

Cloud Prototype

A prototype of the cloud environment.

Getting Started

Prerequisites

Before run the server you need to run mysql first, but no need to create the database.

1. Introduction

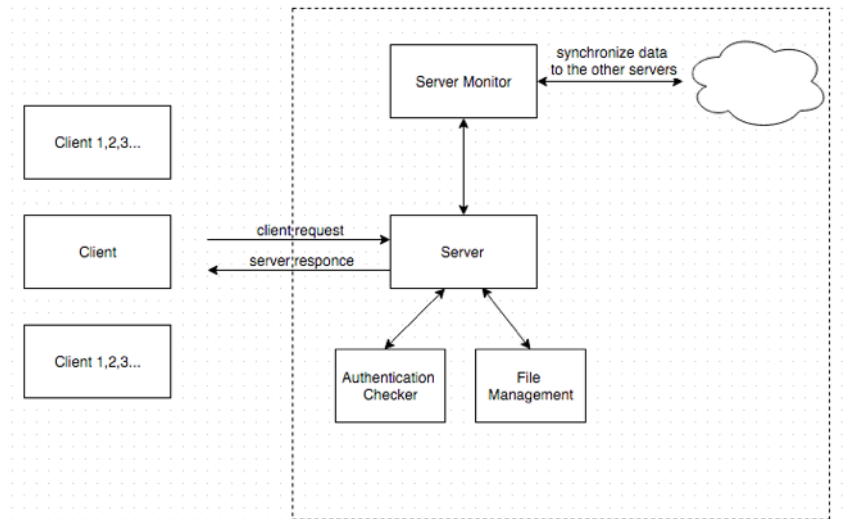


Figure 1: Cloud Architecture

Client

For the client side, we will develop a socket library based on Java socket programming. This library will define the client side protocol which is designed to communicate with the cloud server, and support applications for multiple devices such as the desktop and the mobile phone. Meanwhile, it will also implement the basic functions of cloud client, for instance, login to the account, show the file list of directory and basic file operations.

Server

For the server side, first of all, we will design socket communication with the client side. This implementation will handle all the requests of client side, including login, list the directory, transfer files and so on. We will define a server side protocol which will manage different types of requests, and also it will include the error detecting. Next, we will design socket communication between multiple server sides, in this project, we will assume there are only two servers. We will implement the synchronization of server side by defining a server side protocol which will be discussed in the aim 3. At last, we will consider about the security policy of server side.

2. Backgrounds

Cloud Computing

Cloud Computing is a technology that uses the internet and central remote servers to maintain data and applications. It allows consumers and businesses to use applications without installation and make it possible to access their personal files at any computer with internet access. This technology allows for much more efficient computing by centralizing data storage, processing and bandwidth. In cloud computing, there are three segments which are application, storage and connectivity. Each segment serves a different purpose and offers different products for businesses and individuals around the world. In June 2011, a study conducted by V1 found that 91% of senior IT professionals actually don't know what cloud computing is and two-thirds of senior finance professionals are clear by the concept, highlighting the young nature of the technology. In Sept 2011, an Aberdeen Group study found that disciplined companies achieved on average of 68% increase in their IT expense because of the cloud computing and the only 10% reduction in data center power costs.

Socket Programming

Sockets are network communication channels. The sockets described in this chapter use the TCP network protocol, although you can find Tcl extensions that create sockets using other protocols. TCP provides a reliable byte stream between two hosts connected to a network. It handles all the issues about routing information across the network, and it automatically recovers if data is lost or corrupted along the way. What's more, TCP is the basis for other protocols like Telnet, FTP, and HTTP.

Client Socket

A client opens a socket by specifying the host address and port number for the server of the socket. The host address gives the network location (i.e., which computer) and the port selects a particular server from all the possible servers that may be running on that host. For example, HTTP servers typically use port 80, while FTP servers use port 20. The following example shows how to open a client socket to a Web server

Server Socket

A server socket is a little more complex because it has to allow for multiple clients. The way this works is that the socket command creates a listening socket, and then new sockets are created when clients make connections to the server. Tcl takes care of all the details and makes this easy to use. You give the socket command a callback to execute when a client connects to your server socket. The callback is just a Tcl command. It gets as arguments the new socket and the address and port number of the connecting client.

