

San Jose State University

<http://www.sjsu.edu/>

Course: CMPE-281 Section 99 Cloud Technologies

Professor: sanjay Garje

ISA: Divyankitha Urs

Submitted by-

Hrishikesh Rendalkar

Project Introduction

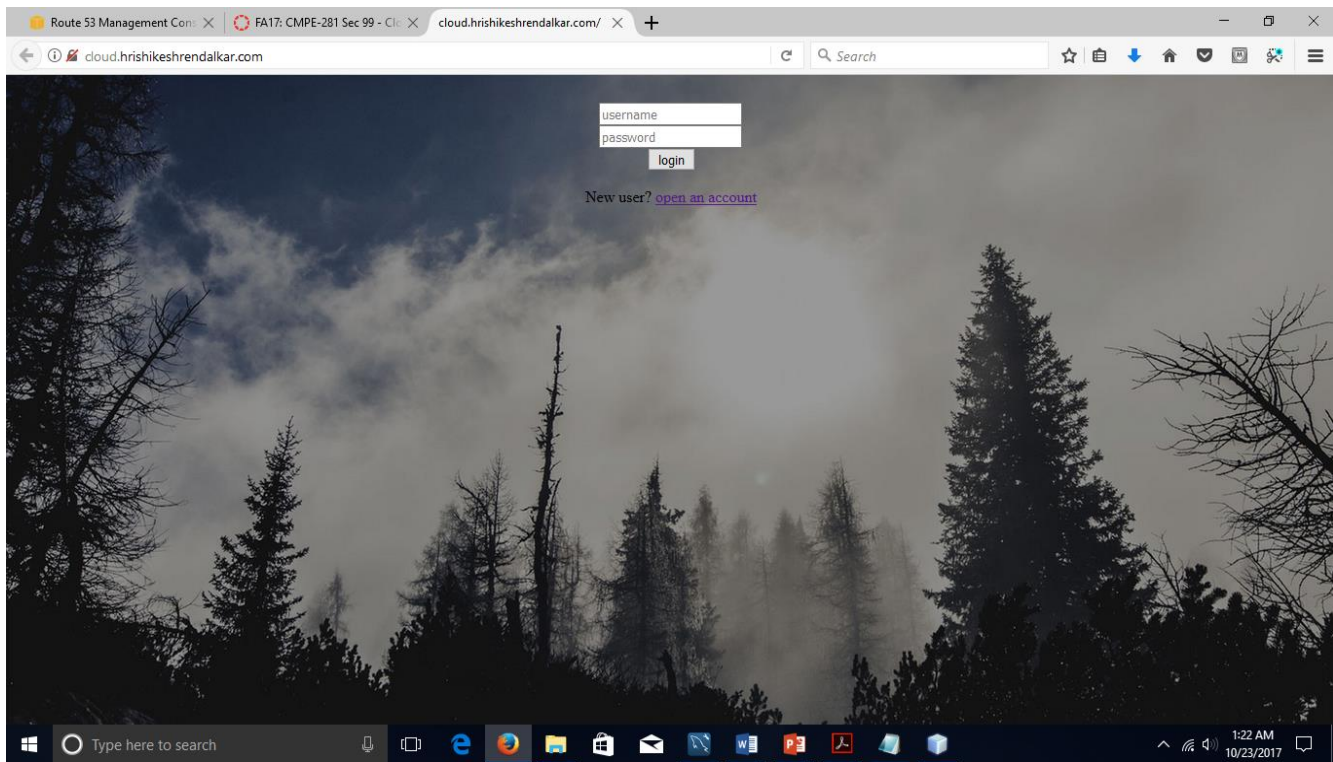
What the application does?

Managing large number of files is tedious now-a-days. Every time user has to take backup of every files he handles. The idea of the project is to build the application which will enable users to upload, download, update and delete their files from any location. As the application will be using all the resources provided by cloud service provider (in this case provider is AWS), user can anywhere and anytime log into the system and can manage his files. Application mainly uses cloud resources so there are no physical resources used to handle the files.

Feature list

1. My application 'Fly Files' provides users to easily access their content anywhere in the world by just one click.
2. The application provides users to create an account with 'Fly Files' and register them. Then the users can choose the files they want and can upload them.
3. Whenever the user logs in again he/she will see the list of files they have previously uploaded.
4. There will be options to update, delete and download the previously added files. They can upload files again if they wish to.
5. Details such as user name, file name, description, date created and time of upload will be provided for the each file uploaded. In this way user can track every details of the file being added.

