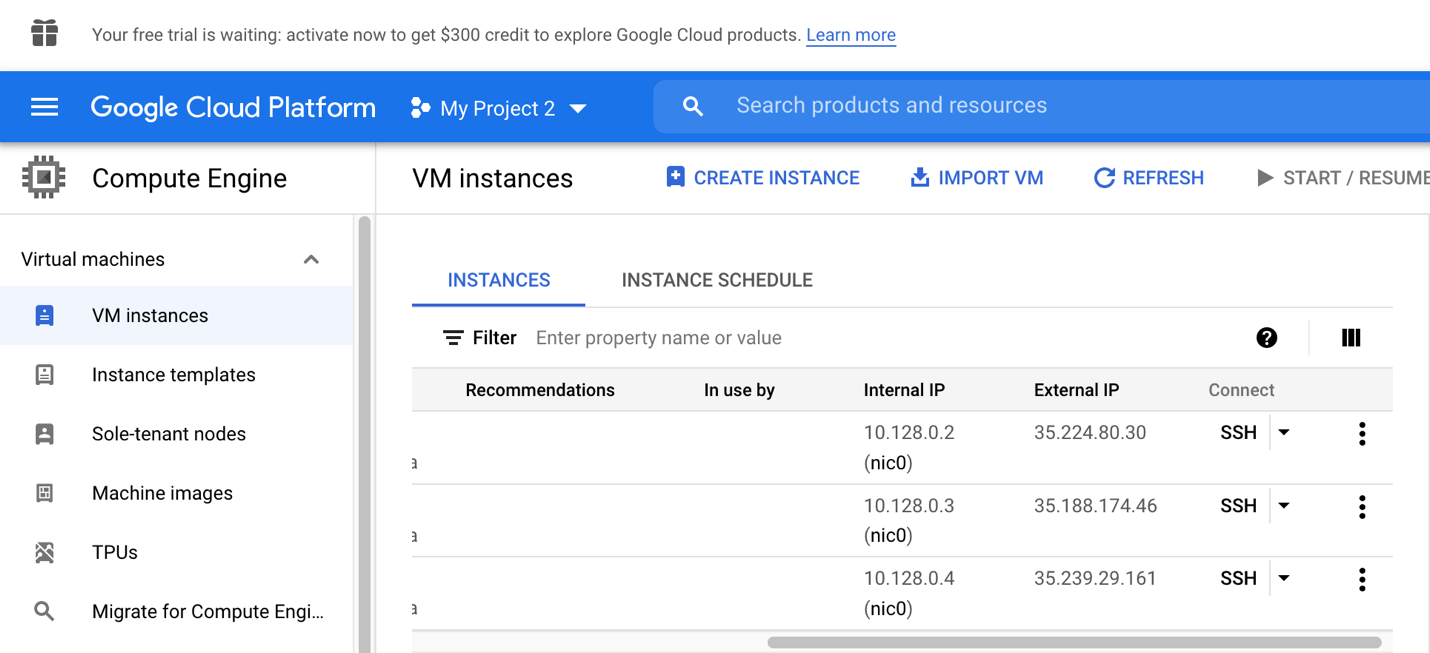
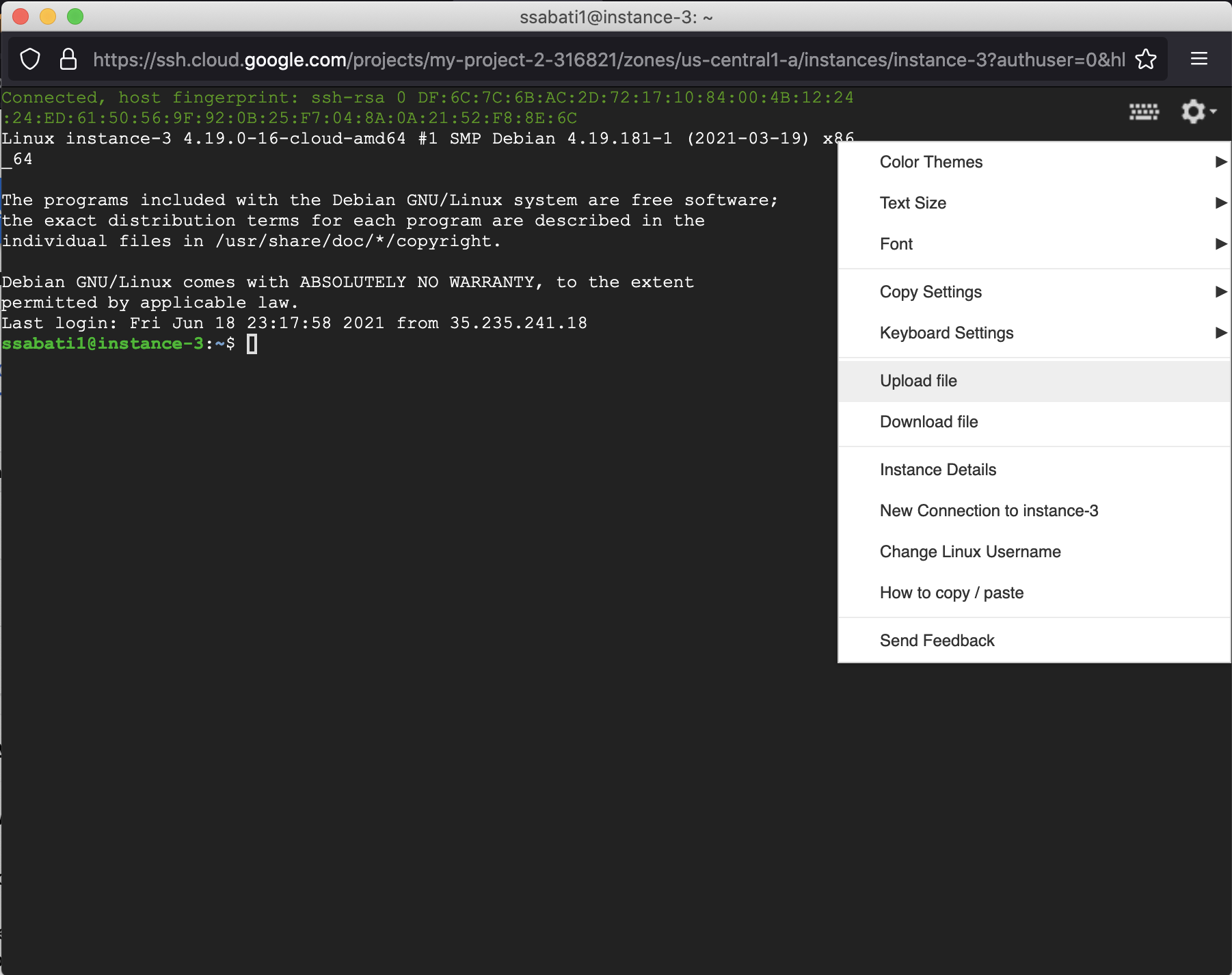
How to run

