Apache Flink Getting Started

Experiment 6: Simple Flink Socket

1.1 Steps to run your next Flink Program

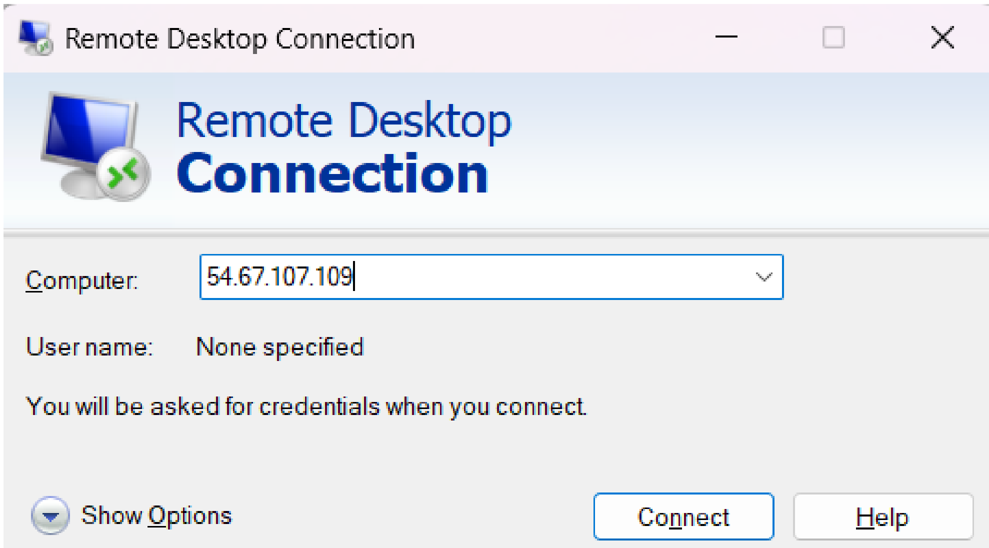
1.1.1 Browse to the GitHub repo that you cloned. This should be cloned to your Windows Jump Box and the Flink Development Server
https://github.com/GeorgeNiece/flink-data-processing-2day

1.1.2 From a command prompt on your jumpbox machine SSH to the Unbuntu server

```
ssh -o ServerAliveInterval=180 -o ServerAliveCountMax=2 -i ansible.pem ubuntu@ip_address_provided
```

1.1.4 Change to the flink folder, verify Flink isn't started, start the Flink dev cluster, and verify that it started

```
ps -ef  | grep flink
cd ~/flink-2.0.0
./bin/start-cluster.sh
ps -ef | grep flink
```

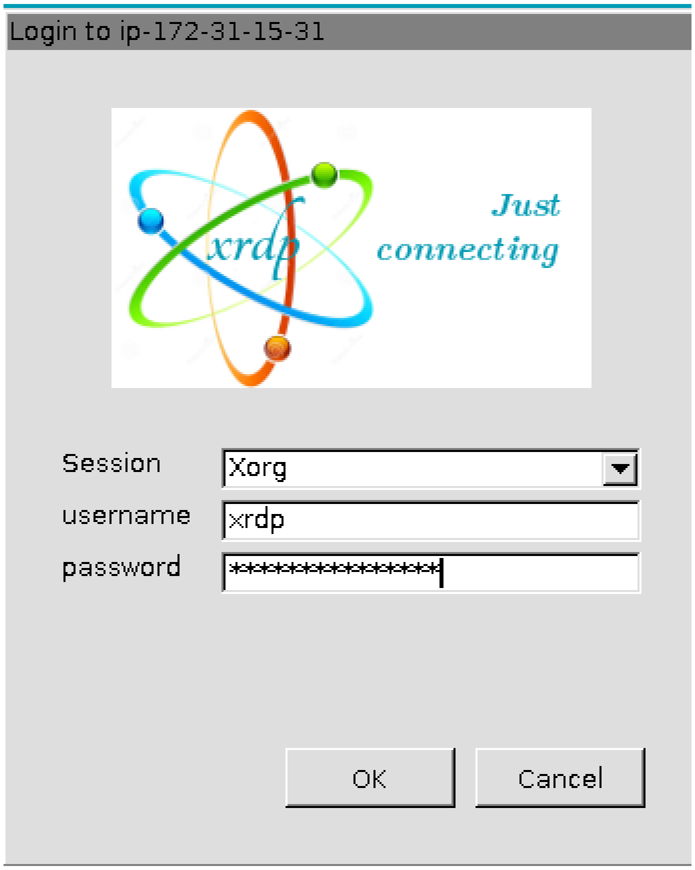


Computer: 54.67.107.109

User name: None specified

You will be asked for credentials when you connect.

