Dr. Andrew Schwarz

ISDS 4130

May 6th, 2023

GoGreen Overview (2017-2019)

GoGreen is a company that provides its end-users with a Customer Relationship Management service designed to make the tracking of customer sales data and documentation easier. This business is understandably storage intensive, and as its customer base grows will become increasingly computationally intensive as its architecture will have to handle an increase in requests for a growing base of customer data. It is unclear exactly what their end-users sell to their customers, but the fact that there is a sales process worth storing for an extended period means it is likely a very financially and chronologically burdening investment. This means availability, fault tolerance, and database durability will be vital moving forward.

Current Architectural Overview

GoGreen utilizes a standard three-tier architecture with a dual storage setup for file and block-level storage. The Access layer of their architecture encompasses the User Devices, Internet, and the DNS Server. The Distribution layer includes a NetScaler and dual Web Servers which split the load distributed by the NetScaler. Next, there are a pair of App Servers which connect directly to a respective Web Server. These then feed into a master Oracle database and attached slave. There is also an attached File System Disks setup that likely is in place for near-line file storage and other frequent access media. There is also an Active Directory system that appears to be spanning both the Distribution and Core layers, as well as a tape backup for the databasing solution.

Current Architectural Assessment

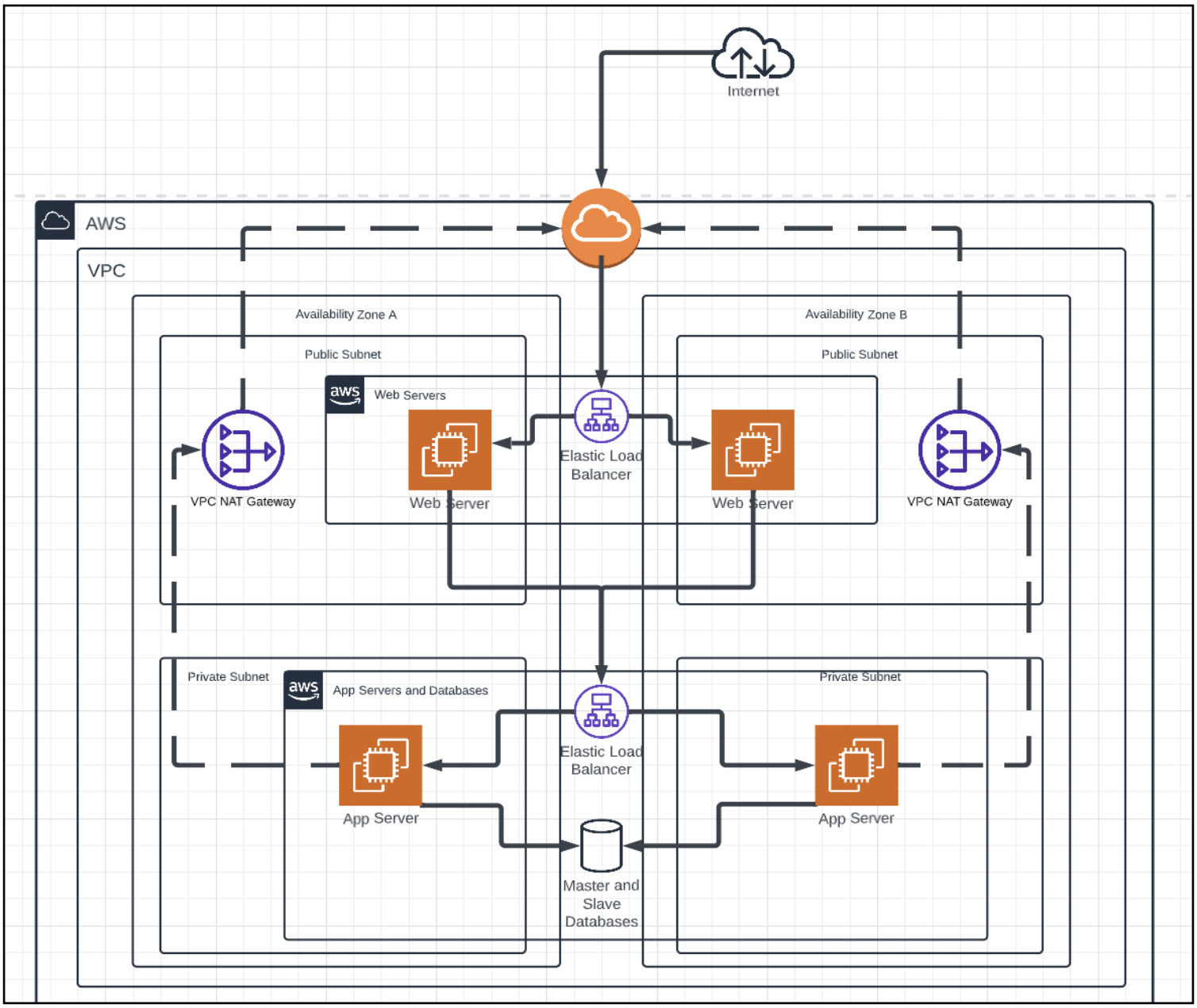
The lack of interconnectedness between Web and App Servers is immediately concerning in terms of fault tolerance. Should a fault occur in a Web Server, its respective App Server is now useless. The use of only a single firewall to protect the master database is also very concerning, as essentially the entire architecture is exposed except for the very core. This leaves DDos and Packet Sniffing opportunities essentially wide open for malicious behavior to exploit. Finally, the use of backup tapes is ancient and rather cumbersome if the data they capture needs to be retrieved. As their business continues to grow, so will the likelihood of backups being needed. Essentially, to meet their RTO needs for increasingly massive backup sizes, Hard Disc Drive (HDD) backups will become necessary.

