**Designing a Cloud Solution: A Medical Company Startup**

**AWS Cloud Architecting**

Spring 2020

**Creation Data**: 26th June 2020

**Submitted By:**

Sharad Sawhney

According to data sources from 2015, there were about 18 cases of security breach in the health care and medical provider industry. If we talk about the number of people whose data was compromised, it will be heart rendering. For citing purposes, the affected people were – 10.5 million!!

Today, we are witnessing a change in the healthcare industry. We are moving from paper-based records and data to electronic health records (EHR). This means healthcare data is being produced at an exponentially increasing rate. It is being forecasted that the size of the data will be about 40 zettabytes by end of 2021. To effectively manage this data, it is imperative that the community manages the data cost effectively and securely for patient care, research reasons, legal reasons, etc. This calls for a pressure on medical companies to move towards adopting a safe, analytical environment. The incoming data can either be structured or unstructured. Thus, it becomes imperative to use a cloud solution to securely store, process and manage the data. Thus, the medical company has handed us the task to build an online social networking and diagnosis assistance application.

As cloud architects, we are expected to:

1. Design an infrastructure architecture in Amazon Web Services (AWS) to meet application needs of Medical Company

2. Connect doctor and users/patients

**Executive Summary**

Given to us that the Medical Company is a startup software as a service (SaaS) company which has built an online medical social networking and diagnosis assistance applicationfor users in APAC, the US, and Europe. The application intends to allow online appointments, remote consultation, remote diagnosis, electronic prescription transfer, and payment services. Additionally, allow customers to upload documents and images. Text would be extracted from documents, and images converted into multiple formats. The application has **not yet been launched publicly**.

Presently, it uses a third-party server to host the medical company for deploying its current development and test infrastructure. It is based on Microsoft Windows server to host its web application and it is linked to Microsoft SQL server. It is expected from the cloud architects who are deploying the cloud solution to be built for business continuity. If one of the servers is unavailable, another one replaces the former. Next, we address the above tasks by:

* Amazon EC2 in Auto Scaling groups for its web application.
* Amazon RDS with MySQL for storing data.
* Amazon S3 to store backups and the storing the uploaded file
* Amazon ElastiCache to improve performance of their web applications.
* Elastic Load Balancing to distribute traffic across applications.
* Amazon SNS for internal applications
* Amazon Cloud Watch for monitoring the system and maintaining the logs and notify other services to handle the issues arise in the system.

To conclude, build highly available platform on AWS which assists:

* Making Online Appointments
* Remote Consultation & Diagnosis
* E- Prescription
* Payment Service

The services we intend to use in our architecture are:



**Requirements & Assumptions Throughout the Model**

The cloud architects intend on constructing a well architected framework which fulfills the following requirements:

* High Availability
* Scalability
* Backup
* Security
* Upload supporting documents
* Multi location support (Asia Pacific, Europe and USA)
* Monitor

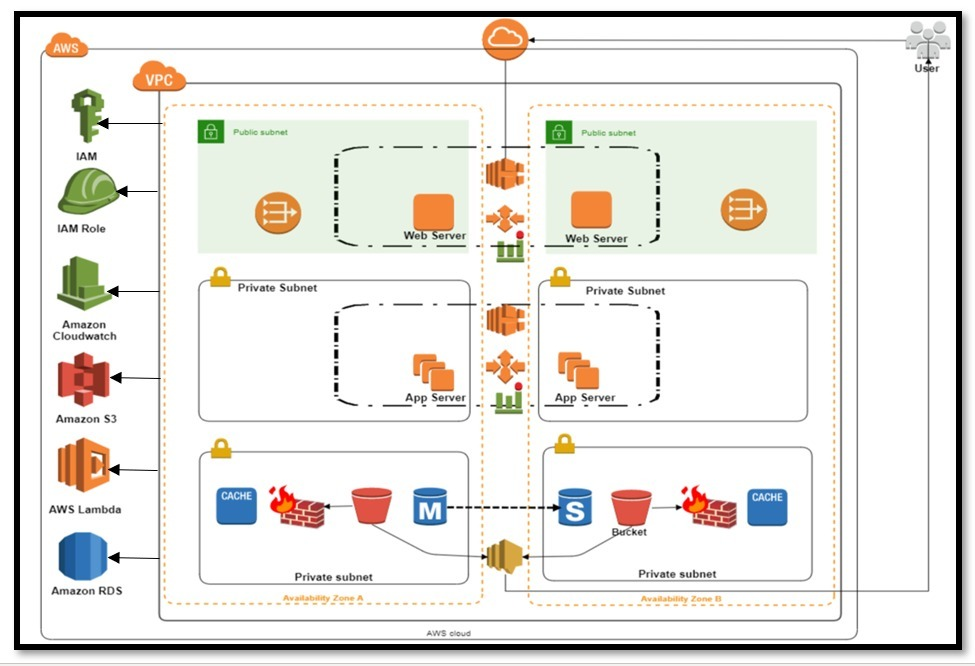
All these requirements above have been discussed below in detail.