1.1.5 Login to the ubuntu dev sandbox using Windows RDP with the xrdp user and the password that you set in Step 1.14



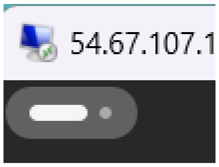
Login to ip-172-31-15-31

Session: Xorg

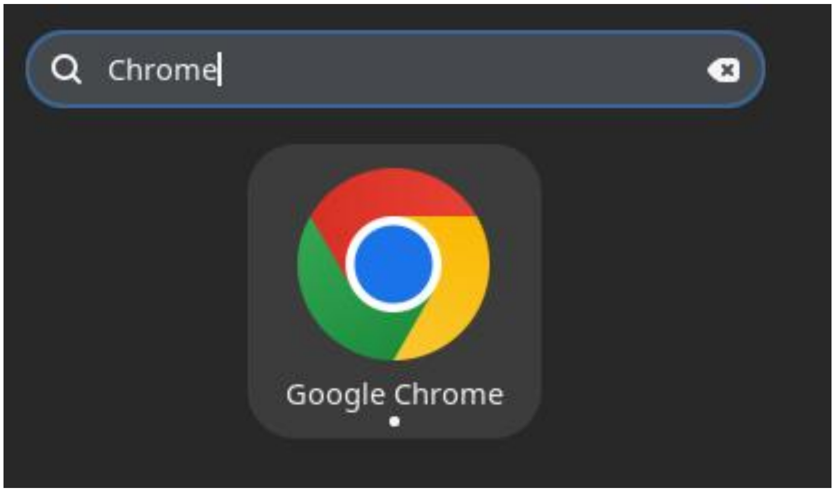
username: xrdp

password: ****************

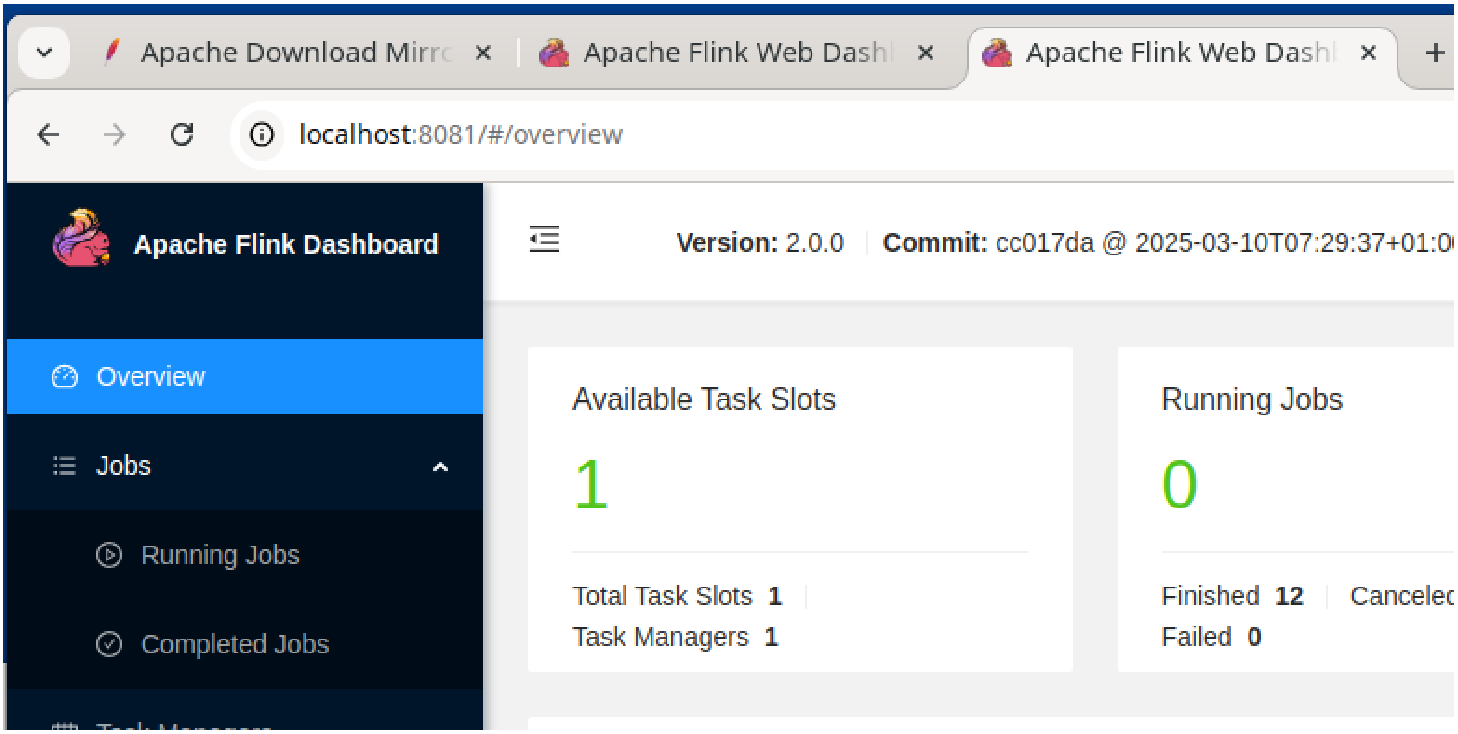
1.1.6 Click the Activities button in the top left corner of the Ubuntu Desktop