In terms of strengths, the use of a Master and Slave database protected by a firewall is optimal. As read requests can be offloaded to the slave during times of high demand so that the Master may focus upon write requests. And the database firewall will deter data-based attacks like SQL Injection or Buffer Overflows. The use of multiple web and app servers in conjunction with a NetScaler is a good start toward distributing network load, but their lack of

interconnectedness in a ‘loosely coupled’ way is a massive fault point. Lastly, the use of File System Disks for near-line storage is a common standard for storing frequently accessed data. It is not specified exactly what setup they are running, but it is a safe assumption to believe it is a Network Attached Storage (NAS) solution in a Redundant Array of Inexpensive Disks (RAID) configuration. If this is indeed the case, then their data storage appears to be mostly sound for the current demand. But as the business continues to grow, horizontal scaling or even an outright upgrade may become a necessity for their file system near-line and ‘deep-freeze’ tape cassette backup storage solutions.

Proposed Architecture

Given the needs outlined for moving the current architecture to AWS, the following diagram was created to model the ideal architecture: *Link for ease of viewership.*



Subnetting and VPCs: Considering the medium size and anticipated growth of this company, utilizing the multi-account approach in coordination with AWS Organizations will be the optimal solution. This means we will create an account with a copy of the VPC outlined above for each region we deploy in. In our case, this means one account for each of the two regions outlined as necessary. We are utilizing one public and one private subnet per Availability Zone, and these subnets will be allocated /24 CIDR ranges for public nets and /18 CIDR ranges for private subnets. Since the network load and projected load are not exactly specified, it will be assumed that the number of devices on the private nets will be high due to the amount customers who request the services at any given time. It is fair to assume the 65,000 addresses of the /16 CIDR block will leave us ample room for allocation for the foreseeable future without compromising efficiency. As the load increases, it will be much simpler to overprovision now and 'grow into' our subnet blocks than attempt to reallocate addresses later. We will use AWS Organizations groups to group together our public and private instances and services across our AZs, as well as user roles to a minor degree for administrative purposes. Though it must be understood that the decision to use multiple accounts was largely made to limit the need for complex user permissions rules.

Data at Rest and In-Transit Security: By utilizing Security Group chaining and Virtual ACLs, our core level infrastructure will be protected from unrestricted internet traffic and cannot even actively interact with external users without passing through our VPC NAT Gateway. All communications to our app servers and the databases they feed into are thoroughly filtered by the many layers of security groups. Virtual ACLs also monitor all traffic on both our public and private subnets, should any suspicious write or read requests target our RDS databases, our ACL will simply reject the malicious activity. The use of a VPN, or Virtual Private Network, will protect our data in transit via encryption so that it will be difficult for interceptors to capture and interpret it.

Performance Efficiency: If increased performance is desired, then the use of M7g General Purpose instances will more than suffice. At up to 30 Gbps, these instances and their processing power would render our Web and App servers respectably fast for the better part of a decade.

