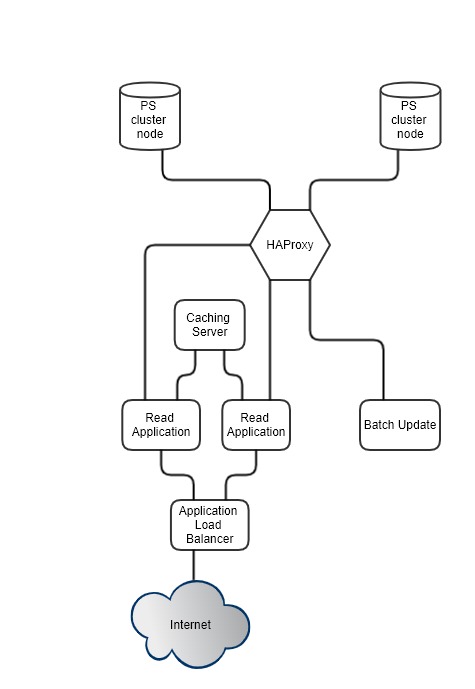
### **Extended service**

* 2.1.1) **How would you design the system?**

For handling heavy load with high availability for the DB operation we can upgrade postgress sql instance into a cluster and application can connect with PS Cluster through the “HAProxy” (Load balancer).

We can split Read/Write operation into two different application. Hence, we can scale read write operation independently. We can use caching server to minimize the DB calls. Hence bottleneck will reduce. We can scale our read application horizontally based on the demand through the **application load balancer** (eg: nginx) the high availability.

We can use PS transaction feature to confirm either all items go in, or none of them do in batch insert.

**BEGIN;**

Batch Insert query;

**COMMIT;**

2.1.2) **How would you set up monitoring to identify bottlenecks as the load grows?**

We can configure NGINX (Application load balancer) to HTTP Health Checks

2.1.3) **How can those bottlenecks be addressed in the future?**

Using the **caching server (Memcached, Redis,…),** we can reduce the bottle for the heavy read operation.

**2.2.1)** **The batch updates have started to become very large, but the requirements for their processing time are strict.**

We can use python multithreading

We can use shell scripting threading (commands end with &)

**2.2.2) Code updates need to be pushed out frequently. This needs to be done without the risk of stopping a data update already being processed, nor a data response being lost.**

We can design the system with below features,

* Take the PS database out from the docker container and host in AWS PS cluster. Then we can deploy code changes only using docker container.

**2.2.3) For development and staging purposes, you need to start up a number of scaled-down versions of the system.**

**?**

**Please address *at least* one of the situations. Please describe:**

* Which parts of the system are the bottlenecks or problems that might make it incompatible with the new requirements?

?

* How would you restructure and scale the system to address those?

?