Network Programming
AWS Assignment -2

Name – Rahul Kumar

Branch – computer Science (3rd year)

Roll No : 171210046

Faculty : Dr. Ravi Kumar Arya

Source Code : Server

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <time.h>
#include <unistd.h>
#include <sys/types.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include<stdio.h>
#include<stdlib.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<string.h>
#include <arpa/inet.h>
#include <fcntl.h> // for open
#include <unistd.h> // for close
#include<pthread.h>
pthread_mutex_t lock = PTHREAD_MUTEX_INITIALIZER;
char data[1025];
void * socketThread(void *arg)
{
 int clintConnt = *((int *)arg);
 read(clintConnt, data, sizeof(data)-1);
printf("server : one client sent me a message          and the message is -> %s \n",data);
```

```
printf("server : enter response message for client -> ");
// Send message to the client socket
pthread_mutex_lock(&lock);
scanf(“%[^\n]%*c”, data);
pthread_mutex_unlock(&lock);
write(clintConnt, data, strlen(data));
printf("server: response message sent to the client\n");
printf("Exit socketThread \n");
close(clintConnt);
}
int main()
{
int sock = 0, clintConnt = 0;
struct sockaddr_in ipOfServer;
sock = socket(AF_INET, SOCK_STREAM, 0); // creating socket
memset(&ipOfServer, '0', sizeof(ipOfServer));
memset(data, '\0', sizeof(data));
ipOfServer.sin_family = AF_INET;
ipOfServer.sin_addr.s_addr=htonl(INADDR_ANY);
ipOfServer.sin_port = htons(2020);
bind(sock, (struct sockaddr*)&ipOfServer , sizeof(ipOfServer));
listen(sock , 3);
pthread_t tid[60];
int i=0;
printf("\nserver is Running.\n");
while(1)
{
```

```
clintConnt = accept(sock, (struct sockaddr*)NULL, NULL);
pthread_create(&tid[i], NULL, socketThread, &clintConnt);
i++;
 }
return 0;
}
```

Source Code: Client

```
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <netdb.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <arpa/inet.h>
int main()
{
   int CreateSocket = 0,n = 0;
   char data[1024];
   struct sockaddr_in ipOfServer;
   memset(data, '0' ,sizeof(data));
   if((CreateSocket = socket(AF_INET, SOCK_STREAM, 0))< 0)
   {
      printf("Socket not created \n");
      return 1;}
```

```
    ipOfServer.sin_family = AF_INET;
    ipOfServer.sin_port = htons(2020);
    ipOfServer.sin_addr.s_addr = inet_addr("127.0.0.1");
    if(connect(CreateSocket, (struct sockaddr *)&ipOfServer, sizeof(ipOfServer))<0)
    {
       printf("Some Error occured\n");
       return 1;
    }
    printf("client is running : Enter ur message for server-> ");
    scanf("%[^\n]%*c", data);
    write(CreateSocket, data, strlen(data));
    memset(data,'\0',sizeof(data));
     read(CreateSocket, data, sizeof(data)-1);
     printf("message from server : %s\n",data);
    return 0;
}
```

AWS Assignment - C Multithreaded Client-Server

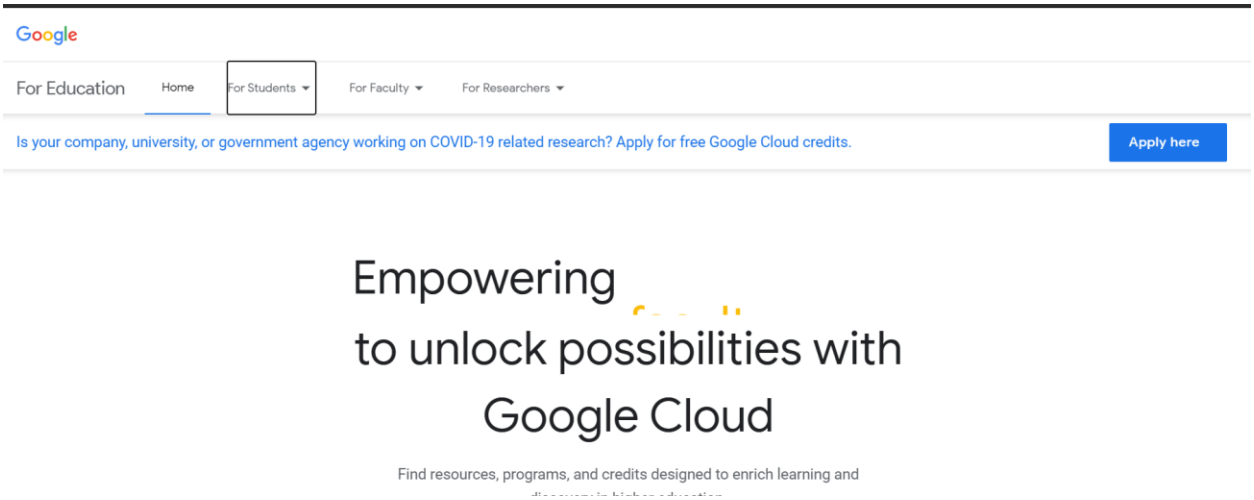
Implement multithreaded (using pthread library) client-server with TCP to allow multiple (2) clients to connect/interact with the server simulataneously.

