1.

**a. git clone:** bu komut ile ile uzakta bulunan bir Git deposunu kendi bilgisayarımıza indiriyoruz. Uzak depoyu klonlandıktan sonraki tüm değişiklikler artık bilgisayarımızdaki klon Git deposunda kayıt altına alınacak, ta ki git push komutu ile değişikliklerimizi uzaktaki depoya geri yazana kadar.

**b. pull :** git pull komutu ile Remote Repo’da yapılan değişiklikleri alıp localdeki repomuzu güncelleriz. Bunu ders dosyalarını güncel takip edebilmek için sürekli yapıyoruz zaten.

**c.fetch :** Sorumuzun cevabı da bu. Remote branch'deki değişiklikleri indirmek için **git fetch** komutunu kullanıyoruz. Git pull’da bunu yapıyor.Buradaki fark şu; remote repoda localinizde bulunmayan tüm değişiklikleri alır ve **bunları yerel repoda saklar**. Ancak, **onları geçerli sizin dosyalarınızla birleştirmez**. Bu özellikle reponuzu güncel tutmanız gerekiyorsa, ancak mevcut dosyalarınız güncellerseniz kaybı sorun olabilecek bir şey üzerinde çalışıyorsanız kullanışlıdır. Git pull komutu ise hem değişiklikleri uzak repodan alır, hem de sizin reponuzdaki dosyalarla onları birleştirir. Soruda “it doesn't integrate any of this new data into your working files” dediği için fetch olacak cevabı.

**d. merge:** Remote branch'deki değişiklikleri local branch'inize entegre etmek için **git merge komutunu kullanıyoruz.**

**e. push:** git push komutu ile uzaktaki repo üzerine kayıt yapabiliriz.

2. AWS, sunucuları yönetmeden kod çalıştırma, verileri yönetme ve uygulamaları entegre etme teknolojileri sunar. Sunucusuz teknolojiler, çevikliği artırmak ve maliyetleri optimize etmek için otomatik ölçeklendirme, yerleşik yüksek erişilebilirlik ve kullanıma göre ödeme faturalandırma modeli içerir. Bu teknolojiler ayrıca kapasite tedariki ve düzeltme eki uygulama gibi altyapı yönetimi görevlerini de ortadan kaldırır, böylece müşterilerinize hizmet sunan kod yazmaya odaklanabilirsiniz. Sunucusuz uygulamalar, 200'den fazla AWS hizmeti ve hizmet olarak yazılım (SaaS) uygulamasıyla yerel olarak entegre edilmiş, olaya dayalı bir işlem hizmeti olan AWS Lambda ile başlar.

AWS Lambda

AWS Fargate

AWS SQS

AWS SNS

AWS API Gateway

S3

DynamoDB

Amazon Aurora

3. What is the service provided by AWS that allows developers to easily deploy and manage applications on the cloud? (Developerlara aplikasyonlarını cloud ortamında kullanma ve yönetme imkanı sağlayan AWS servisi hangisidir?

**b. Elastic Beanstalk:** AWS Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, . NET, PHP, Node. js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.

4. Using API Gateway, you can create SOAP APIs

**What are REST and SOAP?**

REST and SOAP are both web-communication protocols. Only the front-end of applications is stored on the user’s device, while the related data resides on remote servers. REST and SOAP handle how to fetch that data.

**SOAP** *(Simple Object Access Protocol)* is a standardized protocol with pre-defined rules.

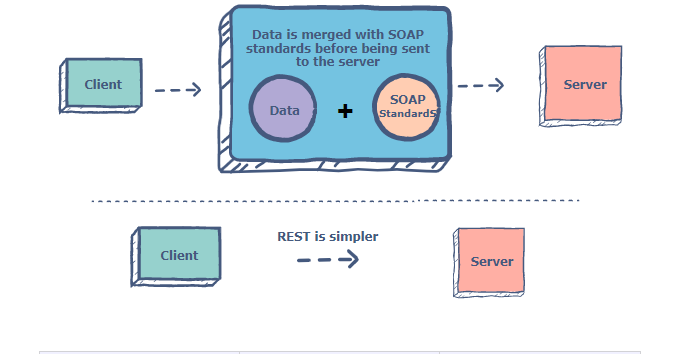
**REST** *(Representational State Transfer)* is an architectural style with loose guidelines, and either one can be used to develop data-retrieval APIs for a system.

