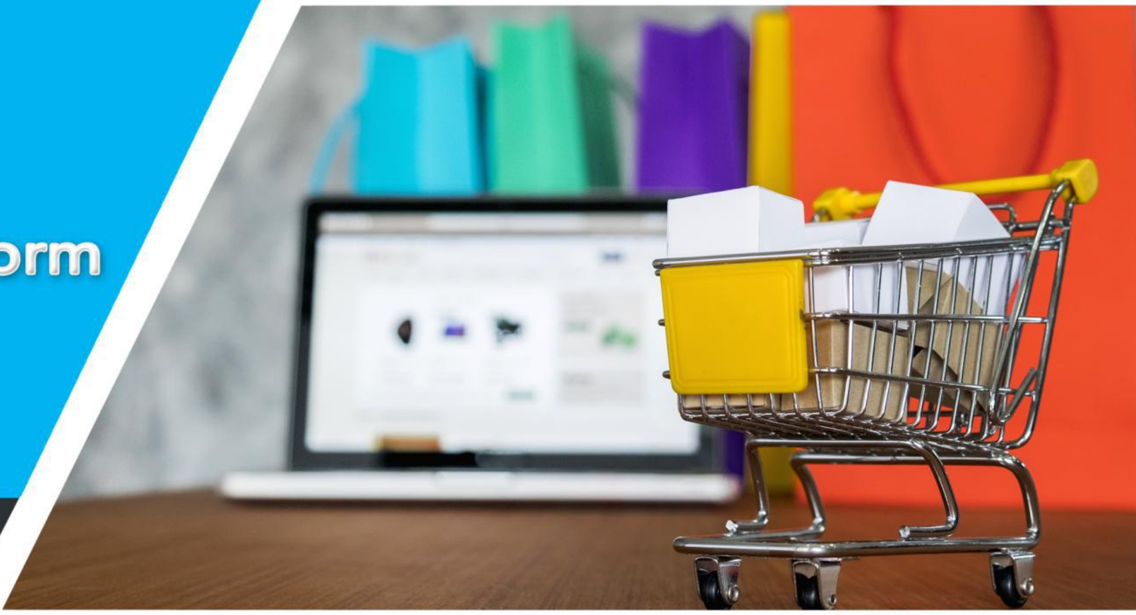
Magento Scalable Platform



info@crozaint.com
www.crozaint.com

Microsoft AZURE PARTNER

amazon web services Partner Network
CONSULTING PARTNER

crozaint

Summary

The client has a multinational e-commerce website developed using Magento 2 through which they sell products worldwide.

Challenges

The client was using a dedicated server and it became an issue at the time when they were running any promotions on their website. It was a challenging task to meet client requirements by keeping it cost effective. They had to spend huge investment upfront in order to meet the server requirements to meet the requirements for promotion days.

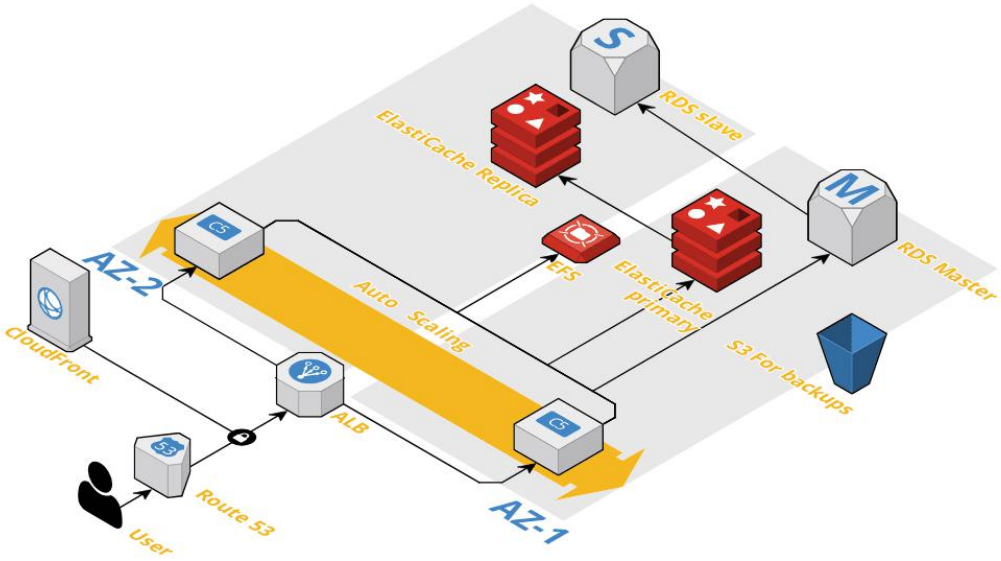
Requirements

Moving the client’ s website to a platform which is highly scalable and cost-effective. Boost database flexibility and scalability to handle peak loads.

Solution

Our team monitored the detailed performance report of the website at the time where the website ran any promotions as well as the normal days. Which gave us a basic idea about the specs. Upon analyzing the details we came to know that to reach client requirements we will have to implement the techniques such as auto-scaling, load balancing and CDN for distributing static contents. Also for the best database performance, we came up with the solution of making use of RDS service which has a great advantage in query caching. To relieve stress from the servers we are planning to use CloudFront CDN services. For user session persistence, we have offloaded the session storage to a Redis instance (elastic cache). Since our client could not tolerate any downtime we came up with an architecture which is highly scalable, secure and fault tolerant. We have implemented EFS to the original plan so that we can keep the Magento stateless so that we can overcome the issues that would have encountered with the autoscaling.

Architecture



Autoscaling and Load balancing

The server was created under an autoscaling group and the load was distributed among them using an application load balancer. We specifically went for Application load balancer since it is more advanced compared to the classic load balancer.

Infrastructure as Code

We made use of terraform because it simplifies the work in making any modifications in the infrastructure.

Monitoring

Since we wanted to keep an eye out so that we can act swiftly at the time of occurrence of any issue in the server we implemented different monitoring tools as Nagios , uptime-bot , cloudwatch & etc.

Backup

Even though our infrastructure is highly scalable and secure still there are a lot of factors that can cause issue in the application servers. Hence to be on the safer side we have implemented multiple backup policies which includes both AWS level and File system level.

Technologies Used

- Magento2
- EC-2
- ALB
- Elastic Cache
- EFS
- RDS
- Auto scaling
- CDN
- Terraform

Business Outcomes

As the result of using our architecture which is highly scalable and durable, client was able to reduce the revenue which was being spend on dedicated servers which was fully utilized only at the time of promotion runs.