Senior Design Fall 2019
Team 4
Client-Server Usage Guide

Luis Bueno, Vagan Grigoryan, Tobias He, Andy Wang

This document details how to run our code. If one hasn't been provided, please request a link to the .zip archive of our project files.

Running Locally

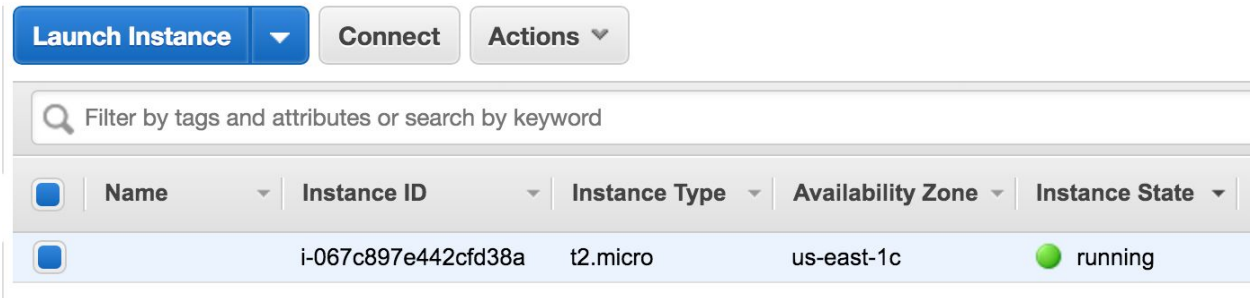
The game can be run locally. In this case, please follow these steps:

1. In terminal (or console on Windows OS), compile the GameServer class within its folder (named Server Side). In MacOS, this command is **javac GameServer.java**
2. In terminal (or console on Windows OS), open a new terminal window. Compile the GameClient class within its folder (named Client Side). In MacOS, this command is **javac GameClient.java**
3. In the first terminal, run the command **java GameServer**
4. In the second terminal, run the command **java GameClient 127.0.0.1** (127.0.0.1 is the IP for localhost, in other words, the first terminal window)
5. The game is now running.

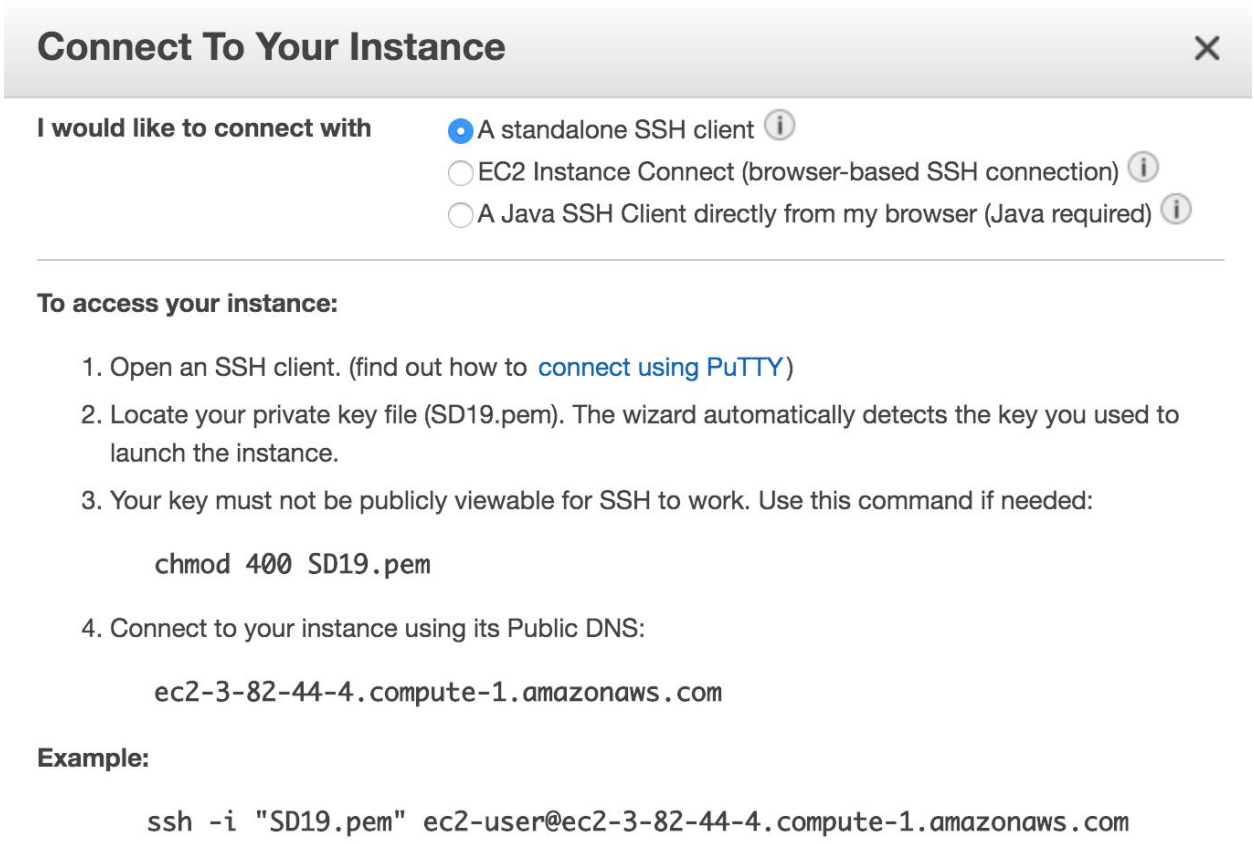
Running Live

The game can be run on a cloud server. We will detail how we set it up with Amazon AWS EC2.

1. We created an Amazon AWS EC2 account. After logging in, we created an EC2 instance of a Linux server that had Java built in.



2. We started the server, and SSH'd in through MacOS terminal using the command at the bottom of this image. Note that the .pem file must be in the directory where this command is run.



3. We put our Server Side code onto the instance, and ran the command **javac GameServer.java**
4. Then we ran the command **java GameServer** which starts the server
5. Then locally on any of our machines, we changed directory into the Client Side folder, and ran **javac GameClient.java** and then **java GameClient <IP>**, where the <IP> is the public IPv4 of the instance. The image below shows where to find this information.

GameServer performance is automatically measured. When a Player disconnects, their stats will be printed out in the terminal that the GameServer is running in. The image below shows this happening.