1.1.7 Wait for the Search Box at the top of the Ubuntu Desktop, and enter Chrome, click on the Launch Logo



1.1.8 Load the Flink Web UI at localhost:8081



Version: 2.0.0 | **Commit:** cc017da @ 2025-03-10T07:29:37+01:0

Available Task Slots

1

Total Task Slots **1**

Task Managers **1**

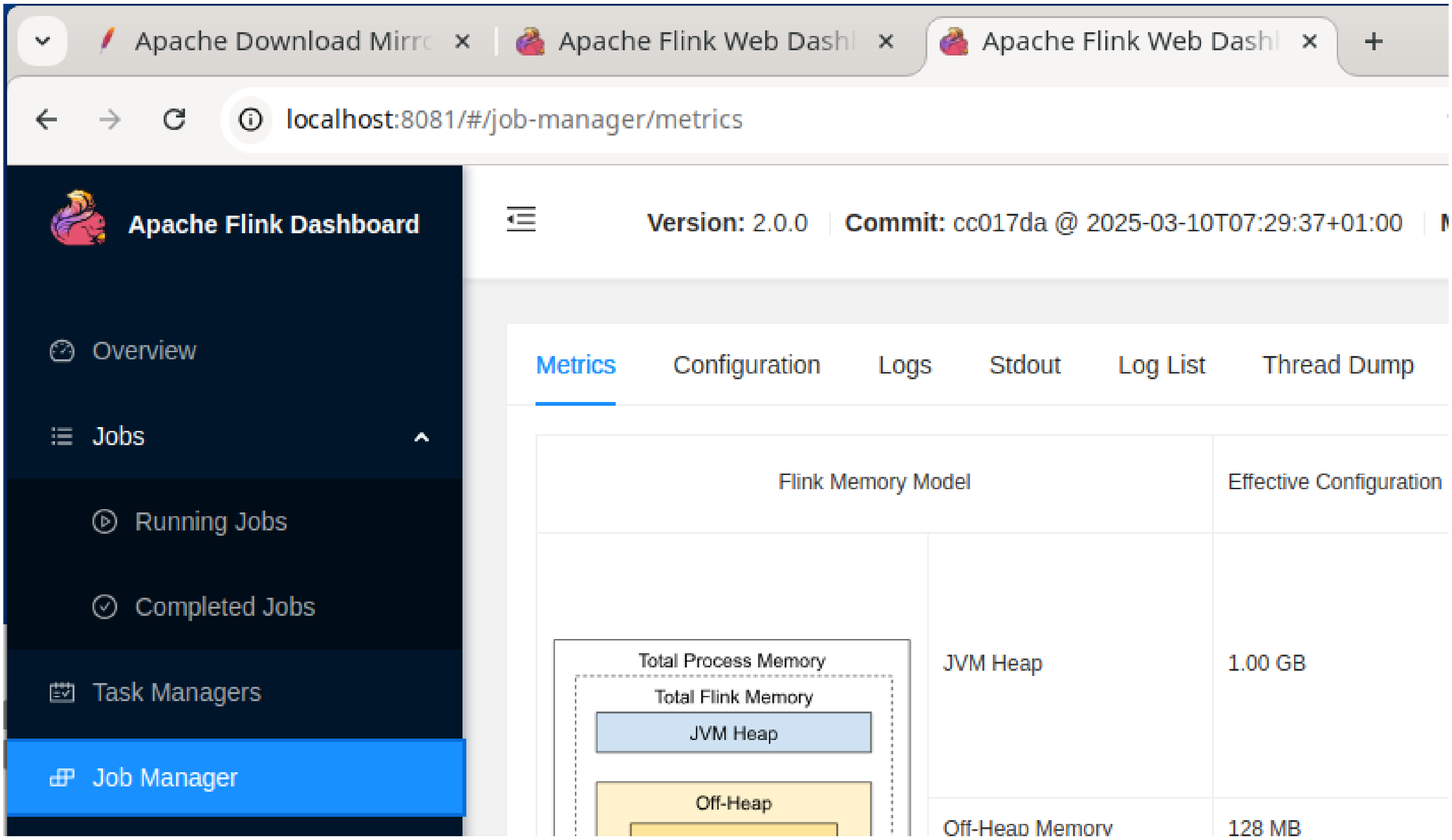
Running Jobs

0

Finished **12** | Canceled

Failed **0**

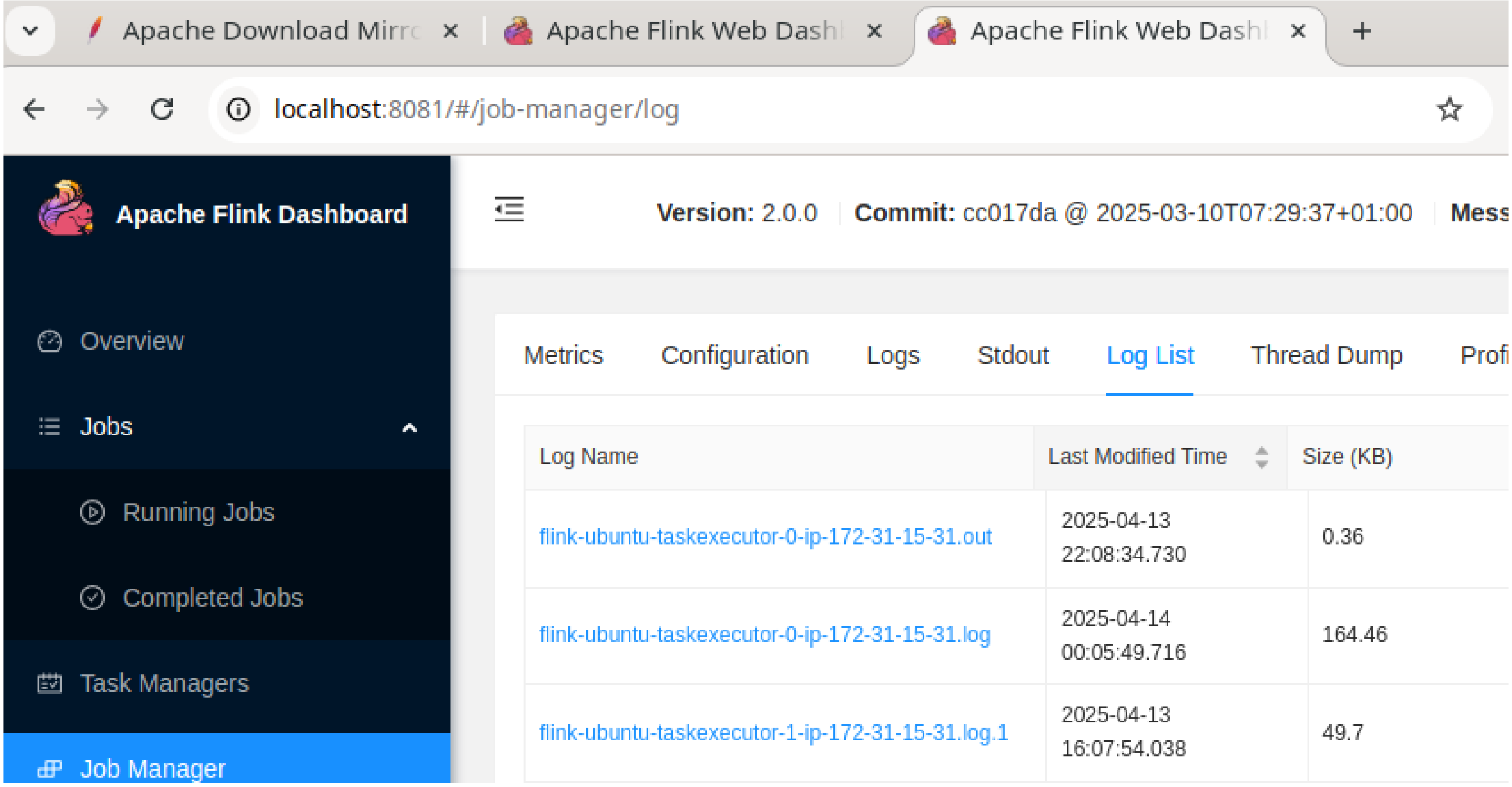
1.1.9 Click the Job Manager in the left hand navigation



