Rapyder Re-Engineers AWS Infrastructure for MilEarth

Introduction

Milearth is an Ed-Tech organization that empowers communities and people everywhere to use technology to realize their dreams. They promise to deliver technology solutions that help them do and achieve more, whether they are at home, work, school, or anywhere in their world. It was set up in 2016, and has grown rapidly in field of an educational technology solution. With its well-researched strategies, educational services and an efficient execution, Milearth stands out from other similar providers with passion, determination, enthusiasm, and quality. With an exceptionally talented pool of employees who are continuously looking to challenge their limits, it is poised to become a leading player in technology solutions.

Business Needs

Milearth's flagship product, MilGrasp aims to help progressive development of all the entities associated with a school, i.e. School Administrator, School Teacher, Parents, Students, by simplifying technology and bringing all the things one needs right into their pocket. This solution was deployed on AWS LightSail using single-tenant architecture. The deployment was being done on separate servers for each client that they were supporting. They were looking to make this existing solution more scalable with High availability multi-tenant architecture to support all their clients with the same single infrastructure. The solution was also be high scalable along with provisioning for content delivery, server caching with the ability to program code to deploy on the server on every commit that they were maintaining in Code Commit.

Solution Approach

The Cloud Consultants of Rapyder did a detailed study of the customer's application and designed the right solution and configuration that could support the seamless functioning of the application to overcome the challenges mentioned above. The consultants of Rapyder proposed the following solution to achieve the desired results:

- Amazon EC2 server was provisioned and all the requirements were carried out.
- Route53 was used for DNS pointing
- RDS created on a private subnet and restored with SQL dump.
- Servers were behind ELB and in Auto-scaling Group for high availability and scalability.
- Auto-scaling launch configuration was created and deployed with a script to fetch code on every new server launch.

- Lambda code was written to copy the code from AWS code commit repository to Server whenever on every commit. For high security, the Lambda function was created within the same network.
- AWS Redis used for providing central server cache so as not to lose session cache.
- AWS CloudFront implemented for content delivery and also restricted bucket only through to CDN for better security.

Reaping Rewards

- Deployment of code was automated, reducing the time to market.
- Auto-scaling reduced the time and manpower required for managing the server monitoring task.

  The client was able to commit many changes on code any time.
- Security-enhanced by restricting access only through CloudFront. SSL certificate applied on the load balancer.