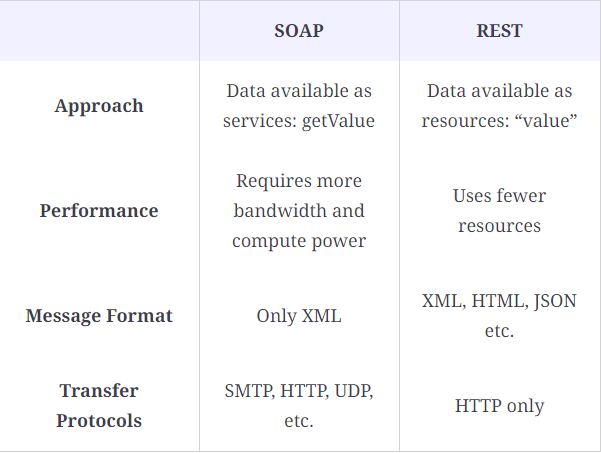
**How are they different functionally?**

SOAP has been a really popular messaging protocol for a long time. It comes with strict rules which make features such as the ACID (Atomicity, Consistency, Isolation, Durability) properties and authorization possible. However, SOAP generally requires a lot of bandwidth and resources, which results in slower operations.

REST was developed to address these problems, and in turn has a more flexible architecture and loose guidelines. As a result, it’s lightweight and better performing.

This has made REST incredibly popular in today’s age, where even a few seconds make a huge difference.





5. A company requires to deploy an existing Java-based application to AWS. Which of the following should be used to fulfill this requirement in the quickest way possible? (Bir şirketin Java temelli bir uygulamasını AWS’de kullanabilmesi için takip edebileceği en hızlı yol nedir?)

B. Use the Elastic Beanstalk service to provision the environment.

Elastic Beanstalk, uygulamanızı AWS üzerinde dağıtmanın en hızlı ve en basit yoludur. Elastic Beanstalk uygulamanızın çalışabilmesi için gerekli kaynakları oluşturur (environment’ı hazırlar). AWS Elastic Beanstalk; Apache, Nginx, Passenger ve IIS gibi bilindik sunucular üzerinde Java, .NET, PHP, Node.js, Python, Ruby, Go ve Docker ile geliştirilmiş web uygulamalarını ve hizmetleri dağıtıp ölçeklendirmek için kullanımı kolay bir hizmettir.

Tek yapmanız gereken kodunuzu yüklemektir; kapasite tedariği, yük dengeleme ve otomatik ölçeklendirmeden uygulama durumunu izlemeye kadar dağıtımın her aşaması Elastic Beanstalk tarafından otomatik olarak gerçekleştirilir. Öte yandan, uygulamanızı destekleyen AWS kaynakları üzerindeki denetim tamamen sizde kalır ve temel kaynaklara dilediğiniz zaman erişebilirsiniz.

Elastic Beanstalk için ek ücret uygulanmaz. Yalnızca uygulamalarınızı depolamak ve çalıştırmak için gerekli AWS kaynaklarına ödeme yaparsınız.

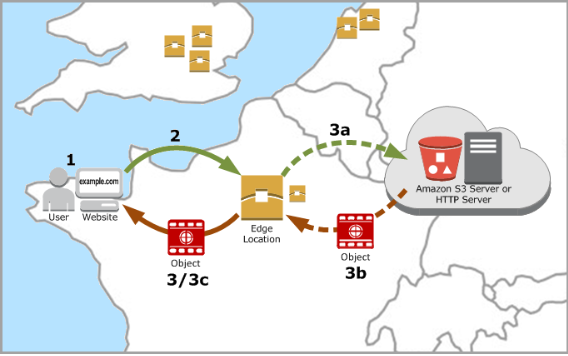
1. You are planning on deploying a video based application onto the AWS Cloud. These videos will be accessed by users across the world. Which of the below services can help stream the content in an efficient manner to the users across the globe? (Video temelli bir uygulamayı AWS Cloud’a yüklemeyi planlıyorsunuz. Bu videolar dünyadaki herkes tarafından ulaşılabilir olacak. Aşağıdaki servislerin hangisi bu içeriğin dünyanın heryerindeki kullanıcılara sağlıklı bir şekilde iletilmesi için en etkili yoldur?)

**C. Amazon CloudFront :** Amazon CloudFront; yüksek performans, güvenlik ve geliştiriciler için kolaylık sağlamaya yönelik olarak oluşturulan bir içerik teslim ağı (CDN) hizmetidir. Amazon Cloudfront verileri, videoları, uygulamaları ve API'leri küresel olarak müşterilere düşük gecikme süresi ve yüksek aktarım hızlarıyla güvenli bir şekilde sunan hızlı bir içerik dağıtım ağı (CDN) hizmetidir.

