Disclaimer

I am a rookie developer and have never documented my code ever. So, you know what I mean. This documentation might be one of the stupidest documentations to come across in your career. Kind apologies for that.

The attached code is a solution for the **Cloud Agonistics problem.**

Which has two modules one being the front end and the other is exactly what you think.

**How I perceived the problem before I started to build the solution?**

I perceived it in a way that, there should be a REST API gateway which should be able to integrate with any existing applications providing access to store, retrieve, update and share files across that platform.

I believe that this solution is for one organisation alone running the application and not like a SAAS.

Also, most importantly it should support multiple cloud providers with some security features to secure the endpoints.

**How the solution is structures?**

The solution has a react application as a mock UI to test the endpoints and also an express.js server which hosts the REST API’s.

The solution is based on the scenario that an organisation with group of people is require a solution to access files across multiple cloud providers seamlessly on one platform and should also be able to do the primary functions related to the functions.

The people who have access to the platform (i.e., people who have account in the system can store, retrieve, delete or share files among themselves).

The detailed explanation is available separately on the documentation folder.

Some basic amounts of the computations are carried out within the frontend itself like the delete function will only be available on the file owner’s dashboard, so, extensive validation of the person who is trying to delete the file can be removed. Also, the autogenerated ID required for the deletion of the file are not even available to the owner, its all attached in the background with the delete request and sent to the API.