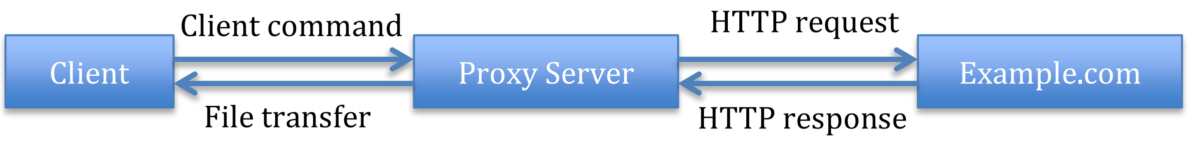
Homework 1: A Proxy Server for Downloading files (Due Oct. 21)

SOLUTIONS ARE LISTED ON THE BOTTOM PAGES

Overview

In this homework, you will use socket programming to develop a proxy server that helps a client program download a file. The server program will be running on a remote server. The client program will connect to the server program, and send a command with the target file's URL. The proxy server will fetch the file and transfer it to the client program.



Homework Task

You need to develop two programs: (1) HW1Server.java and (2) HW1Client.java with the following functions/requirements. "*" indicates the requirements for graduate students. This homework is a practice on socket programming. For the network connection, you must use ServerSocket and Socket class. Other high-level Java classes related to network connection are NOT allowed, such as HttpURLConnection and URL.

- HW1Server.java

1. The server program should be launched as

```
java HW1Server portNumber
```

where the argument portNumber specifies the port of the proxy server.

2. The server program has to support multiple concurrent clients. (refer to MultiThreadEchoServer example)

3. Once a client command is received, the server program needs to first print out the client's IP address. And then, the server program identifies the target HTML file (host and file name) and creates another socket to the remote server following HTTP protocol. The request line and one header line are required. For example,

```
GET /index.html HTTP/1.1
Host: example.com
```

4. The server program will get a HTTP response from the remote server including a status line, some header lines, and the data. The server program needs to transfer the data back to the client.

5. *The server program needs to save a copy of the HTML file received from the web server, named as "proxy-HTML_FILENAME". e.g., in the example above, the target HTML file is "index.html", and the server needs to save the contents to a file "proxy-index.html".

- HW1Client.java

1. The client program should be launched as
   ```
   java HW1Client host portNumber
   ```
where the two argument specify the host and port of the server program.

2. After the connection is established, the user can type in a command with the target file's URL that will be sent to the server, e.g.,
   ```
   GET example.com/index.html
   ```
3. Next, the client will start to receive data from the proxy server and save it in a local file.

4. *Your client program needs to show the downloading progress (any form is OK).

Local Testing

You can test and debug your client and server programs on the same computer by using "localhost" as the host name.

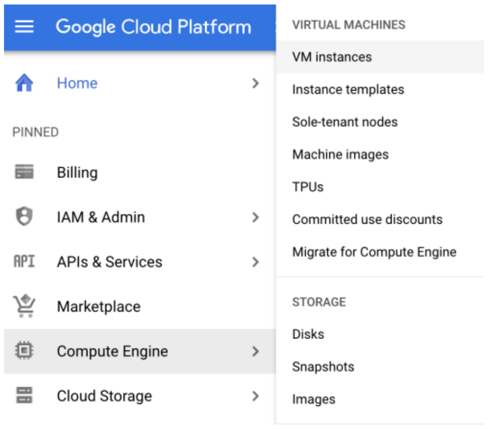
1. Run your server program in a terminal,
   ```
   $java HW1Server 9999
   ```
2. Run your client program in another terminal, e.g.,
   ```
   $java HW1Client localhost 9999
   ```
3. Type in a command from the client's program, e.g.,
   ```
   GET info.cern.ch/hypertext/WWW/Summary.html
   ```
4. Confirm the file is downloaded at the client's side, e.g.,
   ```
   $cat Summary.html
   $cat proxy-Summary.html
   ```
   (for graduate students)