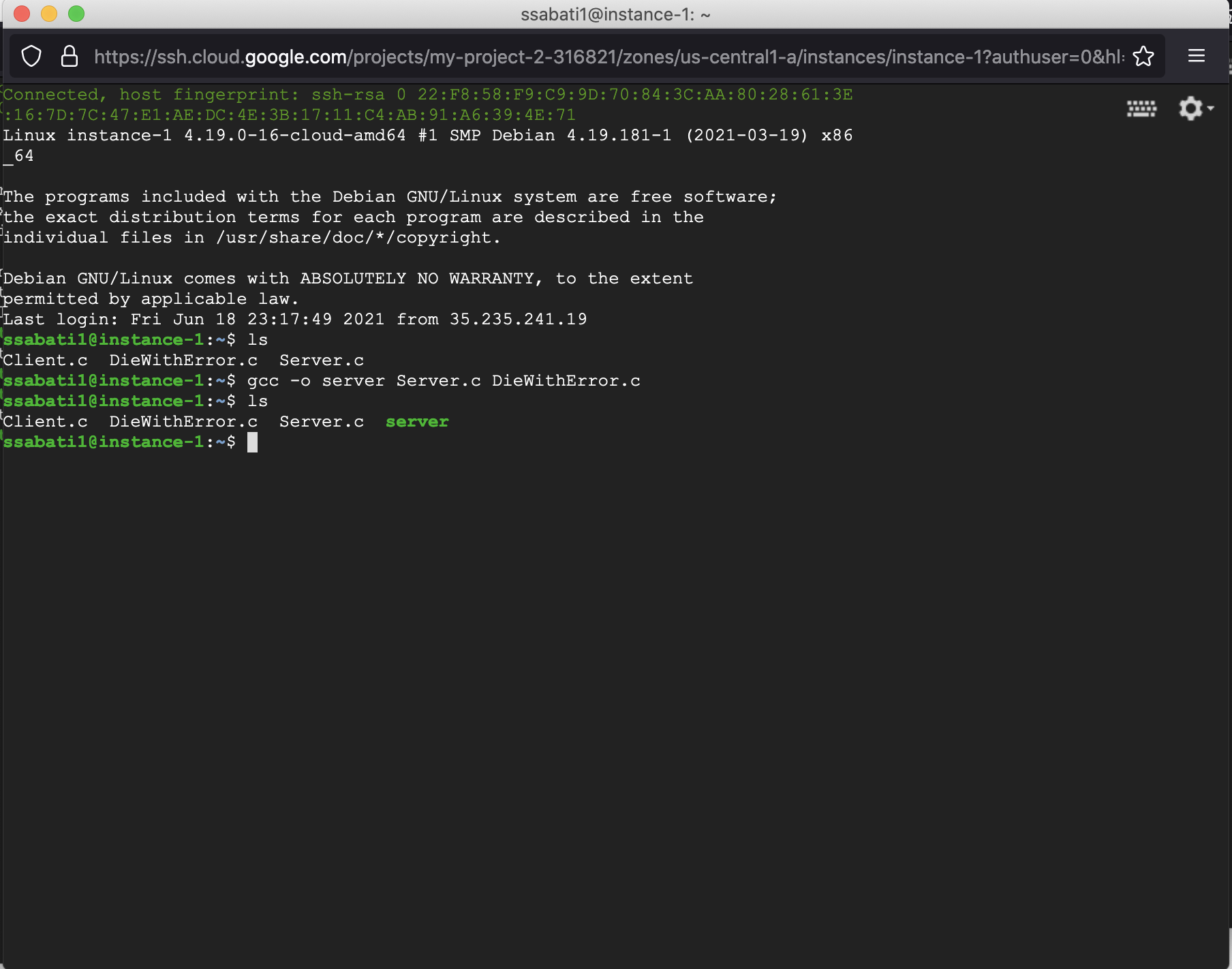
Step 1, create three instances, where one we will use as server and the other two as clients

