator
- Operational overhead ve capacity planning problemini doğrudan çözer

Sorunun kalbi burada

B Web uygulamasını EC2 instance'larına taşımak

Yanlış

- EC2:
 - OS patching
 - Scaling

- Capacity planning gerektirir
- Mevcut sorunu çözmez, hatta tekrarlar

⌚ Amazon CloudFront kurmak

✗ İlgisiz

- CDN:
 - Content delivery
 - Performans artışı
- Operational overhead problemine çözüm değil

▣ Amazon ElastiCache eklemek

✗ Yanlış bağlam

- Performansı artırır
- Ama:
 - Yeni bir katman
 - Yeni yönetim yükü
- Sorunun ana hedefi performans değil

🎯 Sonuç

Bu kelimeleri görürsen otomatik eşleştir:

- Operational overhead
- Capacity planning
- Containerized application

👉 Fargate / Serverless Containers

👉 Managed Database (Aurora)

Kısa Özeti

Katman Doğru Servis

Containers ECS + Fargate

Database Amazon Aurora

Sonuç Minimum operasyon, maksimum ölçülebilirlik

QUESTION 65

An application runs on Amazon EC2 instances across multiple Availability Zones. The instances run in an Amazon EC2 Auto Scaling group behind an Application Load Balancer. The application performs best when the CPU utilization of the EC2 instances is at or near 40%.

What should a solutions architect do to maintain the desired performance across all instances in the group?

- A. Use a simple scaling policy to dynamically scale the Auto Scaling group.
- B. Use a target tracking policy to dynamically scale the Auto Scaling group.
- C. Use an AWS Lambda function to update the desired Auto Scaling group capacity.
- D. Use scheduled scaling actions to scale up and scale down the Auto Scaling group.

Soru:

Bir uygulama, birden fazla **Availability Zone**'a yayılmış **Amazon EC2** instance'ları üzerinde çalışmaktadır. Bu instance'lar, bir **Application Load Balancer** arkasında bulunan bir **Amazon EC2 Auto Scaling** grubunda yer almaktadır.

Uygulama, EC2 instance'larının **CPU kullanım oranı %40 civarında veya bu değere yakın** olduğunda en iyi performansı göstermektedir.

Bir **solutions architect**, Auto Scaling grubundaki tüm instance'larda istenen performansı korumak için ne yapmalıdır?

- A. Auto Scaling grubunu dinamik olarak ölçeklendirmek için **simple scaling policy** kullanmalıdır.
- B. Auto Scaling grubunu dinamik olarak ölçeklendirmek için **target tracking policy** kullanmalıdır.
- C. Auto Scaling grubunun istenen kapasitesini güncellemek için bir **AWS Lambda** fonksiyonu kullanmalıdır.
- D. Auto Scaling grubunu büyütmek ve küçültmek için **scheduled scaling actions** kullanmalıdır.

Soru Analizi:

- Uygulama **Auto Scaling Group** içinde çalışan EC2 instance'larda çalışıyor
- **Application Load Balancer** arkasında
- **En iyi performans koşulu:**
 - 👉 CPU utilization **%40 civarında**

- Amaç:
👉 Tüm instance'larda bu CPU seviyesini otomatik ve sürekli korumak

Yani soru şunu arıyor:

Belirli bir metrik için (CPU = %40) otomatik, sürekli ve dinamik ölçektekleme hangi yöntemle en doğru şekilde sağlanır?



Anahtar ifade	İpucu
<i>performs best when CPU is at or near 40%</i>	 Target value
<i>maintain desired performance</i>	 Sürekli otomatik ayarlama
<i>Auto Scaling group</i>	 Dinamik scaling

Bu ifadeler doğrudan **Target Tracking Scaling Policy**'yi işaret eder.

Sonuç Analizi:



Nedir?

- Belirli bir metriği **hedefdeğerde** tutar
 - Örn:
 - 👉 *Average CPU Utilization = 40%*
 - AWS, instance sayısını **otomatik ve sürekli** ayarlar

Neden doğru?

- Soru açıkça “**CPU %40 civarında**” diyor
 - Tam olarak bu senaryo için tasarlanmıştır
 - Ek CloudWatch alarmı yazmaya gerek yok



“Near X value” → **Target Tracking**



Nedir?

- CloudWatch alarmına bağlıdır
 - “CPU > %70 → 2 instance ekle” gibi **statik kurallar**

- Cooldown süreleri vardır

Neden yanlış?

- CPU'yu **%40 civarında tutmaz**
- Sadece eşik aşılınca tepki verir
- Sürekli ince ayar yapamaz

📌 Sınav yorumu:

“Simple scaling = kaba ve eski yöntem”

✗ C. AWS Lambda ile kapasite güncelleme

Nedir?

- Özel (custom) bir çözüm
- Manuel kod + izleme gereklidir

Neden yanlış?

- Gereksiz karmaşıklık
- AWS'nin hazır sunduğu scaling özelliği varken tercih edilmez
- **Best practice değil**

📌 Sınav yorumu:

“Lambda ile scaling” → %90 yanlış

✗ D. Scheduled scaling actions

Nedir?

- Belirli saatlerde ölçektekleme
(örn: 09:00 scale out, 18:00 scale in)

Neden yanlış?

- CPU yükü **zamana bağlı değil**
- Gerçek zamanlı performansı koruyamaz
- Ani trafik değişimlerine tepki veremez

📌 Sınav yorumu:

“Scheduled scaling” → **predictable workload** gereklidir

⌚ Sonuç

Belirli bir metrik + belirli bir hedef değer

→ Target Tracking Scaling Policy

Target Tracking

Anahtar kelimeler:

- “keep CPU at X%”
- “maintain performance”
- “near a specific value”
- “automatically adjust”

Sınav refleksi:

Bu kelimeleri gördün mü → hiç düşünme, Target Tracking

Step Scaling

Anahtar kelimeler:

- “if CPU is between X and Y”
- “scale more aggressively”
- “different scaling steps”

Kullanım:

Trafik arttıkça **daha büyük adımlarla** ölçekleme

Simple Scaling

Anahtar kelimeler:

- “basic”
- “single alarm”
- “legacy”

Sınav yorumu:

Neredeyse her zaman **yanlış seçenek**

Scheduled Scaling

Anahtar kelimeler:

- “every day at 9 AM”
- “known traffic pattern”

- “business hours”

📌 **Kullanım:**

Trafik tahmin edilebiliyorsa

🧠 **Exam Trick – 5 Saniyelik Karar Rehberi**

Soruda geçen ifade Cevap

“at or near X%” Target Tracking

“maintain metric” Target Tracking

“predictable traffic” Scheduled

“multiple thresholds” Step Scaling

“use Lambda to scale” ✗ Genelde yanlış