Cloud Platform Testing

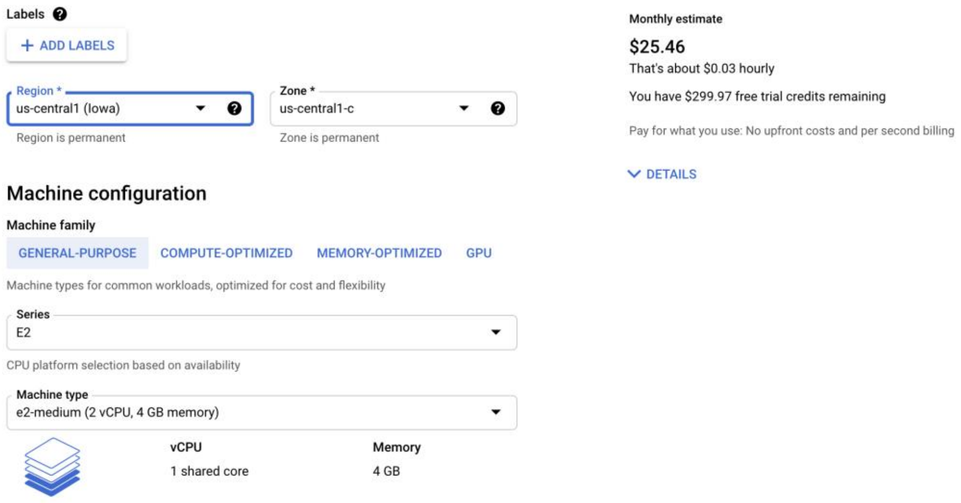
Eventually, you need to test the proxy server program on a remote server hosted on either Google Cloud Platform or CloudLab.

Google Cloud Platform:

1. Create a VM on GCP by choosing "Computer Engine" -> "VM instances"



2. Use the web interface to configure the VM, e2-medium or e2-small are suitable for our test. You may also change the "Region" to pick one with the lowest hourly rate.



Labels

+ ADD LABELS

Region *: us-central1 (Iowa)
Region is permanent

Zone *: us-central1-c
Zone is permanent

Machine configuration

Machine family: GENERAL-PURPOSE | COMPUTE-OPTIMIZED | MEMORY-OPTIMIZED | GPU

Machine types for common workloads, optimized for cost and flexibility

Series: E2
CPU platform selection based on availability

Machine type: e2-medium (2 vCPU, 4 GB memory)

vCPU	Memory
1 shared core	4 GB

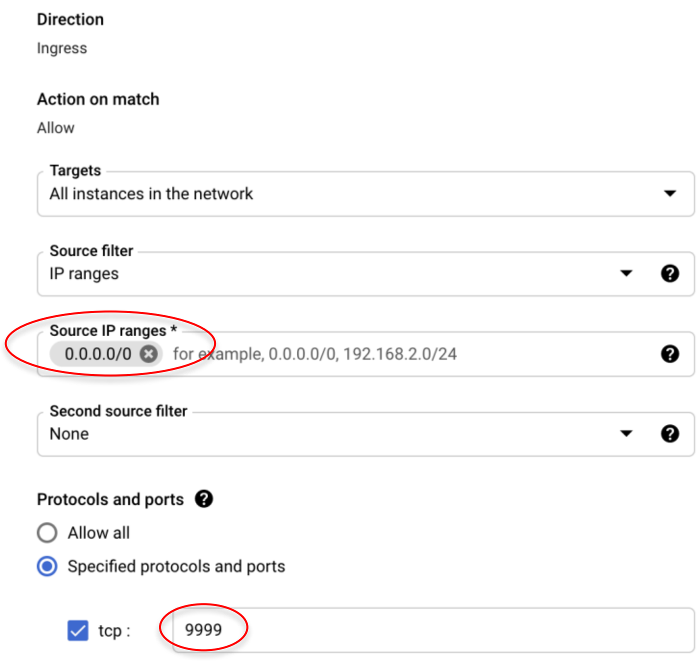
Monthly estimate
$25.46
That's about $0.03 hourly
You have $299.97 free trial credits remaining
Pay for what you use: No upfront costs and per second billing

DETAILS

3. After the instance is created, you can use the web terminal to login, and install JDK,

```
$ sudo apt install default-jdk
```

4. You need to configure the firewall rule for the VM to allow incoming TCP traffic to the port number your server program will use.