Additionally, we want an architecture which aligns with AWS best practices, builds networks which are in sync with these practices and matches the present architecture of the medical company.

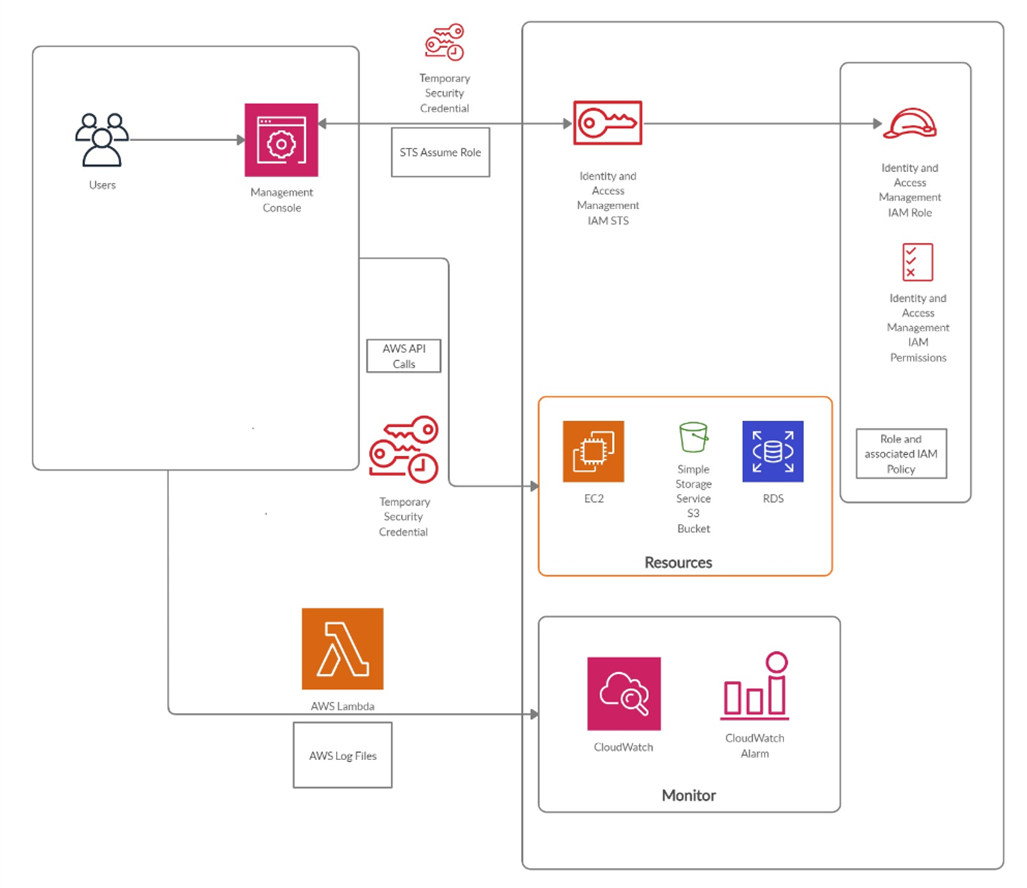
Let’s talk about assumptions-

* Cost Optimization is done considering only AWS Ecosystem
* No third party is involved in our AWS environment
* The theoretical architecture being proposed was validated in a real AWS environment

**Final Architecture Diagram**



Architecture has been constructed based on 5 pillars. This is the final solution to be presented to the Medical Company. The user interacts with the application via VPC NAT Gateway. Elastic Load Balance handles all the incoming traffic. The application has been split into two availability zones such that when we focus on two regions, all three are covered. SNS service assists in messaging. S3 Bucket stores the data. ElastiCache aids high availability performance of architecture. There are two private subnets and one public subnet. Our database is in private subnet. Web server is in public subnet and application server is in the private subnet. Let’s discuss the pillars in detail below:

1. **Network & Security**  
   

The image above shows how we have approached security as a pillar in our architecture. The user interacts with the application. The management console provides an inbuilt user interface to perform AWS tasks like working with Amazon S3 buckets, launching and connecting to Amazon EC2 instances, setting Amazon CloudWatch alarms etc. If the user is using temporary credentials, the query is raised and sent via the STS Assume role using temporary security credential in AWS. AWS API calls for your AWS account and delivers log files to an Amazon S3 bucket. The recorded information includes the identity of the user, the start time of the AWS API call, the source IP address, the request parameters, and the response elements returned by the service. EC2, S3 bucket will be used as a gateway to store data of users. IAM enables security best practices by allowing you to grant unique security credentials to users and groups to specify which AWS service APIs and resources they can access. IAM is secure by default; users have no access to AWS resources until permissions are explicitly granted. AWS API is used to extract files and store them in the database. RDS is used as a database in our case. CloudWatch aids in deriving meaningful insights from the files in our database. CloudWatch alarm sends out notification when a goal has been achieved.

