ACN Team Project

AWS Cloud Connect

Team Members:

1) Chirag Chudasama (cbc140130)
2) Salil Kansal (sxk150430)
3) Sanket Prabhu (srp140430)

Project Description

We have setup a cloud environment using AWS cloud. The basic idea of the application is to maintain the sync of files between the laptop entity with the cloud as well as the android entity with the cloud. This application involves three entities:

1- AWS cloud
2- Laptop
3- Android phone

The block diagram of the project is shown below:

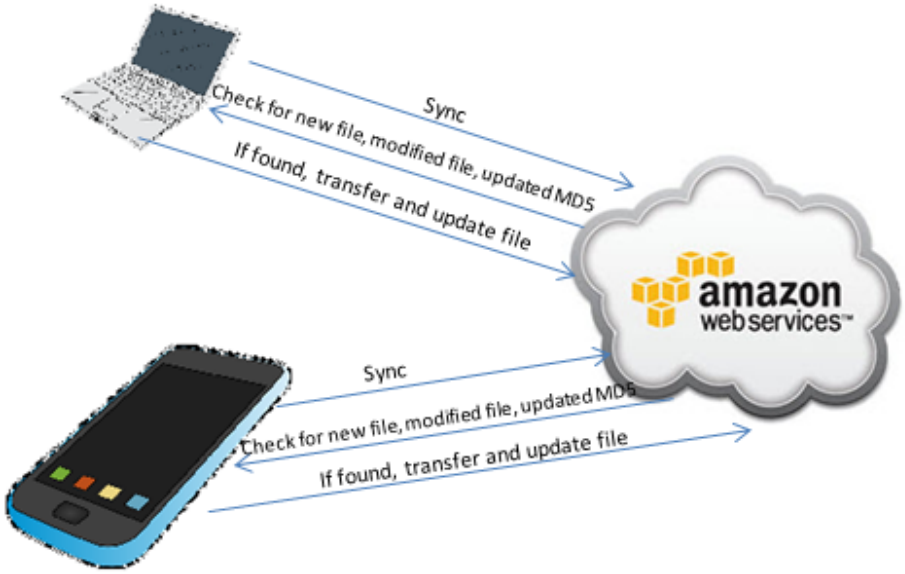


Fig 1: Block Diagram

Entity	Residing in
AWS Cloud	Web
Laptop application	Laptop
Android application	Phone

DESCRIPTION OF COMPONENTS

AWS cloud:

- Resides on the web
- Acts as a storage of files for Android phone and the Laptop
- Performs following functions(Android/Laptop):
    - Receives Sync request
    - Checks for new files on Android/Laptop
    - If new file exists, request transfer
    - Checks for the date modified of existing files
    - If the date modified differs, request transfer
    - Checks for the MD5(checksum) of existing files
    - If the MD5 differs, request transfer

Laptop application:

- Resides on a laptop
- Sends Sync command to the AWS cloud
- Transfers files to the cloud if:
    - There if a new file on the laptop
    - Date modified of the file on the laptop differs from the one on cloud
    - MD5 of the file on the laptop differs from the one on cloud

Android application:

- Resides on an android phone
- Sends Sync command to the AWS cloud
- Transfers files to the cloud if:
    - There if a new file on the phone
    - Date modified of the file on the phone differs from the one on cloud
    - MD5 of the file on the phone differs from the one on cloud

WORKING

The entities work together to provide the required functionality. They communicate among themselves using TCP connection. We have used an array list for the transfer of files. The following major steps define the working of the project and show how entities communicate with each other:

