Assignment 2
Cloud Computing
193RSEG-176-1DL, Fall 2, 2019
Instructor: Ari Davidow
Submitted by: Daniel Joyner & Kevin Markvenas

GROUP C
CLOUD ARCHITECTURE FOR FOR SOCIAL AND EPIDEMIOLOGICAL HEALTH DETERMINANTS ONLINE RESOURCE

November 26, 2019

OVERVIEW

1. PROJECT BACKGROUND AND DESCRIPTION

The MARVENAS Group has implemented the pilot project featuring a public facing searchable database which returns regionally specific statistical information about Life Expectancy, No. of Cardiovascular disease deaths, and % Poverty in regional total population, which are all social determinants of health. The resource is set to serve community and private stakeholders and utilizes Amazon Web Services (AWS) to manage features of storage, security, availability of the application, and others. Below, will be discussed the security and monitoring features that we utilized for this project, as well as going into greater detail about what benefits are obtained by utilizing AWS.

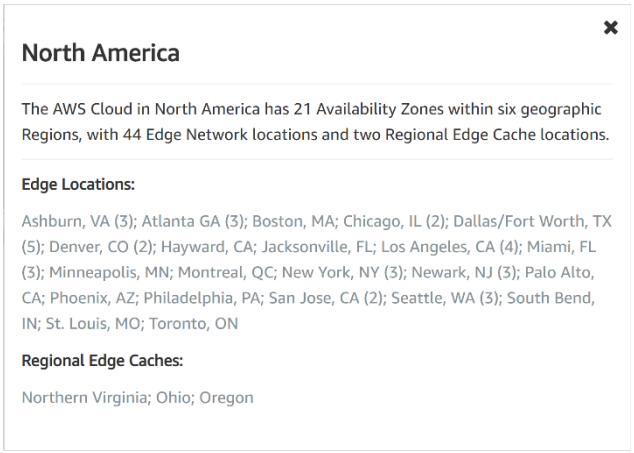
2. DIFFERENCES BETWEEN TRADITIONAL AND CLOUD COMPUTING ENVIRONMENTS[1]

a. IT Assets

By utilizing Amazon Web Services (AWS) our organization can save money on provisioning resources for our application. As our project grows in scope AWS enables us to activate more services to supplement that growth, where a traditional data center would require us to procure expensive resources that increase costs. Everything from additional servers, databases, storage can be instantiated or shut down as needed through their management console. This frees our organization of the responsibility of controlling these assets ourselves, as well as the problem of unutilized resources in the event of downscaling our operations. This also allows us the ability to test features without disruption or major expense, we can easily manage change, and logically build capacity.

b. Global, and Scalable Capacity

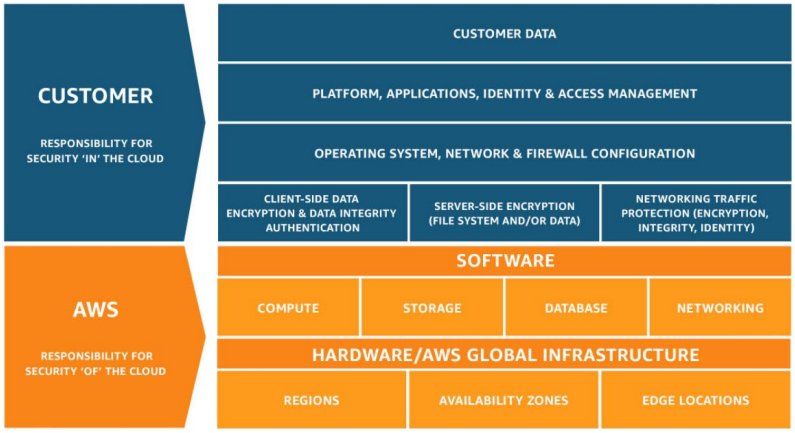
Right now, our solution is relegated to the east coast region of the United States, and only covers the state of Massachusetts. However, this application has the potential to become a national database, and AWS services allows for this potential growth. In AWS we can replicate instances of our application in multiple regions of the country to reduce latency of the user interface. AWS[2]. This capability also allows us to increase availability of our application across multiple data centers, increasing fault tolerance.



[1] Barr, J. (2010, January 14). Architecting for the Cloud. *Retrieved from https://aws.amazon.com/blogs/aws/new-whitepaper-architecting-for-the-cloud-best-practices/.*

[2] AWS. (2019). Global Infrastructure The Most Extensive, Reliable, and Secure Global Cloud Infrastructure. *Retrieved from https://aws.amazon.com/about-aws/global-infrastructure/.*

c. Built-in security[3]



Security at AWS is the highest priority. AWS works through a "shared responsibility model" (see figure left[4]) where Amazon provides security "of" the cloud including: highly secure data centers, world class security monitoring experts who regularly validate their systems to the highest standards, as well as software and encryption services that allow for standing or "in-transit" data encryption options. The other half of the "shared responsibility model" are those security protections that are the responsibility of the customer. This includes client-side data encryption, network traffic protection, and the like. Protections and security implemented by this project team are discussed below.