Step 1: create an account on Google cloud

Problems in creating account on google cloud : it requires all of your information i.e your address and credit card details

Solution of Problem : As we are student google is providing free account for universities student just you have to provide small information and identity proof that you are college student (you can upload your images of your identity card)

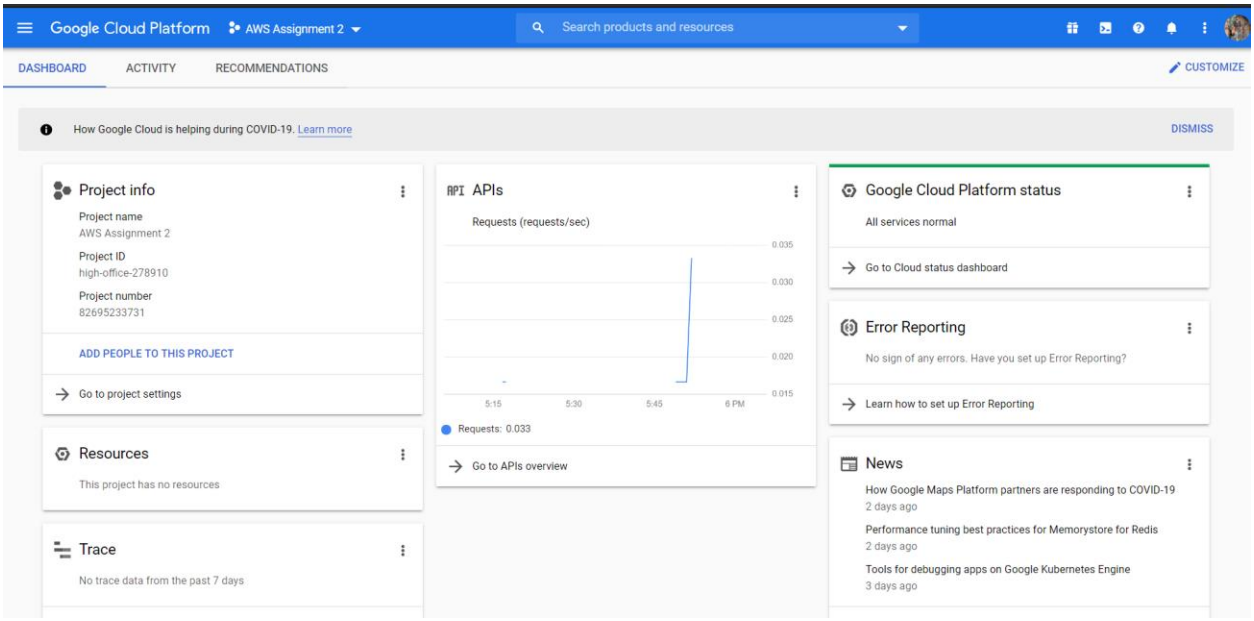
Type google cloud account for students you get this page



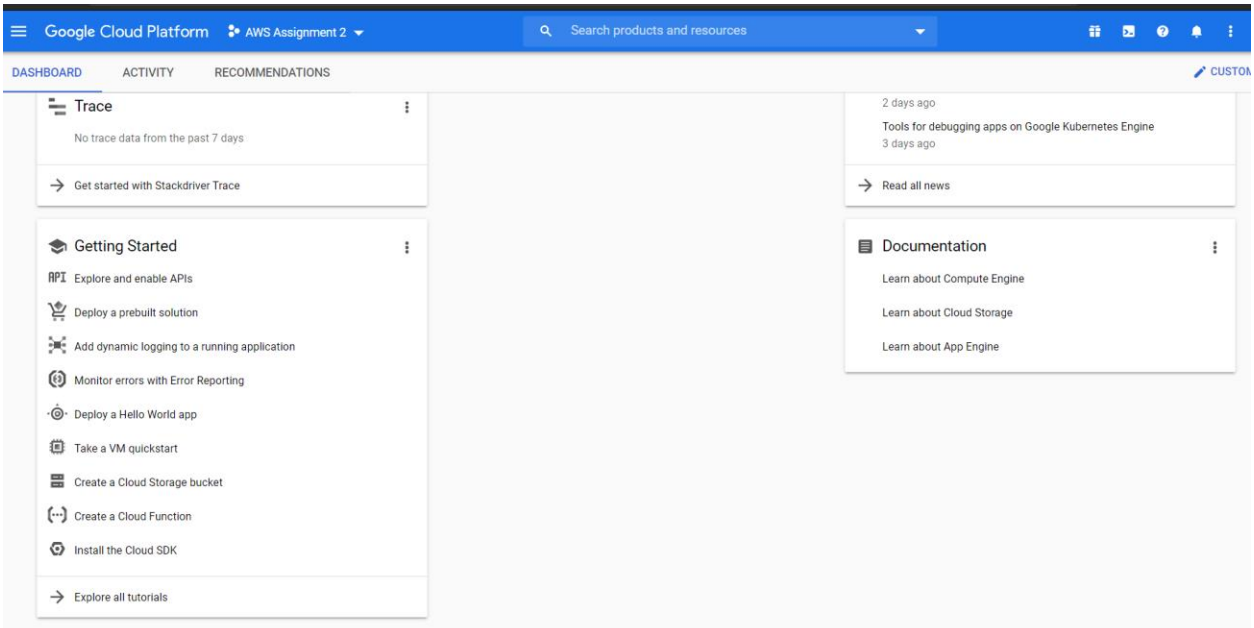
Select students benefit from it and fill your details and upload your photo of your identity card

After some times google verify your identity and now you can use google account free of cost without providing your card Details.

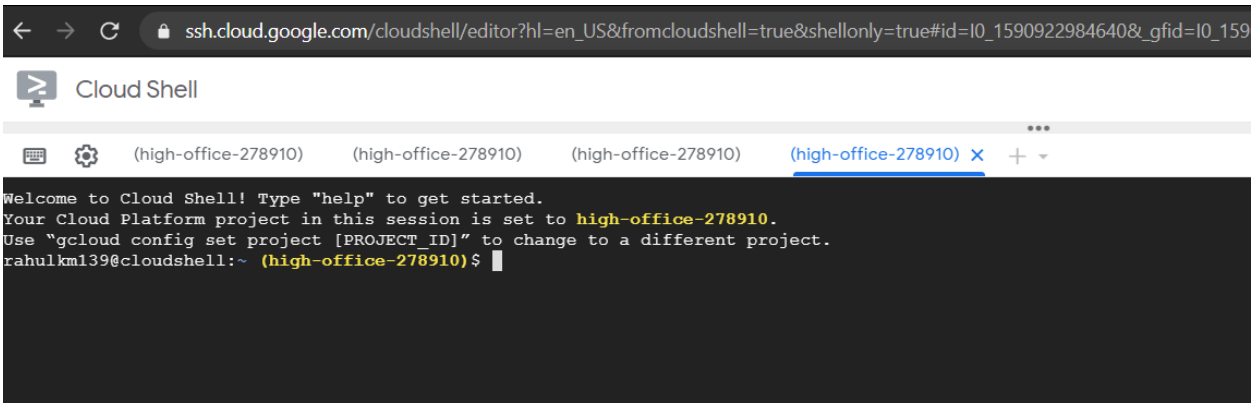
Step 2: now create your Project .