1. **Scalability, HA & Business Continuity**

There are 2 availability zone i.e. us-east-1 and ap-east-1 such that users located in APAC, US and Europe can access it. Users will access application via internet gateway that is used for communication. Elastic Load Balancing automatically distributes incoming traffic across multiple Availability Zones and ensures only healthy targets receive traffic. The 1st ELB distributes incoming traffic to health target of web apps which then passes the traffic to 2nd ELB that again distributes traffic to healthy targets of backend apps. The traffic is then passed to Elasticache and RDS. Elasticache builds data-intensive apps or boost the performance of the existing databases by retrieving data from high throughput and low latency in-memory data stores. RDS database is used to store data. Master and standby RDS are used to have backup in case anyone RDS fails which show reliability. The data is then stored in Amazon S3.

A picture containing map, text

Description automatically generated  
Figure: **Reliability and an online medical social networking and diagnosis assistance application for users in APAC, the US, and Europe**

Amazon S3 is used to store the data of the medical company. The data is then transferred to Lambda for processing in High availability environment which is then transferred to AWS Batch.

AWS Batch allows to build efficient, long-running compute jobs by focusing on the business logic required, while AWS manages the scheduling and provisioning of the work. Running multiple jobs is little challenging so we are using AWS Batch in this case.

Internal working of AWS Batch is shown below:

1) Jobs: The unit of work submitted to AWS Batch, whether it is implemented as a online appointments, remote consultation, remote diagnosis, electronic prescription transfer, and payment services

2) Job Definition: Describes how work is executed as online appointments, remote consultation, remote diagnosis, electronic prescription transfer, and payment services

3) Job Queues: All submitted jobs are listed in the job queues.

4) Compute Environment: The compute resources that run your Jobs. AWS Auto Scaling monitors the applications and automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost. Using AWS Auto Scaling, it’s easy to setup application scaling for multiple resources across multiple services in minutes.

5) The AWS Batch scheduler evaluates when, where, and how to run jobs that have been submitted to a job queue. Jobs run in approximately the order in which they are submitted if all dependencies on other jobs have been met.

Amazon Comprehend is a natural language processing (NLP) service that uses machine learning to find insights and relationships in text. The executed jobs are then stored in S3.

