Northeastern University

ASSIGNMENT FRONT SHEET

Course Name: AWS Cloud Architecting

Professor Name: Carmen Taglienti

CPS Term: Spring 2021

Final Project – Paper

Submission Date: July 3rd, 2021

Group No.6

Anmol Atmaram Desai

Quoc Tuong Dong

Yashmi Shirish Sevak

Statement of Authorship

I confirm that this work is my own. Additionally, I confirm that no part of this coursework, except where clearly quoted and referenced, has been copied from material belonging to any other person e.g. from a book, handout, another student. I am aware that it is a breach of Northeastern University's regulations to copy the work of another without clear acknowledgement and that attempting to do so renders me liable *to disciplinary procedures. To this effect, I have uploaded my work onto Turnitin and have ensured that I have made any relevant corrections to my work prior to submission.*

Team members Introduction:

Team members:

- Anmol Atmaram Desai- Master of Analytics, Data Science (CPS)
- Quoc Tuong Dong- Master of Analytics, Data Science (CPS)
- Yashmi Shirish Sevak- Master of Enterprise Intelligence (CPS)

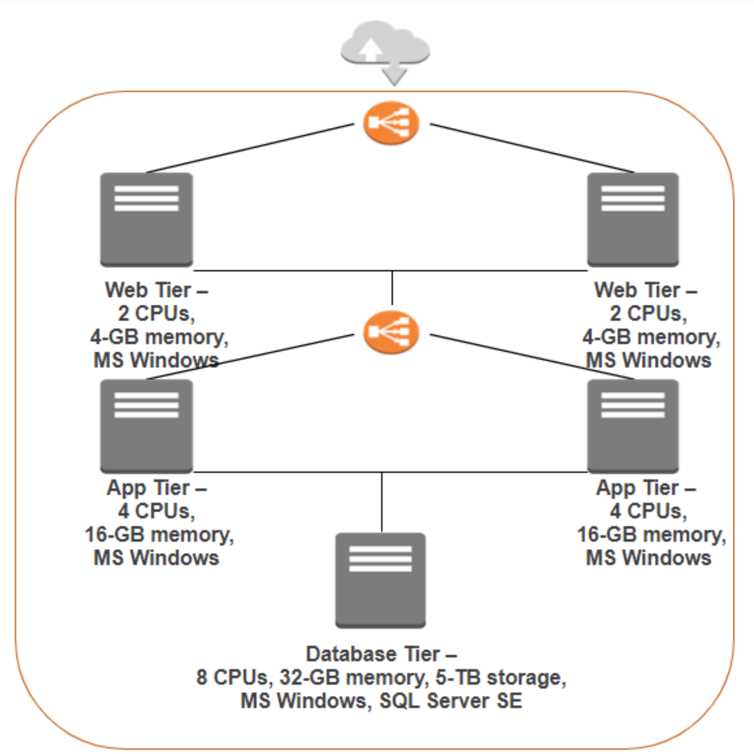
Project Introduction:

Our client is a medical startup SaaS company that has built an online medical social networking and diagnosis assistances application for APAC, US and Europe:

- Connect Patients and Doctors
- Customers can schedule online appointments/consultation/ diagnosis, electric prescription transfer and payment services
- Customers can upload documents and have text extracted to store in database
- Contains PII data and need HIPPA compliance
- Always available with no down time

The company is using Microsoft Windows servers to host their web and application tiers with Microsoft SQL Server Standard Edition backend databases.

- Web Tier: Two physical servers (2 CPUs / 4-GB memory)
- Application Tier: Two physical servers (4 CPUs / 16-GB memory)
- Database Tier: One physical server (8 CPUs / 32-GB memory / 5-TB storage)



Executive Summary:

Current Architecture:

The current system is hosted on premises with the web and application server on Microsoft Windows platform with IIS while the database is running on Microsoft Server Standard Edition.

Proposed solution:

Out of three current popular Cloud platform, AWS is the most suitable choice for our requirements that allows us to secure our data and expand the business internationally

Design

- Designing a scalable and more secure system for the 3-tier architecture which can be accessible to all three regions.
- Providing privilege rights to employees depending on their functional groups and assign them to dedicated AWS service.

Benefits

- Write Infrastructure as code, scale identical solution.
- Cheap solution and you pay for what you use.
- Innovate faster, reduced time to market.
- Globally accessible with no drop times.

