**GSA 18F Technology Stack Write-Ups**

**Amazon Web Services (AWS) Elastic Compute Cloud (EC2) Instances**

We are using a variety of compute resources from AWS EC2 to power our application. Our server instances are Amazon Machine Images (AMI’s) from the AWS Marketplace that are running Ubuntu 14.04 LTS and the MEAN stack (MongoDB, Express.js, Angular.js, and Node.js).

**AWS EC2 Elastic Load Balancing and Auto Scaling Groups**

Our Test and Production environments contain multiple servers that are running in different AWS Availability Zones. The web servers require EC2 load balancers to distribute requests to the servers in order to balance the application’s request load and provide optimal performance and redundancy. The load balancers monitor the health of the web servers and balance requests between the servers. Auto scaling groups are in place to spin up new server instances should the existing servers become overloaded with requests. When the request level decreases beyond a certain threshold, the number of servers is reduced.

**AWS CloudWatch Monitoring**

We are monitoring our EC2 server instances, load balancers, and auto scaling groups using AWS CloudWatch. We are monitoring server health metrics such as CPU utilization to determine if the servers are being overloaded and these metrics are fed into the Auto Scaling configuration to determine whether new servers are needed to help with the load and/or if the number of servers can be reduced if the load is low. The load balancers are also being monitored to determine if HTTP errors are occurring on the servers and if latency is at an acceptable level.

**AWS Route53**

AWS Route53 DNS service is being used to provide a fully qualified domain name to access our application. DNS Alias records are set up for our Test and Production load balancers so that all requests are directed to the load balancers which then pass the requests to our servers.