2.2 Command line tasks:

Java Gradle: The command (gradle tasks --all) lists all the tasks available for Java Gradle. The output has tasks for running the project, build tasks for compiling and testing.

```
$ gradle tasks --all

> Task :tasks

------------------------------------------------------------
Tasks runnable from root project 'JavaGradle'
------------------------------------------------------------

Application tasks
-----------------
run - Runs this project as a JVM application

Build tasks
-----------
assemble - Assembles the outputs of this project.
build - Assembles and tests this project.
buildDependents - Assembles and tests this project and all projects that depend
on it.
buildNeeded - Assembles and tests this project and all projects it depends on.
classes - Assembles main classes.
clean - Deletes the build directory.
jar - Assembles a jar archive containing the main classes.
testClasses - Assembles test classes.

Build Setup tasks
-----------------
init - Initializes a new Gradle build.
wrapper - Generates Gradle wrapper files.

Distribution tasks
------------------
assembleDist - Assembles the main distributions
distTar - Bundles the project as a distribution.
distZip - Bundles the project as a distribution.
installDist - Installs the project as a distribution as-is.

Documentation tasks
-------------------
javadoc - Generates Javadoc API documentation for the main source code.

Fraction Tasks tasks
--------------------
runFraction - Tasks which runs Fraction with no arguments

Help tasks
----------
buildEnvironment - Displays all buildscript dependencies declared in root projec
t 'JavaGradle'.
dependencies - Displays all dependencies declared in root project 'JavaGradle'.
dependencyInsight - Displays the insight into a specific dependency in root proj
ect 'JavaGradle'.
help - Displays a help message.
javaToolchains - Displays the detected java toolchains.
outgoingVariants - Displays the outgoing variants of root project 'JavaGradle'.
projects - Displays the sub-projects of root project 'JavaGradle'.
properties - Displays the properties of root project 'JavaGradle'.
resolvableConfigurations - Displays the configurations that can be resolved in r
oot project 'JavaGradle'.
tasks - Displays the tasks runnable from root project 'JavaGradle'.

Multiply tasks tasks
--------------------
runApp - Tasks which runs Multiply with default parameters
runAppAgain - Tasks which runs Multiply with default parameters or given values

Verification tasks
------------------
check - Runs all checks.
test - Runs the test suite.

Other tasks
-----------
compileJava - Compiles main Java source.
compileTestJava - Compiles test Java source.
components - Displays the components produced by root project 'JavaGradle'. [dep
recated]
dependentComponents - Displays the dependent components of components in root pr
oject 'JavaGradle'. [deprecated]
model - Displays the configuration model of root project 'JavaGradle'. [deprecat
ed]
prepareEmitterBuildcrystalmoon3
processResources - Processes main resources.
processTestResources - Processes test resources.
startScripts - Creates OS specific scripts to run the project as a JVM applicati
on.

Rules
-----
Pattern: clean<TaskName>: Cleans the output files of a task.
Pattern: build<ConfigurationName>: Assembles the artifacts of a configuration.

Deprecated Gradle features were used in this build, making it incompatible with
Gradle 8.0.

You can use '--warning-mode all' to show the individual deprecation warnings and
 determine if they come from your own scripts or plugins.

See https://docs.gradle.org/7.6.4/userguide/command_line_interface.html#sec:comm
and_line_warnings

BUILD SUCCESSFUL in 3s
1 actionable task: 1 executed
```

Gradle Threads: Each line of output starts with Hello from followed by a number and then "loop=" with number of iterations. The output shows that multiple threads are running and printing messages at the same time,

```
algoo@BossHog357 MINGW64 ~/ser321examples/Sockets/JavaSimpleSock2 (master)
$ cd /c/Users/algoo/ser321examples/Threads/FirstThread

algoo@BossHog357 MINGW64 ~/ser321examples/Threads/FirstThread (master)
$ ./gradlew run
bash: ./gradlew: No such file or directory

algoo@BossHog357 MINGW64 ~/ser321examples/Threads/FirstThread (master)
$ gradle run

> Task :run
Hello from 2 loop=0
Hello from 0 loop=0
Hello from 3 loop=0
Hello from 1 loop=0
Hello from 4 loop=0
Hello from 0 loop=1
Hello from 0 loop=2
Hello from 0 loop=3
Hello from 0 loop=4
Hello from 1 loop=1
Hello from 1 loop=2
Hello from 3 loop=1
Hello from 1 loop=3
Hello from 2 loop=1
Hello from 1 loop=4
Hello from 4 loop=1
Hello from 2 loop=2
Hello from 3 loop=2
Hello from 2 loop=3
Hello from 4 loop=2
Hello from 3 loop=3
Hello from 2 loop=4
Hello from 3 loop=4
Hello from 4 loop=3
Hello from 4 loop=4
```