Direction
Ingress

Action on match
Allow

Targets: All instances in the network

Source filter: IP ranges

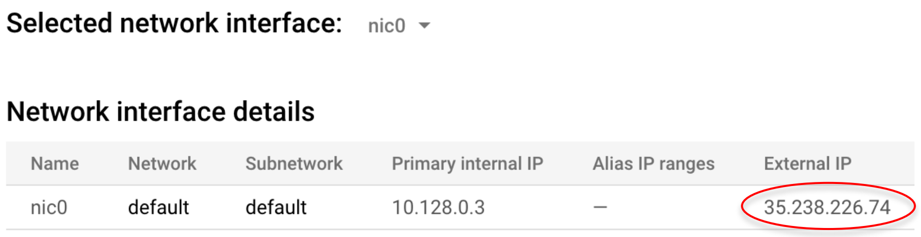
Source IP ranges *: 0.0.0.0/0 — for example, 0.0.0.0/0, 192.168.2.0/24

Second source filter: None

Protocols and ports
- Allow all
- [✓ Specified protocols and ports]

[✓] tcp : 9999

5. Then compile HW1Server.java on the VM and repeat the steps in "Local Testing" and replace "localhost" in step 2 by the IP address of the VM on GCP.


Selected network interface: nic0

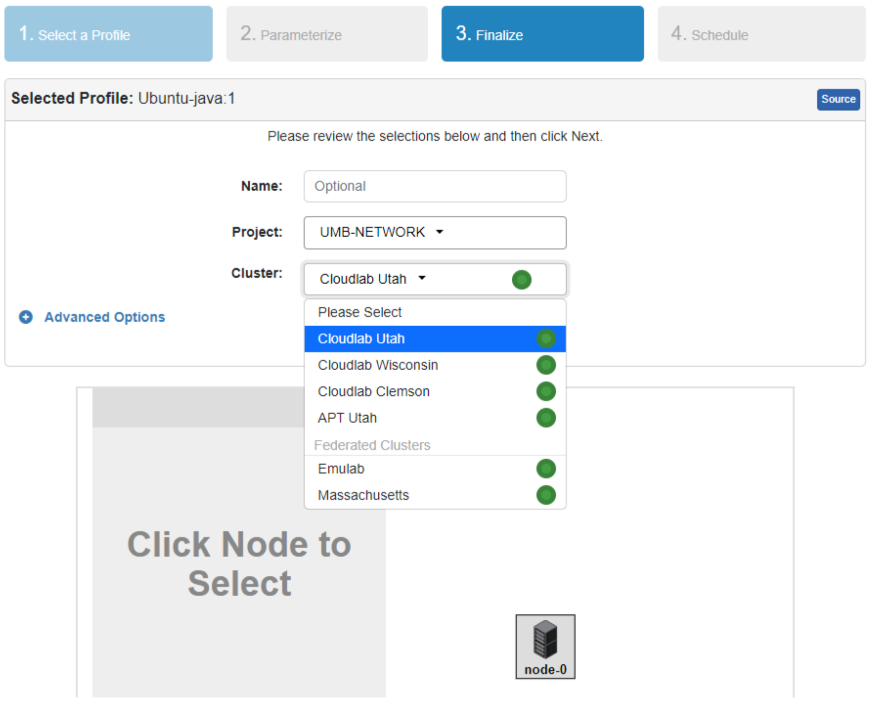
Network interface details

Name	Network	Subnetwork	Primary internal IP	Alias IP ranges	External IP
nic0	default	default	10.128.0.3	—	35.238.226.74