Version: 2.0.0 | **Commit:** cc017da @ 2025-03-10T07:29:37+01:00

Flink Memory Model		Effective Configuration
	JVM Heap	1.00 GB
	Off-Heap Memory	128 MB

1.1.10 Select Log List in the page navigation. These are the logs we'll monitor while we're running some of our experiments



Version: 2.0.0 | **Commit:** cc017da @ 2025-03-10T07:29:37+01:00 | **Mess**

Log Name	Last Modified Time	Size (KB)
flink-ubuntu-taskexecutor-0-ip-172-31-15-31.out	2025-04-13 22:08:34.730	0.36
flink-ubuntu-taskexecutor-0-ip-172-31-15-31.log	2025-04-14 00:05:49.716	164.46
flink-ubuntu-taskexecutor-1-ip-172-31-15-31.log.1	2025-04-13 16:07:54.038	49.7

1.1.11 The first flink-ubuntu-taskexecutor*.out file will be the one we spend the most time looking at. Select that so that we can view there. We could open two browser tabs and watch the Jobs -> Running Jobs page while we run our first experiment.

1.1.12 Navigate back to the SSH terminal to the flink distribution folder

Start and additional terminal and run the netcat to start our provider

```
~:~$ nc -l 9999
```

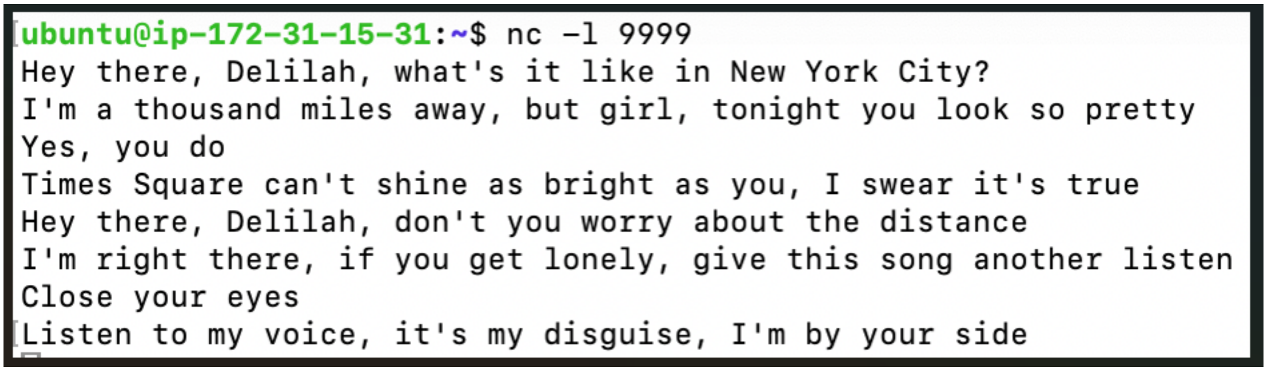
From our original terminal

```
cd ~/flink-2.0.0
./bin/flink run ~/flink-data-processing-2day/experiments/built/SimpleFlinkSocket.jar
```

