FILE STORAGE

By

MAMATHA GUNTU

ABSTRACT:

This project demonstrates a web application, where users can register and sign-In to upload files and perform other operations such as edit, delete and download.

Prerequisites:

* Aws account
* Technical Details :
  + Front end: React (Visual studio code)
  + Back end: Spring boot (Spring Tool Suite)

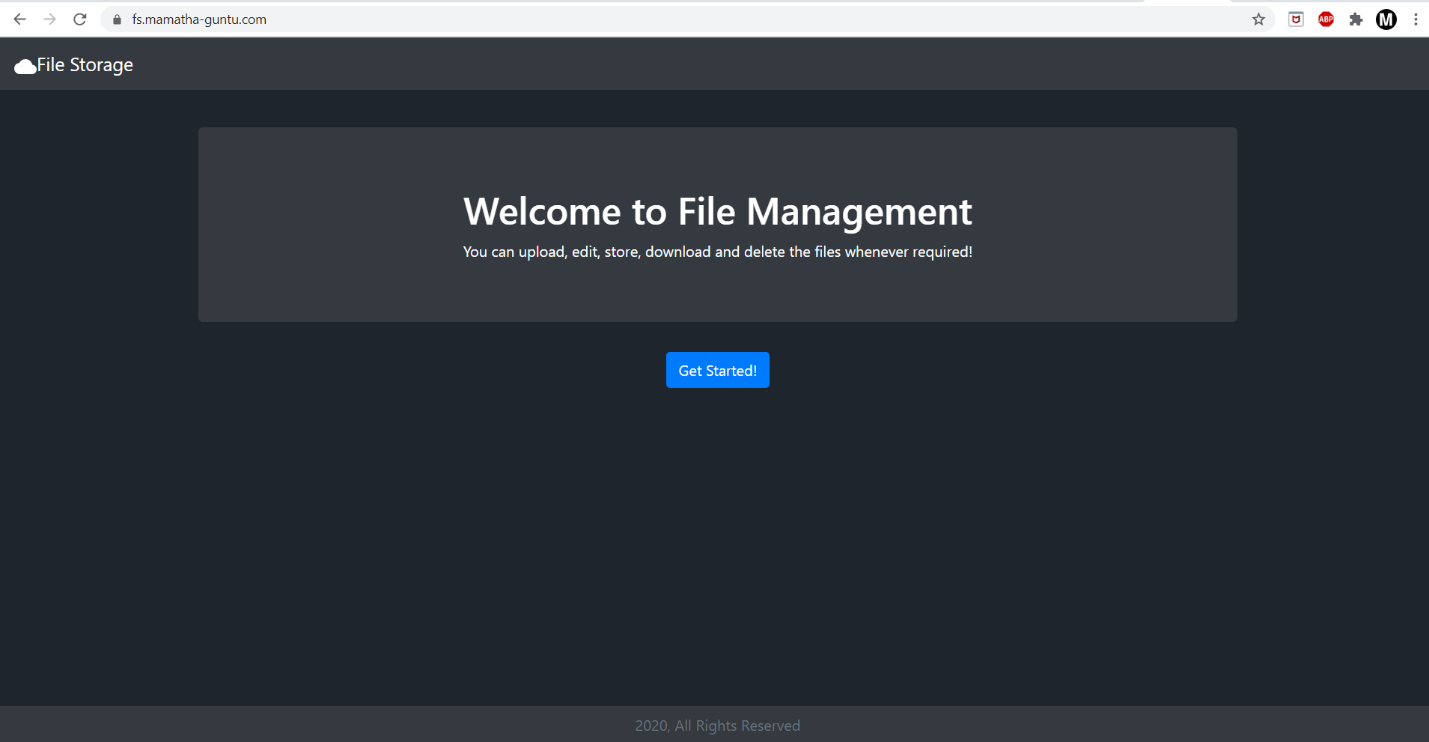
GitHub link: <https://github.com/mamathaguntu/281FSProj/blob/main/WebApp/src/Components/S3ServiceAPI>

## Website link:

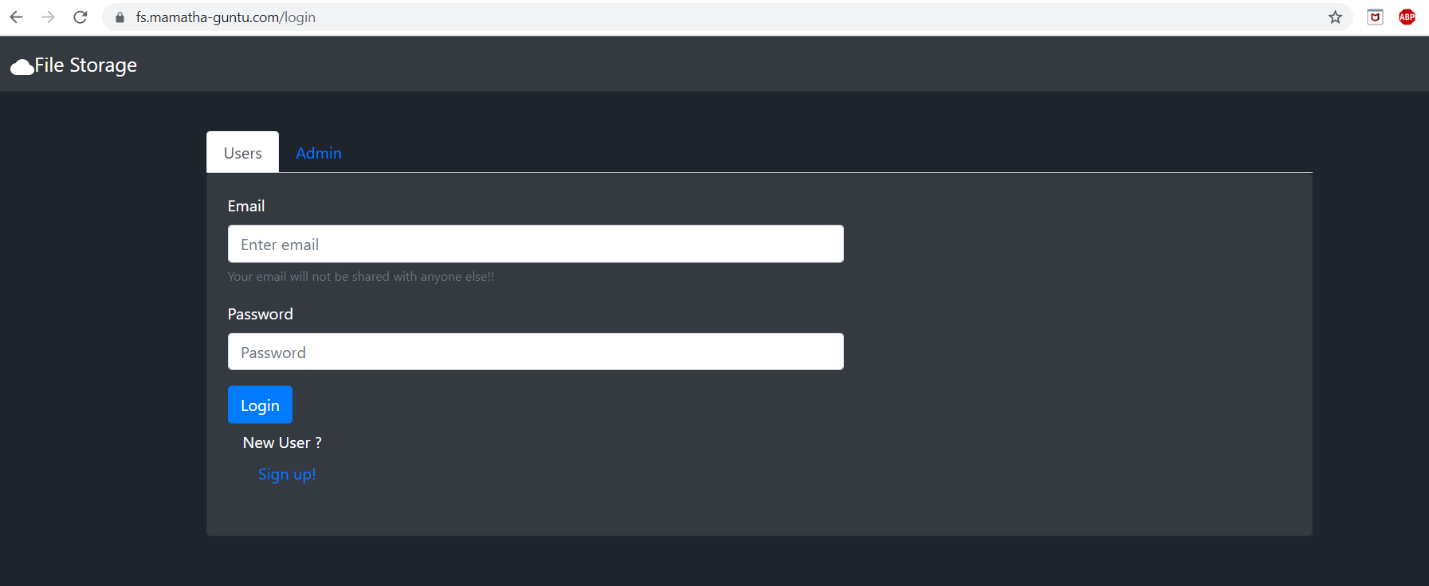
https://www.fs.mamatha-guntu.com

## Demo screenshots:

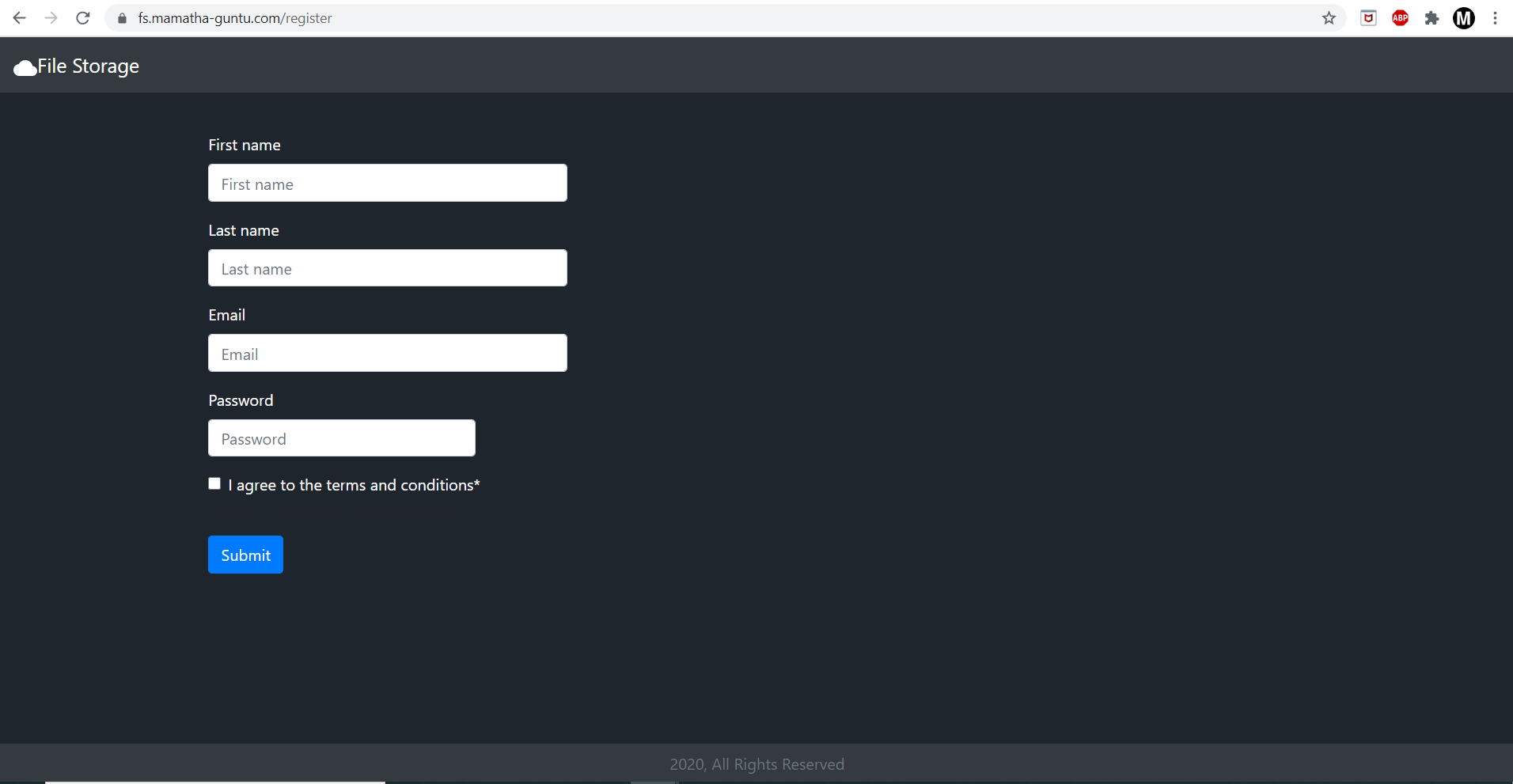
Welcome page:

