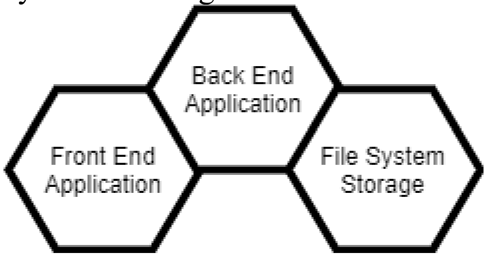
Hangman Web Sockets Project

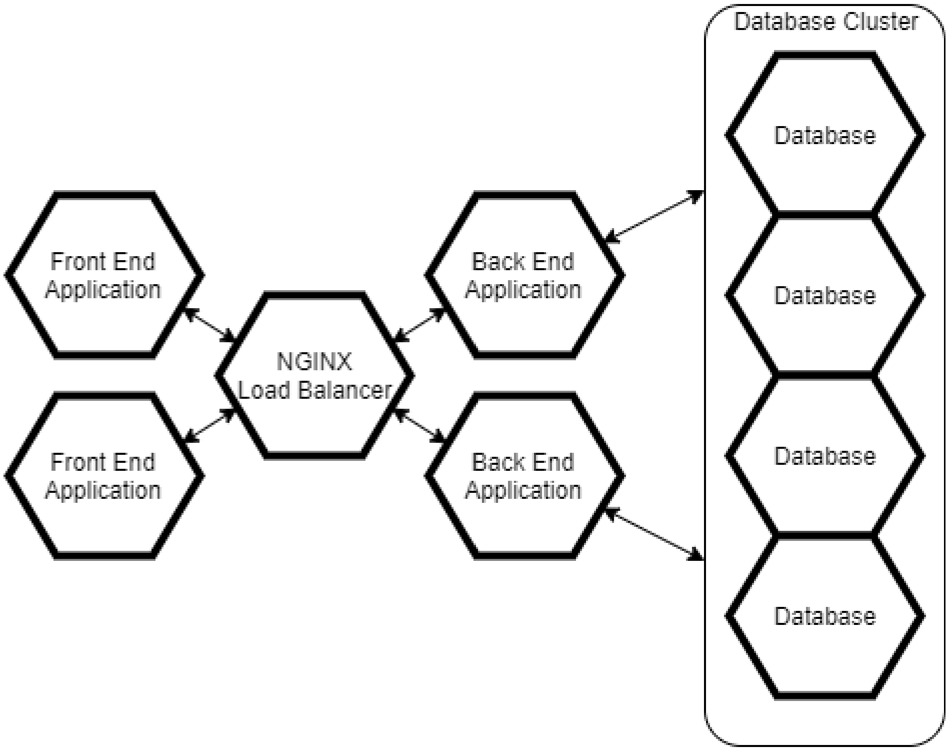
Current Status

It's currently composed of 2 parts, front-end (HTML, JavaScript, CSS) and back-end (Java Spring) in the same application and use file system as storage.



How To Scale?

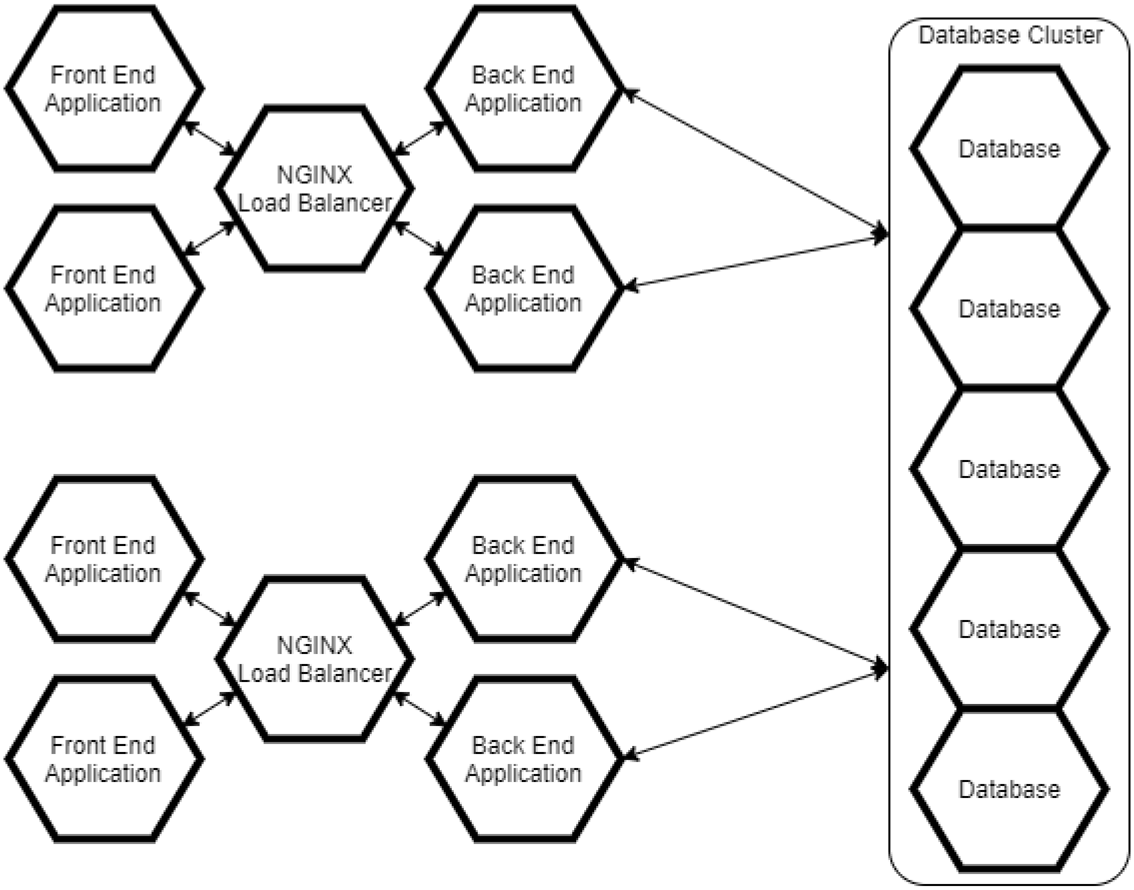
The front-end and back-end can be separated in separate servers and into 2 projects, a database will be required as well instead of file system storage, we could add a NGINX server between the front-end and back-end to work as a load balancing allowing requests to be distributed and reduce overheat, at the same time the database can be implement a cluster setup, this enables our application to work in a highly available environment.



