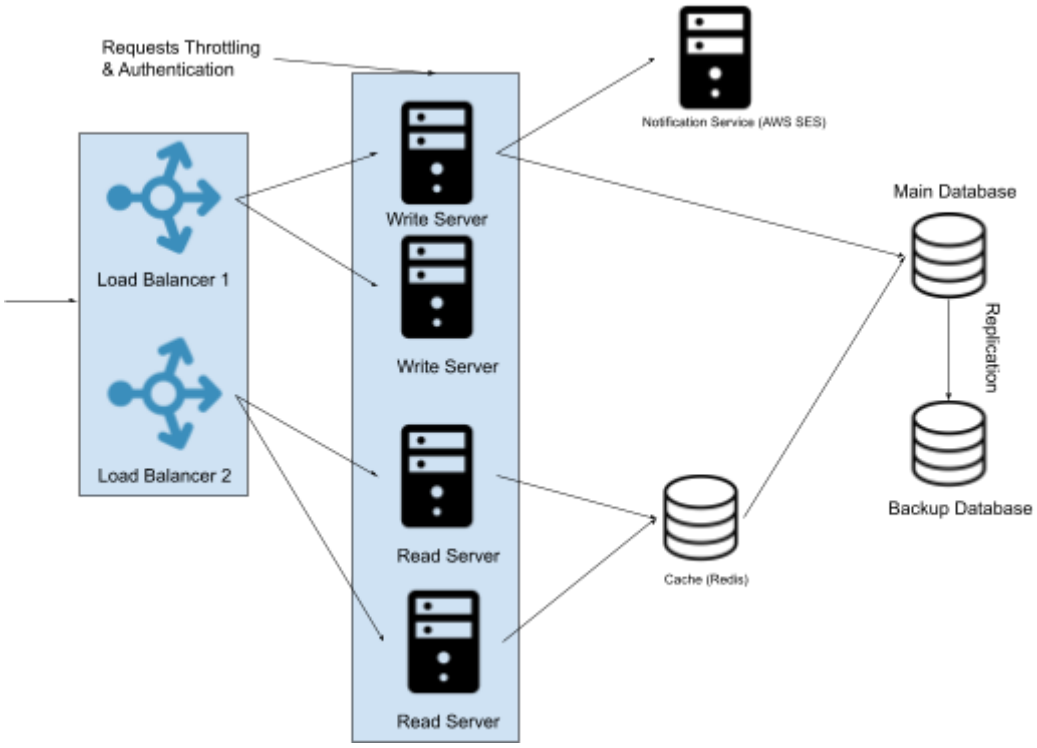
Architecture Proposal ZeBrands Catalog

For the architecture of the Zebrands Catalog Project, what I would do in order to scale the project would be to separate write and read requests in multiple read and write servers in order to apply the three principles of a High Performance Distributed System which are:

- No single point of failure principle.
- Single responsibility principle.
- No bottleneck principle.

Since we have multiple servers which have their own responsibilities separated it would reduce the latency and remove bottlenecks in case a server fails. Also, it would be a great idea to configure a Load Balancer to select between these multiple servers (which would be configured on EC2 by the way) and separate the requests between these multiple servers. In case we have an attacker in the system we should also add request throttling and authentication to prohibit an immense amount of requests and secure those requests (In the test project we configured JWT authentication for its simplicity). Also, if we want to add more types of notifications (SMS, Push, etc) we could configure a separate Notification Service (another project would need to be created) and we would have to configure the AWS SES there (right now this was added to the test project). Also, in the test project a Redis cache was configured in order to reduce the latency of database reads, however if the system scales, it would be great to perform database sharding which would be tenant-based sharding in order to add different data from different geographical locations to the DB. As usual, in scalable and high performance projects, it's important to add a Backup database replicated from the Main database, so if a DB fails we have a backup that can be used instead of the Main DB.



Author: Jorge A. Niño Cabal