6. Remember to terminate the instance after your tests.

<u>CloudLab Platform:</u>

Manual: https://docs.cloudlab.us/
1. After login, choose "Experiments", then "Start Experiment"
2. Click "Change Profile", and search "Ubuntu-java" profile in UMB-NETWORK group.
3. Then choose "Project", and a cluster site to run the experiment, e.g., "Cloudlab Utah"


1. Select a Profile | 2. Parameterize | 3. Finalize | 4. Schedule

Selected Profile: Ubuntu-java:1

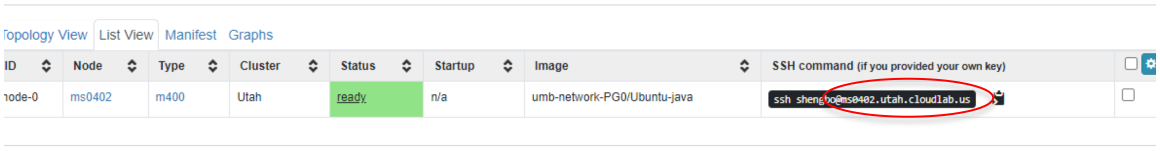
Please review the selections below and then click Next.

Name: Optional

Project: UMB-NETWORK

Cluster: Cloudlab Utah

4. You can find the VM's domain name from "list view", and login via the web shell or your computer's terminal application.



ID	Node	Type	Cluster	Status	Startup	Image	SSH command (if you provided your own key)
node-0	ms0402	m400	Utah	ready	n/a	umb-network-PG0/Ubuntu-java	ssh shengbo@ms0402.utah.cloudlab.us

Submission

Submit the following files on Blackboard:

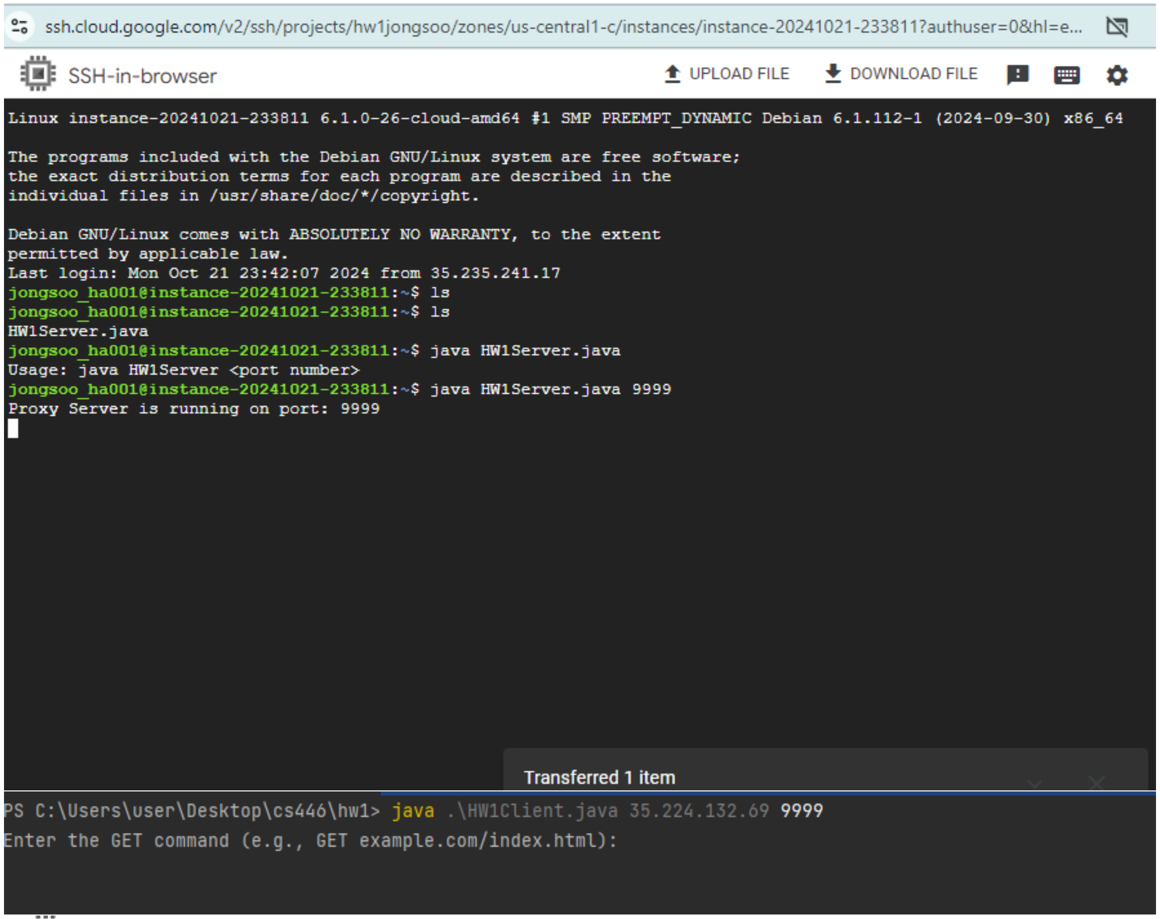
1. the two java files ***HW1Server.java*** and ***HW1Client.java*** ;
2. a word or pdf file that includes the screen snapshots showing the programs are tested with GCP.