Overall Requirements and Assumptions:

In order to successfully migrate the legacy database to the cloud infrastructure, our solution must follow these steps:

- Granting access permissions to the authorized users based on their roles
- Providing network connectivity to the applications in various environments.
- Constructing good architecture capable of autoscaling and handling crisis
- Adding monitoring services, protection to customers data and be HIPPA compliant
- Extracting documents from images

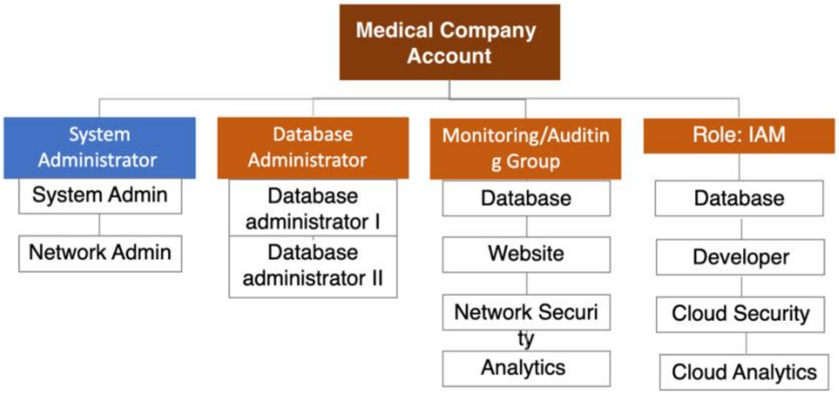
Solution:

1/ Identify AWS Services

- Auto-scaling: Automatically increase or decrease capacity on demand
- AWS Lambda: serverless computing power that allow the code to run automatically without managing or overseeing
- CloudFront: Content delivery networks
- CloudTrail: Tracking users' activities
- CloudWatch: A monitoring and observability service that was created specifically for the DevOps engineers, developers, site reliability engineers (SREs), and IT managers
- EC2: Offering secure, and resizable compute capacity such as Virtual Machine.
- AWS Cognito: Managing Identity inside application
- AWS Web App Firewall: Protecting system against web attacks and improving web traffic visibility.
- AWS firewall Manager: Useful for centrally managing firewall regulations
- AWS DMS: Effortlessly moving data from relation Database and Datawarehouse to the cloud system with minimum modifications to the schema and effort
- Elastic Block Store (EBS): It is a high-performance block storage solution. Handling wide ranges of workloads
- IAM: Establishing users, role and rules related to different roles.
- NAT gateway: Bridging instances in a private subnet to services outside your VPC but external services cannot initiate a connection with those instances.
- RDS: Making database configuration, management, and scaling easy in the cloud. As well as automating tedious tasks
- Route 53: Bridging user requests to infrastructure running in AWS
- S3: Aiding object storage

- VPC: Setting up a reasonably isolated section of the AWS Cloud which can be used to deploy AWS resources at scale in a virtual environment.
- Amazon Macie: Discovering as well as guarding sensitive data
- Secret Manager: Essential for automatic password change
- AWS Backup: Essential for scheduled Backup
- HealthLake: A HIPAA-eligible service enabling healcare practitioners to store, transform, query, and analyze health data at scale.
- AWS Sagemaker: Building machine learning models
- QuickSight: Allowing users to construct and publish interactive BI dashboards that include ML-powered insights
- Personal HealthDashboard: Displaying relevant and accurate information to help manage events in progress, and providing proactive notification to help users plan for scheduled activities

2/ Users Authentication



In this we created three accounts (System Administrator, Database Administrator and Monitoring/Auditing Group) and four Role (Database, Developer, Cloud Security, Cloud Analytics) in order to create user specific rules and provide access as per the requirements. We created.

a. System Administrator: Under System Administrator we created two groups –

1. System Administrator: All access

2. Network Administrator: Read/Write - EC2, AWS Cognito, AWS Web App Firewall, AWS Firewall Manager, CloudWatch, CloudTrail, NAT Gateway, Route 53, VPC, AWS Secret Manager

b. Database Administrator: Read – EC2 & Read/Write - CloudWatch, RDS, S3, DMS, EBS, Redshift, Backup.

c. Monitoring/ Auditing Group: Read – EC2, CloudWatch, RDS, S3, DMS, EBS, Redshift, Backup, AWS Cognito, AWS Web App Firewall, AWS Firewall Manager, CloudWatch, CloudTrail, NAT Gateway, Route 53, VPC, AWS Secret Manager.