3. DESIGN PRINCIPLES

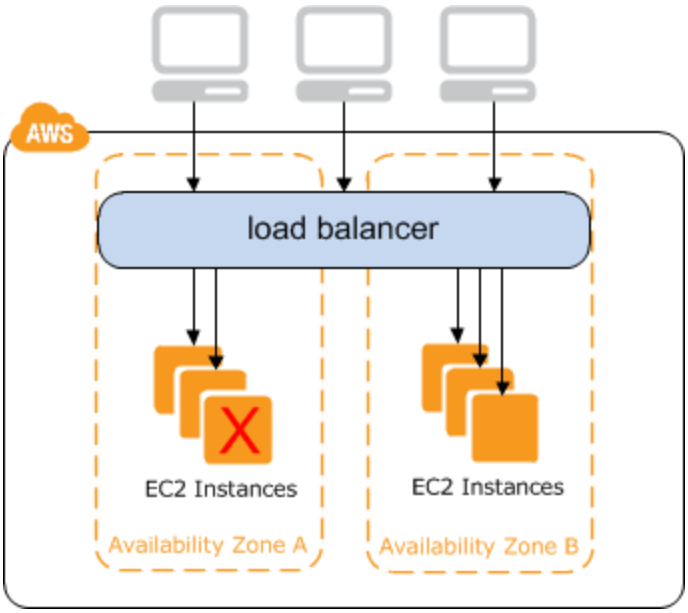
Because AWS employed a "shared responsibility model" for security we wanted to discuss exactly what security options were chosen from the suite of options that AWS offers. Also covered will be solutions we employed for monitoring, automating, and scaling the project.

a. Monitoring

For our project we are utilizing the AWS Load Balancer, which automatically and continuously monitors the health of registered instances. Elastic Load balancers distributes network traffic across multiple EC2 instances. When an instance of the application (in this case our lookup tool) becomes unhealthy, the load balancer reroutes traffic to healthy instances (see figure below[5]). The load balancer uses instances across availability zones, in case the issue is regional. This also increases the availability of our application.

---

[3] Mathew, S. (2019, October). Overview of Amazon Web Services. Retrieved from https://docs.aws.amazon.com/whitepapers/latest/aws-overview/introduction.html?did=wp_card&trk=wp_card.
[4] AWS. (2019). Shared Responsibility Model. Retrieved from https://aws.amazon.com/compliance/shared-responsibility-model/.
[5] AWS. (2019). Elastic Load Balancing: Classic Load Balancers. Retrieved from *https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/elb-classic.pdf#introduction.*



The load balancer features a *Listener* which checks the connection request from our clients.

b. Scalability

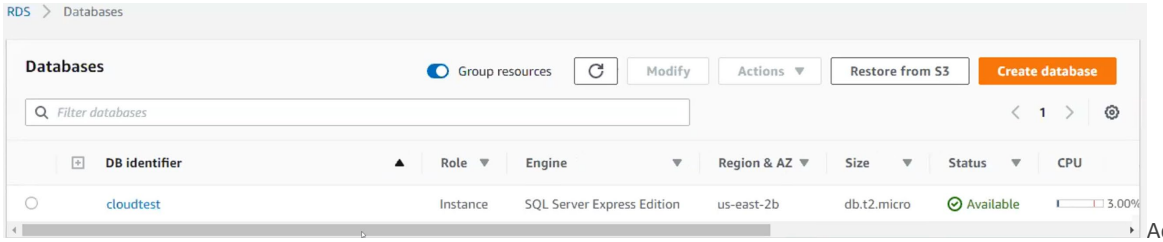
Elastic Load Balancing also scales vertically as traffic to our application changes over time[5] with EC2 Auto Scaling. It does this by distributing the traffic across instances, adding more resources or taking them away on demand as discussed above. This is set up in AWS with multiple instances of our application. Each instance will point to our RDS database.

c. Automation

We did set up EC2 auto recovery through Amazon CloudWatch, which sets up an alarm when any of the EC2 instances gets impaired due to hardware failure or other problem on the AWS side of the "shared responsibility model" (See above). The auto scale service creates an identical instance, as in our Load Balancer, containing the same instant ID and elastic IP addresses[6].