Topology Reliability: Each of our VPCs has two Availability Zones, and it is recommended to have two AZs per region. This ensures reliability on a large scale as if one zone is compromised by a disaster or other event, the secondary zone will be able to serve the region. On the per zone micro-scale, the use of paired elastic balancers at each tier makes routing more dynamic than the rigid 'one-to-one' layout before. This means if one Web server is compromised, the load balancer will be able to direct traffic through the non-compromised server instance, or even auto-scale through an event trigger or other configuration option.

Scaling: By either vertically or horizontally scaling the power/amount of EC2 instances, we will be able to dynamically handle the load placed on our network during spikes.

Scaling Cost Optimization: By utilizing Reserved Instances for our planned computing needs (those depicted in the diagram) we will be able to leverage the discounts provided by AWS for our commitment. In the event we need to scale quickly, the use of On-Demand instances will allow us to combat extreme network traffic until the flow of traffic declines and these instances can be judiciously terminated to avoid cost.

Key Take-Aways

Key Point 1: Money Made. Increased Availability and Fault Tolerance through load-balancing and auto-scaling will result in more money made by our business due to drastically reduced failure frequency and lightning-fast recovery times.

Key Point 2: Money Saved. The auto-scaling and flexibility of AWS allow us to use only what is needed by dynamically provisioning and de-provisioning resources. Resulting in money saved on computing costs while simultaneously earning money.

Key Point 3: Practicality. The services offered by AWS and the model in which AWS presents them is extremely attractive for a multitude of reasons, but the most appreciable trait is the inherent practicality of being able to allocate resources to our company with the click of a button on a management console. As opposed to having to purchase hardware, wait for it to get here, assemble/prepare it for production, integrate it into our data center (that we must pay to rent/maintain), administer any updates or maintenance requirements, pay for facility costs associated with its integration…the list can become never-ending especially as we begin to grow globally. But with AWS, our configurations are saved in AWS Config and all that is needed to deploy to anywhere on the planet is a few minutes of planning, a few more minutes of decision-making, and a couple of pressed buttons. Whereas before you would be looking at **months** of planning and frustration.

End of Part 1

GoGreen Overview (2019-Present)

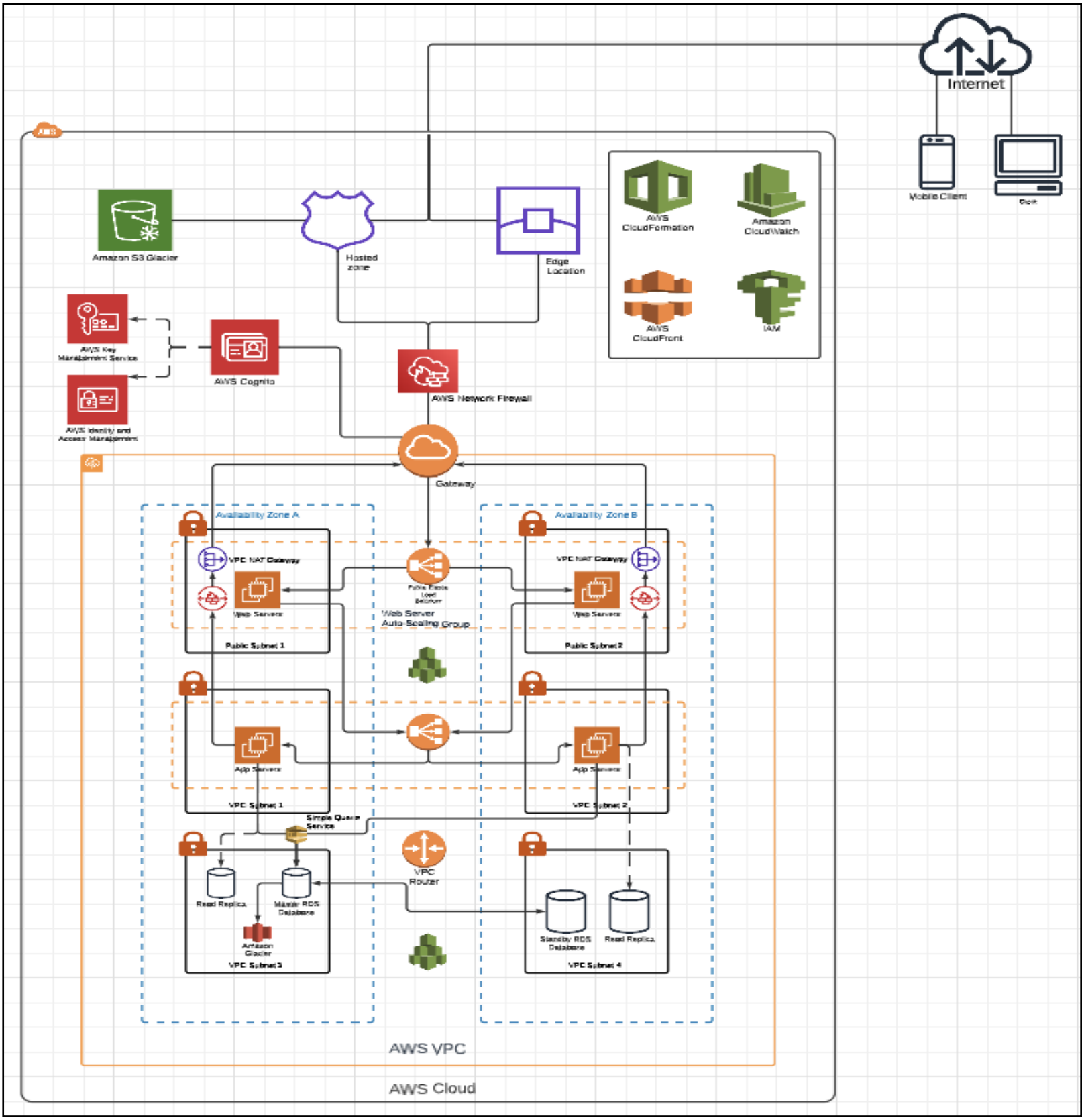
GoGreen has grown tremendously, with potential markets emerging across the United States and in Asia. As a result of this growth, new deployments will be necessary to effectively serve customers with low-latency service. The storage and compute solutions designed previously have worked but are beginning to show signs of inadequacy due to increased rates of simultaneous user access and greatly increased storage needs. This will necessitate the creation of a truly three-tiered architecture with databasing solutions separated from our app servers, as well as the integration of session management and user authentication. Since we are already going to be expanding and generating new deployments, now is an optimal time to redesign our cloud architecture to maximize our effectiveness moving into this new globalized period.