Reference

EchoClient.java, EchoServer.java, MultiThreadEchoServer.java

https://docs.oracle.com/javase/10/docs/api/java/net/Socket.html

Screenshots:



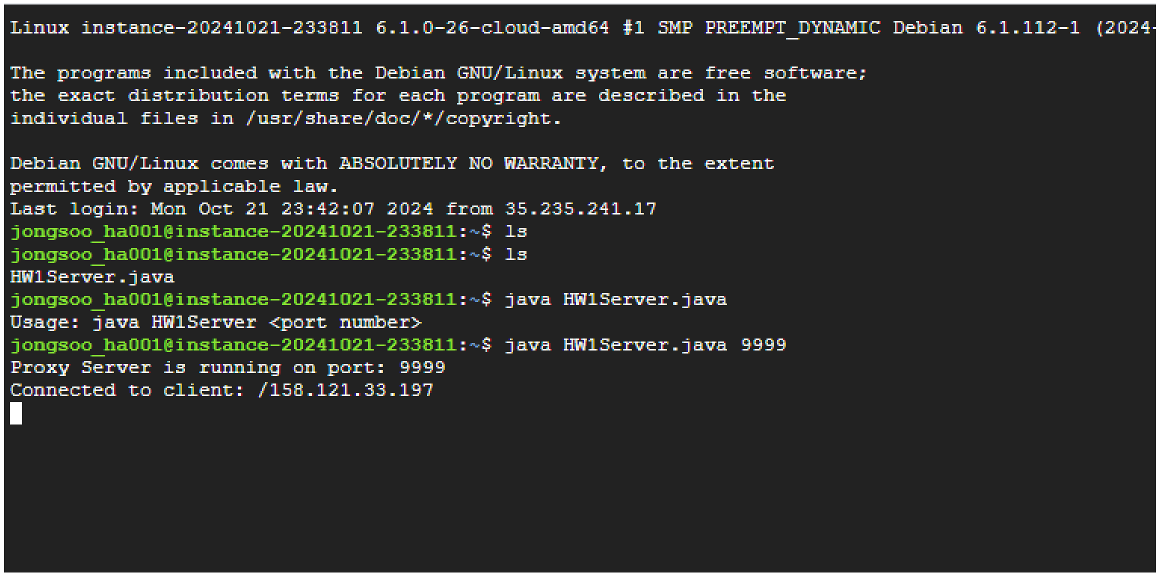
ssh.cloud.google.com/v2/ssh/projects/hw1jongsoo/zones/us-central1-c/instances/instance-20241021-233811?authuser=0&hl=e...
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE
Transferred 1 item

```
Linux instance-20241021-233811 6.1.0-26-cloud-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.112-1 (2024-09-30) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Oct 21 23:42:07 2024 from 35.235.241.17
jongsoo_ha001@instance-20241021-233811:~$ ls
jongsoo_ha001@instance-20241021-233811:~$ ls
HW1Server.java
jongsoo_ha001@instance-20241021-233811:~$ java HW1Server.java
Usage: java HW1Server <port number>
jongsoo_ha001@instance-20241021-233811:~$ java HW1Server.java 9999
Proxy Server is running on port: 9999
```

