

TASK2: Ankit Gupta

Statement:

We are going to develop a commercial website for selling products where:

- Each product has a number of available items in stock.
- The system should be able to process at least $N = 6000$ concurrent requests for viewing or purchasing products.
- The system is only allowed to have at most $S = 6$ servers, where up to 3 servers can be used as relational databases.
- One relational database can serve at most $C = 300$ concurrent connections.
- The system must: (main requirements)
 - ensure data consistency, which means that there will be never such a case where the number of successful purchased items is more than the number of available items in stock.
 - provide real-time feedback to its users with the lowest possible latency.
- It will be good to: (bonus requirements)
 - have the highest availability with no point of failure.
 - be easily scalable in the future when the number of products and users increase.
- Design (draw) the high level architecture of the system.
- Describe what technologies you will use to develop the system.
- Explain in detail how the system works, and how it meets all the listed requirements.
- Note that you must design and analyze based on numbers (N , C , S) provided in the statement.

Solution:

For above statement, I will be using AWS infrastructure to solve key problems.

Please find the architecture below:

- 1) All the traffic from internet will route through ROUTE53 to the Web Servers. (We can also place one ELB before Web server such that we can scale easily)
- 2) After Web servers, request will be pass on to Application servers if necessary through ELB. ELB will helps balance traffic on both servers, and will also helps scaling it afterwards.
- 3) I have created two database, one is the master and other is the replica of it. We will build the application in such a way that, all the write/update operations happen on master and rest all the read queries happens with slave.
- 4) Before request going to RDS, I have placed Elastic Cache between Application server and RDS. Elastic cache will help cache all the read request such that we reduce most of the data serving request from cache itself. When a product is sold or any data is

changed, the request will update the data in Elastic cache as well, such that there will be negligible chance of discrepancies in data.

- 5) Suppose, we got 6K request at a time, I assume that 80% is of view ones, and 20% is of purchase ones. So 4800 request are the request which has to be served to user at a time, but we have placed Elastic cache for our reading purpose. This cache can suppose reduce our db request by 70%. With this means, 1440 request are the only one which are hitting our Read replica.
- 6) Now talking about 20% of the purchase request, so 1200 request from concurrent 6000 request are actually going to write replication, which usually can handle 300 request, but in AWS we can modify DB group parameter of it to handle it.