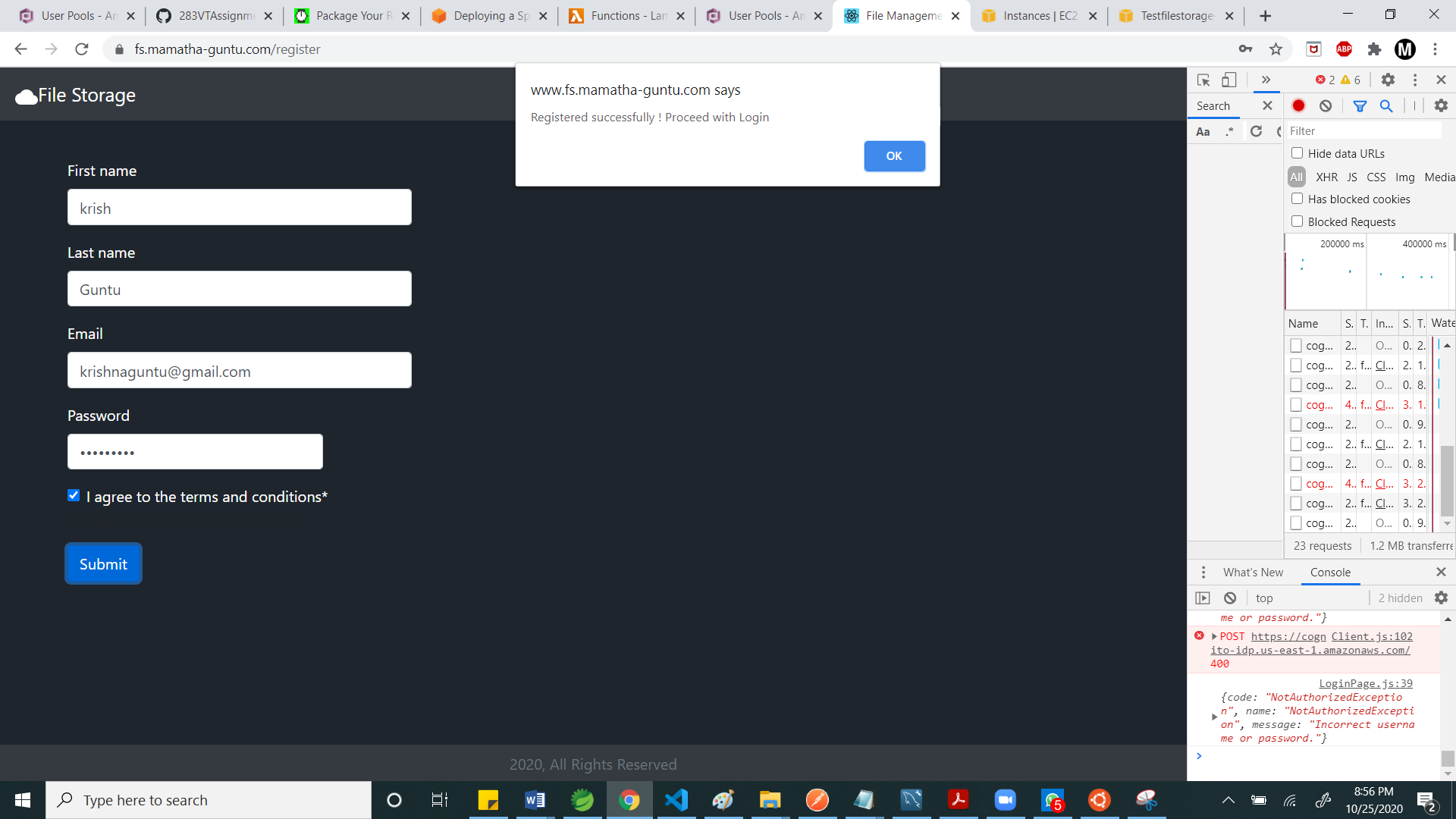


Click ‘Get started!’ and you will be redirected to Login Page

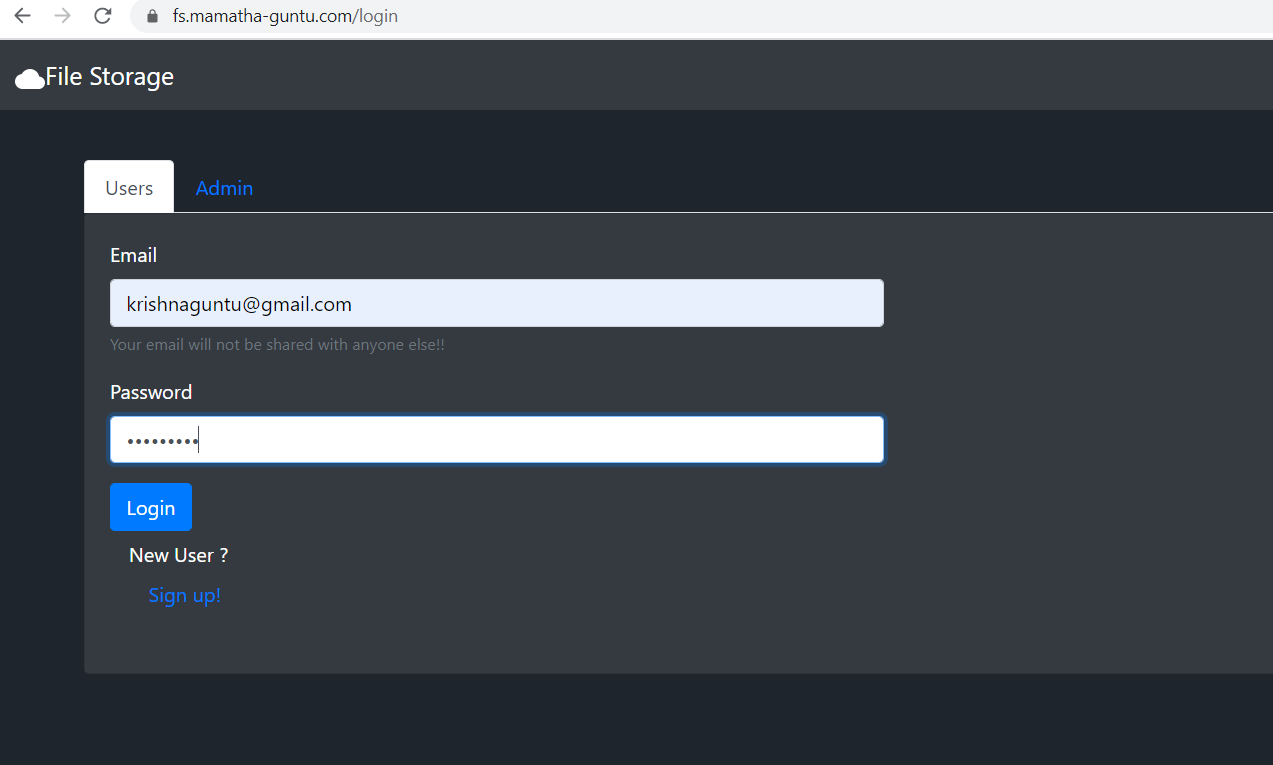


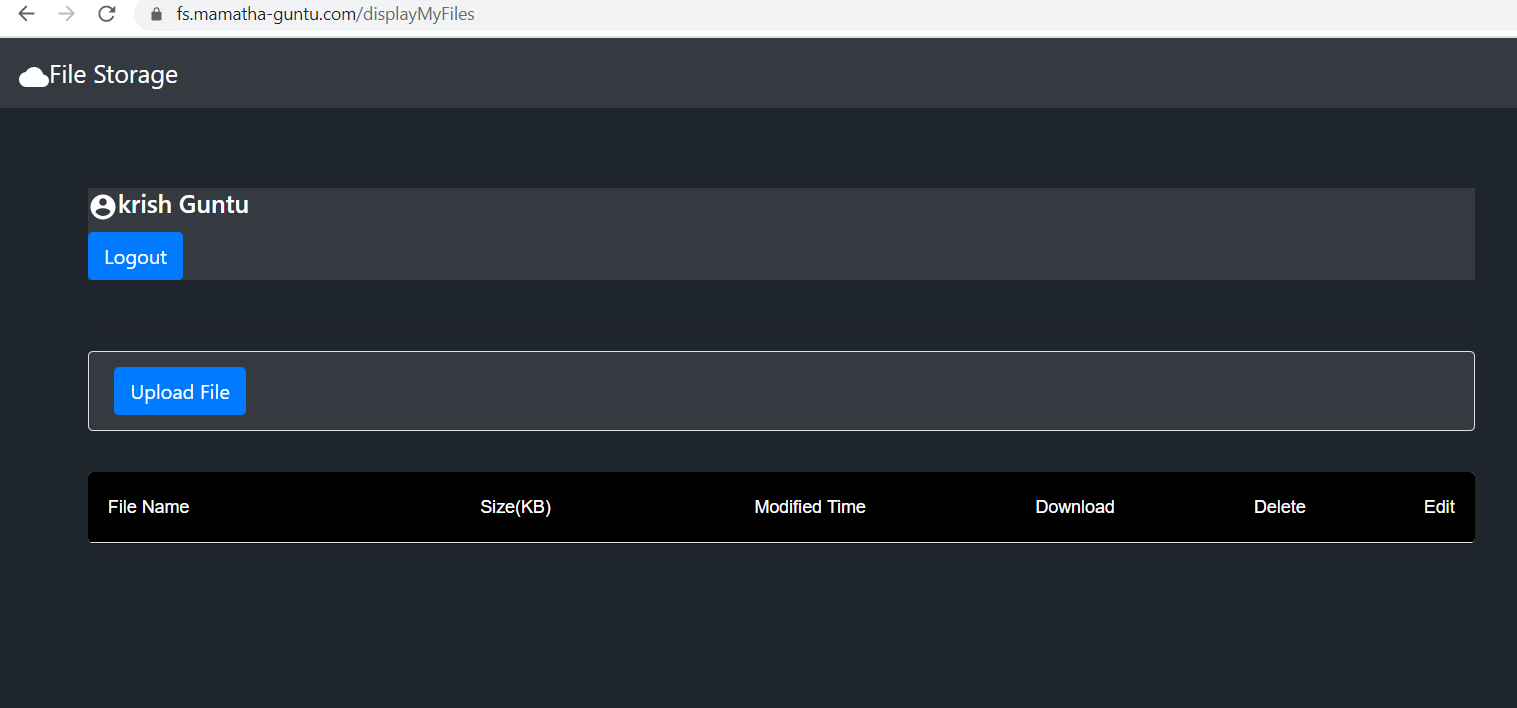
Click ‘sign-up’ to register the account



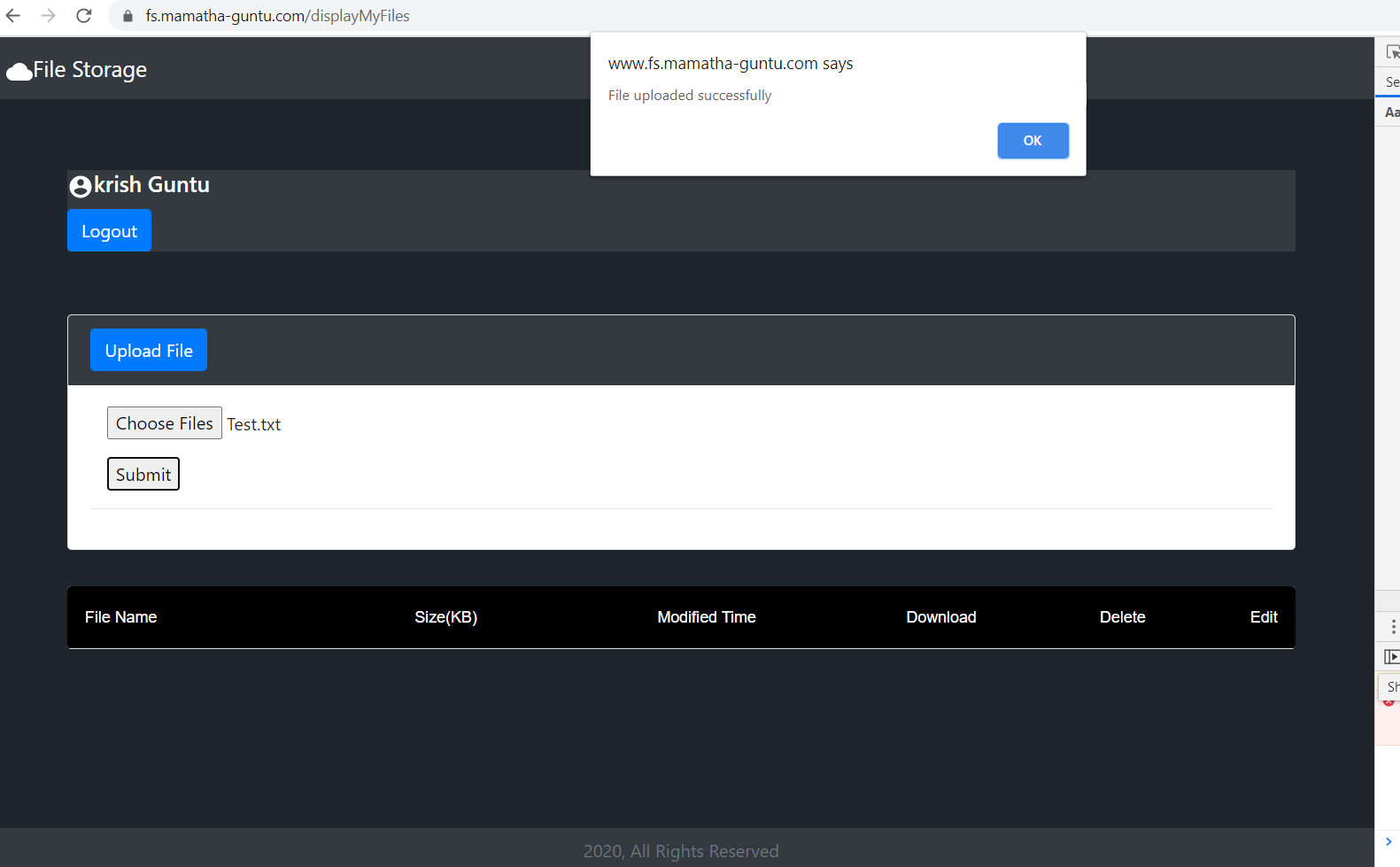
