File Management System

Introduction-File management system allows users to store, update, delete, view files using AWS cloud based web application. Users can upload any number of files or single up to 10 MB max file size. User can upload .txt, excel editable files. Users can update the file contents which can be downloaded immediately. User can check his file last file update time and date, upload time and date and update information. Whenever user wants to delete file it will be deleted. This application basically manages all the files using AWS cloud based components.

Technologies used

1>Front end-JSP/HTML, JavaScript, CSS

2>Server-side processing-Servlets

3>Back end-Mysql RDS with JDBC

4>IDE-NetBeans

5>JDK version-1.8(mandatory)

6>Server-GlassFish 4.1.1

Feature list

1>Bulk upload of multiple files of size less than 10 MB.

2>Insert and update single file.

3>Delete single file.

4>List all the records of the files.

5>Download specific file.

6>Manages multiple users.

Solution

1>To upload max file size of 10 MB-JavaScript validation message will appear and avoid uploading file which is greater than 10 MB.

2>All the information of files will be stored on the S3 bucket and back up will stored on another bucket in another region. (using cross region replication policies).

3>For first 75 days bucket will use AWS Standard access after 75 days it will convert bucket to S3-IA and after 1-year data will be stored on S3 archive. After 2 years data will be deleted in buckets.

4>Each user will register on website first to access file management system using unique username and password. There are 2 tables in database which will separate each user from accessing each other’s content. User table and file info table which will maintain the relationship on one to many. Primary key of users table will be foreign key of file info table. In this way we can identify which user has which files.

5> Website will be registered using AWS Route53.

6>For each insert, update, delete operation the data will be send to S3 bucket and RDS MYSQL instance to maintain record of file. (update time, upload time, description)

7>Components are connected in such way user has faster access to download any file from S3 it will download using CloudFront web distribution.

Components connection: -Route53->CloudFront->Elastic bean stalk (EC2, ELB with auto scaling up to 4 instances and health checkup)

8>The steps we can take to convert single AZ database to multi A-Z database we can take backup of that instance each time we insert data into the database we can take backup of that database using lambda function. Each time user will try to access bucket contents the lambda function will trigger and take back up of the database.

9>Cloud watch billing alarm is set to $0.1 so that it will trigger after this limit and send email notification to user.

10>S3 Transfer acceleration is enabled to get faster speed in uploading and downloading.

11>Multiple EC2 instances are present in multiple availability zones such that in case one instance fails another will boot up auto scaling is set for 4 up to instances such that upper threshold of 6000000 is passed it will spin up new instance to load balance an application.

12>Amazon SNS service is used to send messages to user whenever user download file. The email of user is maintained in the database while user registration.

13>Multiple availability zones and regions are available for S3 bucket and EC2 instances through Elastic bean stalk.

14>For each user application code will check if there are any files already uploaded i.e. if user tried to upload file for the first time it will create new folder for each user for separation on S3 bucket.

Public URL to the application: [www.myfilemanagement.com](http://www.myfilemanagement.com)

Test account credentials-

username- pratik

password- pratikdhumal

source code url to git: <https://github.com/pratikshivajidhumal/myCloudProject>