3. Methods

Protocol (HTTP like protocol)

In this project, we will generate HTTP like protocol.

Request

Several types of requests will be considered in this project, for example, GET, POST, PUT and DELETE. The GET method is used to get a file or get the name list of a directory. The POST method is used to check authentication or close the connect. The PUT method is used to upload a file. The DELETE method is used to delete a file on cloud. Meanwhile, user could define any type of property in the request.

Response

The Status code is the same as HTTP protocol, we will use 200, 401, 404 and so on.

Salt (cryptography)

In cryptography, a salt which is a secure random string that is used as an additional input to a one-way function that “hashes” a password. The main reason using salt is to defend against dictionary attacks versus a list of password hashes and against pre-computed rainbow table attacks. In this project, we use this algorithm to implement the user authentication. In the client side, we use

the username as a salt combined with password and using SHA_256 to hash this combined string and get string H1. Then, pass both username and H1 to the server side. Next, in the server side, we will look for the received username in database and find another salt which used in server side and the H2 which was computed when user registered this account. At the end, we will hash h1 with the new salt and get string H2_, and the user will be validated if H2_ is the same as H2.

4. Results

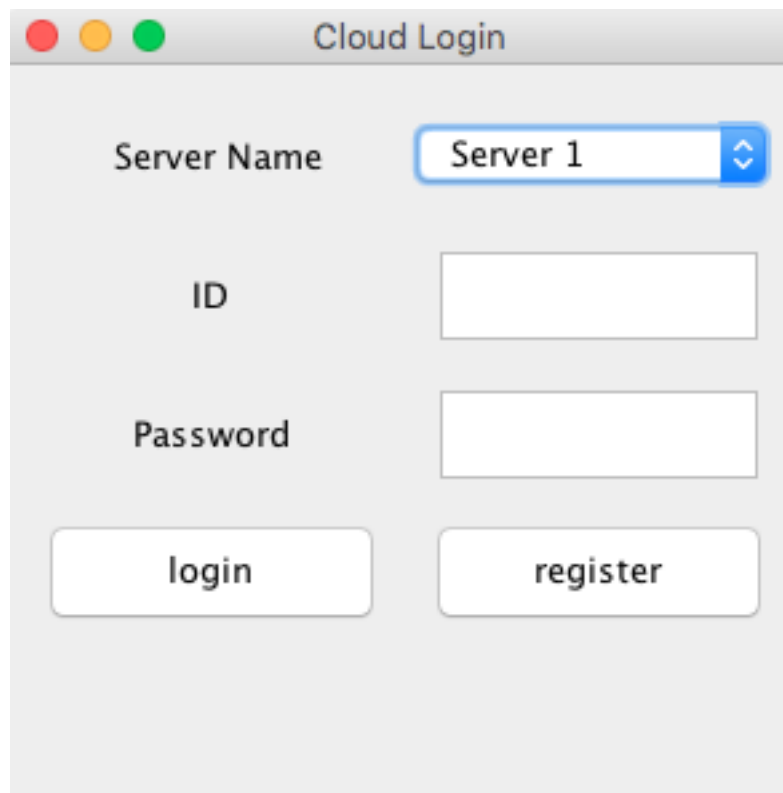
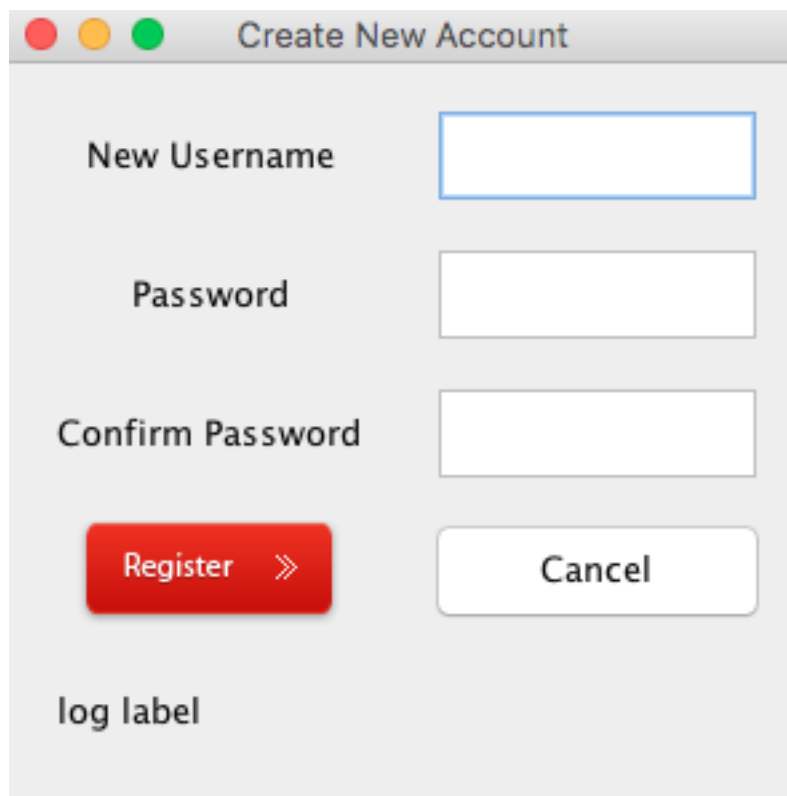
The image shows a window titled "Cloud Login" with a standard macOS-style title bar (red, yellow, and green buttons). The window has a light gray background. It contains four labels: "Server Name", "ID", "Password", and two buttons labeled "login" and "register". The "Server Name" label is positioned to the left of a dropdown menu that currently displays "Server 1". The "ID" and "Password" labels are positioned to the left of empty text input fields. The "login" and "register" buttons are located at the bottom of the form, with "login" on the left and "register" on the right. All text is in a sans-serif font.