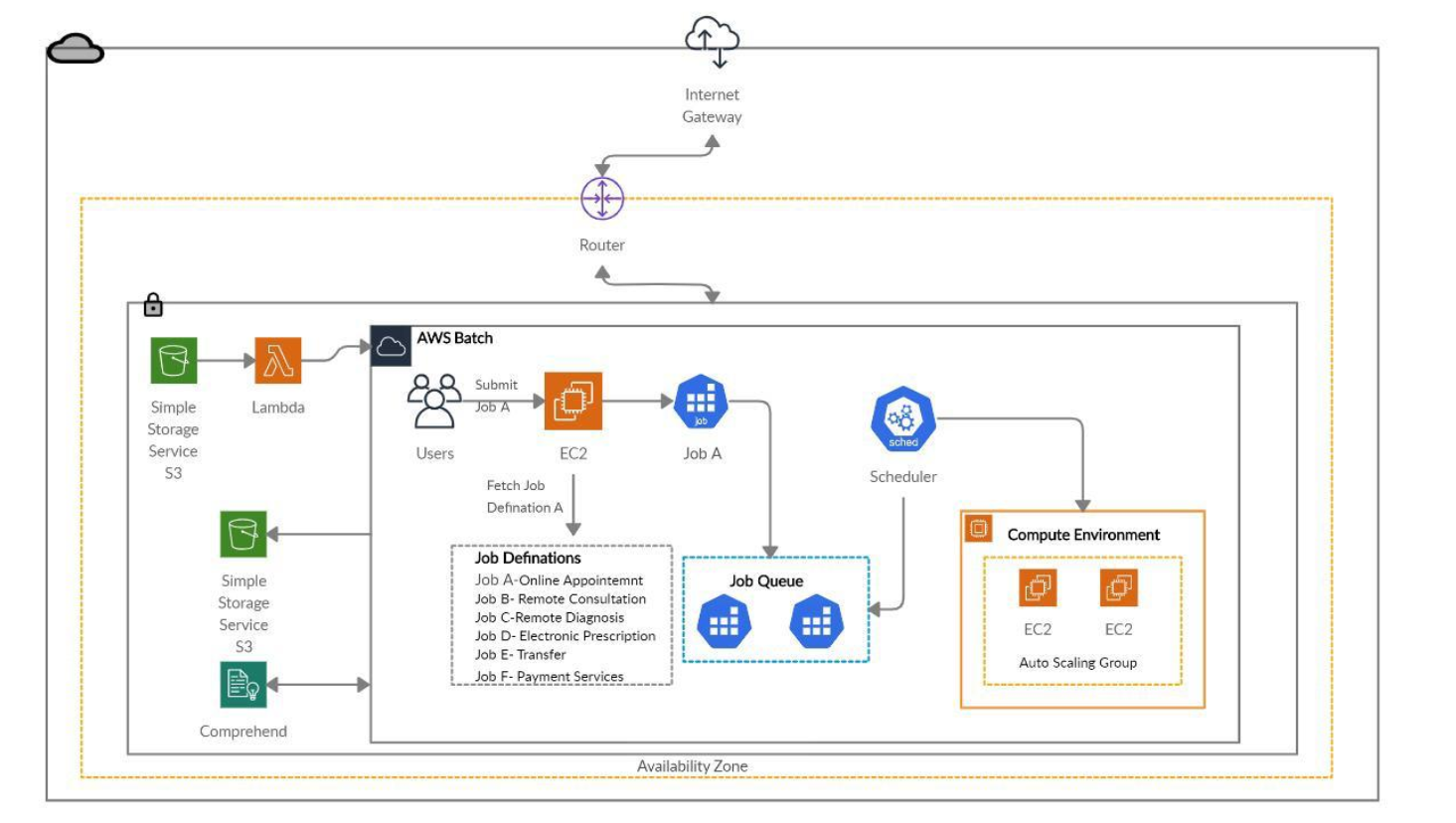


Figure: **Performance Efficiency + Allow online appointments, remote consultation, remote diagnosis, electronic prescription transfer, and payment services.**

**c) Monitoring and Auditing**

The architect above describes how we are planning to design infrastructure for medical companies in order to achieve operational excellence. Based on the users who will be using the application we have decided to divide it different components like:

• Website Component

• Backend Component

• Ops Component

• Analytics Component

A screenshot of a cell phone

Description automatically generated

Figure: **Operational excellence + Monitoring and auditing requirement + Upload document and images requirement**

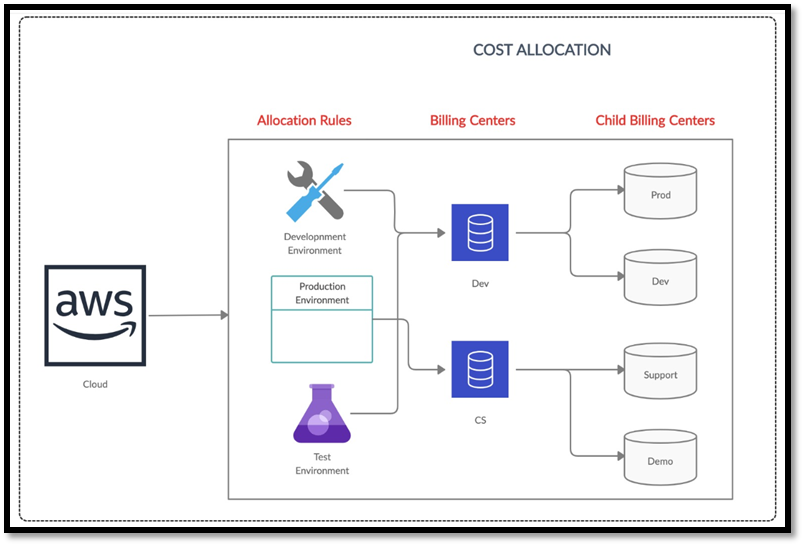
Once the application is hosted, the users will interact with the application using services mentioned in website components like EC2, S3 Website bucket which will further be used as a gateway to store their request and data. Cognito service will be used to authenticate each user login to avoid any theft as the users will be working on sensitive data which includes personal and confidential information. We are using API Gateway which with the help of lambda function extracts the data in the form of text from the document and images and stores it to RDS after converting it to multiple formats. Besides this, it will execute the actions by publishing notifications to Amazon SNS. This feature will help to cater the Upload functionality requirement in which users will get notification after the upload document and images.

For monitoring and auditing purposes, CloudWatch will be used which provide data and actionable insights for AWS as it is capable of collecting and access all performance and operational data in the form of logs and metrics which can be further used by its “Alarm” feature which is capable of receiving and sending notifications when the threshold is achieved. It will also help in initiating the auto scaling options if required.

For DBA purpose, we will be using Amazon Athena which will provide serverless query service which will be used to analyze the data using SQL. As it is serverless, so the medical company will have to pay only for the queries they run. To facilitate this, we have used Amazon kinesis firehose and event bridge.