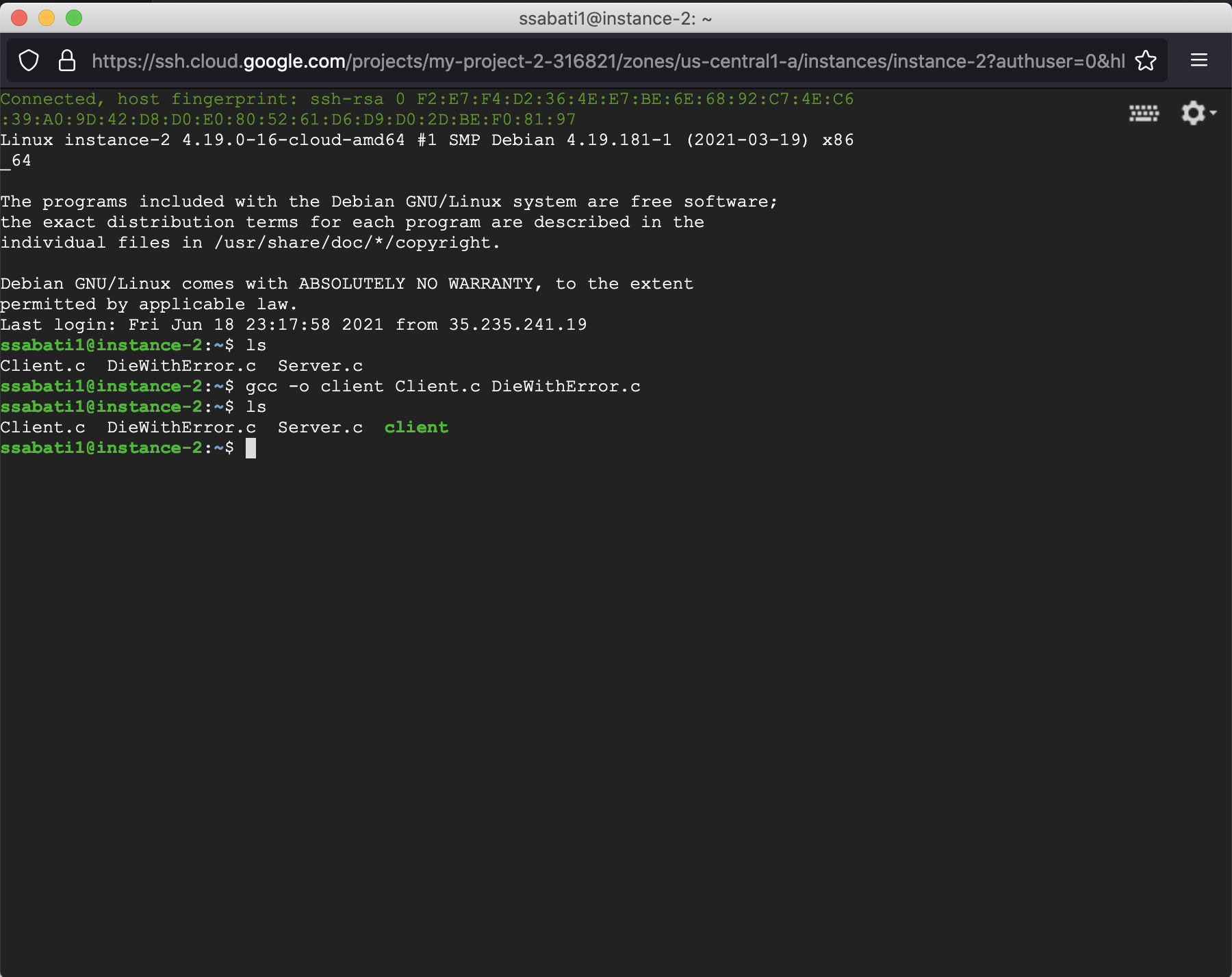


Step 2. Connect to each of them and upload the files Client.c, Server.c, & DieWithError.c



Step 3, once uploaded, Run in instance 1 command gcc -o server Server.c DieWithError.c, and command gcc -o client Client.c DieWithError.c in instance 2 and 3, as shown below