Current Architecture Assessment

The solution presented previously fails to account for more advanced necessities like user authentication, session management, and the need for a true three-tiered architecture. It also does not factor in near-line storage and backup storage necessities. These oversights will eventually harm a business of GoGreen's size, so they must be remedied in this coming initiative by adding additional functionalities to our infrastructure groups and incorporating them into our architecture. It is also worth noting that our cost optimization strategies must be explored more, to provide a full understanding of the tangible and intangible costs associated with every architectural decision made. I believe the current architecture is a good start, with a solidly defined foundation and functional design. But with a few key additions, will be more than enough to handle foreseeable growth.

Proposed Architecture

Given the needs outlined for improving GoGreen's architecture in preparation for a global expansion, the following diagram was created to model the ideal architecture: *Link for ease of viewership.*



Storage: Our production data will be stored in an RDS Block storage instance with read replicas and an entirely separate standby database. The databasing solution will feature snapshots offloaded to Glacier for deep storage in the event records need to be retrieved from a specific point in time. As far as administrative storage like logs and user authentication data, an S3 Glacier bucket will store routing and traffic data from the Hosted Zone instance, and the user authentication data will be stored in AWS Key Management Service.

ELB and Route 53 Benefits: Elastic Load Balancing and the use of Route 53 Hosted Zones enable web traffic to be handled in a dynamic and fault-tolerant way. The functionality of these features allows events to be automatically handled by the provisioned infrastructure in a way that maintains production capabilities.

Importance of Datastore Selection: Since the legacy architecture features a relational database setup, the decision to use RDS was made. RDS is highly scalable and customizable and is easy to set up and maintain. AWS Cognito provides user authentication caching to track sign-in status, and CloudFront caches general media content for low-latency access.

Web-Scale Media Hosting: The use of AWS CloudFront for low-latency caching is vital for our global expansion. When paired with Elastic Load Balancing, we will be getting the most optimal performance for our money. By storing media data near users, we can ensure a satisfying experience while using our services.

Auto-Scaling: CloudWatch will be configured to monitor all our EC2 and load balancer instances for certain levels of performance we deem unacceptable. If these thresholds are exceeded, auto-scaling can be utilized to quickly provision (and then de-provision) on-demand EC2 instances. It may also be configured to distribute SNS alerts to administrators who may then manually provision such instances if that is our desired method.