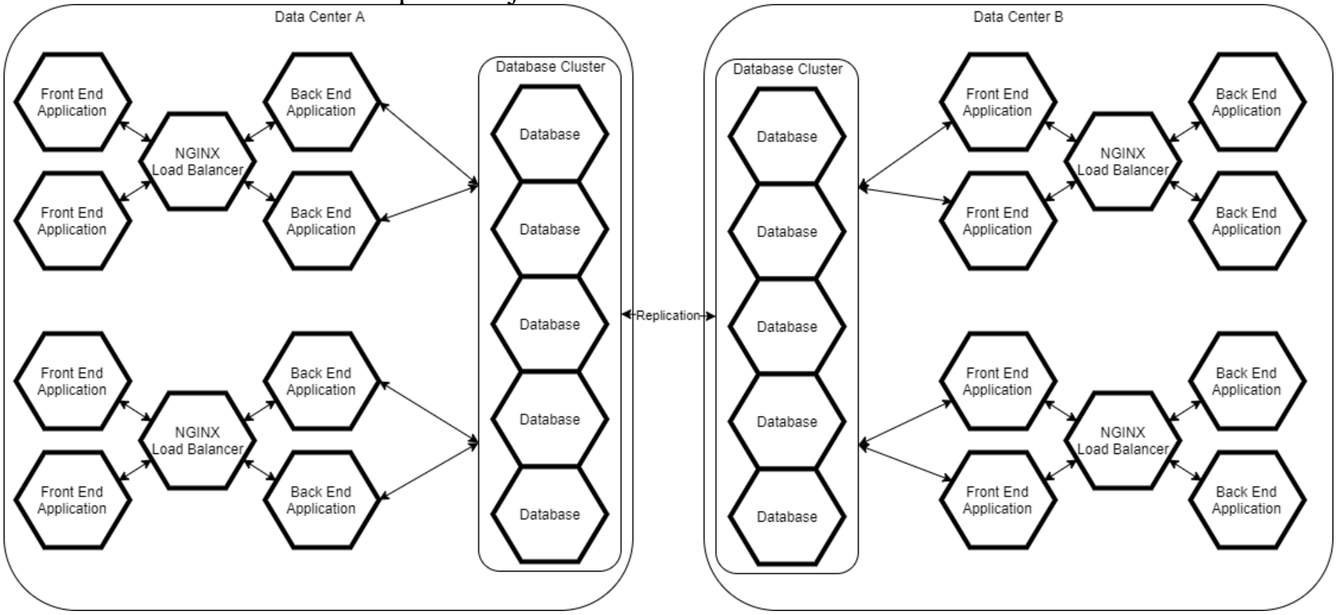
The advantage of this setup is that we could add or remove any of the modules and our application will keep working due to replication and synchronization as long as one instance of the module remain online, this adds flexibility in scalability, if we need to add or remove modules due to a high demand, we could do it without delay by using docker containers for example.

Scaling To Multiple Data Centers

Expanding across the world requires the data to be reachable in acceptable time, the data centers allow to have access to this data in time and form, as well provides the creation of contingency plans to prevent down time in case of disasters.



Using a containerized environment provides an easy way to deploy the same VM configurations by running scripts that will setup the exact same environment, so in theory is relatively easy to deploy a full new data center with a couple of adjustments.


Data Center A
Front End Application
Front End Application
NGINX Load Balancer
Back End Application
Back End Application
Front End Application
Front End Application
NGINX Load Balancer
Back End Application
Back End Application
Database Cluster
Database
Database
Database
Database
Database
Replication
Data Center B
Database Cluster
Database
Database
Database
Database
Database
Front End Application
Front End Application
NGINX Load Balancer
Back End Application
Back End Application
Front End Application
Front End Application
NGINX Load Balancer
Back End Application
Back End Application