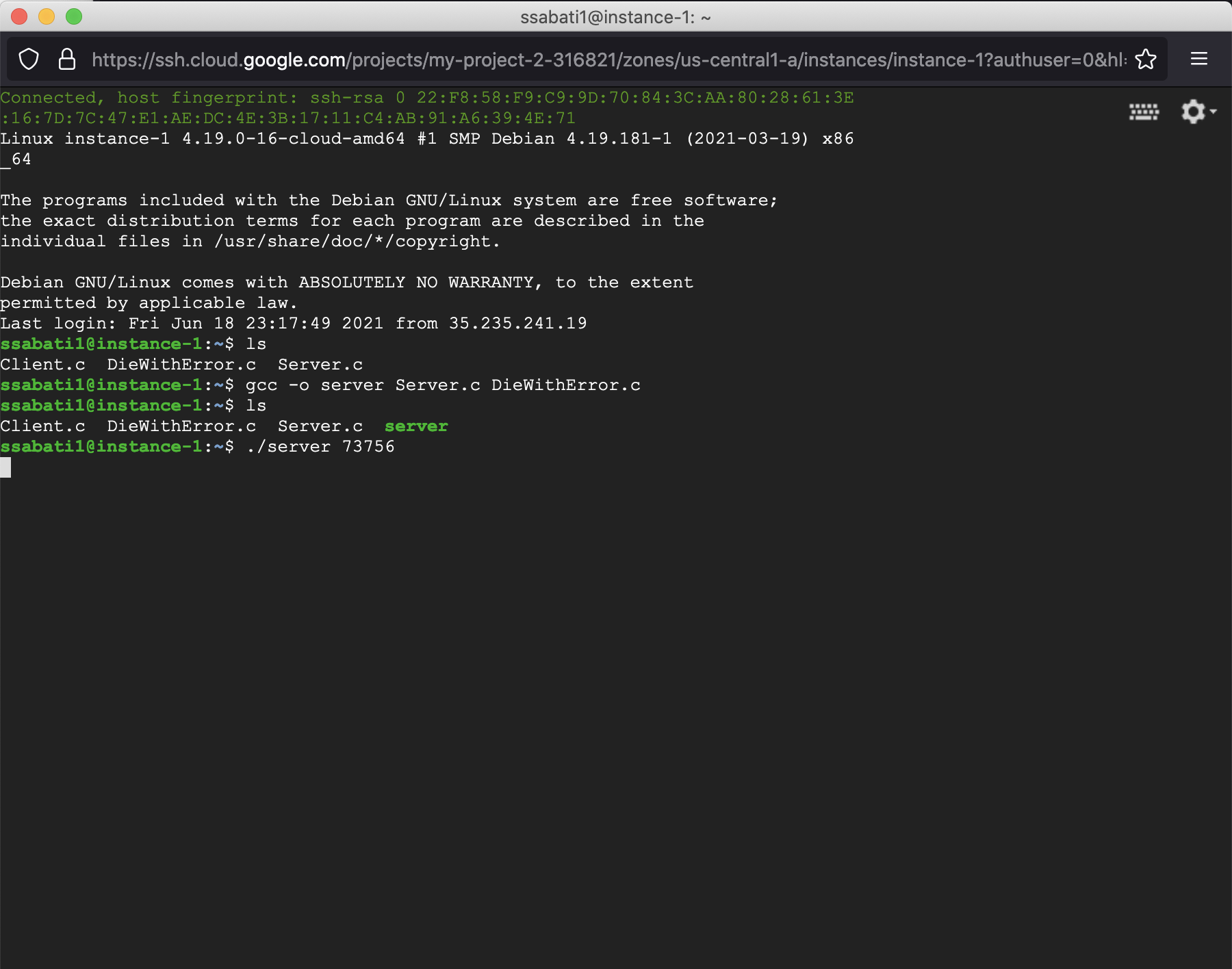


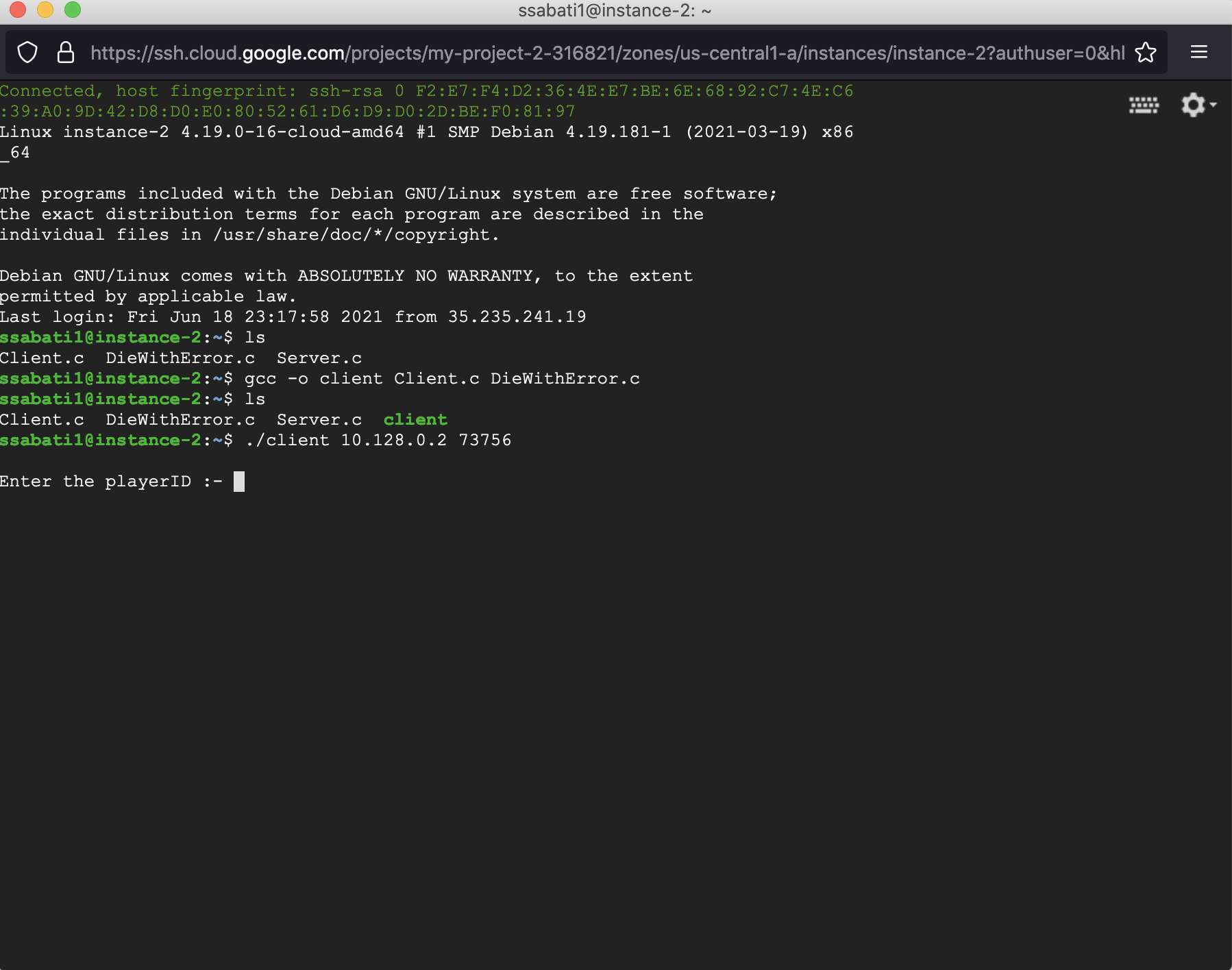


(instance 3 should look like last picture as well)

Step 4, connect to server in instance 1 by running command ./server [Port Number], example ./server 73756, and in instance 2 and 3 connect as clients running the following command

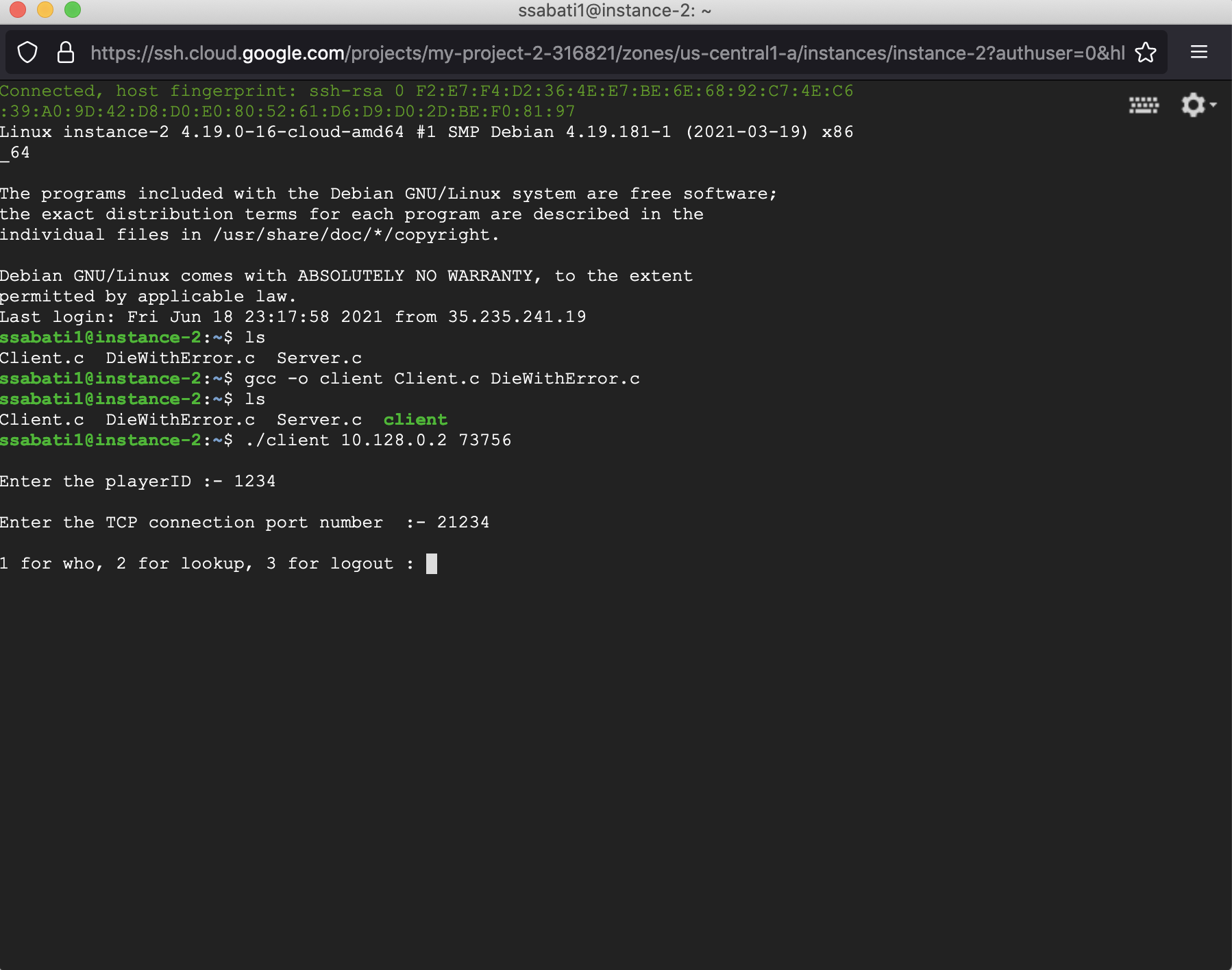
./client [IP address from instance 1] [Port number], example ./client 10.128.0.2 73756, as shown below

