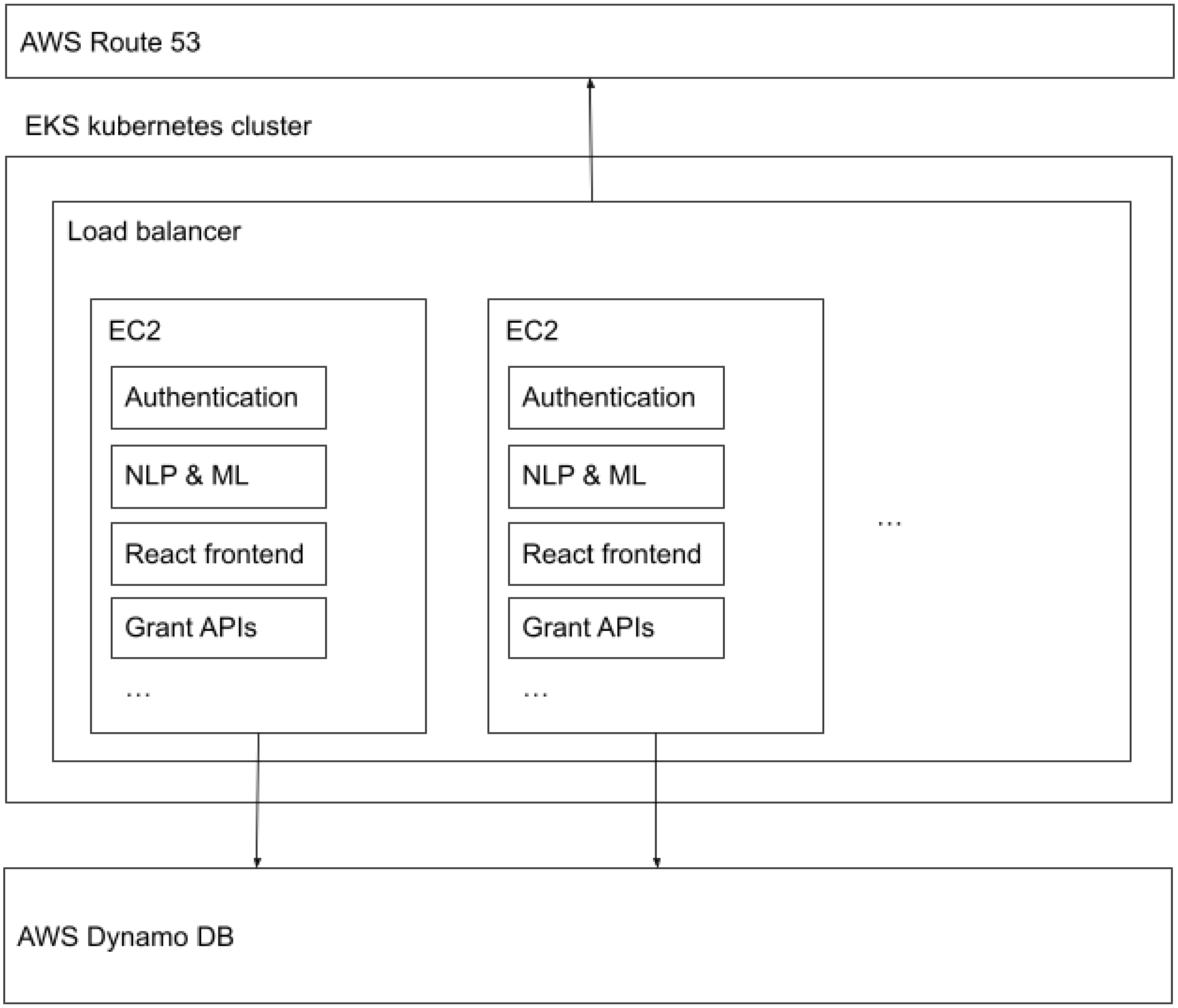
Proposed Architecture



- It is recommend to make use of a Kubernetes cluster with the AWS service EKS (Elastic Kubernetes Service). This way, we can keep the different features of the web application isolated in separate containers, making collaboration on different features very easy. Using EKS, we can orchestrate the containers to create the entire solution.
- The cluster should autoscale to create as many EC2 instances as needed, but perhaps start with 2 and implement a load balancer to manage the traffic between the two instances.
- The nodes within the cluster should make use of EC2 instances, which are readily available and reliable resources for running web applications.
- Within each EC2 node, we can run pods containing the containers that are needed to run the application. This would include authentication, NLP & ML, the React frontend, APIs to fetch grant information and also to analyse whitespace, etc. Using a containerised architecture keeps things very modular and we can easily add new features.
- The containers can also be stored using AWS ECR (Elastic Container Repository)

- DynamoDB would be a good choice to use as a database, since this is AWS’s own document store database offering.
- Finally, AWS Route 53 can be used to route traffic from a domain or subdomain name to the load balancer, so that users can access it through a web browser.