**d) Cost**

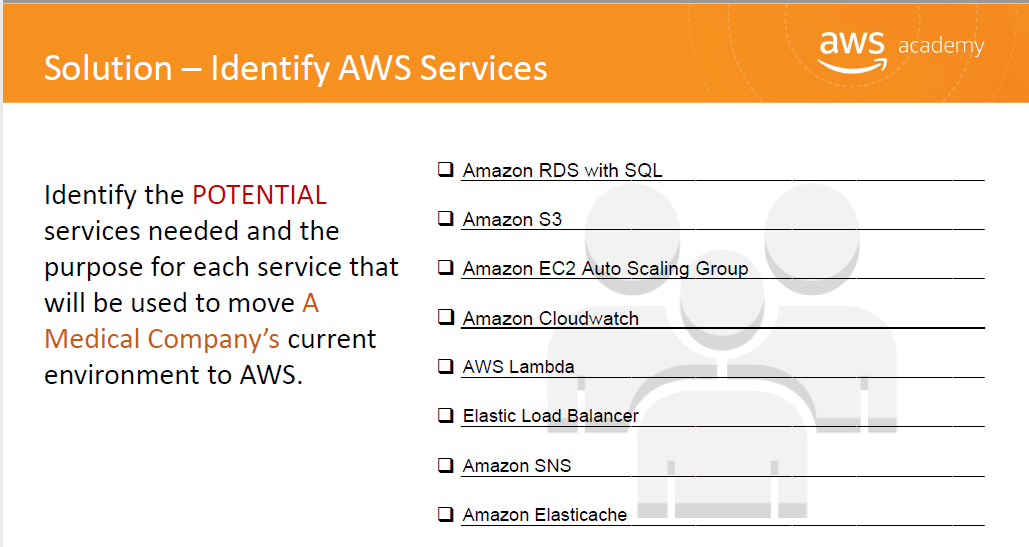
It is another important factor to consider while designing infrastructure as no company would like to invest more than what is required. We have decided to use the organizational model based on the requirement we got from Medical Company.



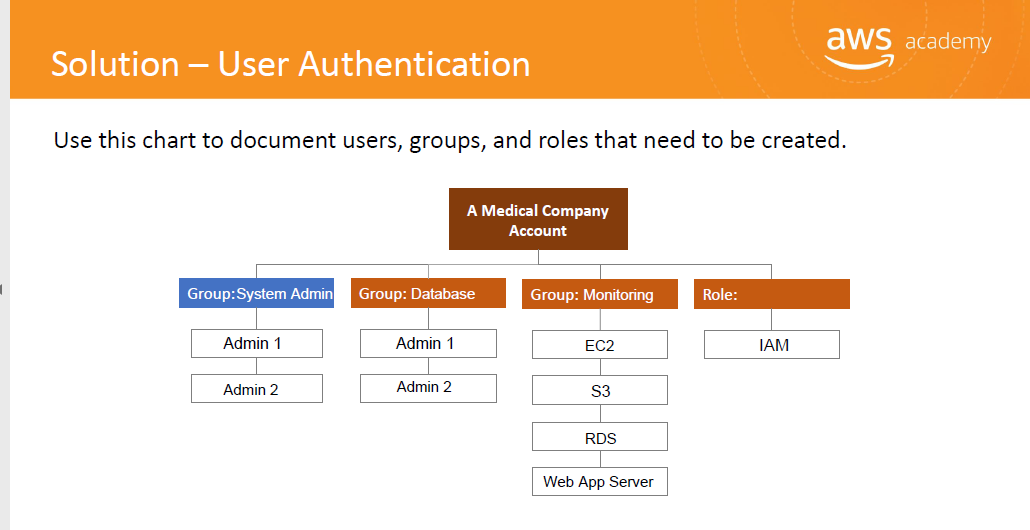
**Figure: Cost Optimization**

As per the allocation rules, we have created different distributed environments architecture for dev, prod and test which will keep the application alive for further future software releases or even for timely patches. We have created different cost or billing centers to optimize cost for our infrastructure setup. It will help to keep track on the usage of services per billing center and enable us to optimize the cost even per resource within the center.