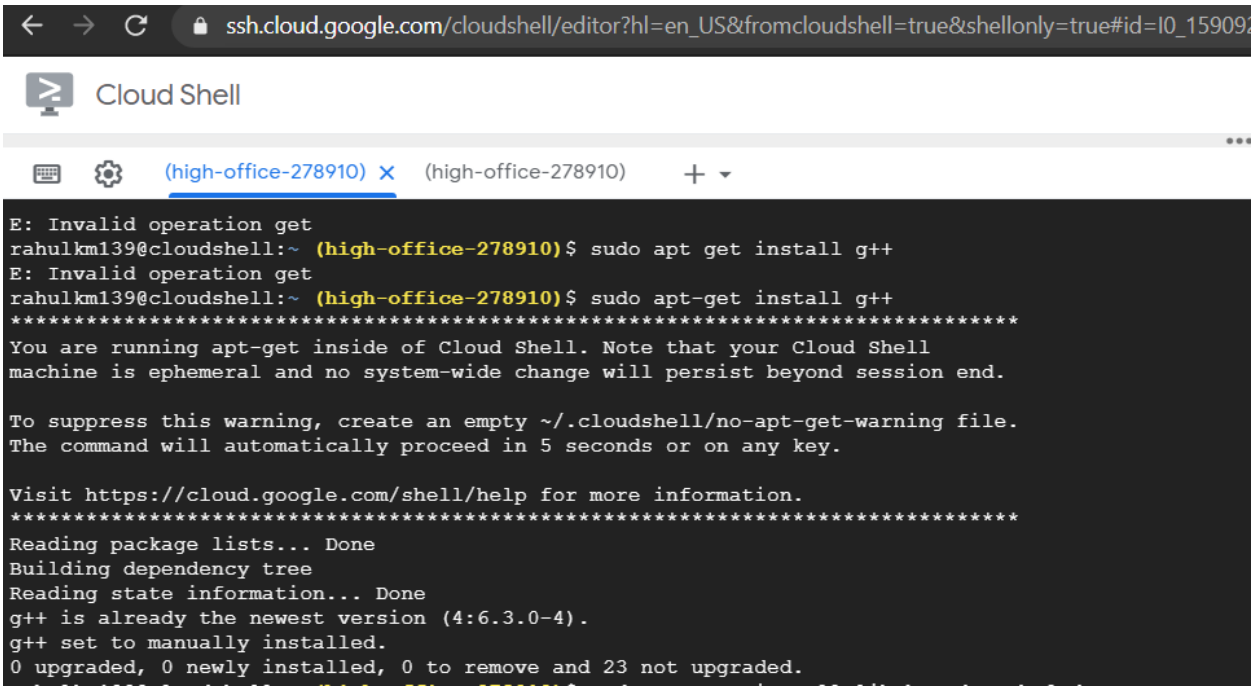
Step 3: Select Take a VM quickstart , it will create your virtual environment.



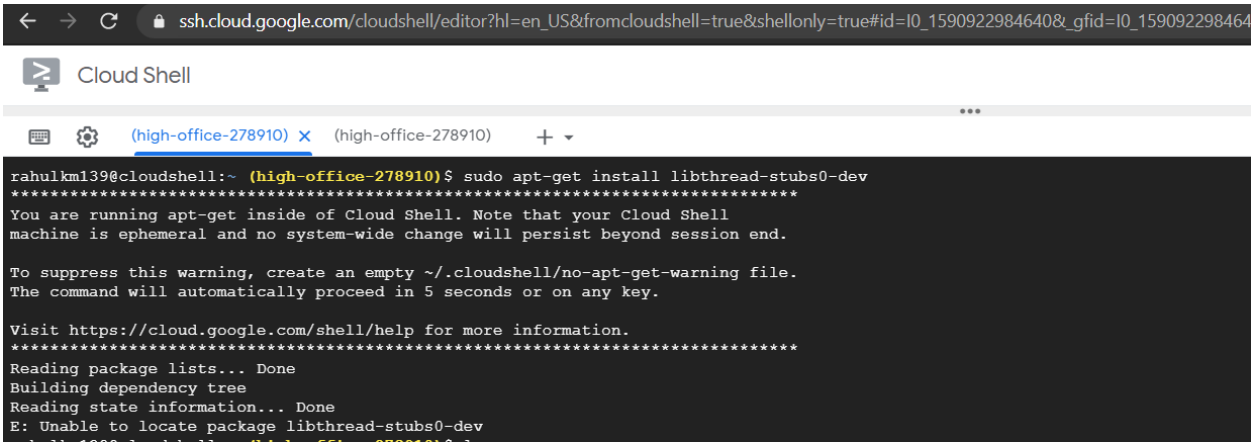
Step 4: Now cloud Shell opened



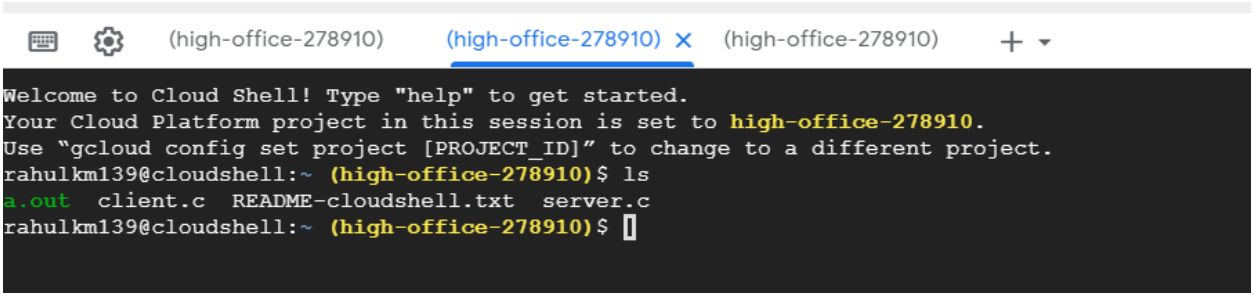
Step 5: install Git