Cloudfront; Amazon S3, Elastic Load Balancing, Amazon EC2 veya herhangi bir özel HTTP kaynağıyla sorunsuz bir şekilde çalışır.

2. Which of the following components of the Cloudfront service can be used to distribute contents to users across the globe? (Aşağıdaki Cloudfront bileşenlerden hangisi bir içeriğin tüm dünyaya dağıtılması için kullanılır?)

**D. Amazon Edge Locations :** Cloudfront sizin içeriğinizi tüm dünyadaki kullanıcılara edge location adı verilen data center’lar sayesinde ulaştırır. Dünya genelinde 265+ Edge Locations var.



3. A professional educational institution maintains a dedicated web server and database cluster that hosts an exam results portal for modules undertaken by its students. The resource is idle for most of the learning cycle and becomes excessively busy when exam results are released. How can this architecture be improved to be cost-efficient? (Profesyonel bir eğitim firması kendi öğrencilerinin sınac sonuçlarını içeren bir database ve web sayfasını yönetiyor. Bu sayfa ve database eğitim döneminin çoğunda hemen hemen hiç kullanılmazken sınav sonuçlarının açıklandığı dönemde aşırı bir yoğunluk yaşıyor. Bu yapıyı finansal olarak en etkin kullanmanın yolu aşağıdakilerden hangisidir?

C. Configure serverless architecture leveraging AWS Lambda functions

AWS Lambda kullanarak hazırladığınız bir kodun sadece gerekli olduğu zamanlarda çalışmasını/trigger sağlayabilirsiniz. Kodunuzu bir EC2 instance ayağa kaldırarak da çalıştırabilirsiniz. Ancak bu durumda EC2 instance’ınız çalıştığı sürece ücretlendirilirsiniz. Oysa aynı işi bir lambda fonksiyonu oluşturarak yaparsanız lambdanın “trigger function” özelliği sayesinde sadece bu kodun çalıştırıldığı/execute zamanlarda ücretlendirilirsiniz. Yani programınızın atıl kaldığı eğitim döneminin çoğunluğu boyunca ücretlendirilmezsiniz.

4. Which of the following is the customer’s responsibility with respect to the AWS Lambda service? (choose 2 options) (Aşağıdakilerden hangi ikisi AWS Lambda servis için kulacının sorumluluğundadır)

A. Lambda function code.

B. Monitoring and logging lambda functions.

C. Security patches.

D. Installing required libraries in underlying compute instances for Lambda execution.

E. Providing access to AWS resources that triggers a Lambda function.

a ve c herhalde, belki a ve b de olabilir. Lambda için biz bir kod yazıyoruz, dolayısıyla a bizim sorumluluğumuzda. B ve c için cevap anahtarına bakılabilir.

5. You have built a REST API using API gateway and distributed to your customers. However, your API is receiving large number of requests and overloading your backend system causing performance bottlenecks and eventually causing delays and failures in serving the requests for your important customers. How would you improve the API performance? (Choose 2 options) API gateway kullanarak bir REST API oluşturdunuz ve müşterilerinize ulaştırdınız. Ama API’nıza çok fazla istek geldiği zaman sisteme aşırı bir yüklenme gerçekleşiyor ve bu da önemli müşterilerinizin taleplerini icrada gecikmelere ve bazen başarısızlıklara neden oluyor. API performansını geliştirmek için aşağıdakilerden hangi ikisi yapılabilir?

A. Enable throttling and control the number of requests per second. ( throttling/daraltma seçeneğini aktif et ve saniye başına düşen istek miktarını kontrol altına al)

B. Create a resource policy to allow access for specic customers during specic time period. (Belirli müşterilerin belirle zamanlarda kaynaklara ulaşabilmeleri için bir policy oluştur)

C. Enable API caching to serve frequently requested data from API cache. (Sıklıkla istek yapılan datalar için API caching aktif et)

D. Enable load balancer on your backend systems. (backend sistemin için load balancer aktif et)

 A developer using your API on a local application may accidentally execute a loop, causing a sudden burst of requests that slow down your performance. It could even be deployed into production! The best way to avoid these problems is to implement a rate-limiting strategy. By measuring the number of transactions per second, per IP address, or token (if each client is authorized before accessing the API), you can cut off API clients that make excessive requests and prevent DDoS-like slowdowns from accidentals.

**Caching** is one of the best ways to improve API performance. If you have requests that frequently produce the same response, a cached version of the response avoids excessive database queries. The easiest way to cache responses is to periodically expire it, or force it to expire when certain data updates happen.