Screen image captures of the working project



After logging for the first time



Uploaded first file 'UploadThisFile.txt'

Route 53 Management Console | FA17: CMPE-281 Sec 99 - Cloud | cloud.hrishikeshrendalkar.com

cloud.hrishikeshrendalkar.com/MainMenuServlet

Hello clouduser

[logout](#)

User	File name	Description	Date created	Download	Delete	Update
clouduser	UploadThisFile.txt	txt	2017-10-23 08:28:08.0	Download	Delete	Browse... No files selected. Submit

Choose file to upload [Browse...](#) No files selected. [Submit](#)

We have uploaded another file 'dice.png' image and now we will now download the first file i.e. 'UploadThisFile.txt'

Route 53 Management Console | FA17: CMPE-281 Sec 99 - Cloud | cloud.hrishikeshrendalkar.com

cloud.hrishikeshrendalkar.com/MainMenuServlet

Hello clouduser

[logout](#)

User	File name	Description	Date created	Download
clouduser	UploadThisFile.txt	txt	2017-10-23 08:28:08.0	Download
clouduser	dice.png	png	2017-10-23 08:30:39.0	Download

Choose file to upload [Browse...](#) No files selected. [Submit](#)

Opening E_UploadThisFile.txt

You have chosen to open:

E_UploadThisFile.txt

which is: Text Document (52 bytes)
from: http://cloud.hrishikeshrendalkar.com

What should Firefox do with this file?

☐ Open with Notepad (default)

☒ Save File

☐ Do this automatically for files like this from now on.

[OK](#) [Cancel](#)

We will now delete the second file we uploaded i.e. 'dice.png'. So now in bucket also there will be only one file

The screenshot shows a web browser window with the URL `cloud.hrishikeshrendalkar.com/DeleteServlet`. The page has a light blue background and displays "Hello clouser" with a [logout](#) link. Below the header is a table with columns: User, File name, Description, Date created, Download, Delete, and Update. The table contains one row for user 'clouser' with file 'UploadThisFile.txt' (description 'txt', date '2017-10-23 08:28:08.0'). The 'Download' and 'Delete' buttons are visible. The 'Update' column shows a 'Browse...' button and the text 'No files selected.' Below the table, there is a 'Choose file to upload' section with a 'Browse...' button and a 'Submit' button.

User	File name	Description	Date created	Download	Delete	Update
clouser	UploadThisFile.txt	txt	2017-10-23 08:28:08.0	Download	Delete	Browse... No files selected. Submit

Choose file to upload No files selected.

We can update previously updated file just by clicking the browse button in front of the file we want to update and click on submit. Both the copies of file are kept. One can differentiate it by looking at the time when the file was uploaded.

The screenshot shows a web browser window with the URL `cloud.hrishikeshrendalkar.com/MainMenuServlet`. The page has a light blue background and displays "Hello clouser" with a [logout](#) link. Below the header is a table with columns: User, File name, Description, Date created, Download, Delete, and Update. The table contains two rows for user 'clouser'. The first row has file 'UploadThisFile.txt' (description 'txt', date '2017-10-23 08:28:08.0'). The second row has file 'uploadThisText.txt' (description 'txt', date '2017-10-23 09:06:26.0'). Each row has 'Download' and 'Delete' buttons. The 'Update' column shows a 'Browse...' button and the text 'No files selected.' Below the table, there is a 'Choose file to upload' section with a 'Browse...' button and a 'Submit' button.

User	File name	Description	Date created	Download	Delete	Update
clouser	UploadThisFile.txt	txt	2017-10-23 08:28:08.0	Download	Delete	Browse... No files selected. Submit
clouser	uploadThisText.txt	txt	2017-10-23 09:06:26.0	Download	Delete	Browse... No files selected. Submit

Choose file to upload No files selected.

Resources needed to configure in AWS account:

- First create IAM. After creating the IAM account, login with that account credentials.
- S3 bucket: Create bucket in any region and enable versioning and transfer acceleration located in properties of the bucket you create. Also apply lifecycle policy to the objects stored in the bucket. The content is used by the user for first 75 days so keep it in the bucket. After 75 days the objects will move to standard-IA and the contents will be stored in glacier for 1 year before they are discarded.
- Relational database service(RDS): Launch an Database instance and connect it with your MySQL workbench.
- Elastic Beanstalk: Create a new application in elastic beanstalk which includes services such as EC2, S3, CloudWatch, autoscaling and Elastic Load Balancer.
- Route 53: If you want to host your website use route 53. First register your domain and then launch your application on that domain.
- CloudFront: Create a CloudFront distribution which when connected to your bucket helps downloading content much faster as it stores data locally to consumers.
- Cloudwatch: Create a dashboard and add widget's to it so that you can monitor your resource utilization.
- Simple Notification Service(SNS): Create a topic and subscribe to an particular event so that you can receive notification when that event occurs.

Software's to download locally

- Java JDK 1.8.0
- NetBeans IDE
- MySQL workbench
- Internet Browser

Local configuration

- Download the project from GitHub and change the following things
 1. AWS access key and secret key
 2. Bucket name
- Connect MySQL workbench with AWS RDS Instance. Go to database in MySQL and open connect to database enter endpoint path, port, username and password given on AWS RDS instance you created.
- Create a web application in NetBeans IDE and compile It on local Glassfish server.

How to setup and run project locally

- Open NetBeans IDE on your windows machine import the project and make the necessary changes mentioned above.
- Open MySQL and create the necessary tables.
- Go to the run option and select compile and build project. After the build success message run the project.
- In your web browser the project will start working i.e. locally. Fill in the details and keep checking the MySQL database for the updates.