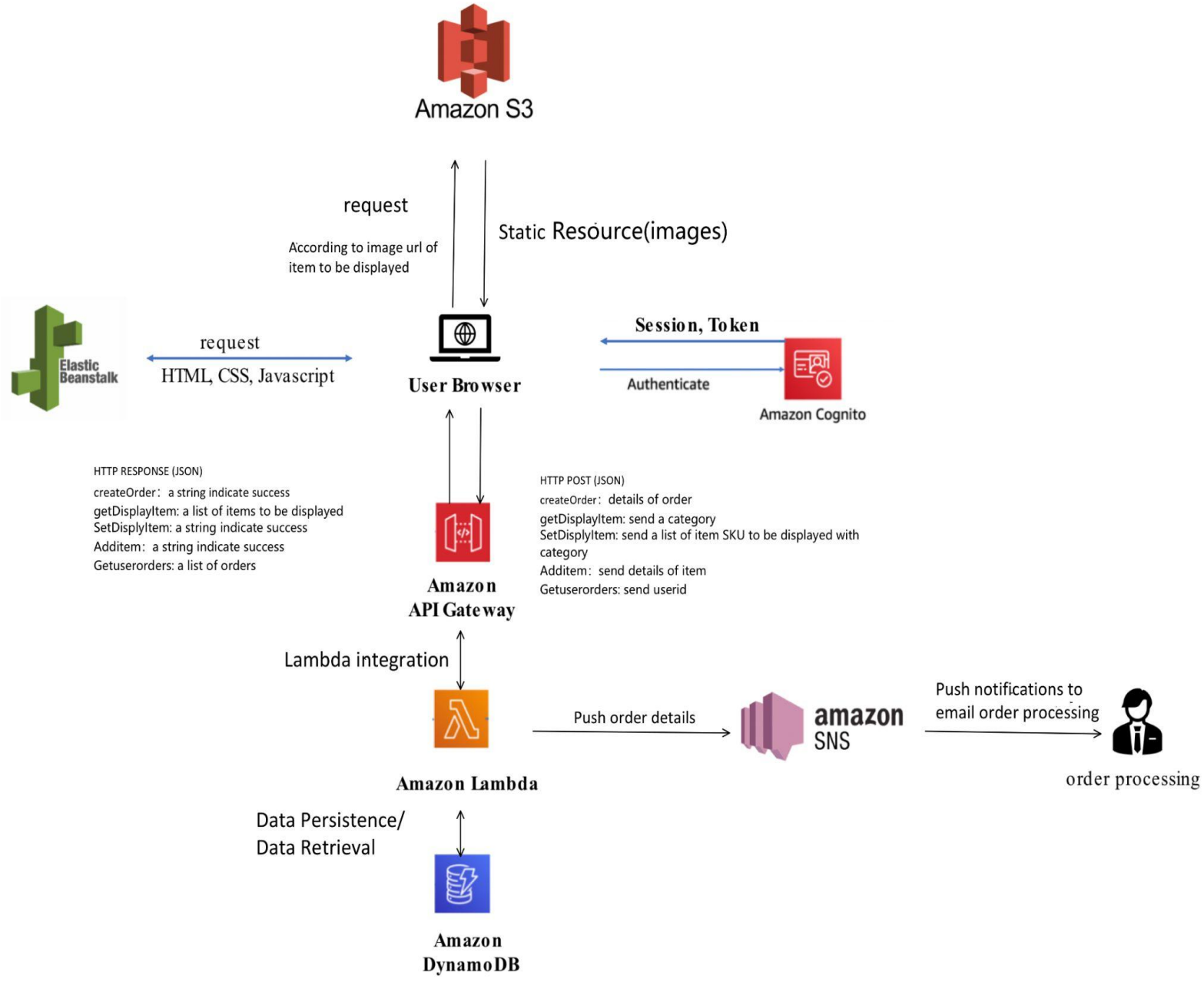
Architecture Critical Analysis and Response

Group Member:

Archer Zhou B00806294
Junqiao Qu B00817232
Kessel Zhang B00809478

Architecture of System



Important aspects:

1. Elastic Beanstalk is our most important component in our architecture. It is the platform where we host our website page, html css, and javascript. Every user should communicate with Elastic Beanstalk when opening our website. So we assume it will need to handle the most traffic and therefore, need the highest availability in our system.

2. API Gateway is our second most important component in our architecture. It is called to process every request made by registered users. Most of our functions, like creating orders, retrieving items' information, will need to access the API in this service. We assume this service is required to handle a large amount of request, therefore need a high availability as well.