Fill in the details and click submit

Click ok and the page will be redirected for login

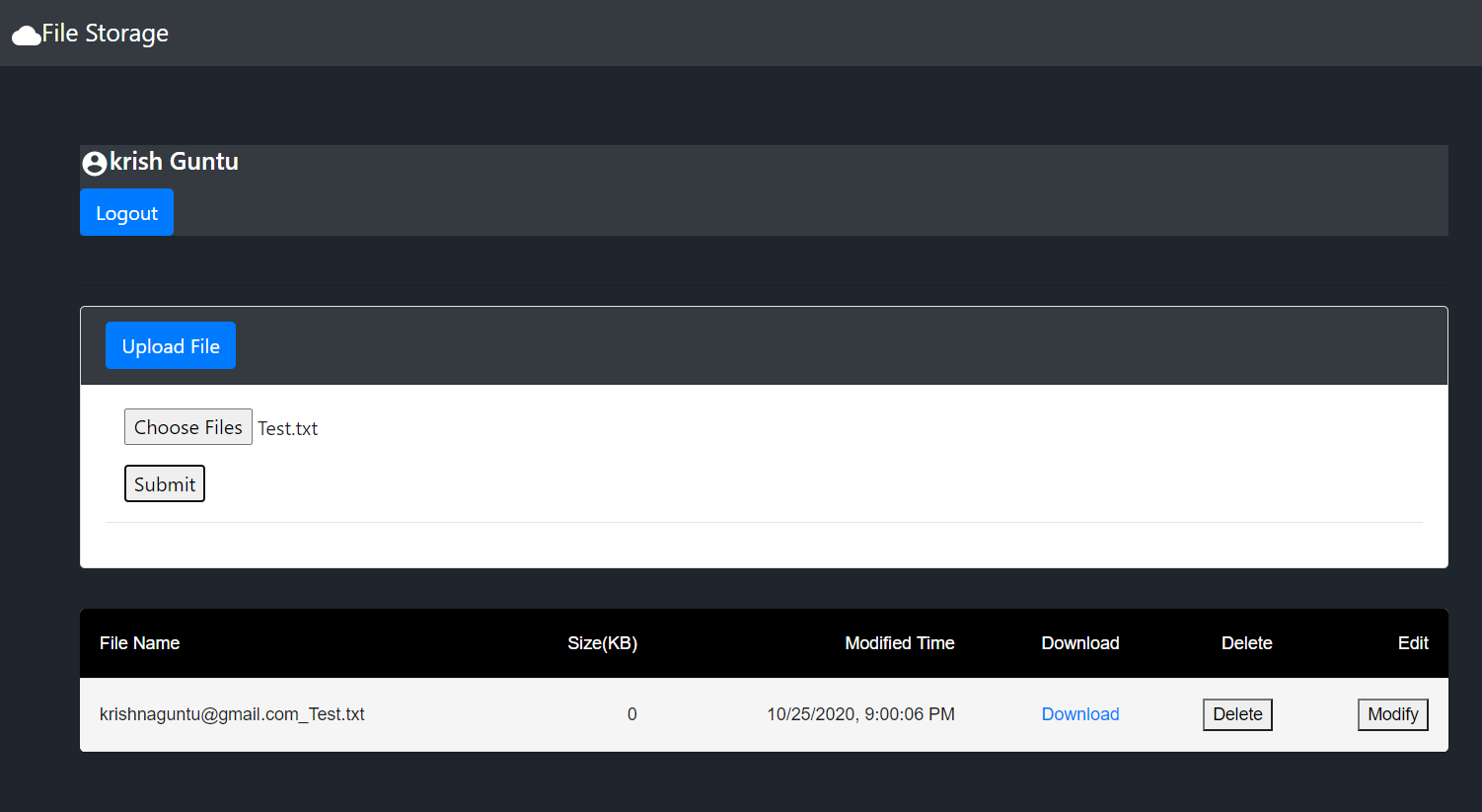




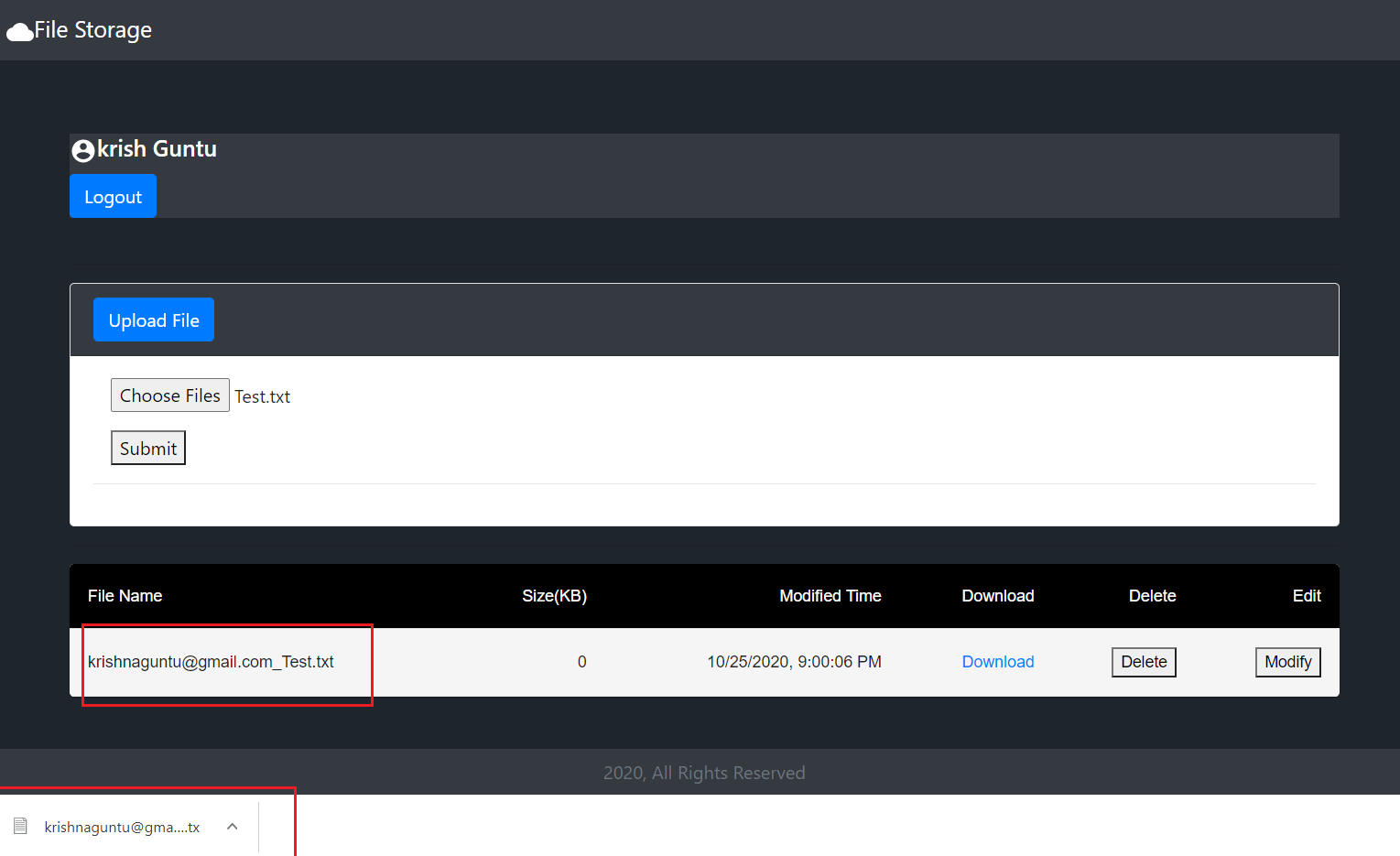
Upload a file



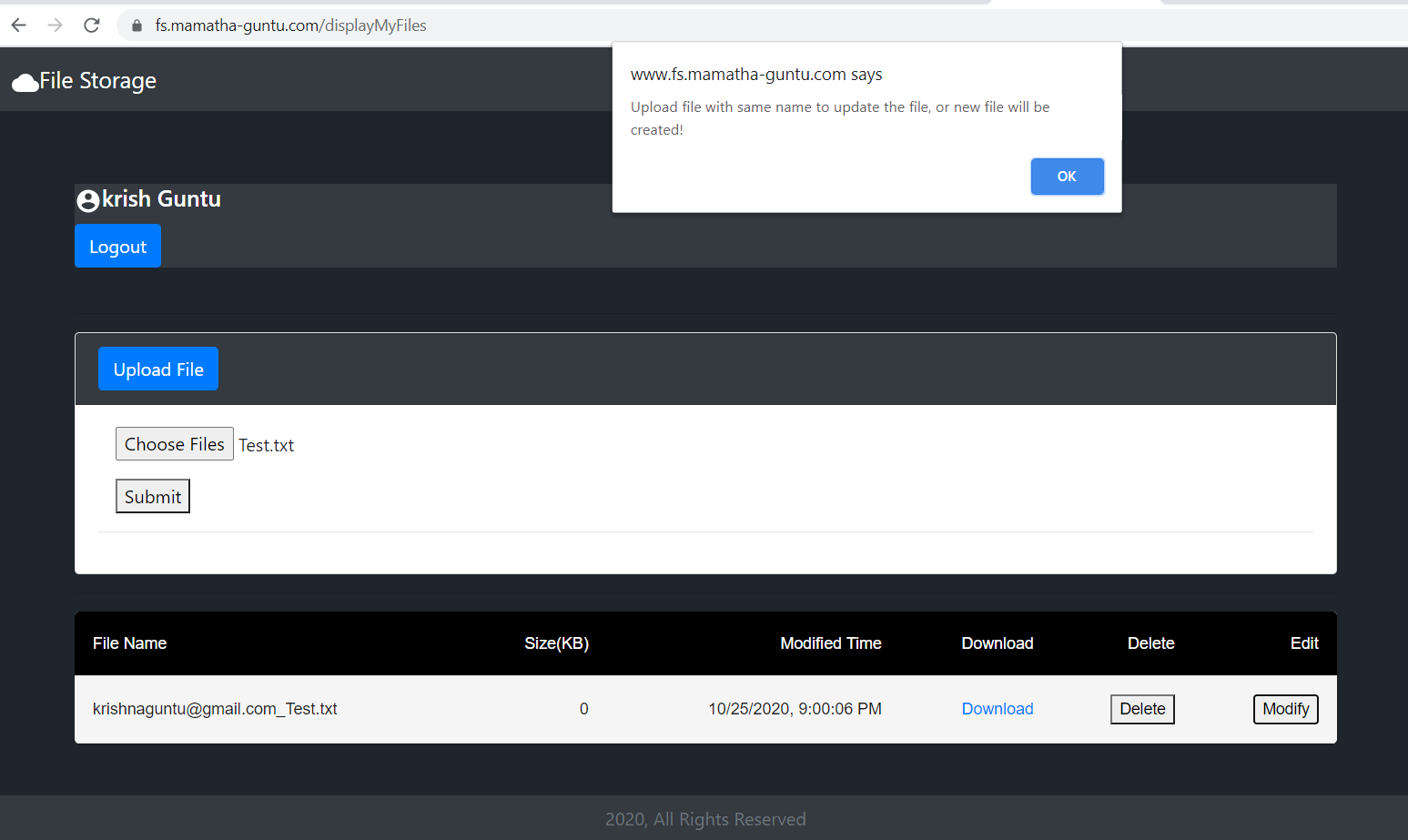
Verify Uploaded file is displayed in the Grid