Figure 2: Cloud Login

5. Discussion

Built With

- Makefile



A macOS-style dialog box titled "Create New Account". It features three standard window control buttons (red, yellow, green) in the top-left corner. The dialog contains three text input fields: "New Username", "Password", and "Confirm Password". Below these fields are two buttons: a red "Register" button with a double-right arrow icon, and a white "Cancel" button with a gray border. At the bottom left, there is a "log label" text element.

Create New Account

New Username

Password

Confirm Password

Register »

Cancel

log label

Figure 3: Cloud Register

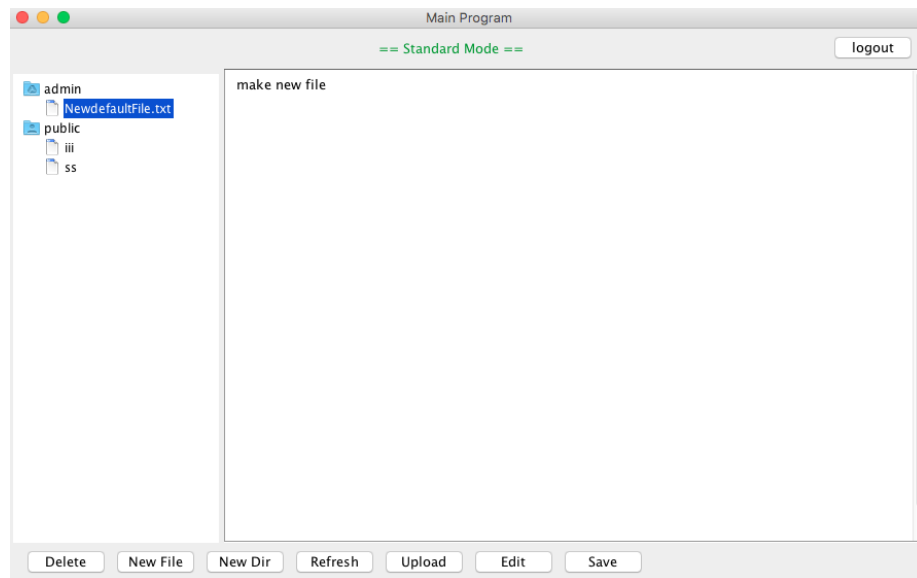


Figure 4: Cloud Main

```
make serverClean - will create the required database and tables every time and start the server
make server - will create the required database and tables at first time and start the server
make client - start the client
make clean - remove all class files
```