The 5 Pillars

Pillar 1: Operational Excellence. The incorporation of CloudWatch for event monitoring, KMS for user credential storage, and a Glacier S3 bucket for routing logs ensure working statistics are easily presented to administrators. These measures will enable more expeditious failure recovery and judicial operations monitoring.

Pillar 2: Security. All data in transit within our network is encrypted by default since it is within a VPC. CloudFront also ensures data remains protected once it leaves our network and is cached in an edge location. Our network-wide firewall and respective firewall endpoints ensure both inbound and outbound traffic is filtered of harmful content. Cognito ensures user authentication, and CloudWatch monitors all traffic for any signs of danger. Our defense-in-depth is 5 layers deep by the time it reaches our core databasing and adheres to industry standards throughout with a total of 6 layers of defense across total traffic. If desired, data stored within our databasing can remain encrypted with keys stored in our KMS instance, this may be implementable for customer payment or personal information and would push

our tally to 7 layers of defense. Anything short of an extremely hardened state actor with extensive computing power would be hard-pressed to mount a notable attack on this architecture.

Pillar 3: Reliability. Elastic Load Balancing between multiple Availability Zones, optional CloudWatch Alarms, and multiple databasing solutions with read replicas and deep freeze snapshots will ensure absolute reliability in the event of all but the most catastrophic of events. No amount of load can outpace on-demand computing power at the hands of our administrators with AWS's impressive computing capabilities.

Pillar 4: Cost Optimization. Auto-Scaling functionality and judicial monitoring by our administrators (who are aided by CloudWatch) will ensure we do not overprovision computing power during spikes in demand, and the reservation of reserved instances will keep our baseline computing costs to a minimum. The decision to connect a standby and read replica databases is one that may be slightly more costly but will ensure an extremely robust failure response that will save us more money than it will cost in the long run. The utilization of Simple Queue Service in tandem with the RDS Database will ensure data is not lost in the case of a filled database or of a database at capacity in times of intense load. This saves money as well, as we no longer need to provision our database to a high rate, but instead to an average rate that makes the most of the queue's temporary storage.

Pillar 5: Performance Efficiency. With reserved instances running M7g General Purpose Processors, and the option for nearly unlimited on-demand computing power, it would be difficult to overwhelm our infrastructure by nature of sheer computing load. Elastic Load Balancing and Edge Caching ensure the load is distributed evenly, and in proximity to users, to keep our latency as low as possible for a quick and clean experience.

Key Take-Aways

Key Point 1: Enhanced Security. The addition of AWS Cognito, CloudWatch, and a Network Firewall enhances the overall security of our platform. Not only are our subnets encrypted, but all activities are now monitored by CloudWatch so that any breaches or suspicious behavior will now be captured for administrators to monitor. The use of Cognito requires users to authenticate, increasing security by limiting web and app server access to authenticated users. This service works in tandem with Key Management Service and Identity and Access Management to encrypt and securely store user authentication information in an easily accessible and functional user pool. Finally, our network firewall and VPC NAT Gateways ensure that incoming malicious traffic cannot penetrate our network. Essentially, all traffic entering the gateway will be authenticated as legitimate. Even if malicious traffic enters the private subnets, the firewall endpoints will intercept any attempts to steal data on the way out into the public subnet. The abundance of security measures is known as "Defense in Depth" and makes a cyber-attack on our infrastructure so difficult as to make it not worthwhile for any attacker.

Key Point 2: Robustness. This solution includes more robust routing, storage, and administrative components. The integration of an S3 Glacier Bucket to store hosted zone traffic logs, CloudWatch alarm ability, CloudFront containerization, IAM user pooling, and Glacier block storage make this architecture much easier to monitor and troubleshoot and much more stable in terms of integrity. Core customer data is offloaded onto Glacier for deep storage in the event of a recall being necessary. This will make error handling and fault resolution simpler moving forward.

Key Point 3: Reliability and Performance. The integration of CloudFront and CloudFormation allows our engineers to deploy and maintain a high degree of low-latency service to our customers. Instances can be quickly deployed and hosted in edge locations as close as possible to our highest-demand areas. The use of multiple load balancers means that traffic will be dynamically redirected in the event of a fault, ensuring no lapses in our service. These load balancers may also be modified to distribute SNS notifications to administrators in the event of a failure for remediation. Finally, database read replicas and a standby database ensures high availability and performance.

"The Network is the Computer." – John Gage, Sun Microsystems