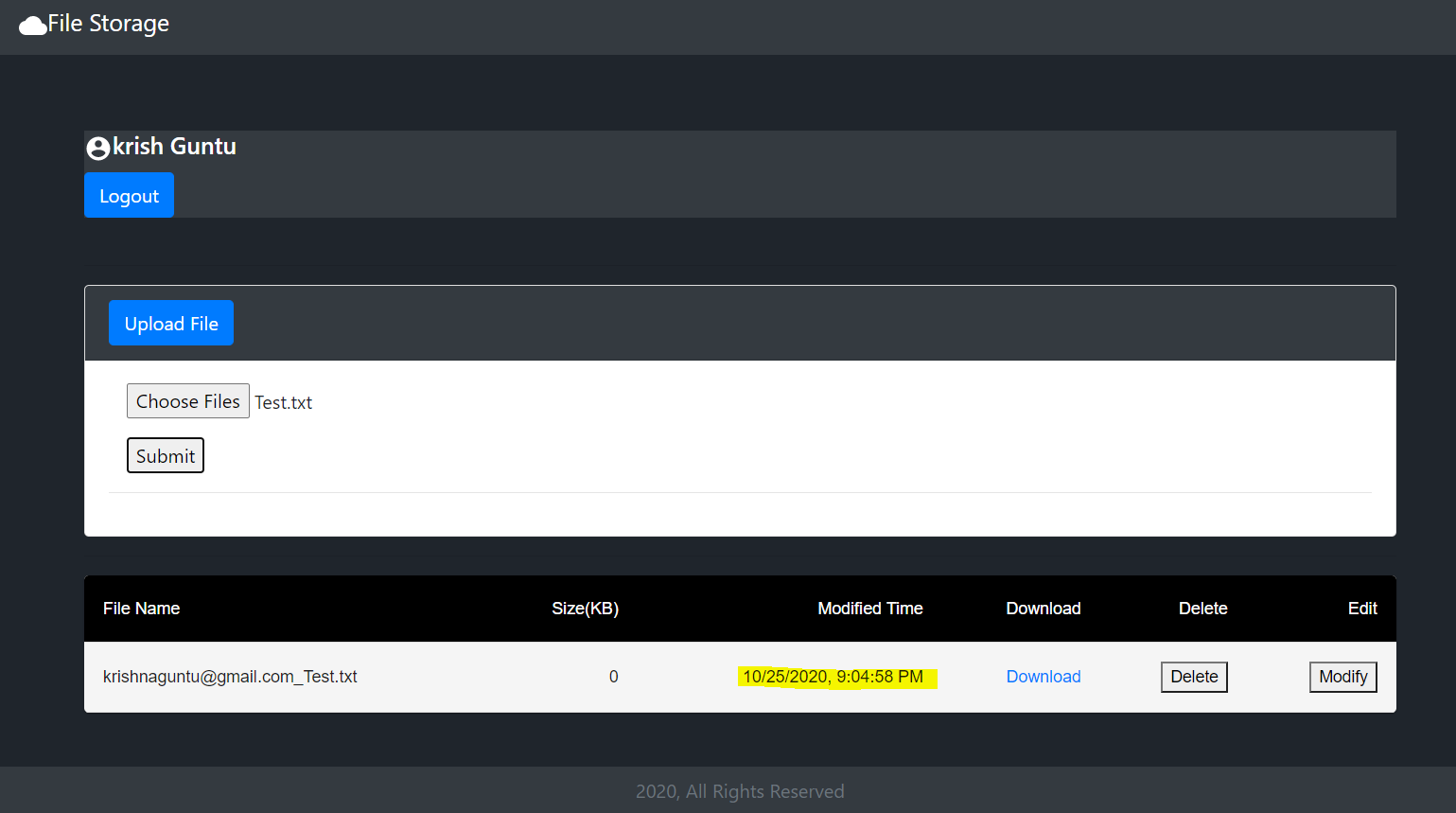
Download the file



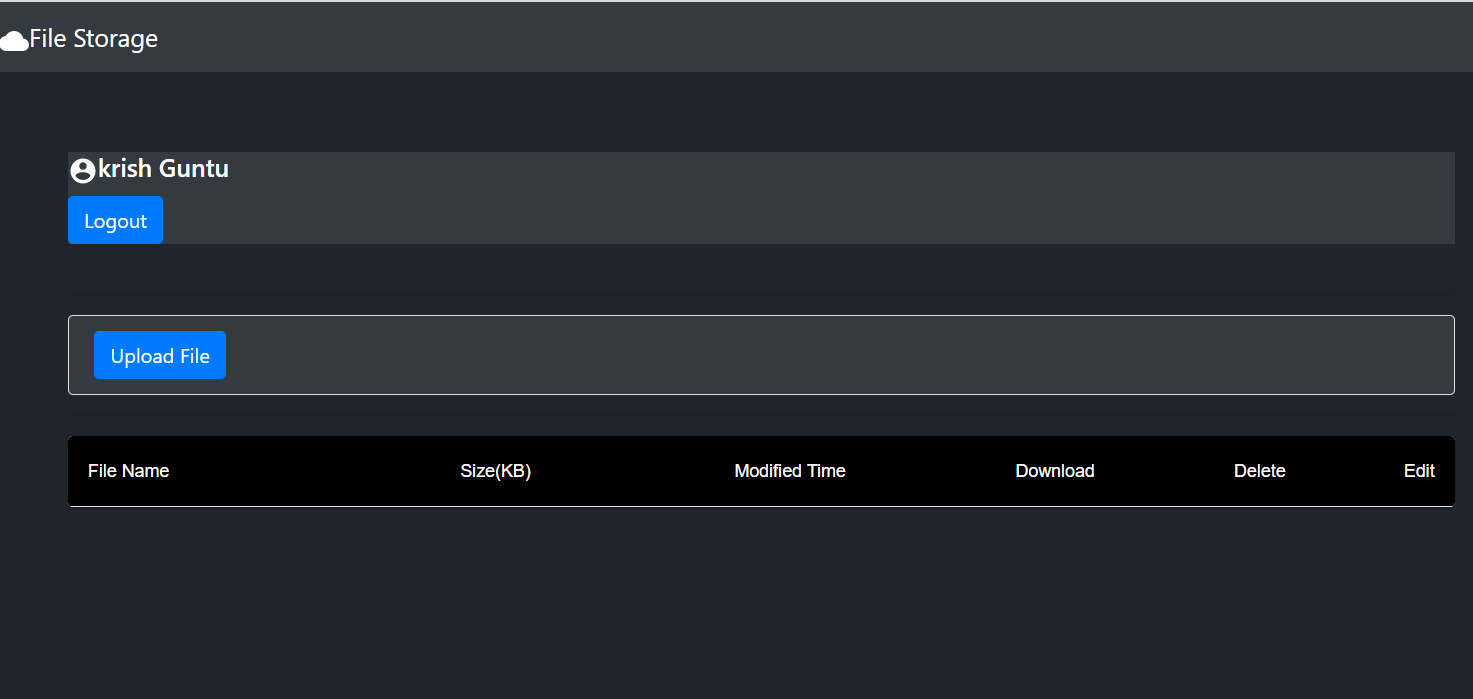
Modify the file



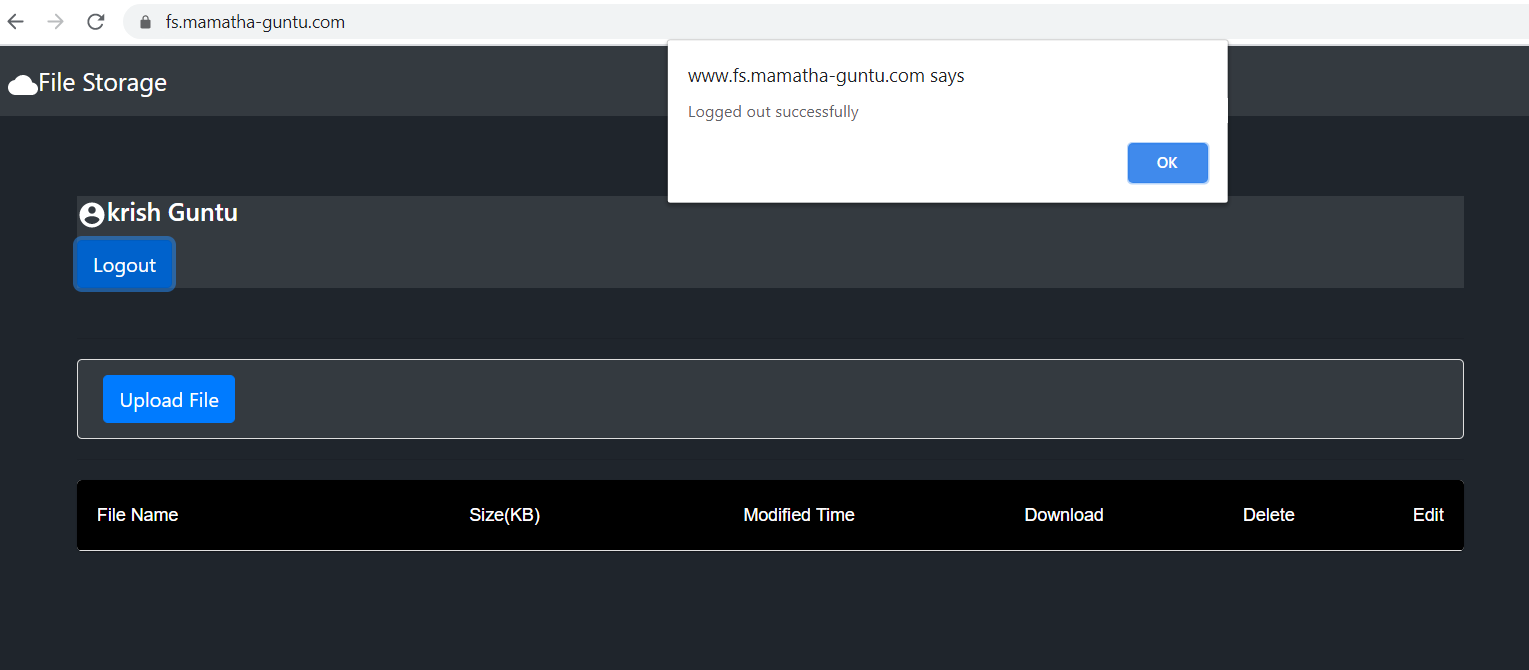
Upload Modified contents file with the same name, else a new one will be created. This is to maintain version control for the file.

