**Your company's on-premises content management system has the following architecture:**

- **Application Tier – Java code on a JBoss application server**
- **Database Tier – Oracle database regularly backed up to Amazon Simple Storage Service (S3) using the Oracle RMAN backup utility**
- **Static Content – stored on a 512GB gateway stored Storage Gateway volume attached to the application server via the iSCSI interface**

**Which AWS based disaster recovery strategy will give you the best RTO?**

A) Deploy the Oracle database and the JBoss app server on EC2. Restore the RMAN Oracle backups from Amazon S3. Generate an EBS volume of static content from the Storage Gateway and attach it to the JBoss EC2 server.
B) Deploy the Oracle database on RDS. Deploy the JBoss app server on EC2. Restore the RMAN Oracle backups from Amazon Glacier. Generate an EBS volume of static content from the Storage Gateway and attach it to the JBoss EC2 server.
C) Deploy the Oracle database and the JBoss app server on EC2. Restore the RMAN Oracle backups from Amazon S3. Restore the static content by attaching an AWS Storage Gateway running on Amazon EC2 as an iSCSI volume to the JBoss EC2 server.
D) Deploy the Oracle database and the JBoss app server on EC2. Restore the RMAN Oracle backups from Amazon S3. Restore the static content from an AWS Storage Gateway-VTL running on Amazon EC2

**An ERP application is deployed in multiple Availability Zones in a single region. In the event of failure, the RTO must be less than 3 hours, and the RPO is 15 minutes. The customer realizes that data corruption occurred roughly 1.5 hours ago. Which DR strategy can be used to achieve this RTO and RPO in the event of this kind of failure?**

A) Take 15-minute DB backups stored in Amazon Glacier, with transaction logs stored in Amazon S3 every 5 minutes.
B) Use synchronous database master-slave replication between two Availability Zones.
C) Take hourly DB backups to Amazon S3, with transaction logs stored in S3 every 5 minutes.
D) Take hourly DB backups to an Amazon EC2 instance store volume, with transaction logs stored in Amazon S3 every 5 minutes.

**The Marketing Director in your company asked you to create a mobile app that lets users post sightings of good deeds known as random acts of kindness in 80-character summaries. You decided to write the application in JavaScript so that it would run on the broadest range of phones, browsers, and tablets. Your application should provide access to Amazon DynamoDB to store the good deed summaries. Initial testing of a prototype shows that there aren't large spikes in usage. Which option provides the most cost-effective and scalable architecture for this application?**

A) Provide the JavaScript client with temporary credentials from the Security Token Service using a Token Vending Machine (TVM) on an EC2 instance to provide signed credentials mapped to an Amazon Identity and Access Management (IAM) user allowing DynamoDB puts and S3 gets. You serve your mobile application out of an S3 bucket enabled as a web site. Your client updates DynamoDB.
B) Register the application with a Web Identity Provider like Amazon, Google, or Facebook, create an IAM role for that provider, and set up permissions for the IAM role to allow S3 gets and DynamoDB puts. You serve your mobile application out of an S3 bucket enabled as a web site. Your client updates DynamoDB.
C) Provide the JavaScript client with temporary credentials from the Security Token Service using a Token Vending Machine (TVM) to provide signed credentials mapped to an IAM user allowing DynamoDB puts. You serve your mobile application out of Apache EC2 instances that are load-balanced and autoscaled. Your EC2 instances are configured with an IAM role that allows DynamoDB puts. Your server updates DynamoDB.

D) Register the JavaScript application with a Web Identity Provider like Amazon, Google, or Facebook, create an IAM role for that provider, and set up permissions for the IAM role to allow DynamoDB puts. You serve your mobile application out of Apache EC2 instances that are load-balanced and autoscaled. Your EC2 instances are configured with an IAM role that allows DynamoDB puts. Your server updates DynamoDB.

**You are building a website that will retrieve and display highly sensitive information to users. The amount of traffic the site will receive is known and not expected to fluctuate. The site will leverage SSL to protect the communication between the clients and the web servers. Due to the nature of the site you are very concerned about the security of your SSL private key and want to ensure that the key cannot be accidentally or intentionally moved outside your environment. Additionally, while the data the site will display is stored on an encrypted EBS volume, you are also concerned that the web servers' logs might contain some sensitive information; therefore, the logs must be stored so that they can only be decrypted by employees of your company. Which of these architectures meets all of the requirements?**

A) Use Elastic Load Balancing to distribute traffic to a set of web servers. To protect the SSL private key, upload the key to the load balancer and configure the load balancer to offload the SSL traffic. Write your web server logs to an ephemeral volume that has been encrypted using a randomly generated AES key.
B) Use Elastic Load Balancing to distribute traffic to a set of web servers. Use TCP load balancing on the load balancer and configure your web servers to retrieve the private key from a private Amazon S3 bucket on boot. Write your web server logs to a private Amazon S3 bucket using Amazon S3 server-side encryption.
C) Use Elastic Load Balancing to distribute traffic to a set of web servers, configure the load balancer to perform TCP load balancing, use an AWS CloudHSM to perform the SSL transactions, and write your web server logs to a private Amazon S3 bucket using Amazon S3 server-side encryption.
D) Use Elastic Load Balancing to distribute traffic to a set of web servers. Configure the load balancer to perform TCP load balancing, use an AWS CloudHSM to perform the SSL transactions, and write your web server logs to an ephemeral volume that has been encrypted using a randomly generated AES key.

**You are designing network connectivity for your fat client application. The application is designed for business travelers who must be able to connect to it from their hotel rooms, cafes, public Wi-Fi hotspots, and elsewhere on the Internet. You do not want to publish the application on the Internet.**

**Which network design meets the above requirements while minimizing deployment and operational costs?**

A) Implement AWS Direct Connect, and create a private interface to your VPC. Create a public subnet and place your application servers in it.
B) Implement Elastic Load Balancing with an SSL listener that terminates the back-end connection to the application.
C) Configure an IPsec VPN connection, and provide the users with the configuration details. Create a public subnet in your VPC, and place your application servers in it.
D) Configure an SSL VPN solution in a public subnet of your VPC, then install and configure SSL VPN client software on all user computers. Create a private subnet in your VPC and place your application servers in it.

**Your company hosts an on-premises legacy engineering application with 900GB of data shared via a central file server. The engineering data consists of thousands of individual files ranging in size from megabytes to multiple gigabytes. Engineers typically modify 5-10 percent of the files a day. Your CTO would like to migrate this application to AWS, but only if the application can be migrated over the weekend to minimize user downtime. You calculate that it will take a minimum of 48 hours to transfer 900GB of data using your company's existing 45-Mbps Internet connection.**

**After replicating the application's environment in AWS, which option will allow you to move the application's data to AWS without losing any data and within the given timeframe?**

A) Copy the data to Amazon S3 using multiple threads and multi-part upload for large files over the weekend, and work in parallel with your developers to reconfigure the replicated application environment to leverage Amazon S3 to serve the engineering files.
B) Sync the application data to Amazon S3 starting a week before the migration, on Friday morning perform a final sync, and copy the entire data set to your AWS file server after the sync completes.
C) Copy the application data to a 1-TB USB drive on Friday and immediately send overnight, with Saturday delivery, the USB drive to AWS Import/Export to be imported as an EBS volume, mount the resulting EBS volume to your AWS file server on Sunday.
D) Leverage the AWS Storage Gateway to create a Gateway-Stored volume. On Friday copy the application data to the Storage Gateway volume. After the data has been copied, perform a snapshot of the volume and restore the volume as an EBS volume to be attached to your AWS file server on Sunday.