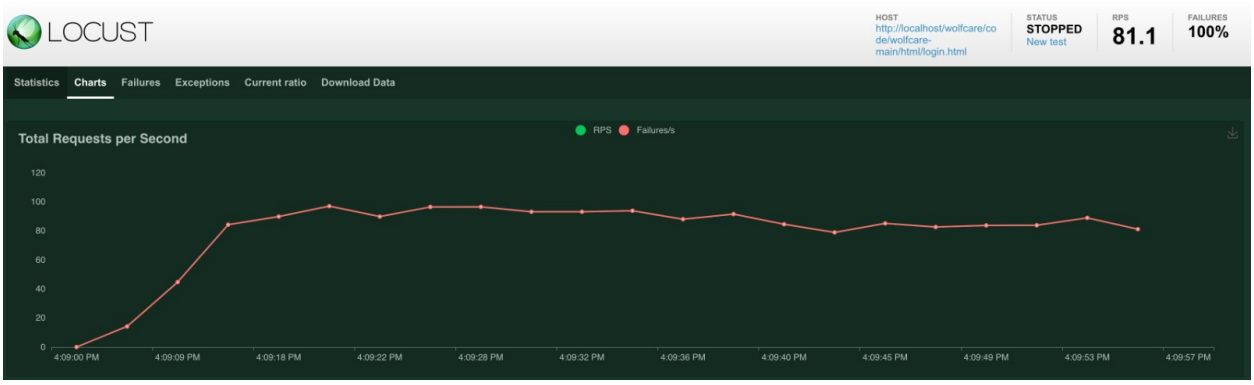
Application Scalability Design

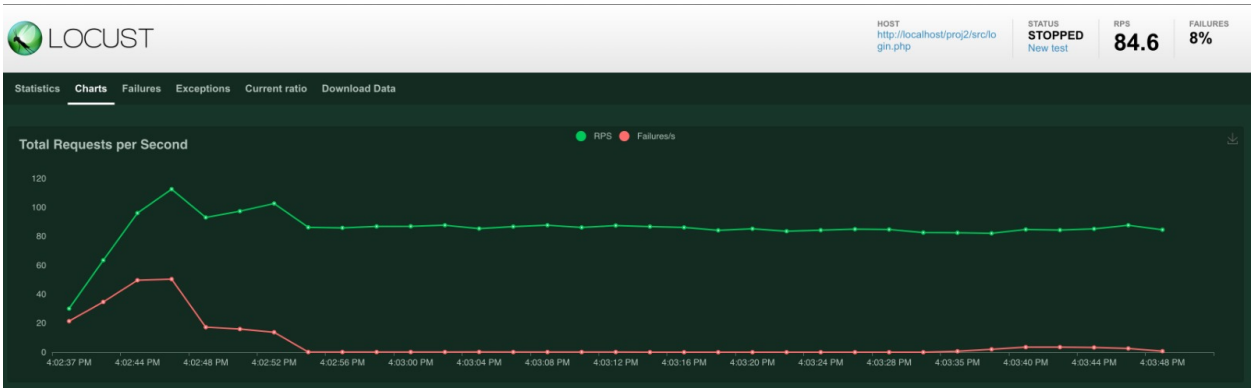
We performed the load testing on wolfcarev1 and observed that the application can handle 10 user requests per second. This load was generated using locust where the inputs are given as number of requests and the target application is wolfcarev1 which is hosted on the local server.