d. Roles:

1. Database: Read – EC2 & Read/Write - CloudWatch, RDS, S3, DMS, EBS, Redshift, Backup.

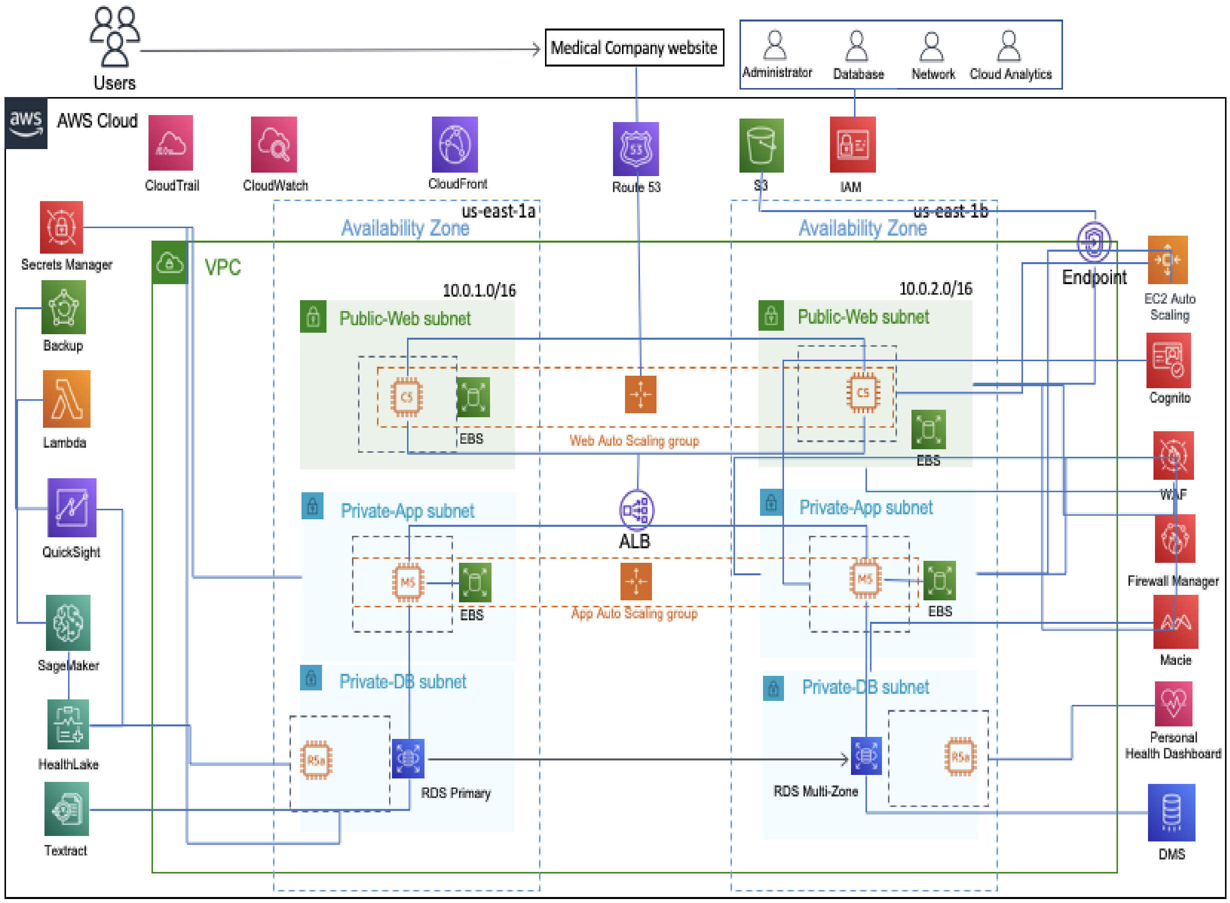
2. Cloud Analytics: Read/Write - AWS HealthLake, AWS SageMaker, AWS QuickSight, AWS Personal Health Dashboard.

3. Cloud Security: Read/Write- AWS Web App Firewall, AWS Firewall Manager, CloudWatch, CloudTrail, NAT Gateway, Route 53, VPC, AWS Secret Manager.

3/ Network Security

For better security and stability, we considered 6 Regions servers for this project as the company want to establish US, Europe and APAC market. For US region we used 6 subnets- four for deployment, two for disaster management and one for development and training. For APAC region we used 9 subnets (Six for deployment, three for disaster management) Finally for Europe we used 3 subnets (two for deployment, one for disaster management). For Publicweb-Subnet1, PrivateApp- Subnet2, PrivateDB- Subnet3 for each region.