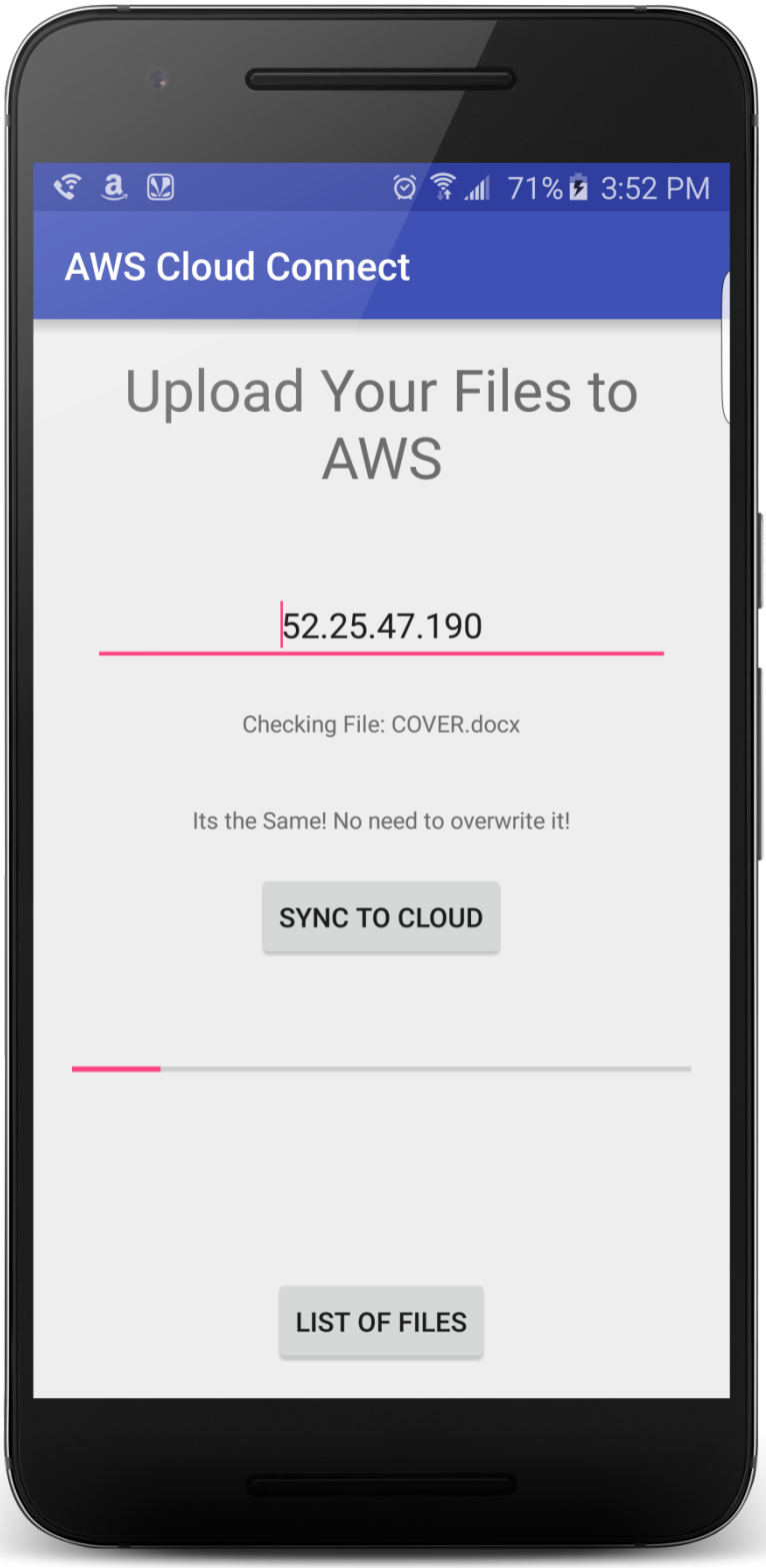
1- Laptop/Phone sends sync command to the AWS server with its IP.
2- We have used a variable that allows the AWS server to determine whether the sync command has been generated via laptop (0) or phone (1).
3- Laptop compares the files from the “filesonlaptop” folder in the laptop to the “laptop” folder on AWS server. Phone compares the files from the “filesonandroid” folder in the laptop to the “android” folder on AWS server.
4- Laptop/Phone checks if the name of all files matches the name of all the files on AWS server,
   - If matches, it checks for the date modified and MD5 of the files
   - If date modified and MD5 matches, no action is taken
   - Else we include the name of the file in the array list
   - Also, if the name of the file differs on AWS server, we include file in the array list too
5- Now, the laptop/phone transfers the same array list to the AWS server using our protocol

Considerations:

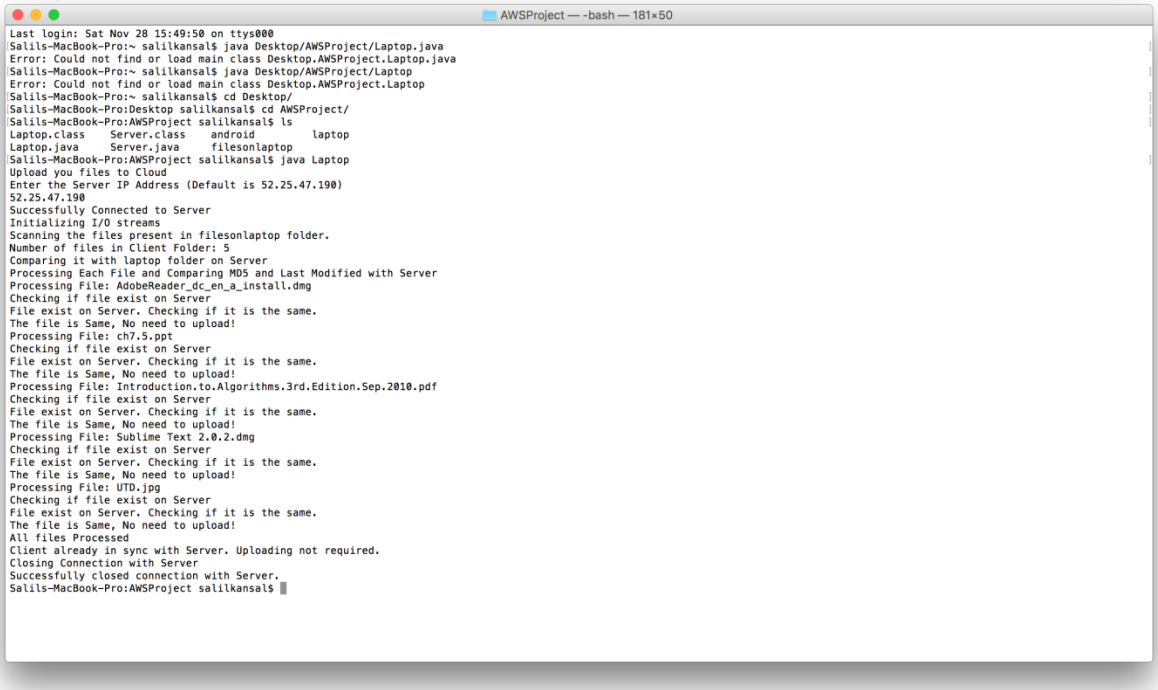
- The command which user inputs into the laptop application should be specific. Otherwise, the AWS server will not be able to interpret the request correctly
- Since, we do not have the scenario of DNS server, we assume that laptop application already knows about IP address of AWS server so that it may initiate connection with it
- Similarly, android phone should also know full address of AWS server