Gradle Task:

The "Configure project:" part that makes custom print statements that are made to display when the project is being set up.

```
al@al-VirtualBox:/media/sf_ser321examples/Gradle/JustGradle$ gradle task1

> Configure project :
Hello task 1
Hello task 2
Hello World
Hello you

> Task :task1
first
last


BUILD SUCCESSFUL in 0s
1 actionable task: 1 executed
al@al-VirtualBox:/media/sf_ser321examples/Gradle/JustGradle$ gradle task2

> Configure project :
Hello task 1
Hello task 2
Hello World
Hello you

> Task :task2
last
first


BUILD SUCCESSFUL in 0s
1 actionable task: 1 executed
al@al-VirtualBox:/media/sf_ser321examples/Gradle/JustGradle$ gradle task1
```

2.3: Understanding Gradle

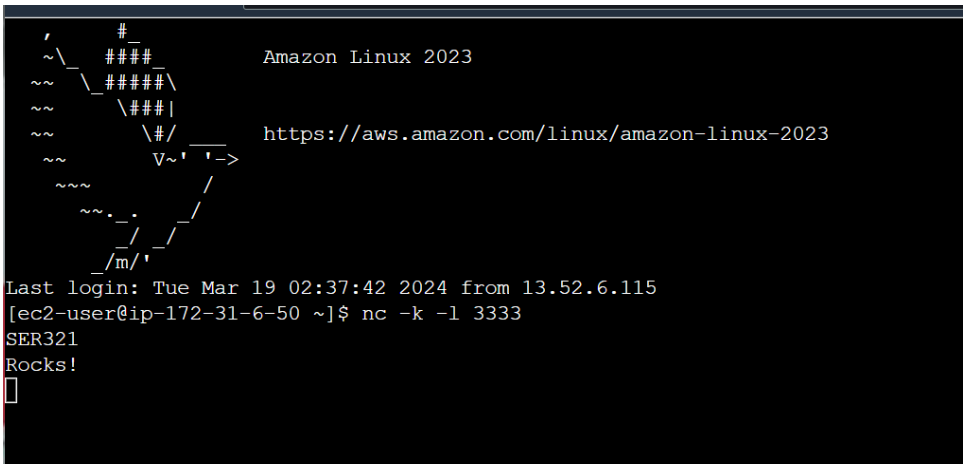
```
al@al-VirtualBox:/media/sf_ser321examples/Gradle/JavaGradle$ gradle runFraction

> Task :runFraction
The fraction is: 1/3
```

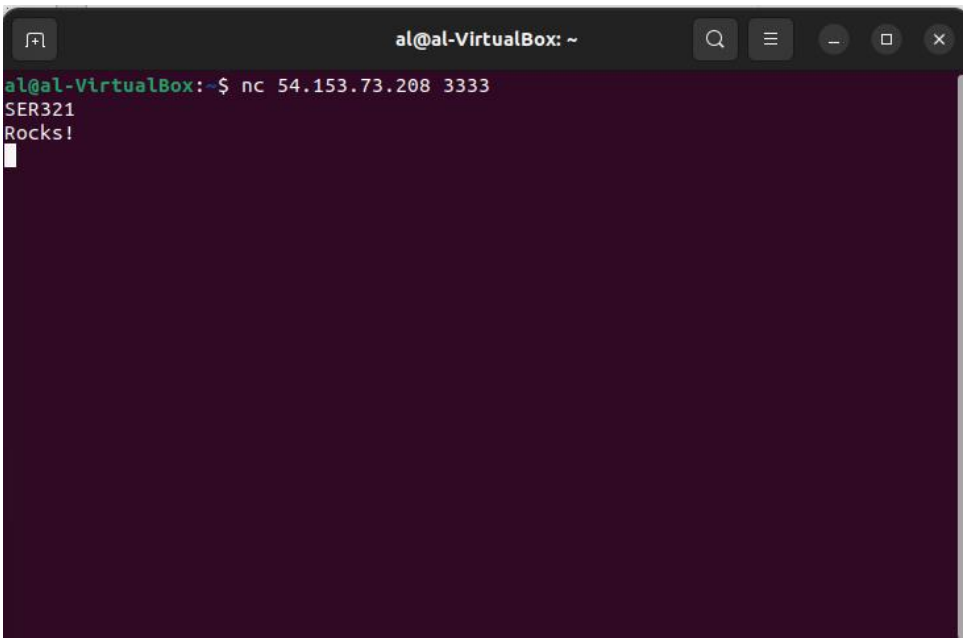
2.4: Socket Server and client run from Linux VirtualBox; IP address set from AWS