4/ Web and Application Tier (Architecture Diagram



1

The architecture diagram as shown above consists of two availability zones in the us-east-1 region- us-east-1a and us-east-1b. We use Public Subnet for the Web tier and Private Subnet for Database tier with CIDR range as 10.0.0.0/16.

We use Route53 as a bridge between the users to access the website from the AWS infrastructure. IAM roles such as Administrator, Database, Network and Cloud Analytics as used. We use Autoscaling for all the three tiers for meetings the requirements as per demand without usage of resources.

AWS WAF and AWS Firewall Manager is used for protecting the website and application from web attacks ot any external threat where the Firewall manager manages these firewall regulations.

AWS Macie is used for all the three tiers to manage and guard sensitive data.

For Application tier we use AWS secret manager to change the password by a certain time interval for data protection. The data from web and app tier goes from endpoint of VPC to AWS S3.

For Database tier, we use AWS Backup services for emergency if the data is lost. AWS Healthlake is used to store, transform, query, and analyze health data at scale and then the developers or data scientists can perform machine learning algorithms using this data on AWS Sagemaker.

Data from AWS Healthlake can even be shipped to AWS Quicksight and AWS lambda users to construct and publish interactive BI dashboards that include ML-powered insights.

AWS textract is used to extract data from documents by scanning text or written data which can be used to store patient information for future purposes or to perform machine learning tehniques for health requirements stored in the database tier.

AWS Personal Health Dashboard is used to fetch data from the database tier to visualize information to manage on going events and give notifications for the users to schedule accordingly.

AWS DMS is used for the database tier to migrate the data to AWS RDS for efficiency.

AWS Cloudfront is a monitoring and observability service that was created specifically for the DevOps engineers, developers, site reliability engineers (SREs), and IT managers.

5/ Web and Application Tier

For Web and Application Tier we are using the latest Microsoft Windows Server 2019. Under web tier we considered c5.large instance as it can be used to handle web traffic and provide better website visibility and to lower the loads, we will be considering 2-4 instances. For Application tier we considered m5.xlarge instance as it can handle general purpose loads and to handle the load with any downtime, we will be considering 2-4 instances. For DB layer we are considering r5a.2xlarge instance as will help handle highly intensive DB workload with 1 master and 1 standby instance.

For security reasons we have created 2 separate security groups naming web-elb-sg for Web tier and app-elb-sg for Application tier connecting both to the Application load balancer with rule set to Port 443. For Database tier we create security group named db-tier-sg which has App security group as a source and applying the rule Port 1443.

6/ Business Continuity

For business continuity we have created a separate disaster management subnet we if we come up with any downtime on any of the instances, we will divert all our traffic to the respective instances of every region.

7 Auditing

CloudTrail- continuously monitor, and retain account activity related to actions across your AWS infrastructure

Various Events used to record activities-

Data events

Using Data event logging, we record various API activities. We can also fetch details on who initiated the request, its location and time. These are high volume activities which include Amazon S3 object level APIs, AWS Lambda function Invoke APIs and Amazon DynamoDB item-level APIs.

Management events

Management events give the details into the control panel operations that run on resources in the AWS account.

CloudTrail insights

Identify unusual activity in your AWS accounts.

Monitoring

- CloudTrail integration with CloudWatch Logs delivers management and data events captured by CloudTrail to a CloudWatch Logs log stream in the CloudWatch Logs log group you specify.
- CloudTrail logs are segregated according to regions we specify and then redirected to S3 bucket.
- These logs are recorded and then used for analysis.
- We also use AWS Elasticsearch and Kibana. The logs from CloudTrail are shipped to AWS Elasticsearch and then using Kibana Dashboard, we can aggregate, analyse and visualize the data.

Next Steps and Conclusion:

For this medical company we have created an AWS cloud architecture considering all 3 tiers that is Web, Application and Database to provide better scalability, security and flexibility.

- Using Machine Learning Services we have created a flexible environment to create and access the data dashboard.
- Cost Optimized solution.
- No server down time due to more than one AZ

Next Steps

- Migrating data from the company's Microsoft SQL server to AWS RDS Using AWS DMS
- Hosting website on AWS server
- Automating the system by configuring AWS lambda for machine learning using Sagemaker