After update , Observe that the date and time is changed

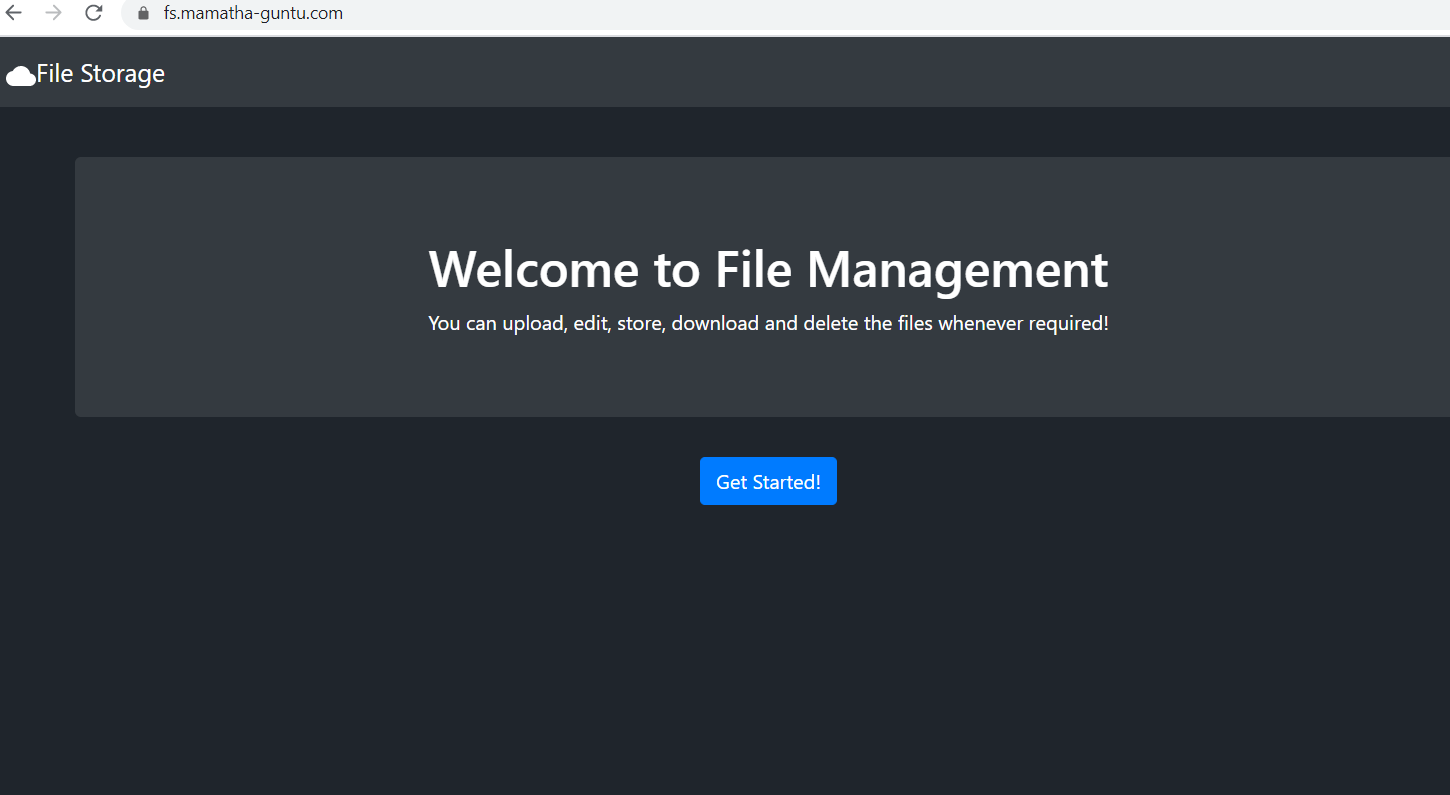
Click delete and you can see the file getting deleted



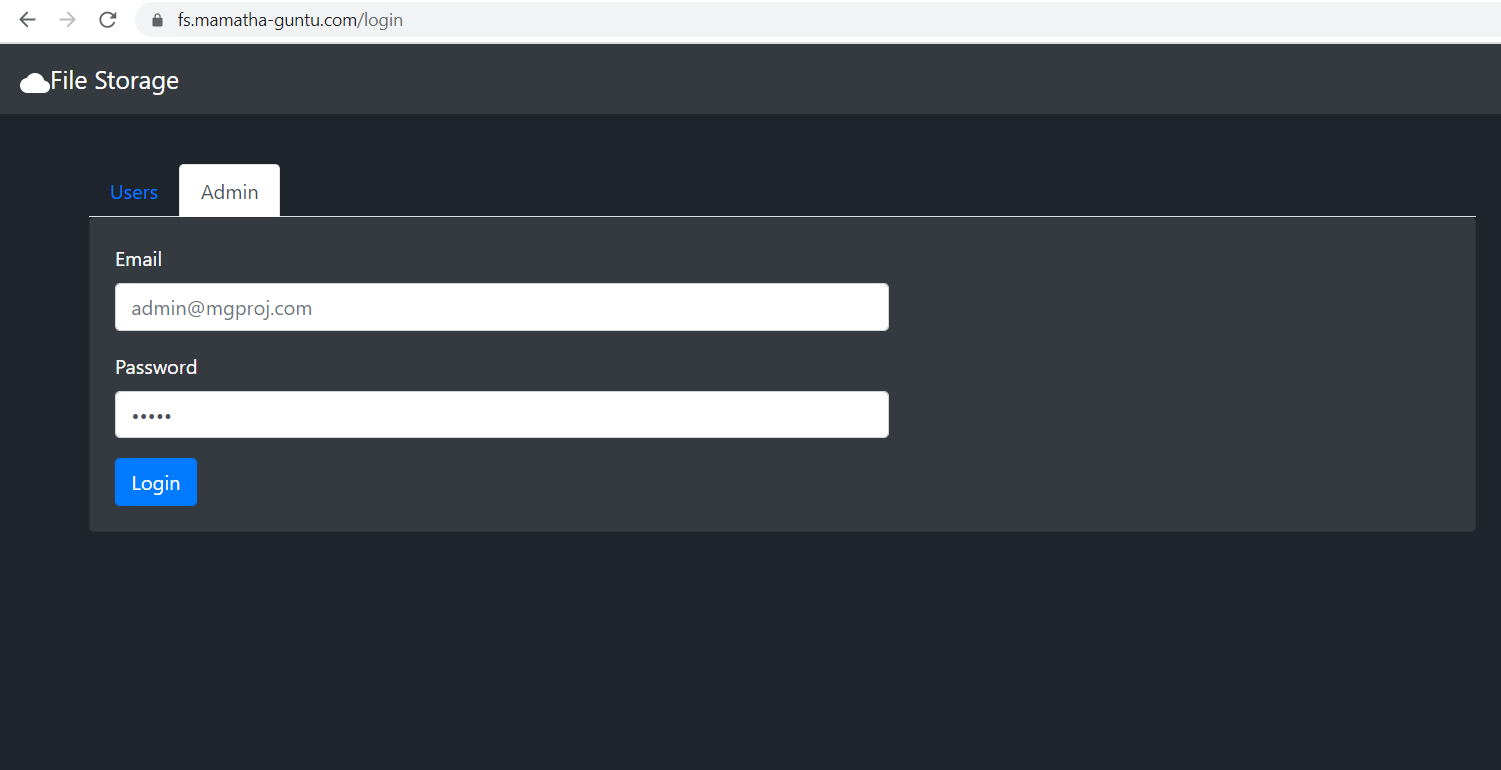
Log out of the account



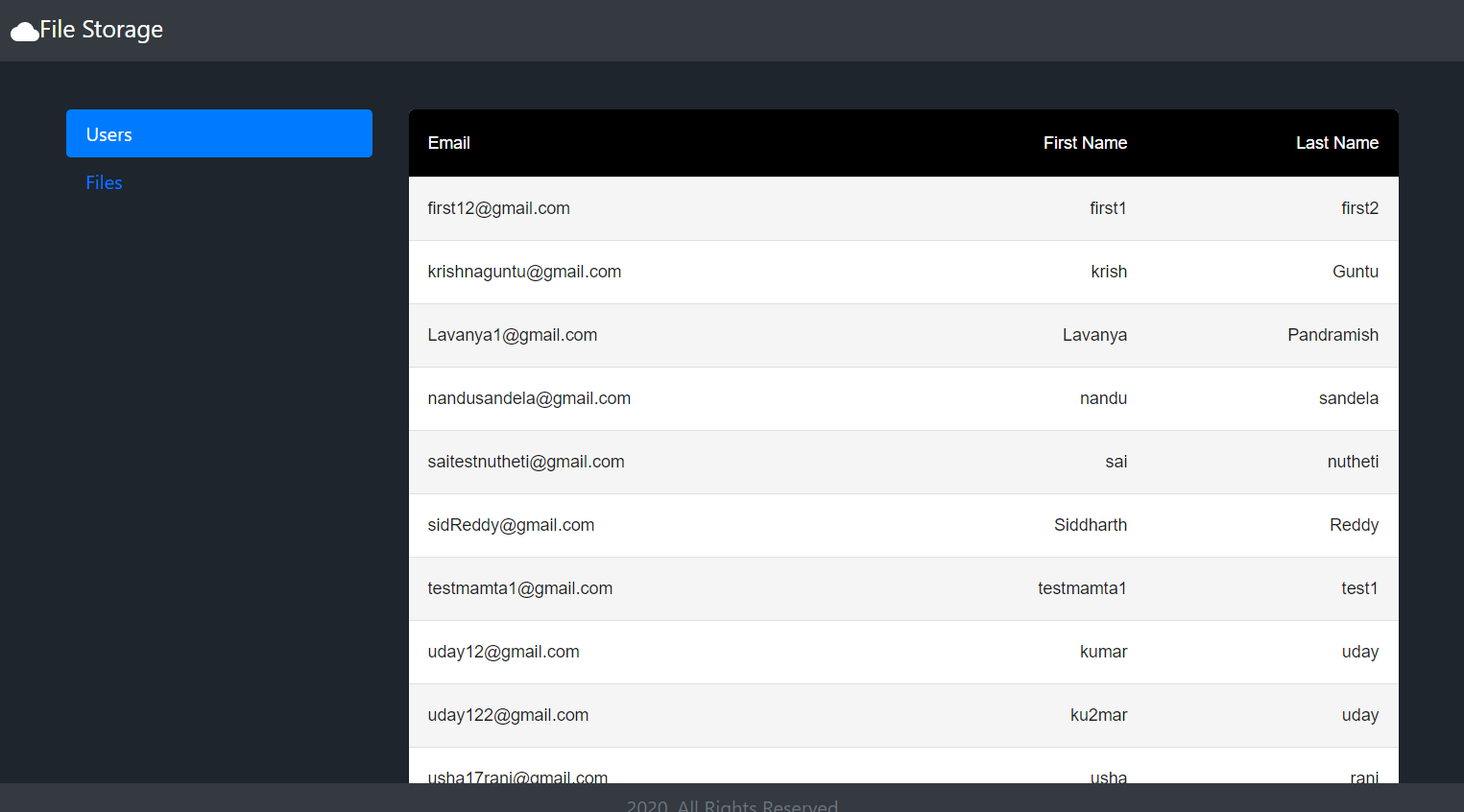
It will be redirected to the home page once , ok is clicked on the alert



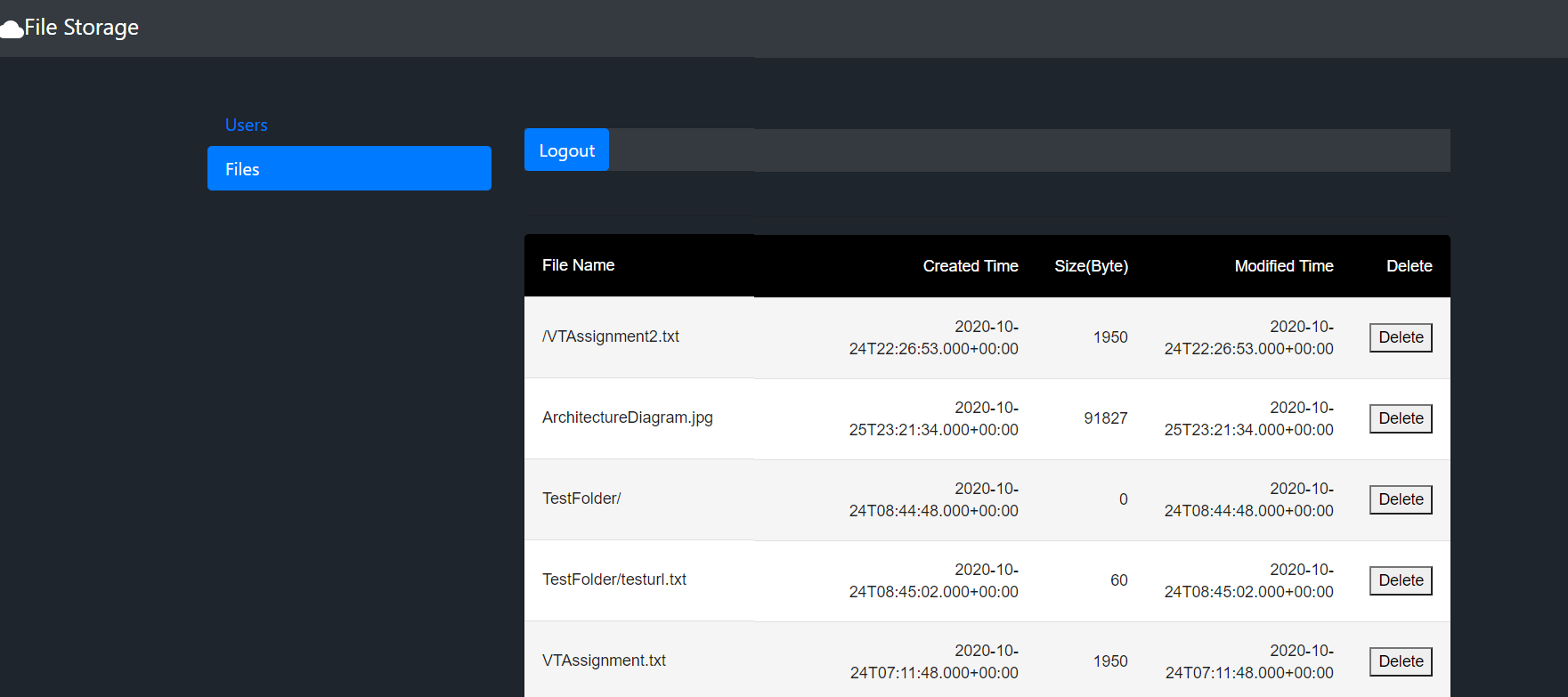
Click get started to get into admin account, select admin tab, and enter credentials



