AWS Case Study

Problem Statement:

A firm wants to launch their new application. To reduce cost it has been decided that the entire application will be hosted on AWS. It's a photo and video storage application. This application will store movies and videos, and customers can also upload photos and their custom made videos using this application.

To buy videos, movies and wallpapers stored in the application customers have to pay some amount using their credit card. Customer will have to provide their home address and some other personal information as well.

Following are some of the requirements:

1. The most important requirement is that the application should be very much secure and none of the servers (front-end UI, back-end application, database) should be available over Internet.

2. Application should handle sudden spike in customer activity without any problem.

3. Customer information should be kept secure as much as possible.

4. It has been decided that 2 types of NoSQL databases would be used to store metadata information about videos and photos. 3 Relational databases would be used for transactional purposes.

5. Performance is vital for the success of the product. Make sure that users get low latent responses wherever possible.

6. Photos and videos should be stored securely at rest.

7. Access to AWS resources should be authorized as per business need, e.g. application developers should not have access to database servers. DBAs should not have any access to application servers.

8. Whenever customer uploads a photo/video he or she should get an email notification.

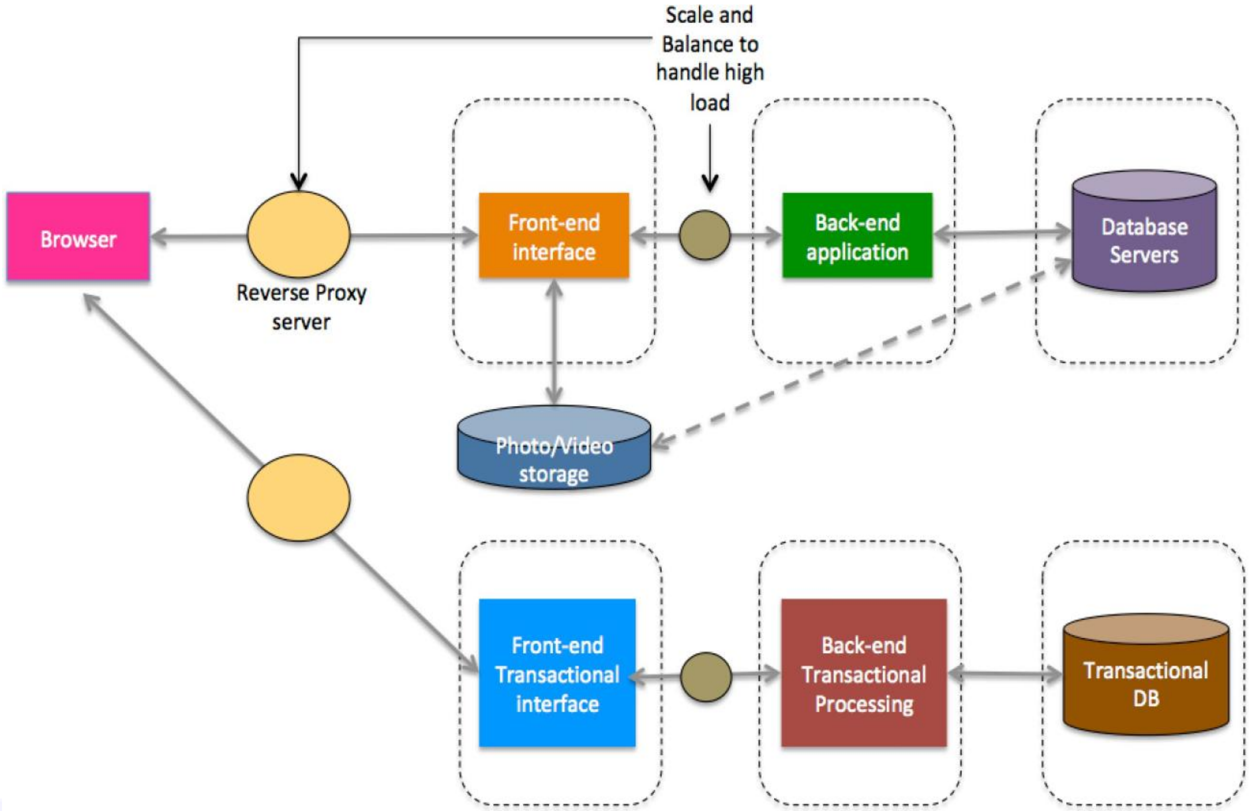
9. Photos/videos older than 2 years should be archived to save cost.

10. Application should be highly available. It should not go down in the event of any data center failure. Disaster Recovery has to be set up to mitigate regional failures as well.

11. Transactions should be as secure as possible.

12. All the SysOps out there configure and use AWS CLI to provision all services.

As AWS administrators and architects you will have to make sure everything in the requirement is taken care of. Make sure to have provision to apply patches on the servers periodically. Also, make sure to have future provisioning for technology refresh. Following is the high-level architecture diagram suggested by the senior solutions architect. Set up the entire AWS environment based on this. Factor in DR setup as well



Lab Environment:

Set up AWS environment with the following components. Remember, you DO NOT have to create the application, just set up the architecture using AWS services and make sure you understand which services to use where. This is a typical media services domain application, but with much less complexities.

1. AWS Account
2. VPC components
3. EC2 and EBS
4. Load Balancer, Auto Scaling groups and scaling policies
5. S3
6. DynamoDB
7. RDS DB Instance
8. Route 53
9. SNS
10. Elastic Beanstalk (Optional, if you choose PaaS for your front end and back end applications. Better to use Elastic Beanstalk for one of the tiers)
11. IAM Users, Roles and Groups
12. Glacier
13. ElastiCache Cluster

Hint:

Architecture has already been given; you have to first map the architecture with AWS services. Check the requirements multiple times; you have to implement all the provided requirements in AWS. Some services have to be pulled in and placed in the architecture to fulfill all the requirements. Lab environment already mentions all the required services, try to use those in the architecture and create the application.

You DO NOT have to write any code or create the application as such. It will need a lot of coding and development knowledge. The purpose of this project is to make sure that you know how to implement AWS services to create an end-to-end application. Draw the diagram first using AWS services and then provision the services as needed. You might incur some AWS charges, but those will be minimal.

MAKE SURE TO TERMINATE ALL THE SERVICES ONCE YOU ARE DONE WITH THE EXERCISE.