SF Match Migrates Legacy Infrastructure to AWS to Modernize and Scale Operations

Executive Summary

For nearly 40 years, SF Match has provided residency and fellowship matching services in various specialties and subspecialties across the spectrum of medicine including ophthalmology, plastic surgery, movement disorders, critical care anesthesiology, microsurgery, and many others. They work to connect people with organizations using a proven applicant review and ranking methodology as well as a Centralized Application Service that streamlines the processing, distribution, and review of applications with a simple user interface.

> “ClearScale was there to answer our questions and provide the background and technical expertise that led to a very organic and natural migration. Throughout the process, each step was explained, and as we approached completion of the project and migration off of our legacy hardware, ClearScale provided the needed support to ensure that our first experience would be fully supported.”
>
> **Dennis S. Thomatos,** Manager of the SF Matching Program

The Challenge

SF Match had been experiencing performance degradation and availability issues due to their continued organic growth, aging hardware, legacy technologies and several single points of failure in their architecture. When a hardware refresh cycle came due for their servers and infrastructure, SF Match decided it was the perfect time to migrate to AWS in order to provide the flexibility they needed to grow and modernize their operations which were predominantly being run in Microsoft Windows and with IIS/.NET applications.

High level requirements:

- Reduce outages/increase availability
- Increase Scalability and Elasticity of services
- Reduce IT management overhead
- Automate infrastructure and application deployment
- Increase Recoverability/Disaster Recovery

The ClearScale Solution

SF Match chose ClearScale based on our Premier Partner status with AWS and our extensive experience performing migrations away from legacy infrastructure. We began with a detailed analysis of the client's requirements, then we began planning the architecture design, automation implementation and deployment in AWS, including data migration using real-time file sync and real-time database replication.

In this case, because of the client's organic growth over the last 40 years and the diminishing returns they were experiencing with their existing architecture, a "lift and shift" migration methodology was not an option if we were to meet their specific requirements. As opposed to simply porting their applications to the cloud with minimal, if any, changes made to their underlying function, it was necessary to wholly reevaluate their needs and design a new architecture that would meet those needs while supporting continued growth.

AWS services implemented were: CloudFormation, Amazon Machine Images (AMI), CodeDeploy, Elastic Load Balancers (ELBs), CloudWatch, CloudTrail, AWS Config, Identity and Access Management (IAM), Virtual Private Cloud (VPCs), Availability Zones (AZs), Simple Storage Service (S3), Elastic Compute Cloud (EC2), Auto scaling, MSSQL Multi-AZ RDS, MySQL Multi-AZ RDS, NAT Gateway, VPN Gateway, Simple Email Service (SES).

Infrastructure Design

Throughout the infrastructure design and implementation, we leveraged PCI compliant AWS services for enhanced security. An important goal of the new architecture was to provide improved scaling for future growth, along with the necessary redundancy to ensure smooth operations. AWS auto-scaling meets this need by automatically increasing or decreasing the number of application instances needed to meet demand.

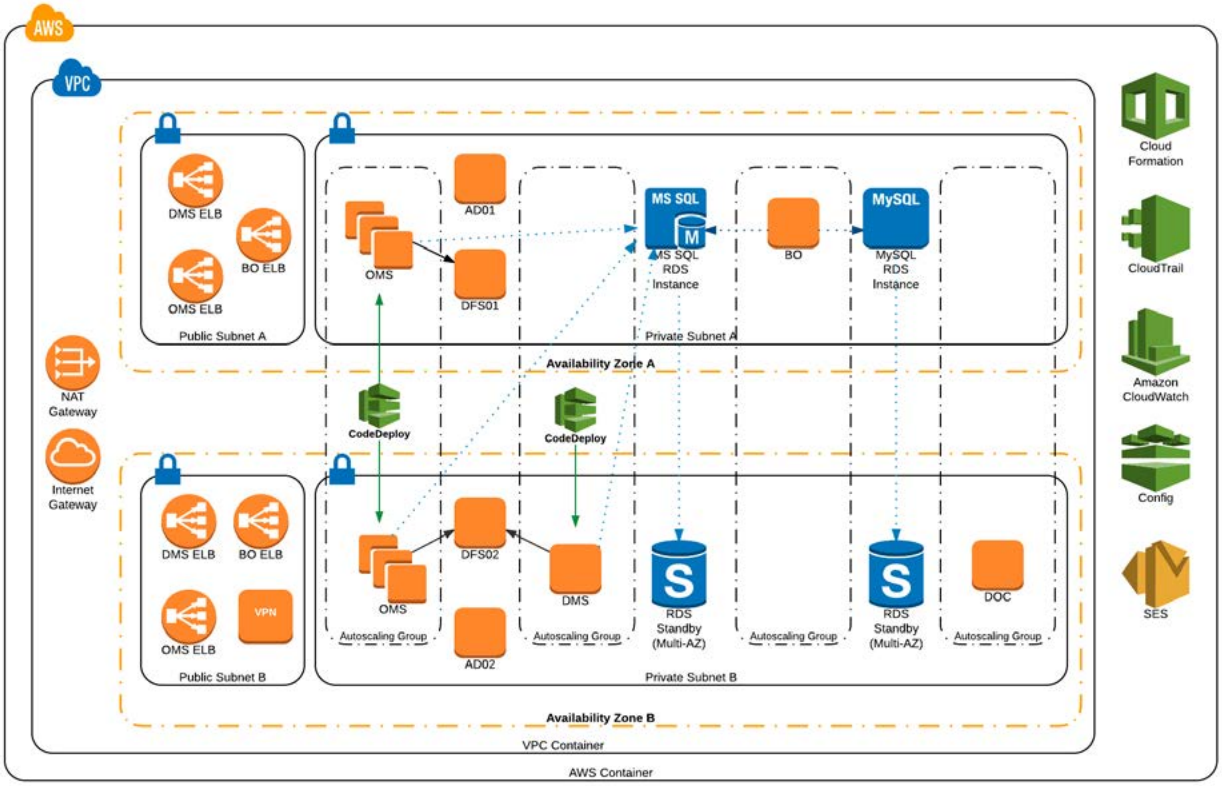
AWS Availability Zones (AZ) were also implemented to further the goal of constant uptime and redundancy. AWS allows for application resources, such as instances and relevant data, to be placed in multiple AZs. This ensures that if Amazon experiences a failure in one data center, SF Match's application will still function.

