Name: Elizabeth Muirhead
Student ID: 30220085

I created a TCP client and server with python that implements the network time protocol to synchronize the client clock with the remote server clock.

Server

The server is capable of holding multiple connections if there are multiple instances of the client. The server must be started first. It listens for incoming connections on port 10085. When the server receives incoming requests, it creates a timestamp (t2), then it sends a response with t2 and another timestamp (t3).
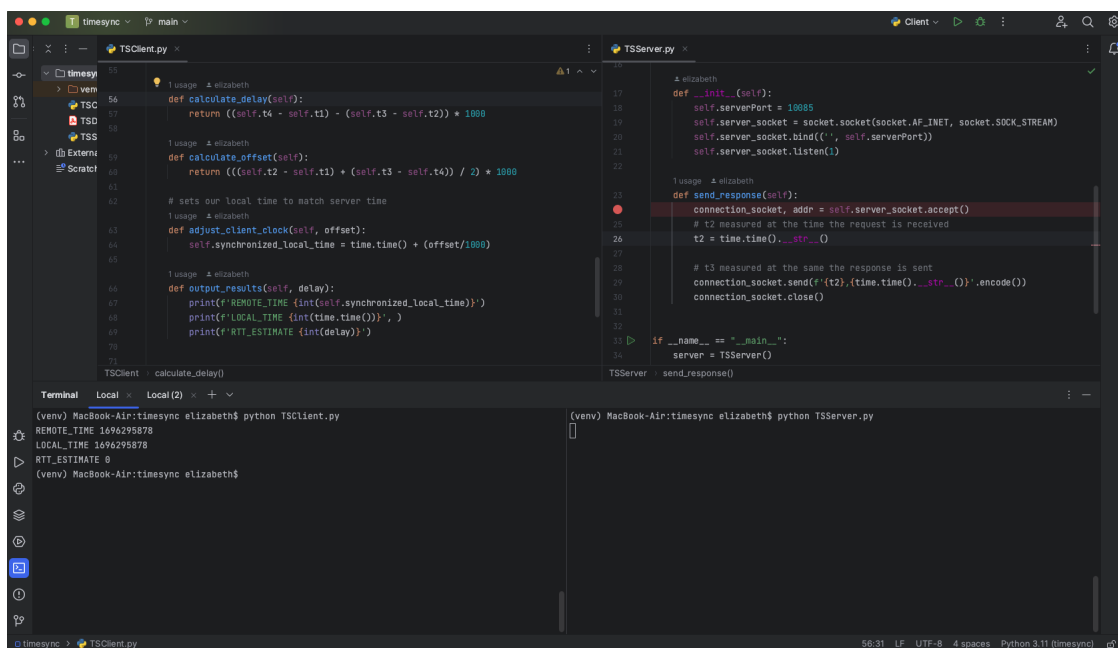
Client

When the client is started, it accepts one command line argument to specify a server name. If a server name is not specified, it defaults to localhost. It initializes a client, i.e. it establishes a TCP socket connection with the server.

Then it creates timestamp 1 (t1), sends a request to the server, and awaits a response. Once the client has received the response, it creates timestamp 4 (t4). Then, it parses the response from the server to get t2 and t3. After all four timestamps have been created, the client uses the network time protocol to calculate the round trip delay and offset.