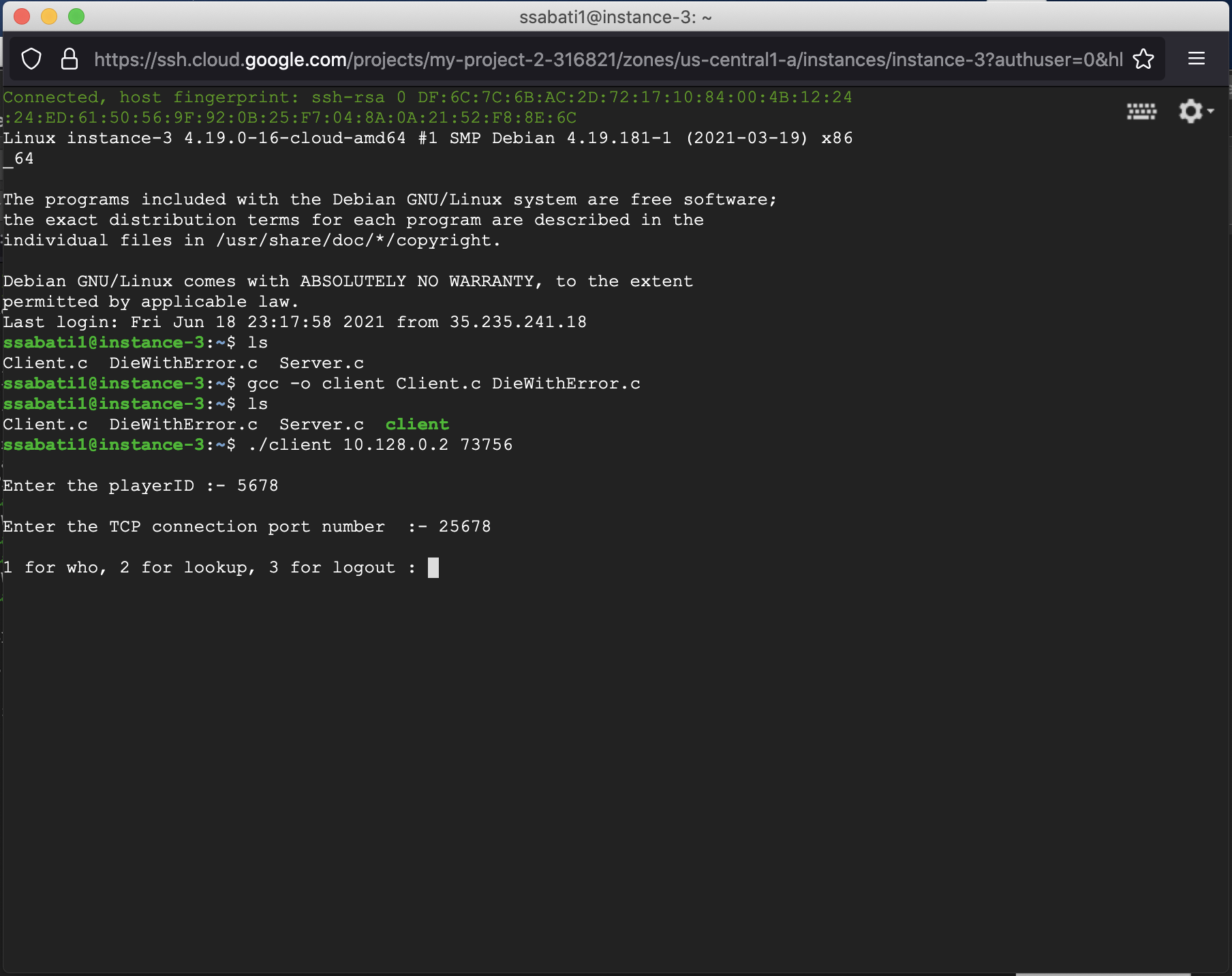




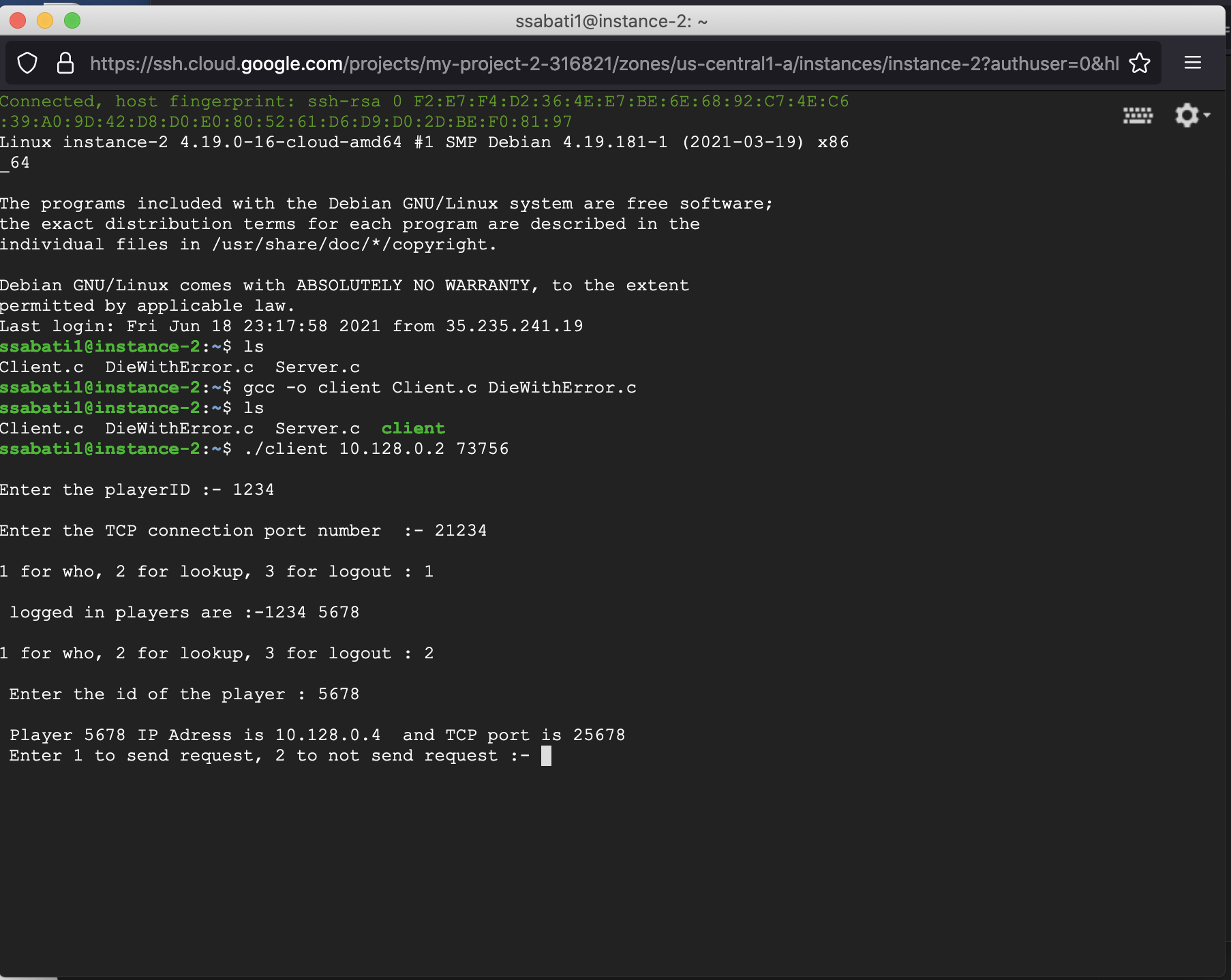
(instance 3 should look like last screenshot)