Because the new infrastructure runs on multiple AZs, Amazon RDS for MySQL was incorporated to help ensure that the underlying database could scale to the demands of the application. We implemented Amazon RDS to automatically maintain synchronous copies of the database in each of the AZs in use.

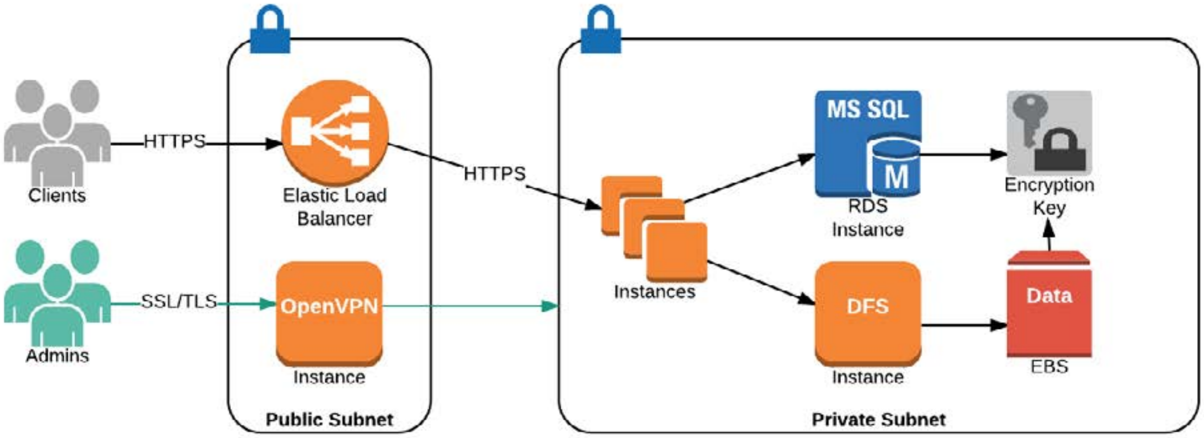
A Cold DR Region was used for backups of code, snapshots and AMIs, and Microsoft DFS clustering was implemented for HA file sharing and PCI compliance.

The integration of these technologies also provided a much more robust path to disaster recovery, as there were multiple synchronous copies in the cloud, as opposed to the legacy in-house solution previously used.

AWS Deployment Diagram:



Data Flow Diagram:



Automation

CloudFormation was used to provide developers and admins with an easy way to create, deploy, and manage AWS resources. Using stack templates (Master, VPC, Application Level, Database), CloudFormation manages the various dependencies or runtime parameters an application needs to function properly.

AMIs, in turn, contain the information needed to launch an instance or virtual instance on AWS. The instance can include an operating system, application server and the applications to run, along with the various permissions that control access to the applications.

CodeDeploy was integrated to automate the process of deploying new code. This speeds up deployment time, reduces downtime associated with updates and minimizes errors that can occur with manual deployment.

Implementation and Migration

To ensure the smoothest transition possible, ClearScale used a three-step process for implementation.

1. Data migration occurred once the application had been designed, coded, and deployed. This involved running a real-time file sync from the Datacenter to AWS.
2. A mock cutover scenario was run to identify any potential issues that might arise and address them before the production cutover.
3. The production cutover to the new service was performed once all bugs and potential issues had been identified, while the existing system was temporarily left in place while testing and documentation was finalized.

The Benefits

SF Match has matured their infrastructure capabilities, reduced IT management overhead, and implemented a controlled/automated method for releasing new code. All single points of failure have been replaced with HA systems or mitigated with automatic failover configurations. The new environment is elastic and highly scalable, and infrastructure automation drives quality and repeatability of deployments. SF Match chose to leverage managed services, which drastically reduced IT overhead and maintenance costs. Ultimately, their new architecture eliminates single points of failure and provides quality, high availability service to their users.