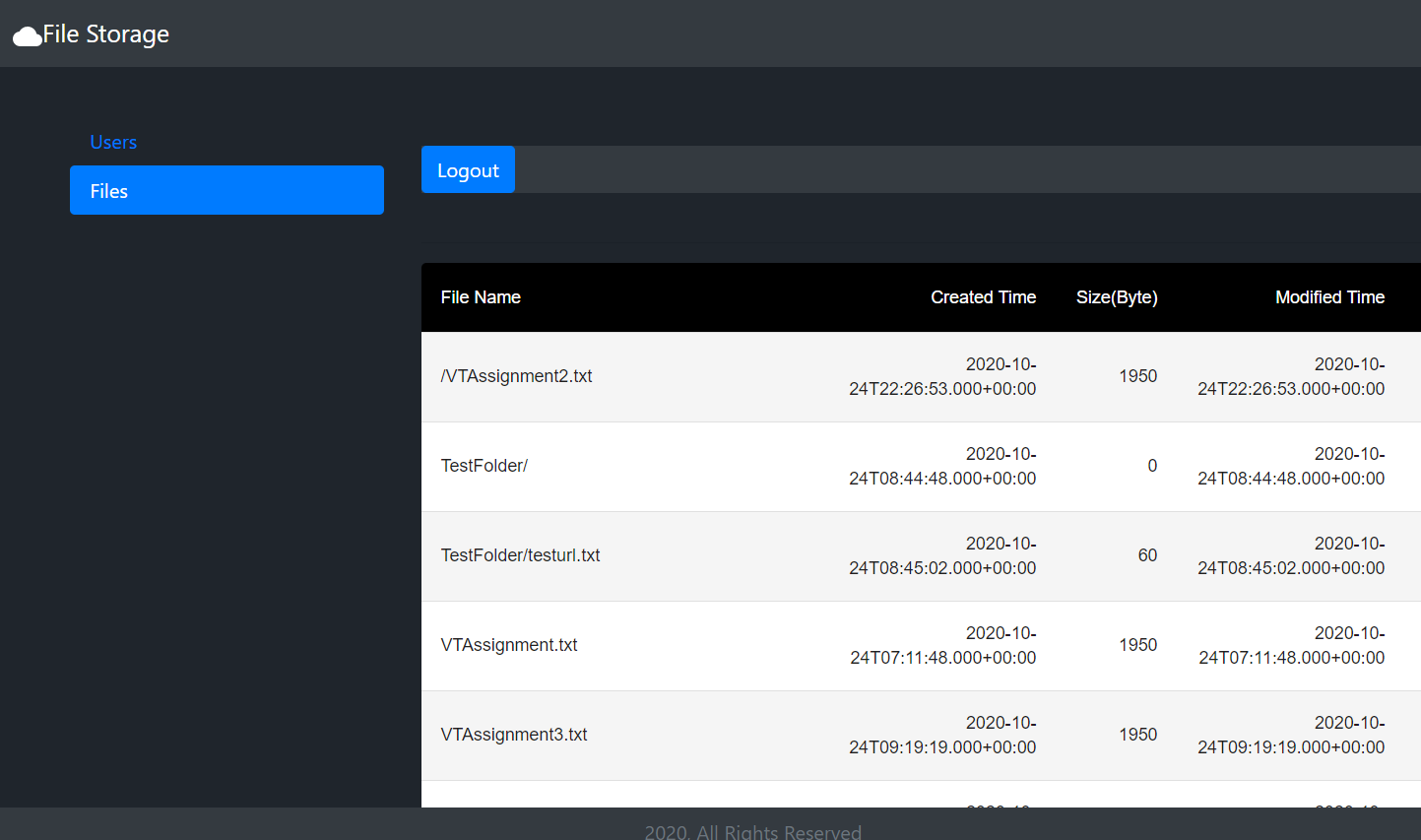
View all the user details



Click on the files tab and View all the files uploaded



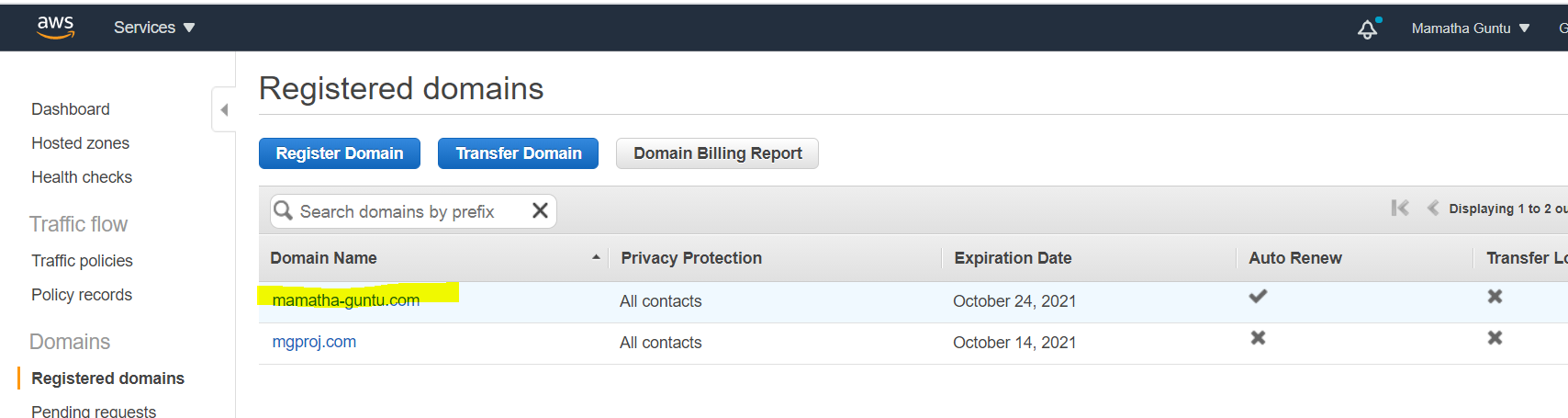
Delete



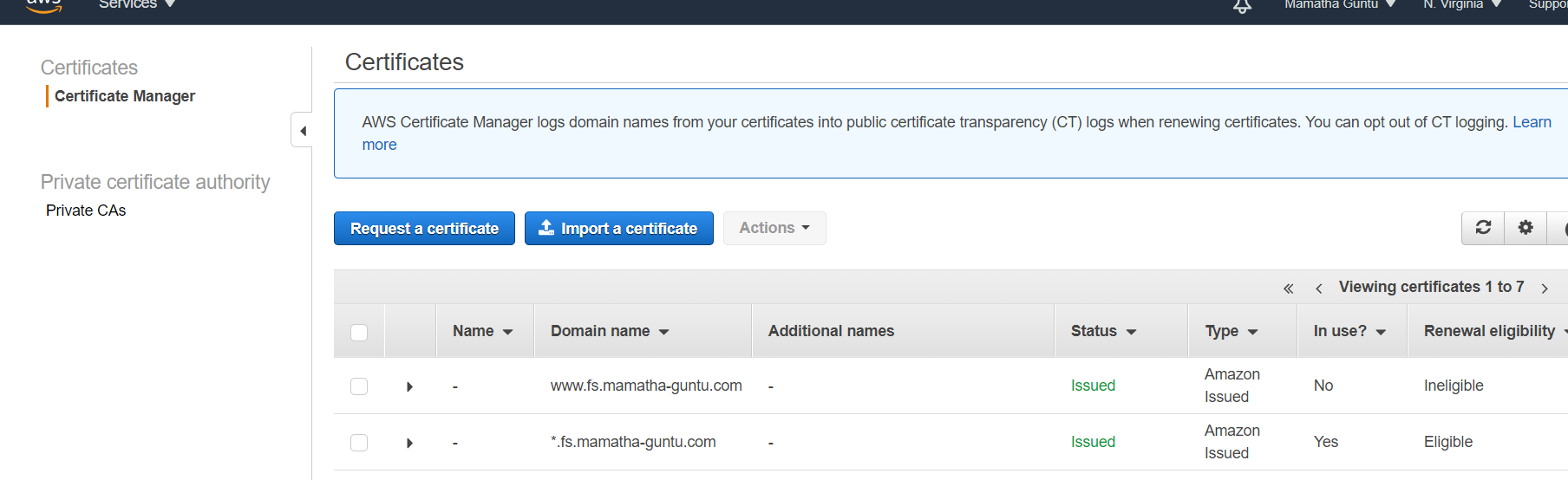
Logout will take to the home page

AWS Services and configuration

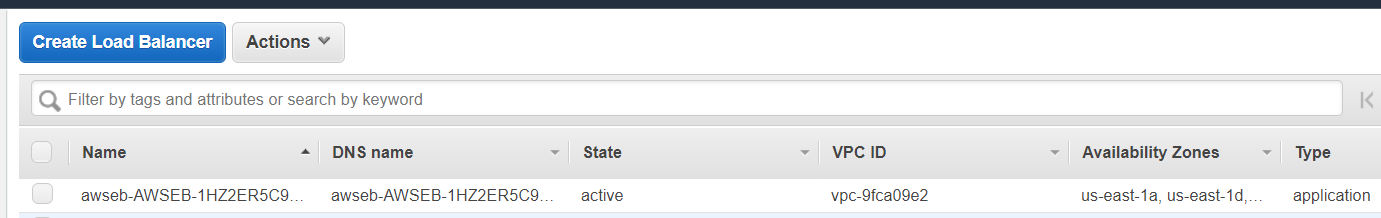
* Create R53 domain



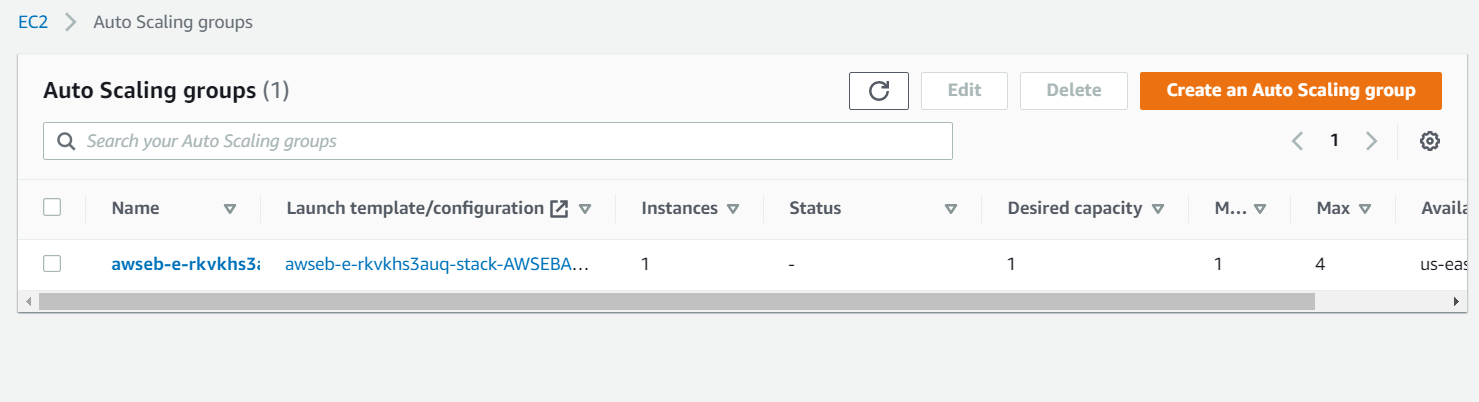
and create SSL certificate for the domain created