Screenshot

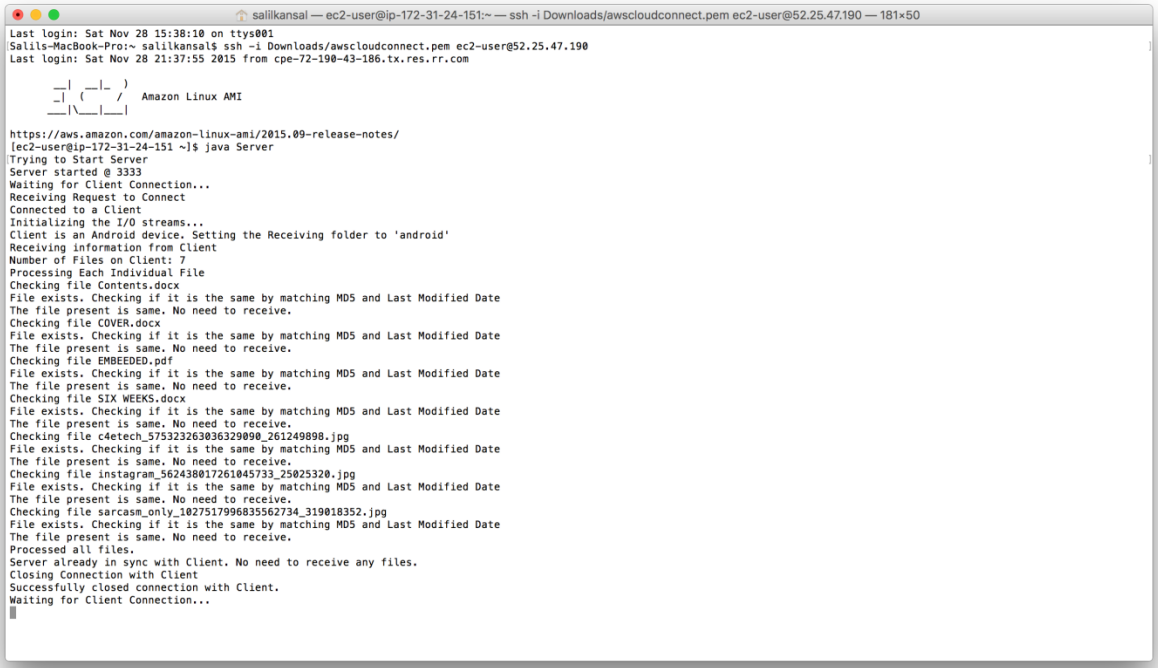
Android Phone:



Laptop:



Server:



Contributions from team members

From the beginning, we divided what each member is supposed to do for the completion of the project. All three of us had the input in designing the protocol for file transfer. The brief idea of what each member did:

- **Salil Kansal:**
  Salil dedicated himself to android programming and created an android application that can send commands to the AWS server and that can sync the files on the phone with the AWS server. He also worked on the Protocol and wrote its code single handedly. Salil also designed the algorithm for the Server application.

- **Sanket Prabhu:**
  Sanket was basically responsible for implementation of the protocols we discussed. He solely debugged all the errors and built error free client and server java files that can communicate with the AWS server.

- **Chirag Chudasama:**
  Chirag was mainly responsible for signing up on AWS for server side storage. He learnt how to transfer files to the cloud, install java on cloud and run the server file on the cloud that allows the cloud to be online for transfer of the files. He was also responsible for the documentation of the report.