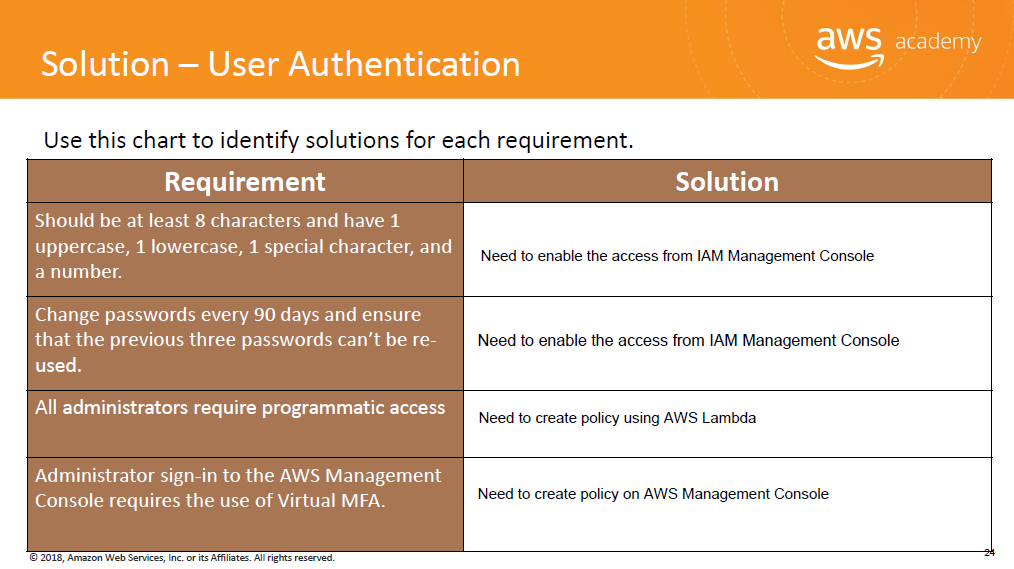
**Solution**



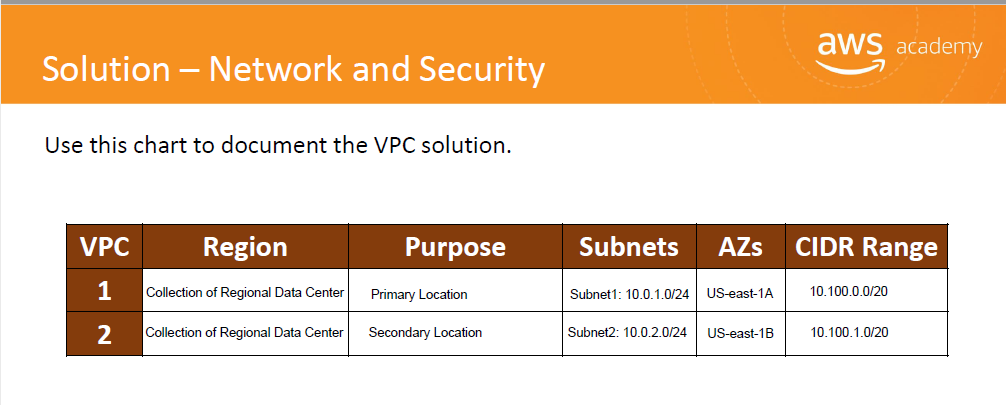
We have identified all the required services based on the requirements mentioned by medical company and their detailed purpose is explained above with the design pillars.

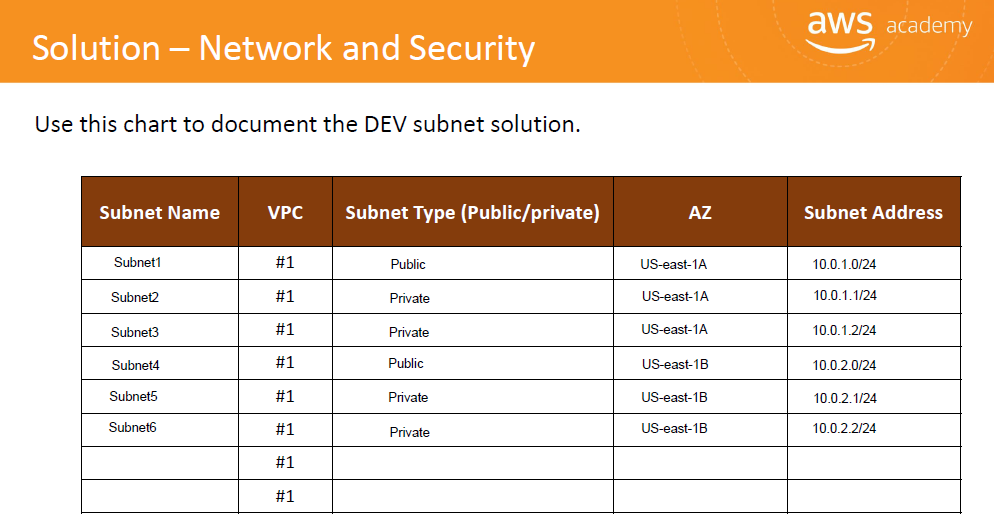


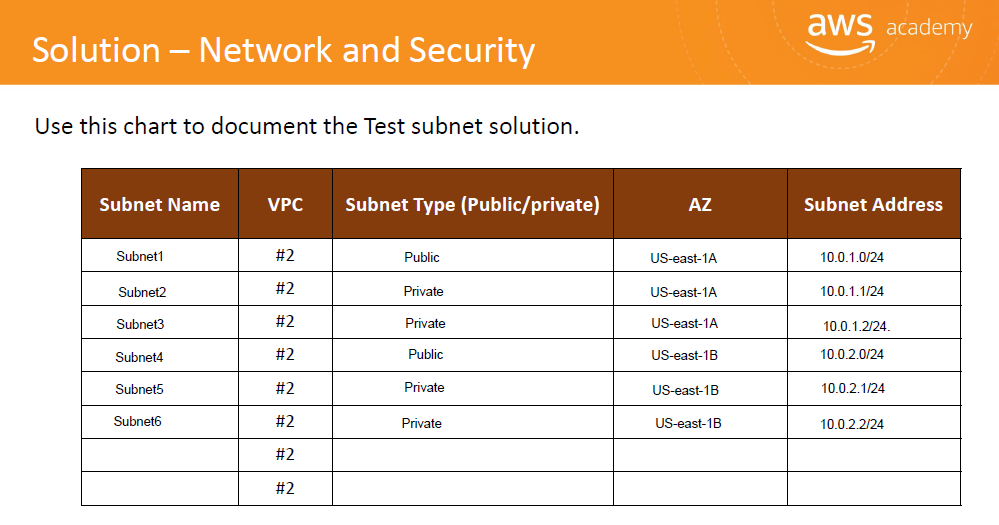
We have defined different groups like System admin, Database and one group for Monitoring based on the authenticity and privileges to use the system’s functionality and assigned services accordingly.