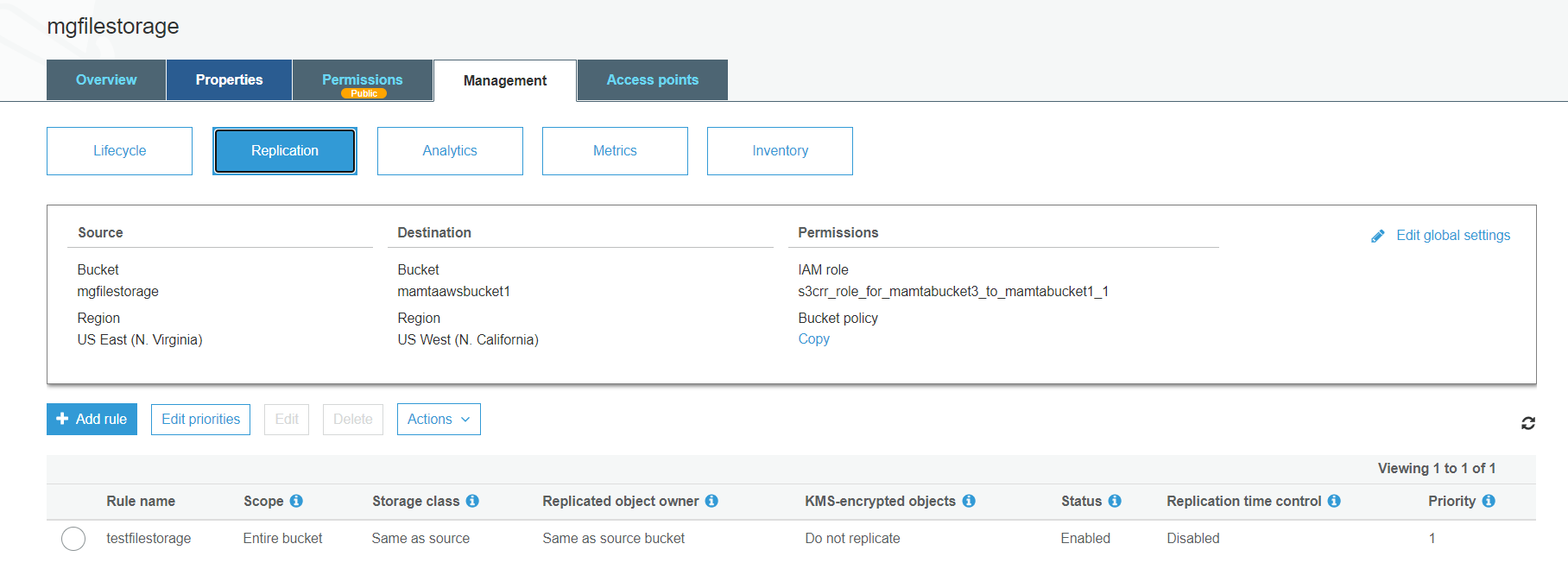
* Classic load balancer



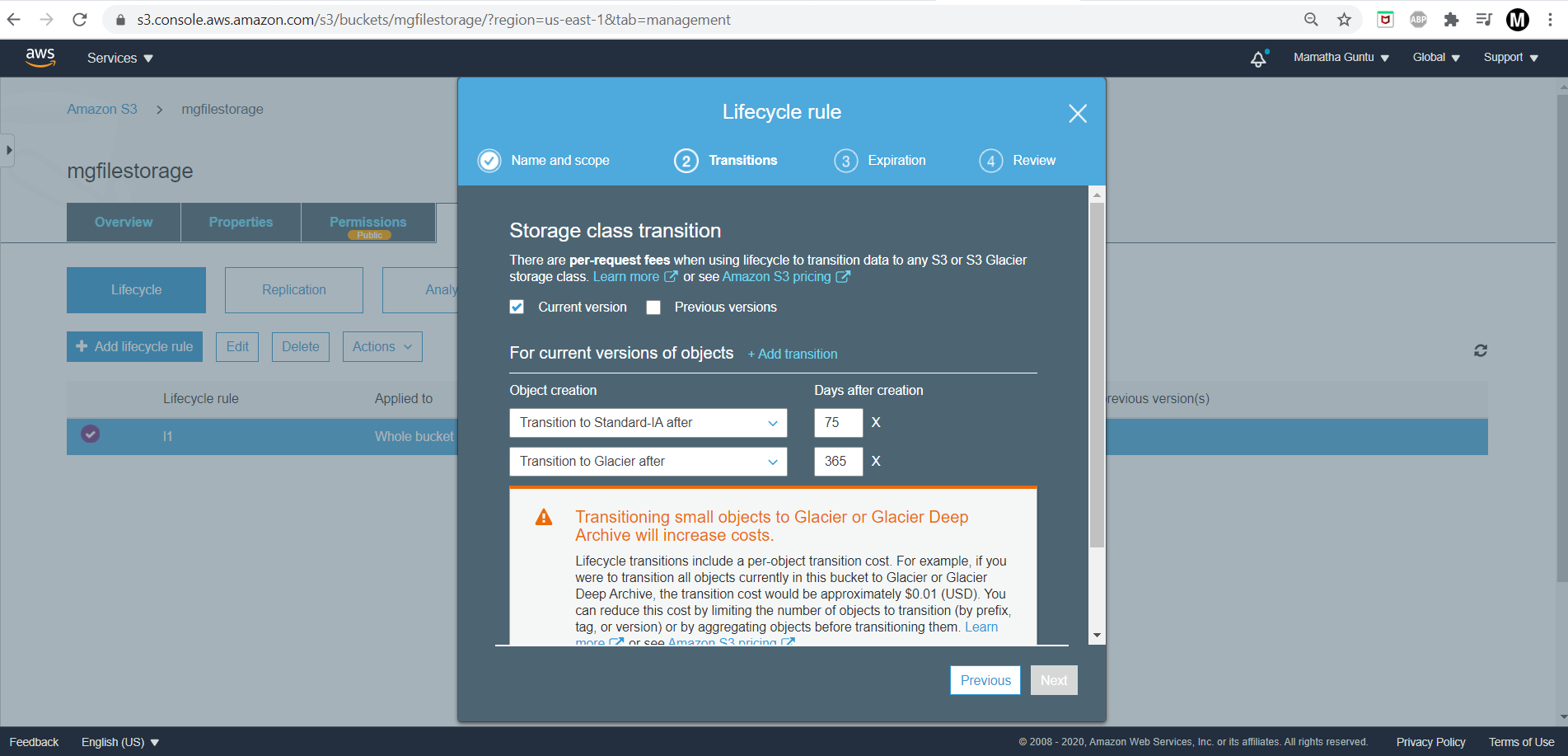
* Auto scaling group



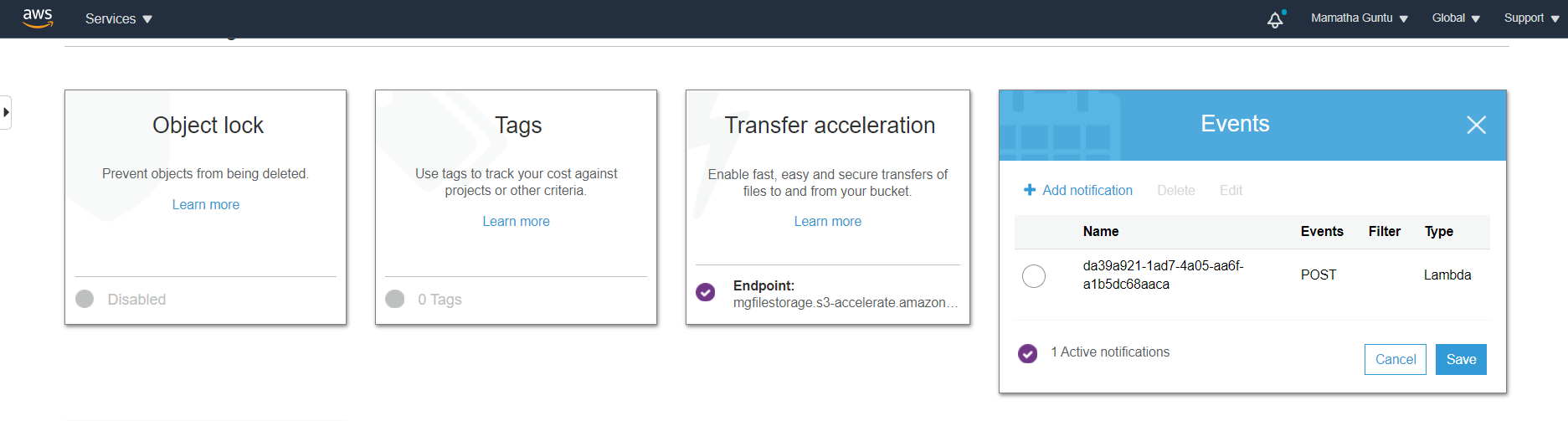
* S3 bucket with Replication for disaster recovery