$$delay = (t4 - t1) - (t3 - t2)$$
$$offset = ((t2 - t1) + (t3 - t4)) / 2$$

Then it creates a local clock and adjusts it by adding the offset. This newly adjusted clock should match the remote time on the server.
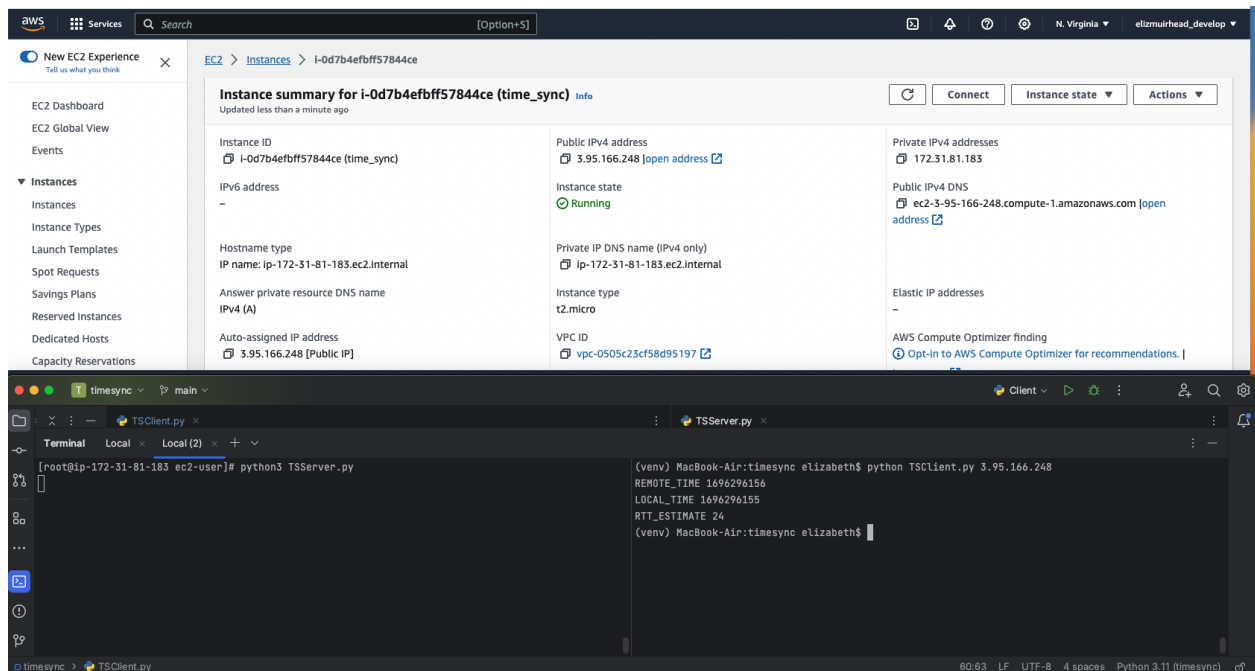

This image shows the client and server running locally.

This image shows the server running in AWS and the client running locally. The server is running on the bottom left hand side. The terminal prefix points to my EC2 instance, and the client on the right is pointing towards the EC2 instances public address.
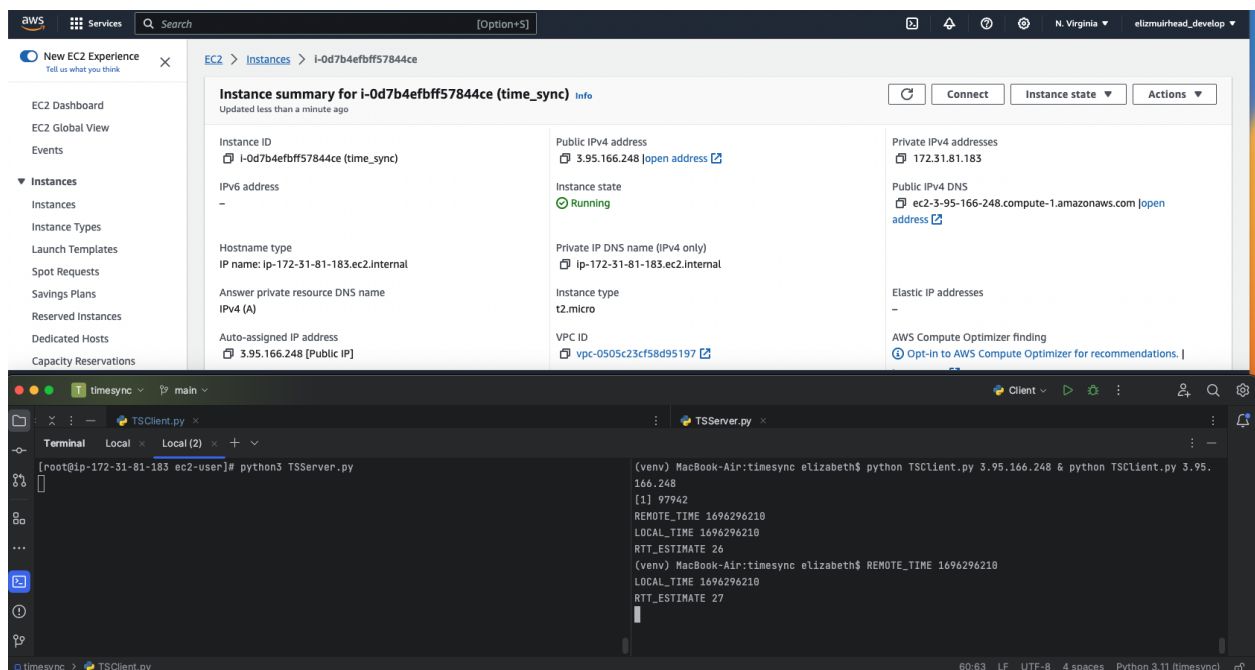
This image shows the server running in AWS and two clients running locally.