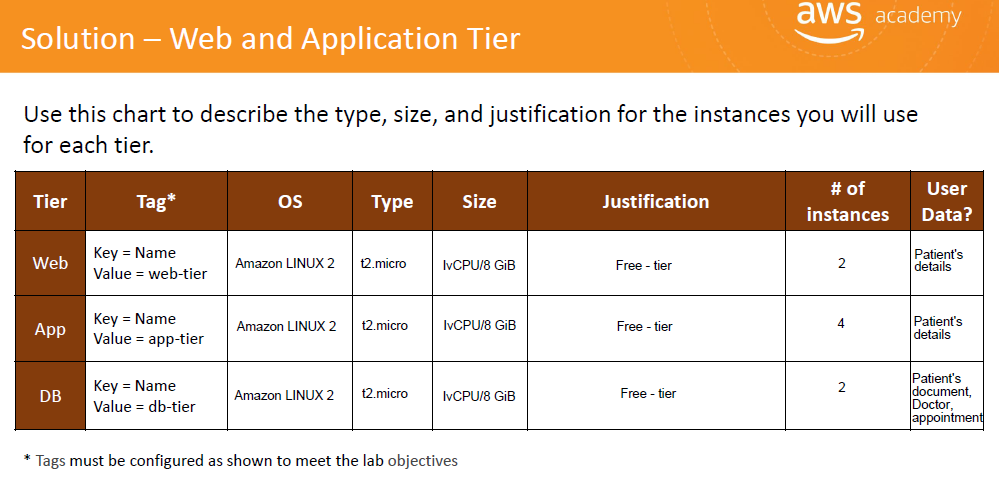
Above requirements are the system validations which we are taken care by providing the above-mentioned corresponding solutions.