Now we type into our netcat terminal any sentences or words we choose. We'll see those echo'd out to our output log.

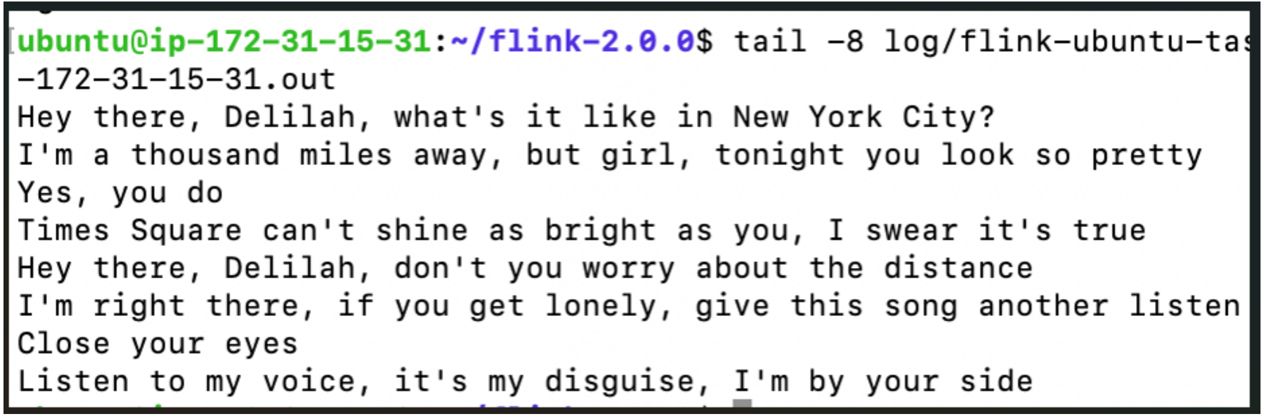
```
Hey there, Delilah, what's it like in New York City?
I'm a thousand miles away, but girl, tonight, you look so pretty
Yes, you do
Times Square can't shine as bright as you, I swear it's true
Hey there, Delilah, don't you worry about the distance
```

I'm right there if you get lonely, give this song another listen
Close your eyes
Listen to my voice, it's my disguise, I'm by your side



```
ubuntu@ip-172-31-15-31:~$ nc -l 9999
Hey there, Delilah, what's it like in New York City?
I'm a thousand miles away, but girl, tonight you look so pretty
Yes, you do
Times Square can't shine as bright as you, I swear it's true
Hey there, Delilah, don't you worry about the distance
I'm right there, if you get lonely, give this song another listen
Close your eyes
Listen to my voice, it's my disguise, I'm by your side
```

View the **flink-ubuntu-taskexecutor*.out,** this time we'll just tail the file which shows us the following. For my sandbox environment that would be
~/flink-2.0.0:$ **tail log/flink-ubuntu-taskexecutor-0-ip-172-31-78-140.out**
Which shows us the following



```
ubuntu@ip-172-31-15-31:~/flink-2.0.0$ tail -8 log/flink-ubuntu-tas
-172-31-15-31.out
Hey there, Delilah, what's it like in New York City?
I'm a thousand miles away, but girl, tonight you look so pretty
Yes, you do
Times Square can't shine as bright as you, I swear it's true
Hey there, Delilah, don't you worry about the distance
I'm right there, if you get lonely, give this song another listen
Close your eyes
Listen to my voice, it's my disguise, I'm by your side
```

We can view the logs to see the last written with the unix command `ls -alrt`

Cleanup Note: Stopping the netcat by Ctrl-C or killing that additional terminal will stop our running flink job, otherwise it will run indefinitely.