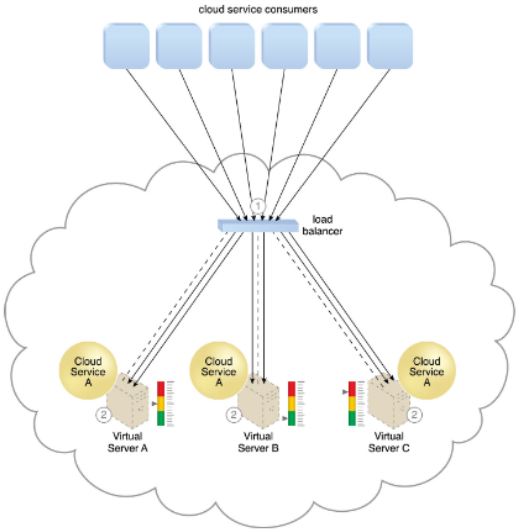
Architecture Comparison

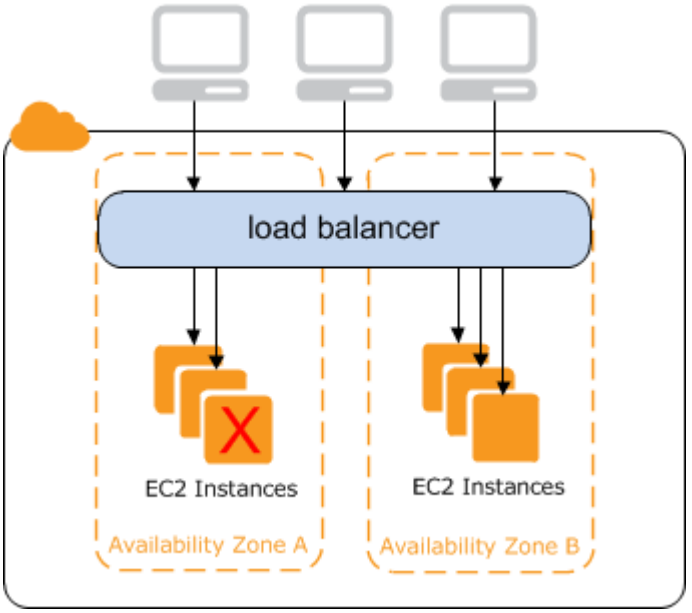
Similar Architecture:

The most similar architecture to our project in class is service load balancing. We use the Elastic Beanstalk for our website page (html, css and javascript). In the Elastic Beanstalk, there is a built-in load balancer in the environment . We use a Classic Load Balancer and it will distribute traffic and route HTTP, HTTPS request traffic to different ports on EC2 instances [1]. It is very similar to the definition of service load balancing architecture, a specialized variation of the workload distribution architecture that was designed specifically for scaling cloud service implementations. Redundant deployments of cloud services are created, with a load balancing system added to dynamically distribute workloads [2].

Differences:

According to the document, a load balancer in Elastic Beanstalk distributes incoming application traffic across multiple EC2 instances [1]. An EC2 instance is a virtual server. So We think there is no difference between our architecture and service load balancing architecture.





Architecture Analysis

1. **Rapid Provisioning Architecture** is more performant architecture than ours. The architecture establishes a system that automates the provisioning process. When a demand is going to increase significantly, like the release of a limited edition, we can use this architecture to quickly provision a wide range of IT resources. It will enable us to handle higher but unusual demand and traffic, therefore it increases the performance of our application.

2. **Cloud Balancing Architecture** is providing higher availability than ours because it utilizes multi cloud to prevent failure. It will have higher availability than our load balance architecture using a single cloud. Besides, cloud balancing architecture provides a failover system that adds resiliency [3], if a failure happens, the traffic can be rerouted to another cloud. It is an architecture providing almost always-on service and more stable compared to our architecture.
3. **Dynamic Scalability Architecture** should theoretically show better scalability, it could scale up and down (when more compute capacity is needed) or scale out and in with workload [2]. This architecture could dynamically relocate IT resources, it is flexible when the IT resources need to be changed.

Reference:

[1] “Load balancer for your Elastic Beanstalk environment” AWS Elastic Beanstalk Developer Guide[Online]. Available: https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/using-features.managing.elb.html [Accessed: 20-Mar-2022].

[2] R. Hawkey, Class Lecture, Topic: "Fundamental Cloud Computing Architectures”, Faculty of Computer Science, Dalhousie University, Halifax, Mar. , 04, 2022.

[3] R. Hawkey, Class Lecture, Topic: "Advanced Cloud Computing Architectures”, Faculty of Computer Science, Dalhousie University, Halifax, Mar. , 11, 2022.