Step 5, Enter player ID, TCP connection Port number in the clients’ instances, as shown below

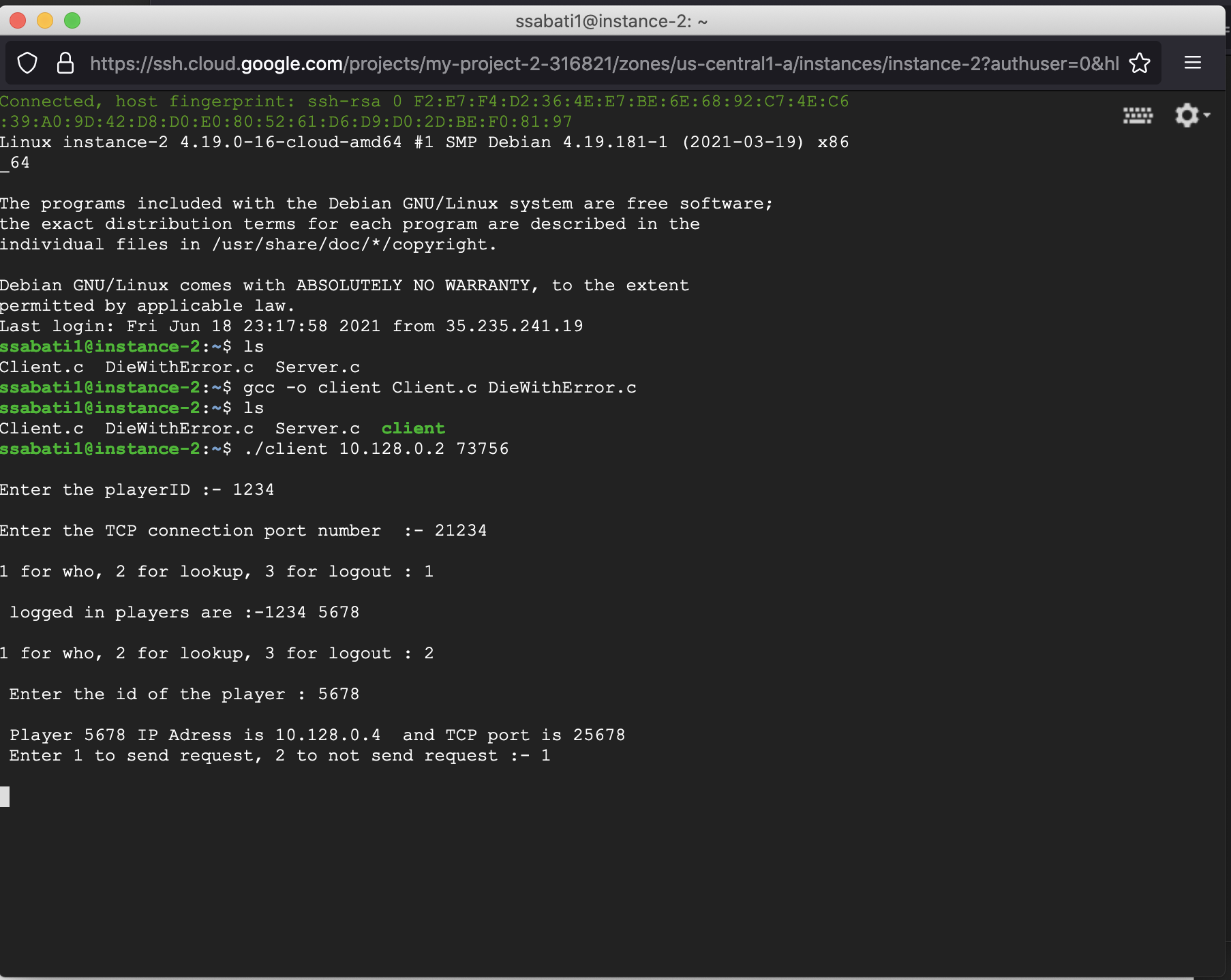




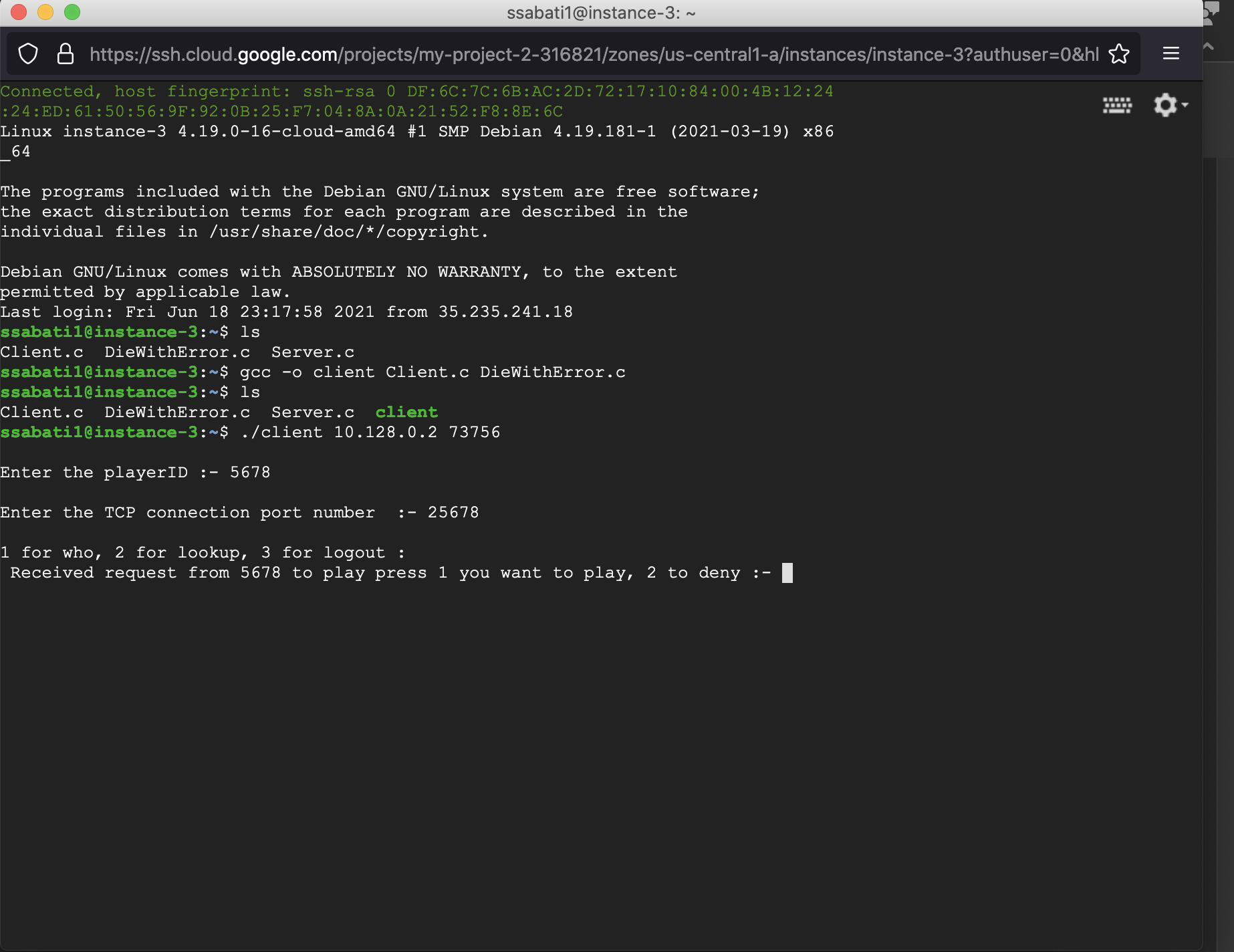
Step 6, shows options 1 (for identifying other players that are logged in), option 2 (for looking player by ID), option 3 to log out. Choosing option 2 is shown below to for looking up the player 5678 that client 1234 wants to play with