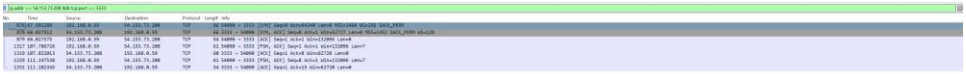
```
[ec2-user@ip-172-31-6-50 JavaSimpleSock2]$ gradle SocketServer

> Task :SocketServer
Server ready for 3 connections
Server waiting for a connection
Received the String HI
Received the Integer 100
Server waiting for a connection
```

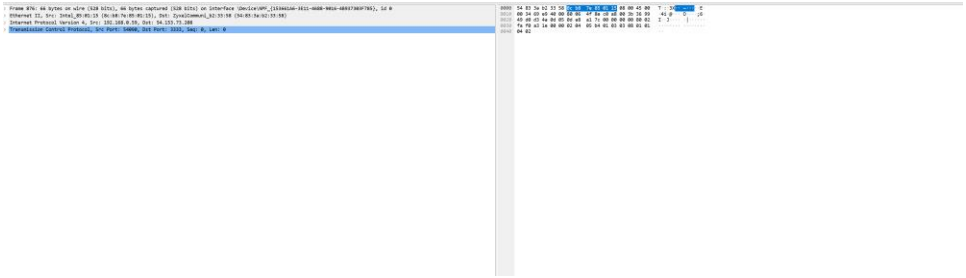
```
al@al-VirtualBox:/media/sf_ser321examples/Sockets/JavaSimpleSock2$ gradle Socket
Client

> Task :SocketClient
Got it!
```

3.2 TCP


Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023
Last login: Tue Mar 19 02:37:42 2024 from 13.52.6.115
[ec2-user@ip-172-31-6-50 ~]$ nc -k -l 3333
SER321
Rocks!


al@al-VirtualBox: ~
al@al-VirtualBox:~$ nc 54.153.73.208 3333
SER321
Rocks!





5a: The c -k -l 3333 command from AWS Instance communicated to nc 54.153.73.208 to relay comments from Linux virtual machine

5b. 4 frames

5c. 4 packets for 4 frames

5d. 4

5e. 7 bytes

5e. 7

3.3.1:

Video link: https://www.kapwing.com/w/iSgZSoogEW

Screen shot of TCP activity

ip.addr == 54.153.73.208 && tcp.port == 3333

No.	Time	Source	Destination	Protocol	Length	Info
181	29.733537	192.168.0.59	54.153.73.208	TCP	62	62335 → 3333 [PSH, ACK] Seq=1 Ack=1 Win=516 Len=8
182	29.767384	54.153.73.208	192.168.0.59	TCP	54	3333 → 62335 [ACK] Seq=1 Ack=9 Win=490 Len=0
190	34.140614	192.168.0.59	54.153.73.208	TCP	61	62335 → 3333 [PSH, ACK] Seq=9 Ack=1 Win=516 Len=7
191	34.174532	54.153.73.208	192.168.0.59	TCP	54	3333 → 62335 [ACK] Seq=1 Ack=16 Win=490 Len=0
202	38.180745	192.168.0.59	54.153.73.208	TCP	59	62335 → 3333 [PSH, ACK] Seq=16 Ack=1 Win=516 Len=5
203	38.216060	54.153.73.208	192.168.0.59	TCP	54	3333 → 62335 [ACK] Seq=1 Ack=21 Win=490 Len=0
719	90.792428	192.168.0.59	54.153.73.208	TCP	58	62335 → 3333 [PSH, ACK] Seq=21 Ack=1 Win=516 Len=4
720	90.827193	54.153.73.208	192.168.0.59	TCP	54	3333 → 62335 [ACK] Seq=1 Ack=25 Win=490 Len=0
735	103.501785	192.168.0.59	54.153.73.208	TCP	81	62335 → 3333 [PSH, ACK] Seq=25 Ack=1 Win=516 Len=27
736	103.536595	54.153.73.208	192.168.0.59	TCP	54	3333 → 62335 [ACK] Seq=1 Ack=52 Win=490 Len=0
850	117.948612	192.168.0.59	54.153.73.208	TCP	91	62335 → 3333 [PSH, ACK] Seq=52 Ack=1 Win=516 Len=37
851	117.982544	54.153.73.208	192.168.0.59	TCP	54	3333 → 62335 [ACK] Seq=1 Ack=89 Win=490 Len=0
900	122.276741	192.168.0.59	54.153.73.208	TCP	68	62335 → 3333 [PSH, ACK] Seq=89 Ack=1 Win=516 Len=14
912	122.310538	54.153.73.208	192.168.0.59	TCP	54	3333 → 62335 [ACK] Seq=1 Ack=103 Win=490 Len=0
924	126.086890	192.168.0.59	54.153.73.208	TCP	69	62335 → 3333 [PSH, ACK] Seq=103 Ack=1 Win=516 Len=15
925	126.120324	54.153.73.208	192.168.0.59	TCP	54	3333 → 62335 [ACK] Seq=1 Ack=118 Win=490 Len=0
1047	158.124922	192.168.0.59	54.153.73.208	TCP	81	62335 → 3333 [PSH, ACK] Seq=118 Ack=1 Win=516 Len=27
1048	158.158306	54.153.73.208	192.168.0.59	TCP	54	3333 → 62335 [ACK] Seq=1 Ack=145 Win=490 Len=0
1156	170.437178	192.168.0.59	54.153.73.208	TCP	77	62335 → 3333 [PSH, ACK] Seq=145 Ack=1 Win=516 Len=23
1157	170.470401	54.153.73.208	192.168.0.59	TCP	54	3333 → 62335 [ACK] Seq=1 Ack=168 Win=490 Len=0
1234	182.596348	192.168.0.59	54.153.73.208	TCP	60	62335 → 3333 [PSH, ACK] Seq=168 Ack=1 Win=516 Len=6
1235	182.631279	54.153.73.208	192.168.0.59	TCP	54	3333 → 62335 [ACK] Seq=1 Ack=174 Win=490 Len=0
1297	193.732378	192.168.0.59	54.153.73.208	TCP	79	62335 → 3333 [PSH, ACK] Seq=174 Ack=1 Win=516 Len=25
1298	193.766521	54.153.73.208	192.168.0.59	TCP	54	3333 → 62335 [ACK] Seq=1 Ack=199 Win=490 Len=0
5020	350.794695	192.168.0.59	54.153.73.208	TCP	66	62335 → 3333 [PSH, ACK] Seq=199 Ack=1 Win=516 Len=12
5021	350.834371	54.153.73.208	192.168.0.59	TCP	60	3333 → 62335 [ACK] Seq=1 Ack=211 Win=490 Len=0
5081	354.179194	192.168.0.59	54.153.73.208	TCP	63	62335 → 3333 [PSH, ACK] Seq=211 Ack=1 Win=516 Len=9
5082	354.212682	54.153.73.208	192.168.0.59	TCP	54	3333 → 62335 [ACK] Seq=1 Ack=220 Win=490 Len=0
6918	552.324671	192.168.0.59	54.153.73.208	TCP	66	62335 → 3333 [PSH, ACK] Seq=220 Ack=1 Win=516 Len=12[Malformed Packet]
6919	552.363468	54.153.73.208	192.168.0.59	TCP	60	3333 → 62335 [ACK] Seq=1 Ack=232 Win=490 Len=0

For the gradle call to run I had to type gradle SocketServer on one terminal and Gradle SocketClient from the shared folder.

3.3.3

There could be issues with the firewall blocking the IP address, the ports would have to be set correctly for it to run.

3.3.4:

It would be different since the IP addresses are not the same, it would have to connect from the AWS IP Instance. For the server to connect the ports would have to send incoming traffic to a private IP address,