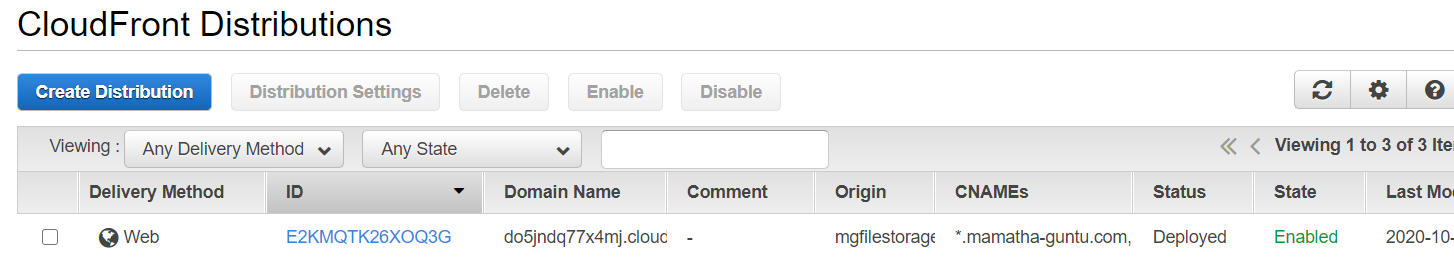
Lifecycle policy



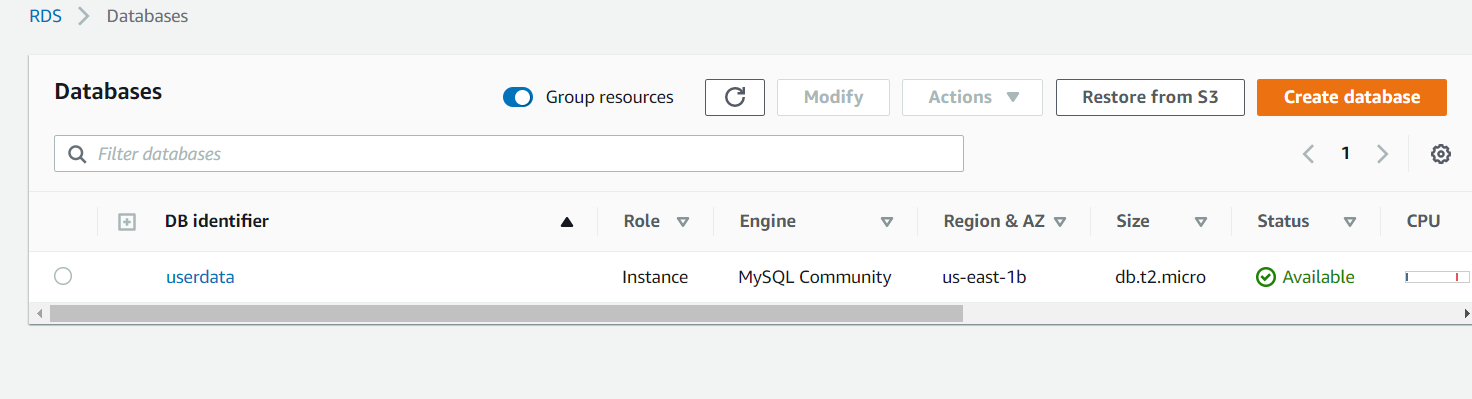
* Lambda for Post operation

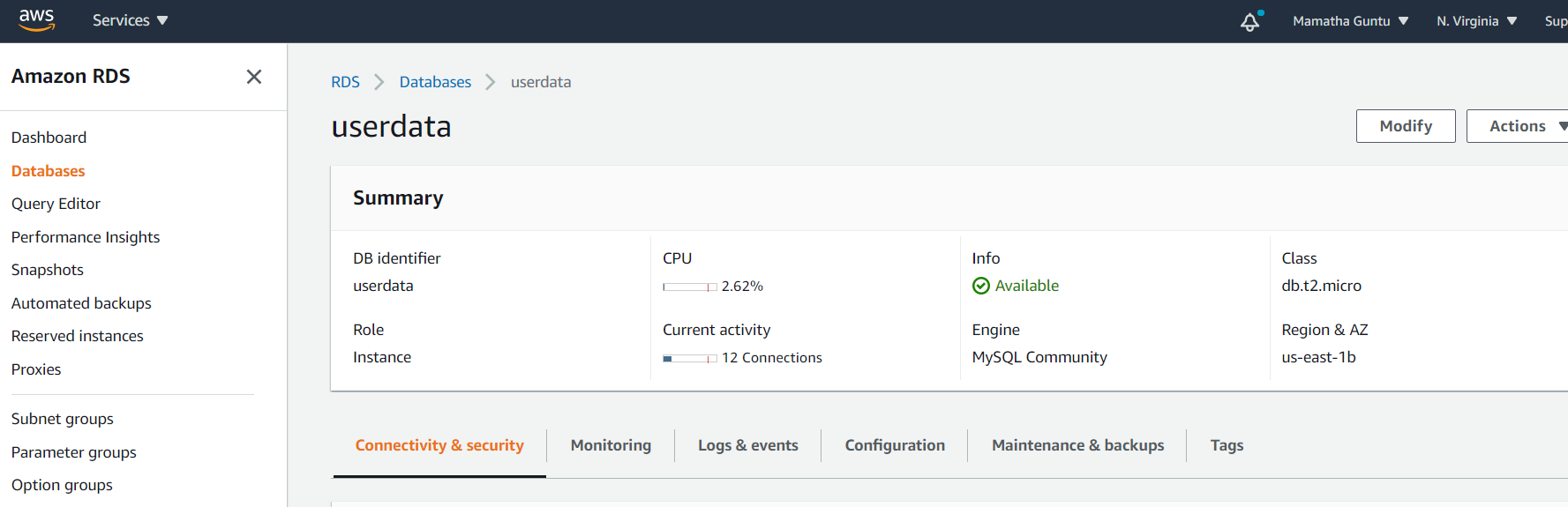


* Cloudfront distribution to access the S3 files for download quickly

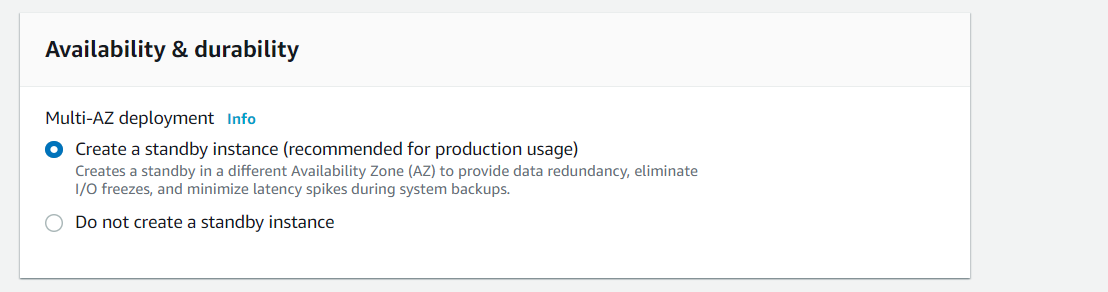


* RDS- MYSQL to maintain user data

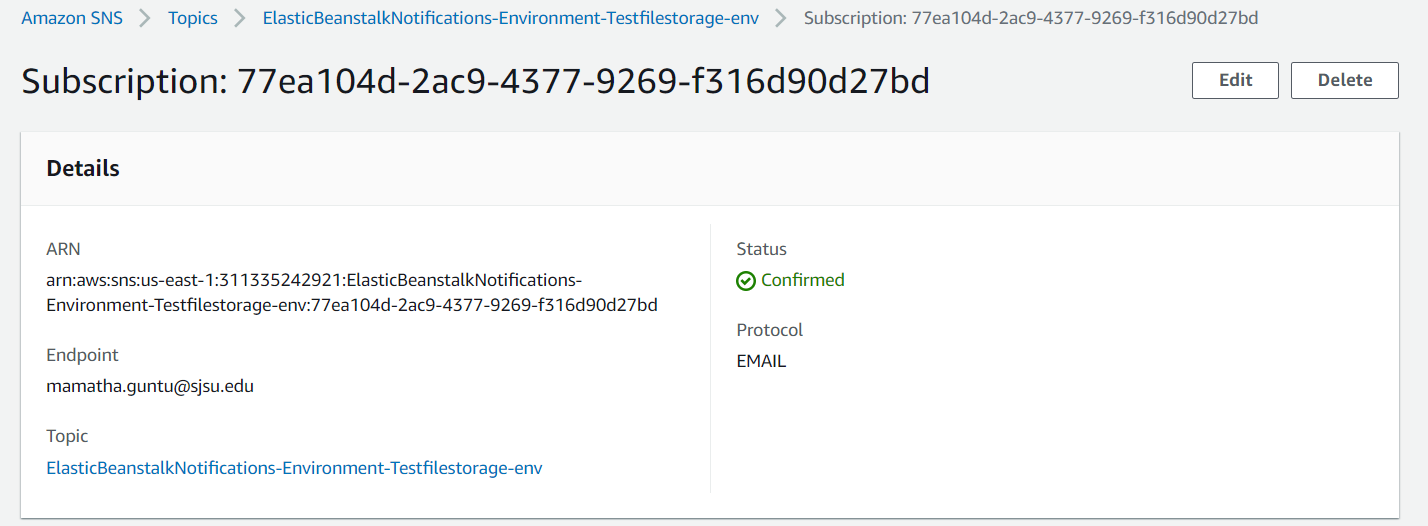




RDS Multi AZ deployment

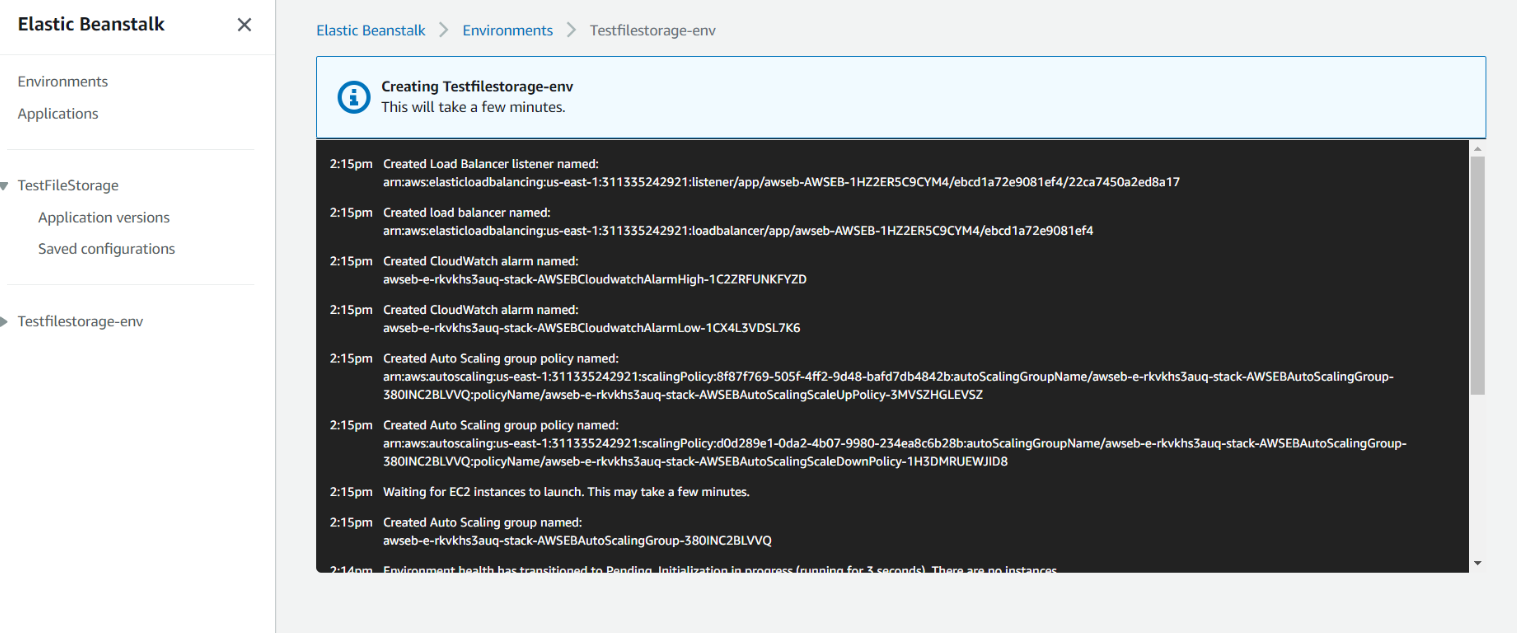


SNS for Deployed environment status

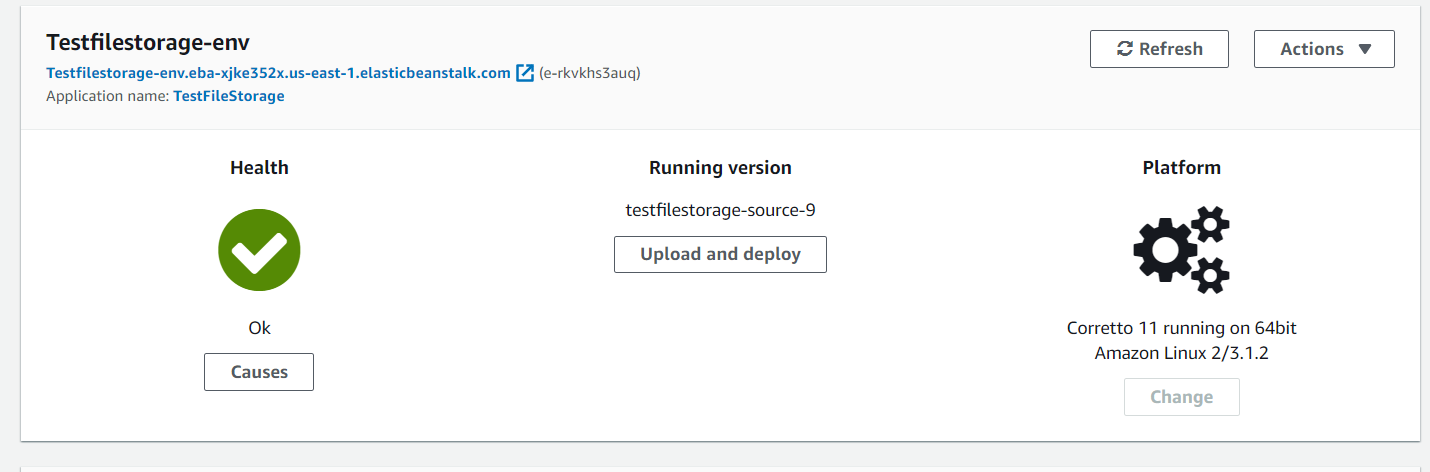


Deployment steps:

* Clone the github repo
* npm install
* maven clean
* maven install
* verify .jar file is uploaded in target folder
* create Elastic beanstalk web application with Java and upload the jar file
* Wait till the environment is created



Once the environment is ready, we can see the url with which we can access webapp



Create a simple routing policy record with our registered domain to use custom domain instead of beanstalk url 