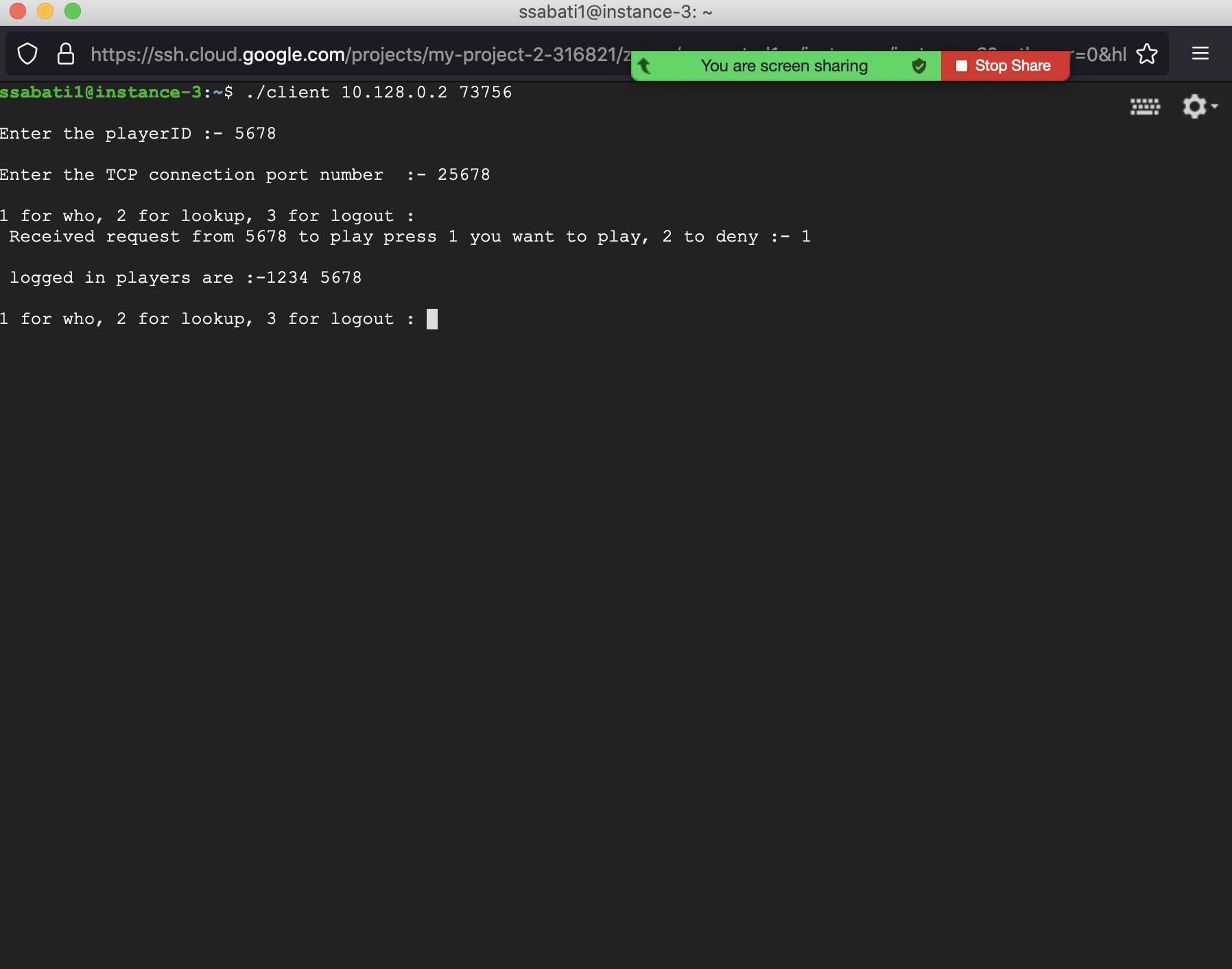
Step 7, Gives option to send request to play by entering 1, or 2 if they don’t want to send request. While client 1234 in instance 2 send request, client 5678 in instance 3 receives the request.



