S3C Deployment

1.) Where to get S3C?

Github link: Click Here
Folder link: Click Here

2.) Multiple ways to run:

a.) Open the projects in an IDE in the directories S3C-Master/S3C/SemanticSearchCloud and S3C-Master/S3C/SemanticSearchEdge respectively. Run the cloud then the edge in your IDE. **This has potential to mess up the library dependencies of the Edge code. If you choose to use this way and the libraries are not linked, you must manually add each jar file to the library of the java project in your IDE.

b.) Import the projects in an IDE from the ZIP files located in S3CZIPs. This should preserve the library jar files that the Edge code uses. Run the cloud then the edge in your IDE.

c.) In a terminal:

java -jar S3CMaster/S3C/S3CJars/SemanticSearchCloud.jar

- This will run the Cloud code in one terminal

java -jar S3CMaster/S3C/S3CJars/Client/SemanticSearchEdge.jar

- This will run the Client/Edge code in another

d.) Running on the web server:

**Get file system set up from grad students account.

In client directory:
- All S3C-Master/WebDemo files
- All contents of S3C-Master/S3C/S3CJars/S3Client

In cloud directory:
- All contents of S3C-Master/S3C/S3CJars/S3Cloud

Run “java -jar public_html/S3C/S3Cloud/SemanticSearchCloud.jar &” in one terminal and access your personal PHP displayed client via: “https://teaching.cmix.louisiana.edu/~**your-login**/S3C/S3Client/home.php”. Search or upload.

3.) What can you do?

i.) The cloud is just a server. It does not do anything by itself. It listens on three different ports (assigned in config.properties). From time to time, the Cloud code will not quit when the terminal that it is running in is closed. To kill the process:

ps aux || grep java : This will find all running processes with java as the file type. Find the process named SemanticSearchCloud.java
kill -9 ***ProcessIDFromAboveCommand*** : Ends it

ii.) The Client is going to be given a list of possible commands. Each does what is expected but there are a few caveats:

-k : This will make the key files used to index the encrypted text files. This MUST be called before -u is used to actually upload to any of the destinations. The methods called here used to be combined with the -u command but adding multiple repositories changes at which time the key files need to be made in the PHP code because of inadequate file permissions. (Will go into this next)

-f : Used exclusively for Google Drive or Dropbox searching. Since we do not locally have the files like in a regular upload, we must fetch them from the repository.

-reset : USE WITH CAUTION. This will completely delete all the files in the Google Drive service account since they cannot be seen like in the Dropbox. This was implemented for testing purposes and is very useful but dangerous.

4.) File system explanation:

- <u>inputEncrypted:</u> Searching or fetching will put the encrypted file in this directory to be further accessed.
- <u>uploads:</u> All files in this folder will have a key made out of it with the -k command and all files in this folder will be uploaded to the chosen destination when -u is called.
- <u>storage</u>: This will hold all key files that are uploaded regardless of destination repository as well as any encrypted text file saved on our local server.
- <u>utilities</u>: This will hold the IndexFile and DocSizesFile used by the cloud code. To refresh the server, simply delete them. **This will remove all memory of uploaded files!

****** FILE PERMISSIONS ******

Everything about this project relies heavily on proper file permissions on the different directories. Setting all directories permissions to 777 makes your life infinitely easier, but is obviously unsafe. If some weird bug pops up, you probably do not have write permissions in some folder. The folders not mentioned above are generally not important for basic usage of the project but do things during the upload process. Ask the grad students for help manipulating the file permissions in each of your folders if necessary.

Google Drive and Dropbox

Login (same for both):

Email/Username: s3cdropbox@gmail.com
Password: hpccull16

Edit Dropbox settings:
https://www.dropbox.com/developers

Edit Google Drive settings:
https://console.cloud.google.com

Currently, on the basics of the Google Drive service account and normal Dropbox account are in place and connected to the java via APIs. There is a ton of functionality available that you can play with or read about through documentation.

Links to documentation:

Google Drive Service Account

Dropbox API