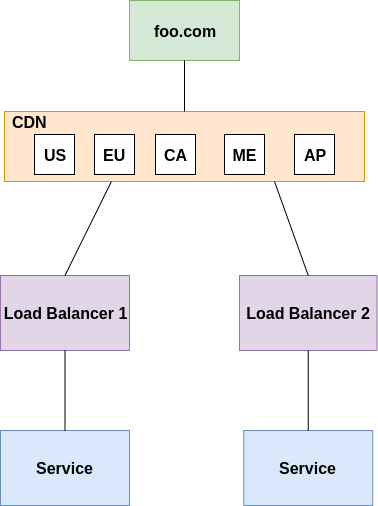
1. We have a high availability application running with the following Architecture.



* **Explain the Architecture model of this and the Benefit**

I think this is the active active architecture model for a high availability application like foo.com where same replicas of the services are running actively and traffic is split between these replicas based on load balancer fanout mechanism and then the service is offered to the end user based on the request placed

Some benefits of such kind of architecture is that

a) no single point of failure - Even if one service instance fails at a particular reigion, other one can drive the traffic

b) better traffic control: if certain regions have higher traffic/or have other regulatory requirements/or is having a new release features, traffic from that can be driven to the required service with this approach

* **How do you roll out an update to the service without any downtime impact to the app foo.com**

We could do a one at a time roll out for this type of architecture. First we remove one service from the load balancer and perform the update while the other service serve the customer requests. Once the update is done it can be put back to serve the traffic and take out the next service for the update.

* **If you have to deploy this App to the Cloud what services would you use and explain the Architecture. You can mention for Cloud of your choice (AWS / Amazon / Google Cloud (GCP) / Oracle / Alibaba Cloud)**

If we want to deploy this application architecture to AWS, first we need to use route 53 for DNS mapping and then use CloudFront for content delivery network on AWS. Later the traffic can be controlled using an elastic load balancer. We could use EC2 instances as compute engines to run the services on cloud and the resulting data could be stored on S3 buckets or in databases like dynamo DB.