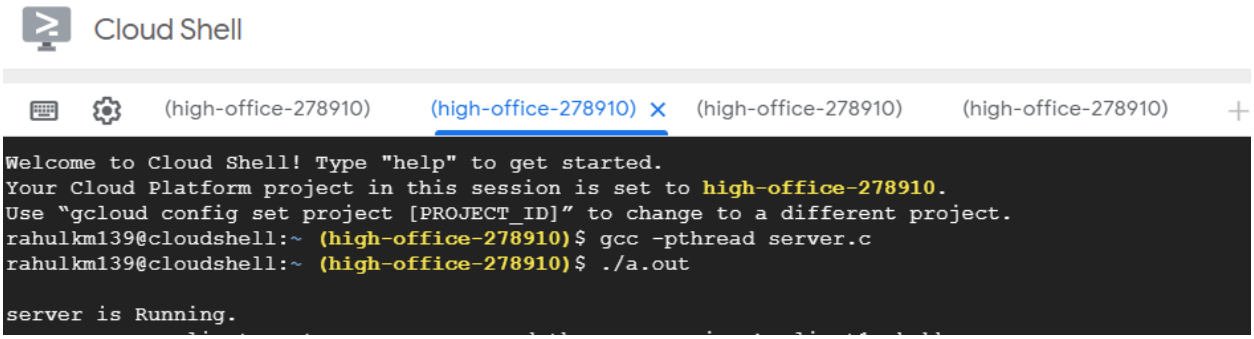
Step 6: install pthread Library



Step 7: Upload client program and server program

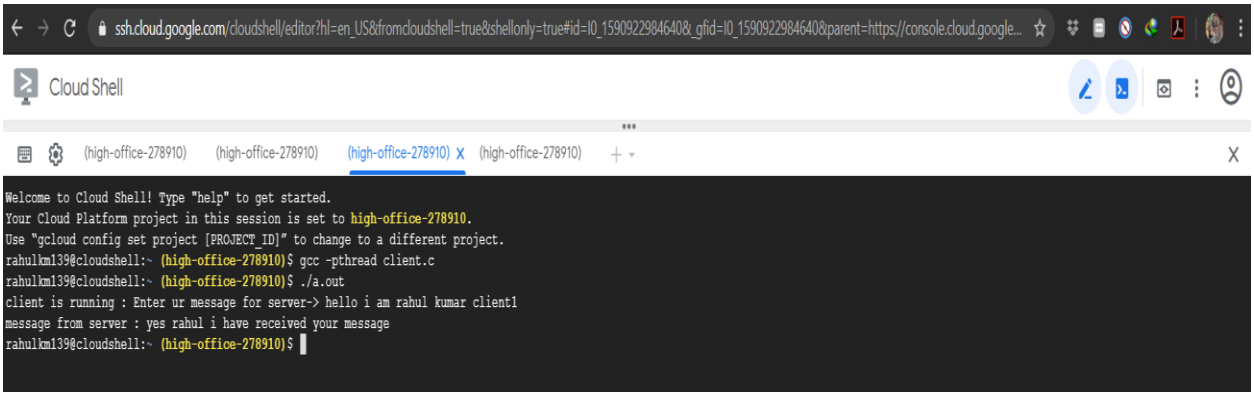


Step 8: Run server program using gcc -pthread server.c

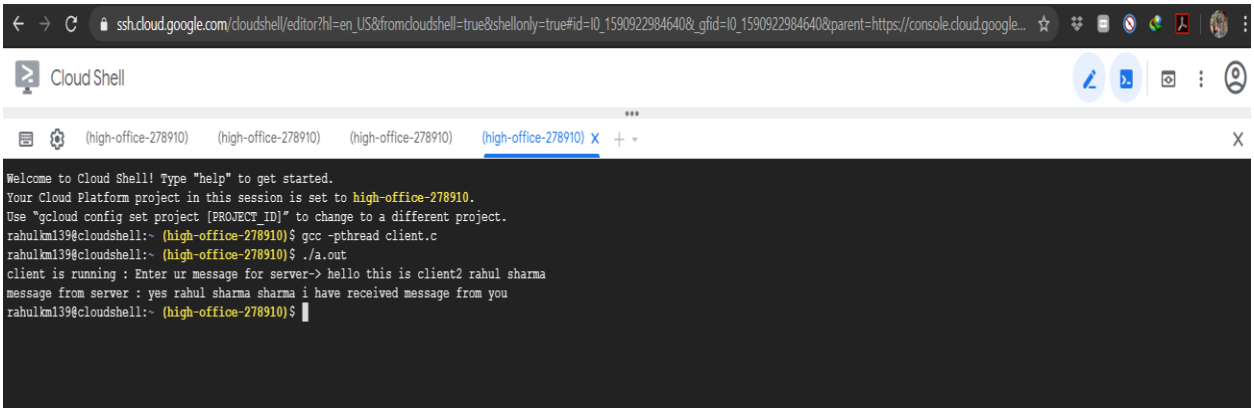