d. Databases

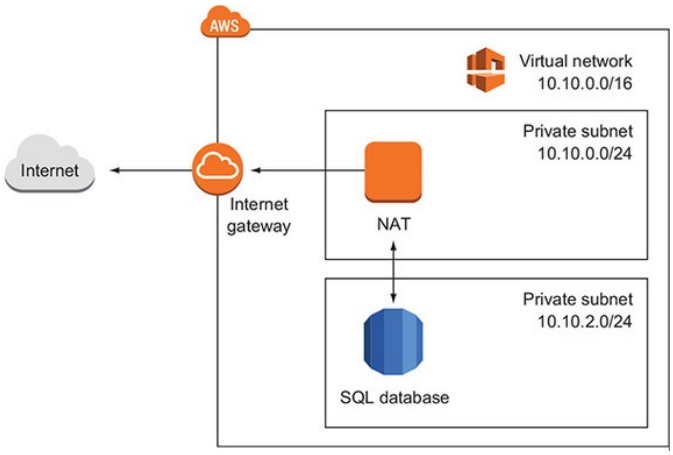
Our RDS database is associated with our Virtual Private Cloud (VPC) through AWS. As discussed in the project proposal: The interface for the searchable database will be a Website that we setup and maintain. In AWS we created an RDS DB instance of a SQL server in the Ohio East region. It is named "cloudtest".



Access to The "cloudtest" instance is controlled by a Virtual Private Cloud (VPC) security group and firewall rules (see Figure below). The

[6] AWS. (2019). Recover Your Instance. Retrieved from https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-recover.html.

website will be hosted on an Elastic Cloud Computing (EC2) server. Right now, our instance of the SQL server is running on a template provided by AWS on a virtual network (Microsoft Windows Server 2012 R2 Base), utilizing .NET development environment. We launched an EC2 server with Microsoft Server 2012 R2 and Internet Information Systems (IIS). We also included the database on AWS so that the site is all Amazon driven. All the image files are being stored in an S3 storage bucket (not shown). The public website accesses the database through the VPC instance (Figure below).



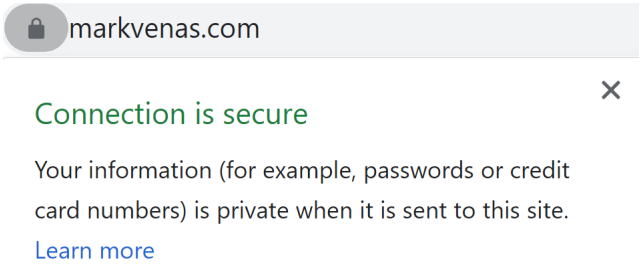
Figure[7] (Adapted from Wittig,2018)

e. Removing Single Points of Failure

Our use of the Elastic Load Balancer (ELB) feature eliminates single points of failure by routing traffic to healthy endpoints[3]. Amazon CloudWatch performs health checks for our instances, and we have alerts set up to notify us of any issues with performance. Additionally, the Autoscaling function of the ELB seamlessly replaces unhealthy nodes so the user will not experience any disruptions in service.

f. Security

For security, we added an SSL (Secure Sockets Layer) certificate which secures our site, enabling encrypted communication from our website. Therefore, we can be assure the anonymity of our users that interact with our service who may enter in personally identifiable information like their Zip code or town of residence. This can be seen my navigating to markvenas.com, a secure website designation is now displayed on the site, and the DNS is preceded by "HTTPS (See image below)."



Additionally, AWS allows us the ability to incorporate security features into some of the automated services they offer. Therefore, in order to secure our Load Balancer, we set up a key value pair. AWS used Keys to encrypt login information. There is a public Key that is tied to the instances we create. This key serves to encrypt the data on AWS. The private key decrypts the

[7] Wittig, Andreas & Michael. (2018). Amazon Web Services in Action, 2e. *Manning Publications Co.*

data and is tied the machine utilizing the service[8]. Additionally, our application uses secure ports, that we designated, to access the RDS database hosted by AWS. This function of only being accessible through a secure port, adds an additional layer of security on the client side, along with the SSL certificate and the AWS automatic encryption.

4. CONCLUSIONS

By taking advantage of offerings provided by AWS, and being diligent about the parameters of shared responsibility offered by their model, we have gained an invaluable partner in AWS. Our application is taking advantage of their world class security and monitoring service with the highest quality and most up-to-date protocols to protect our application. Additionally, we have the advantage of setting up automated monitoring services which also enable built in responses like fault protection through our Load Balancer. We are also, built to scale by enabling Auto scaling, so we can ensure optimization of experience for all of our users at peak traffic times, and we are not wasting resources during low traffic. We are also taking advantage of the wide availability of AWS servers, so when our solution goes national, we can easily service all areas of the country. Please feel free to share this document with stakeholders, as it is our certain belief that we have no better partner than Amazon Web Services.

5. TEST THE SITE

Click below to be redirected to the testing page or follow the link to:

https://www.markvenas.com/

Regional Health Data Lookup 

[8] AWS. (2019). Amazon EC2 Key Pairs. Retrieved from https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html.