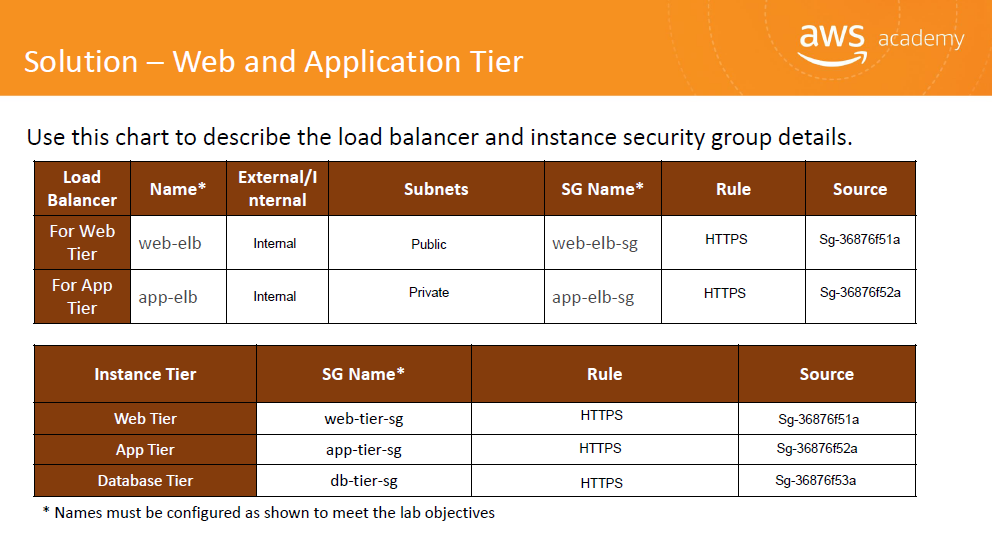




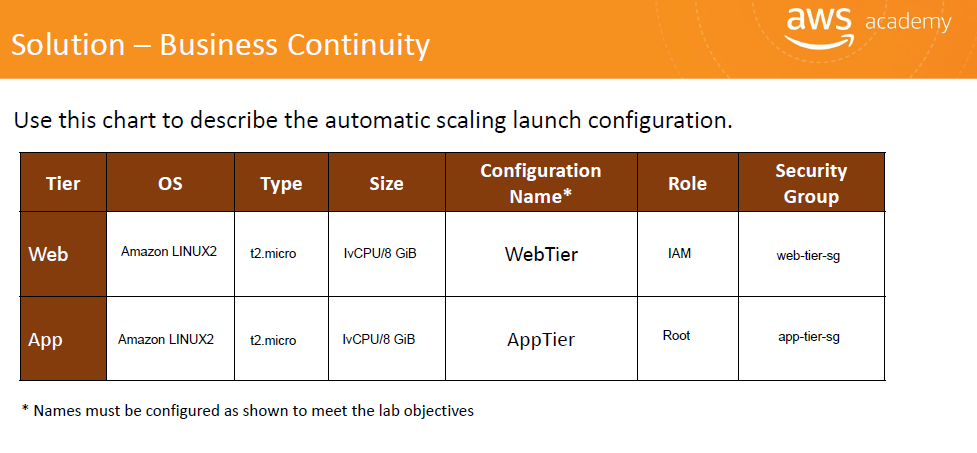


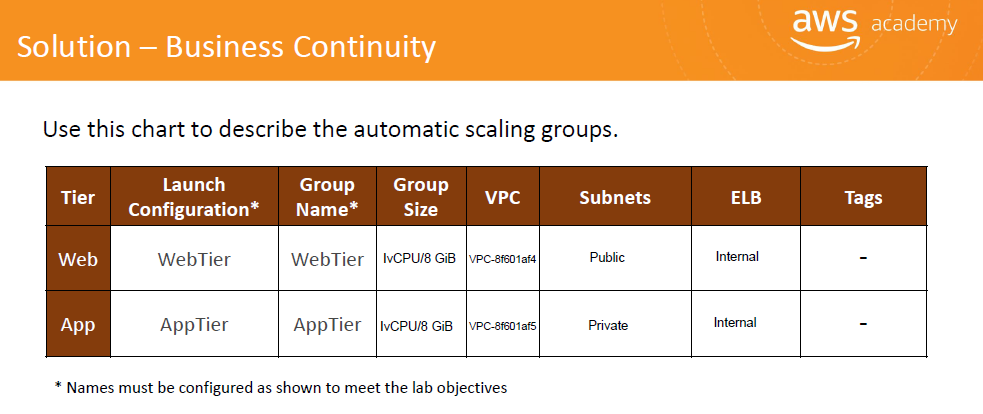
Above worksheets depicts how we have catered the requirements for providing optimal network coverage and high-class security to target audience. There are two private subnets and one public subnet. Our database is in private subnet. Web server is in public subnet and application server is in the private subnet. In short, it describes the technicalities we come across while designing high performance and most reliable architecture by accommodating two different available zones with above configuration mentioned in different types of environment including dev and test environment.



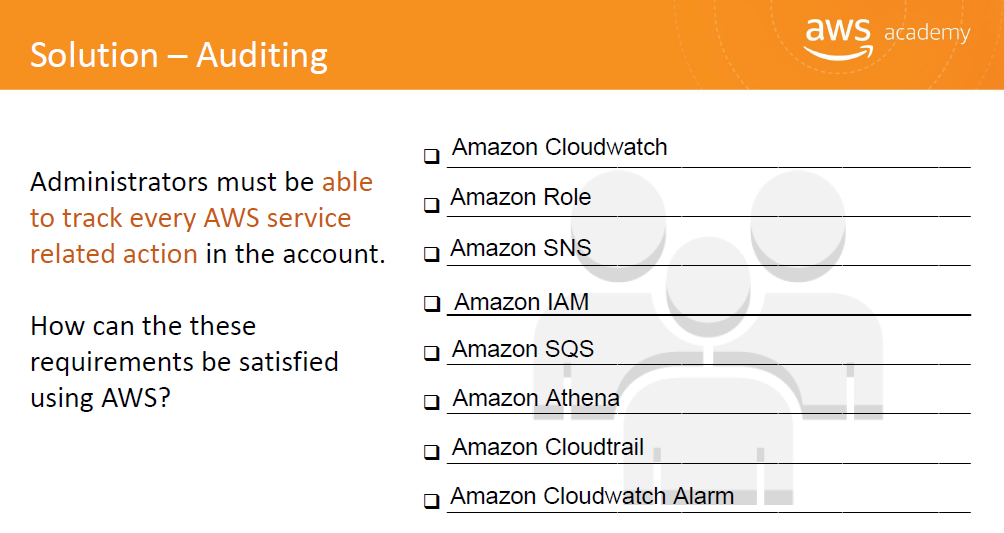


Above worksheet clearly describes the justification and usage of different instances used in each tier including Web and application tier.





This is how we have planned to maintain the auto scaling configuration options if required. This will help user and medical company to enjoy uninterrupted services even if there is a case of internal system failure.



Above mentioned services will enable admin user to perform auditing to keep track on the system.

**Next Steps & Conclusion**

The company can consider the possibilities of introducing AWS services such as AWS Lambda, AWS IoT, and Amazon Machine Learning, as well as making use of data lakes and big data via AWS Glue. The theoretical architecture constructed should be practically viable in a real AWS environment. Using AWS cost estimation calculator, the exact costs which will be incurred can be concluded. The cloud architects will consider accommodating third party vendors to the AWS environment by exploring additional AWS service to increase functionality, if need be.

With the use of AWS cloud formation architecture, IT department is moving from simply providing stable infrastructure to a position from which it can contribute to the business itself. We have translated customer requirements into a proposed technical solution. This is our team’s proposed solution to the customer.

**References**

<https://aws.amazon.com/blogs/startups/architecting-your-healthcare-application-for-hipaa-compliancepart-1/>

<https://aws.amazon.com/blogs/architecture/store-protect-optimize-your-healthcare-data-with-aws/>

<https://medium.com/weareservian/getting-started-with-aws-batch-3442446fc62>