Step 9 : Create Multiple Client and make them communicate with server

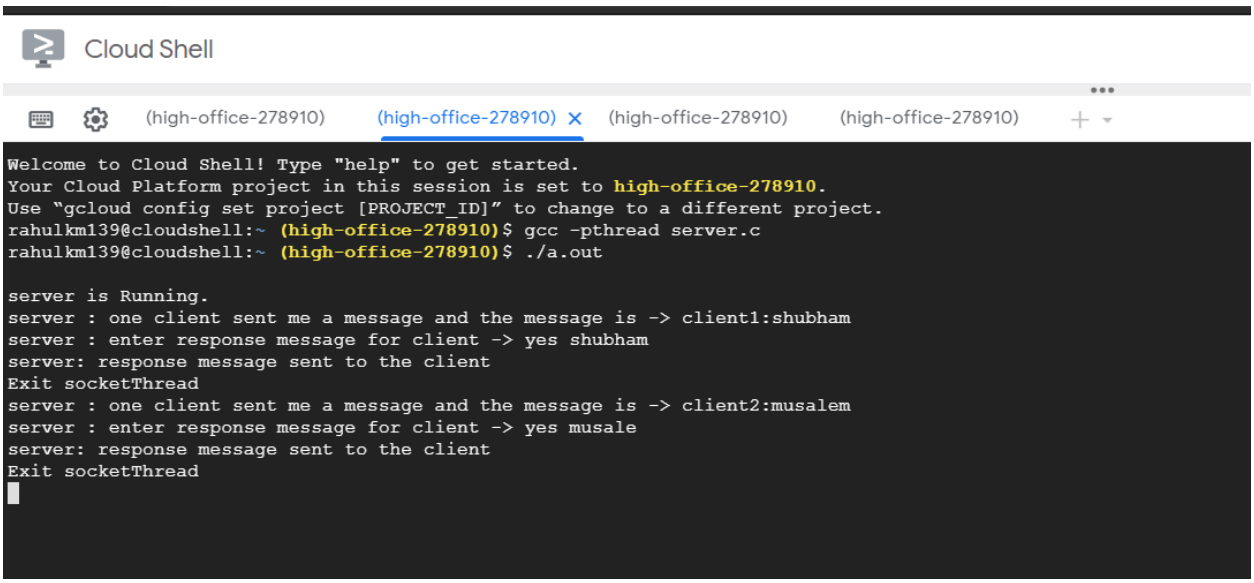
Client 1:



Client2:



Step 11: Now Start communication with server and multiple client



Code Link : https://github.com/rahulkmsharma/Aws2