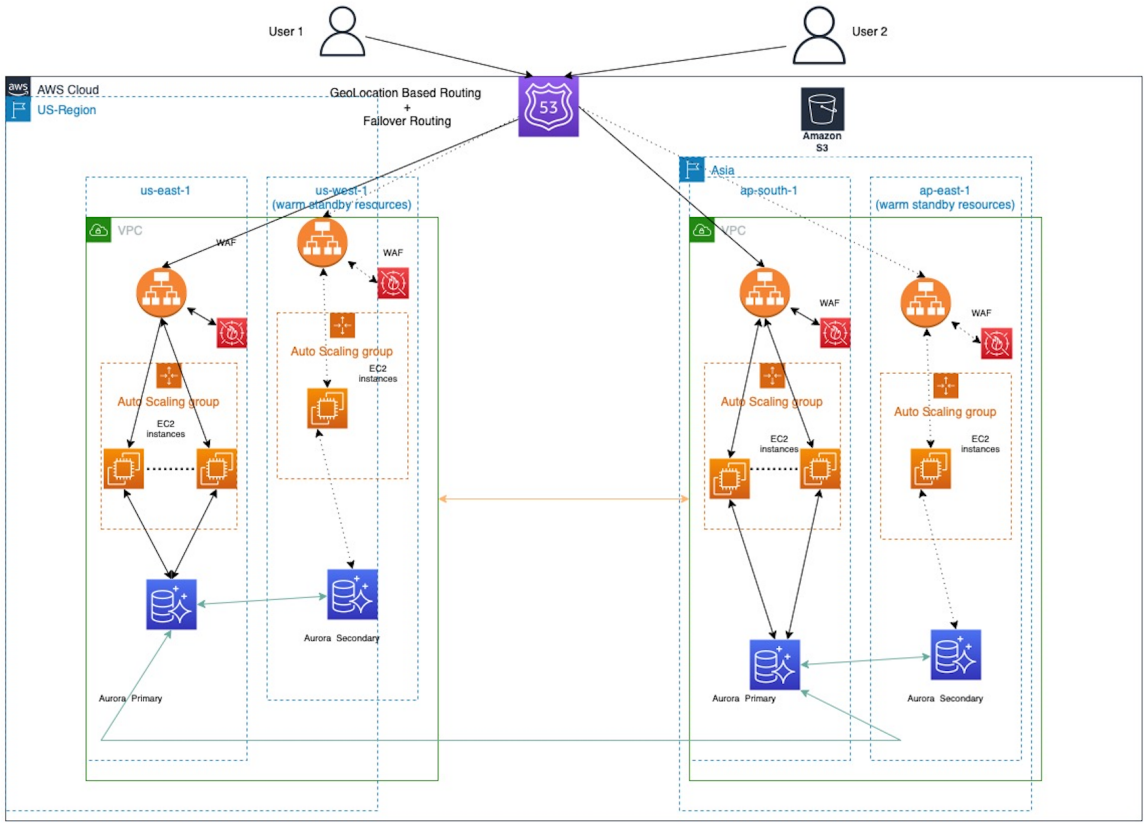
For wolfcarev2 we considered two key parameters for scaling the applications.

- Number of requests per second
- Faster response times

We observed that wolfcarev2 can handle approximately 85 user requests per second.



The application can be hosted on AWS cloud to provide better scalability.



To Scale In or Scaler Out the resources based on the average number of requests, Amazon EC2 Autoscaler is added to the design. Autoscaler subscribes to this metric from AWSCloudWatch and compares it with the threshold. If the metric has crossed more than threshold or reduced less than threshold, autoscaler launches or terminates Amazon EC2 instances and tends to ensure sufficient EC2 instances are available to handle the load for our application. These new instances are clones from the AMI Images created for the wolfcare application.
Amazon Route 53 provides a functionality to create and manage our public DNS records and subdomains as well. To reduce the average latency between the source of request and the compute resource that serves the request, our design incorporates the GeoProximity based routing feature in the AWS route 53. Instead of routing all the traffic to a single availability zone, AWS Route 53 service with GeoProximity feature routes traffic to the resources based on the geographic location of the workload requests being received from and the compute resources that are needed to serve the request. Two major geographic locations our application users are considered are in Asia and North America. With Route 53 workload requests would be forwarded to the closest of the three geographic locations.

To avoid a single point of failure in all three geographic locations, the loadbalancer along with other resources that it would forward to are deployed in two different availability zones. One availability zone acts as primary and another availability zone acts as backup i.e secondary.