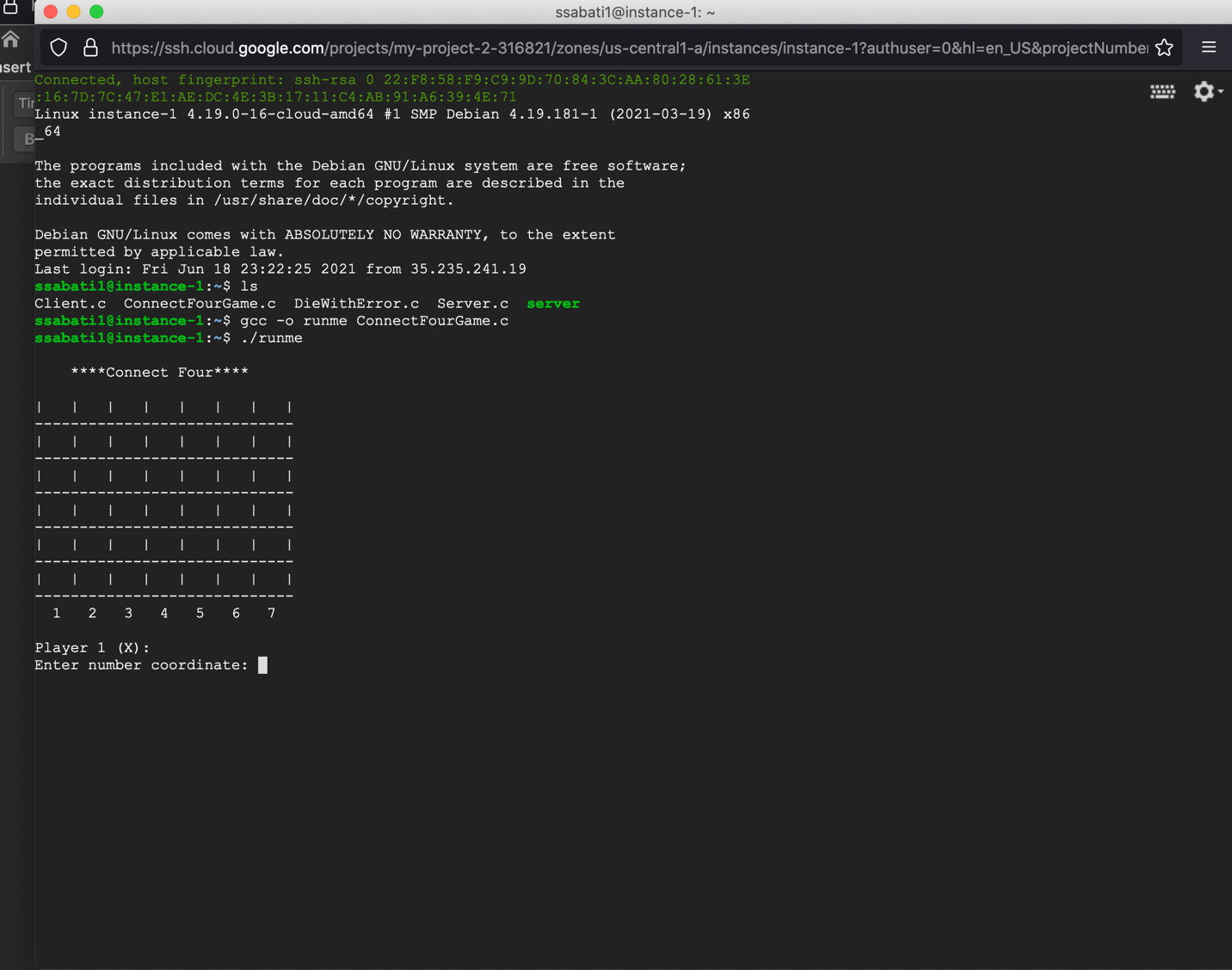
Step 8, Client 1234 was able to send request and 5678 was able to receive request



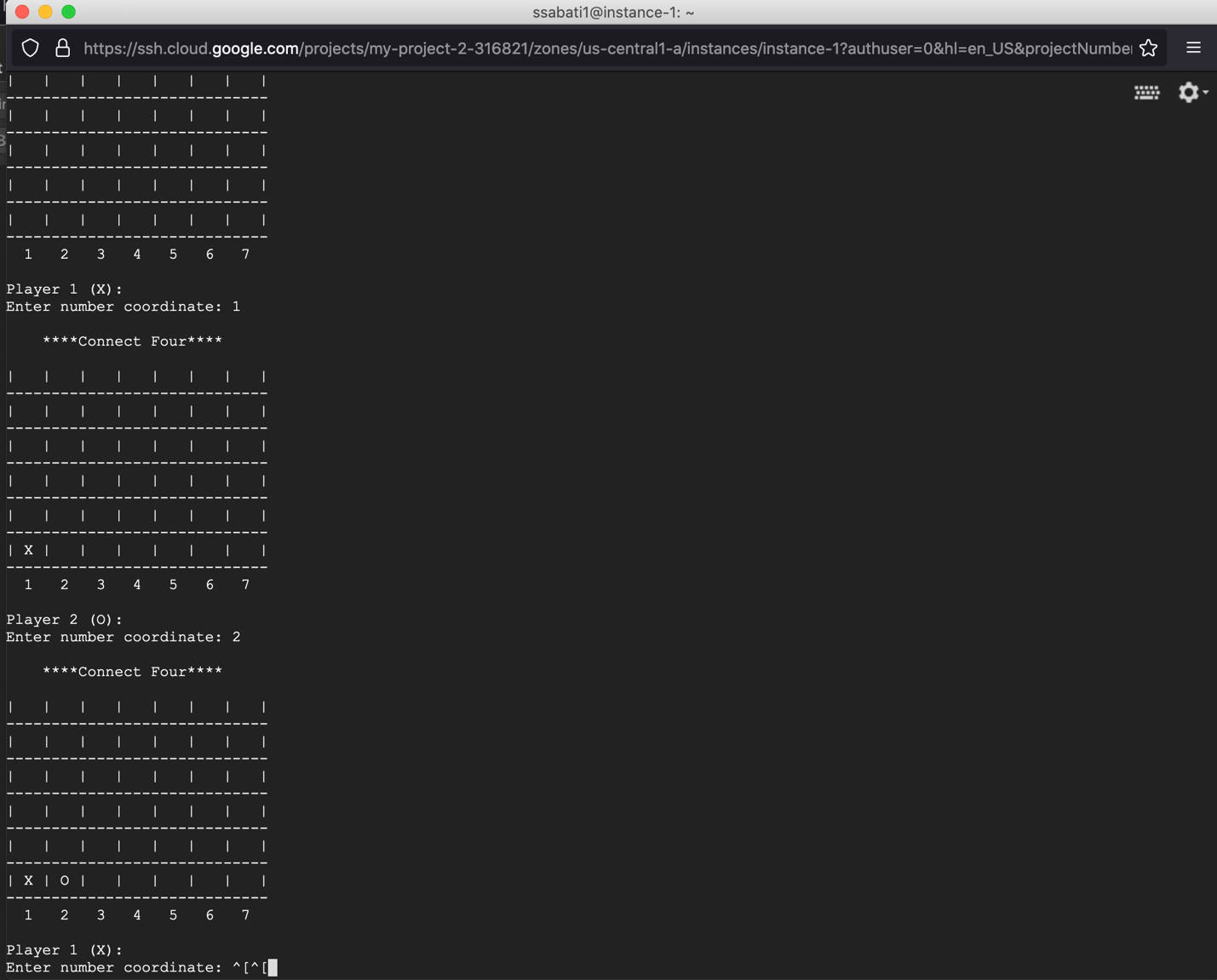
Error we are getting is when client 5678 receives the approval/deny request, the scanf does not work properly, it considers the scanner for the parent process and prints the “who list”



However the game on its on works, please look attached photos



More…



And so on by the time x wins