1.2 Steps to build your next Flink Program

1.2.1 We have the source for the Flink program in both the flink-project structure and the executable jar. **SimpleFlinkSocket** uses the DataStream API to create a socket source that eats any data from an existing socket. This simple version just relays those sentences to the stdout in our flink-*.out files in the log folder of our Flink Developer Sandbox.

1.2.2 To compile from command line with Java you would need to reference the flink distribution jar files in your classpath, either directly as noted here or in a build tool like Maven or Gradle. A sample POM file is included in our course GitHub repo in the flink-project

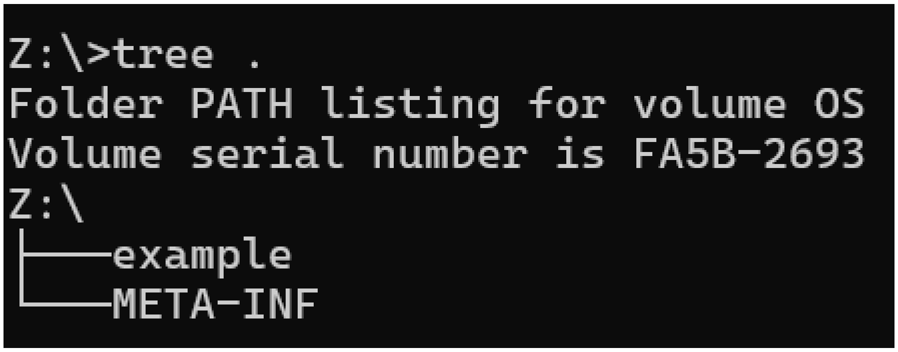
1.2.3 To compile from command line with Java you would need to reference the flink distribution jar files in your classpath, either directly as noted here or in a build tool like Maven or Gradle. A sample POM file is included in our course GitHub repo in the flink-project

```
javac -classpath C:\lib-2.0\flink-cep-2.0.0.jar;C:\lib-2.0\flink-
connector-files-2.0.0.jar;C:\lib-2.0\flink-csv-2.0.0.jar;C:\lib-
2.0\flink-dist-2.0.0.jar;C:\lib-2.0\flink-json-2.0.0.jar;C:\lib-
2.0\flink-scala_2.12-2.0.0.jar;C:\lib-2.0\flink-table-api-java-
uber-2.0.0.jar;C:\lib-2.0\flink-table-planner-loader-
2.0.0.jar;C:\lib-2.0\flink-table-runtime-2.0.0.jar;C:\lib-
2.0\log4j-1.2-api-2.24.1.jar;C:\lib-2.0\log4j-api-
2.24.1.jar;C:\lib-2.0\log4j-core-2.24.1.jar;C:\lib-2.0\log4j-
slf4j-impl-2.24.1.jar;C:\lib-2.0\flink-streaming-java-
1.20.1.jar;C:\lib-2.0\flink-runtime-2.0.0.jar;
example/SimpleFlinkSocket.java
```

1.2.4 To package the executable jar for the Flink program we create a MANIFEST.MF that we'll use in the packaging, notice that the

```
Manifest-Version: 1.0
Implementation-Title: Flink : Examples : Simple Stream
Implementation-Version: 2.0.0
Archiver-Version: Plexus Archiver
Built-By: geoniece
Specification-Vendor: Innovation in Software
Specification-Title: Flink : Examples : Simple Stream
Implementation-Vendor-Id: com.innovationinsoftware
program-class: example.SimpleFlinkSocket
Implementation-Vendor: Innovation in Software
Created-By: Apache Maven 3.8.6
Build-Jdk: 1.11.0_312
Specification-Version: 2.0.0
```

1.2.5 We have a folder structure with our package



```
Z:\>tree .
Folder PATH listing for volume OS
Volume serial number is FA5B-2693
Z:\
├───example
└───META-INF
```

1.2.6 To package the executable jar for our Flink program we do the following

```
Ubuntu-@ip-172.15.50.23:~$  jar --manifest=META-INF/MANIFEST.MF --create --file c:\users\Geo\SimpleFlinkSocket.jar example/*
```

1.2.7 **Congratulations, time to celebrate** you ran another Flink program in our session