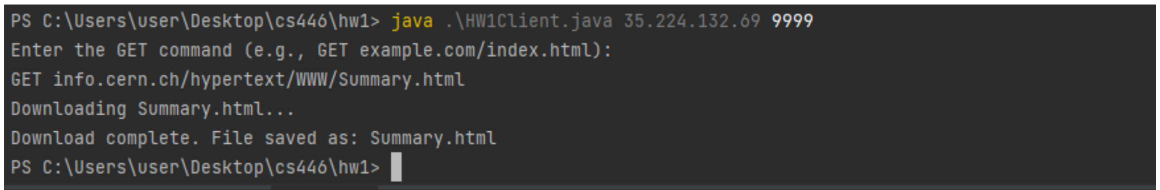
```
PS C:\Users\user\Desktop\cs446\hw1> java .\HW1Client.java 35.224.132.69 9999
Enter the GET command (e.g., GET example.com/index.html):
```



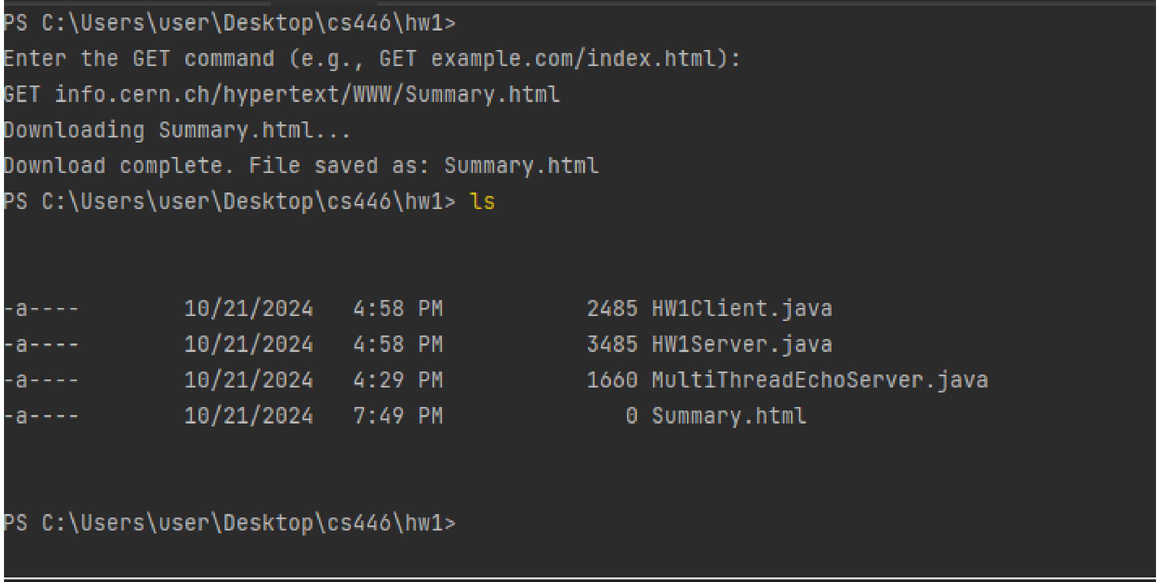
```
Linux instance-20241021-233811 6.1.0-26-cloud-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.112-1 (2024

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Oct 21 23:42:07 2024 from 35.235.241.17
jongsoo_ha001@instance-20241021-233811:~$ ls
jongsoo_ha001@instance-20241021-233811:~$ ls
HW1Server.java
jongsoo_ha001@instance-20241021-233811:~$ java HW1Server.java
Usage: java HW1Server <port number>
jongsoo_ha001@instance-20241021-233811:~$ java HW1Server.java 9999
Proxy Server is running on port: 9999
Connected to client: /158.121.33.197
```



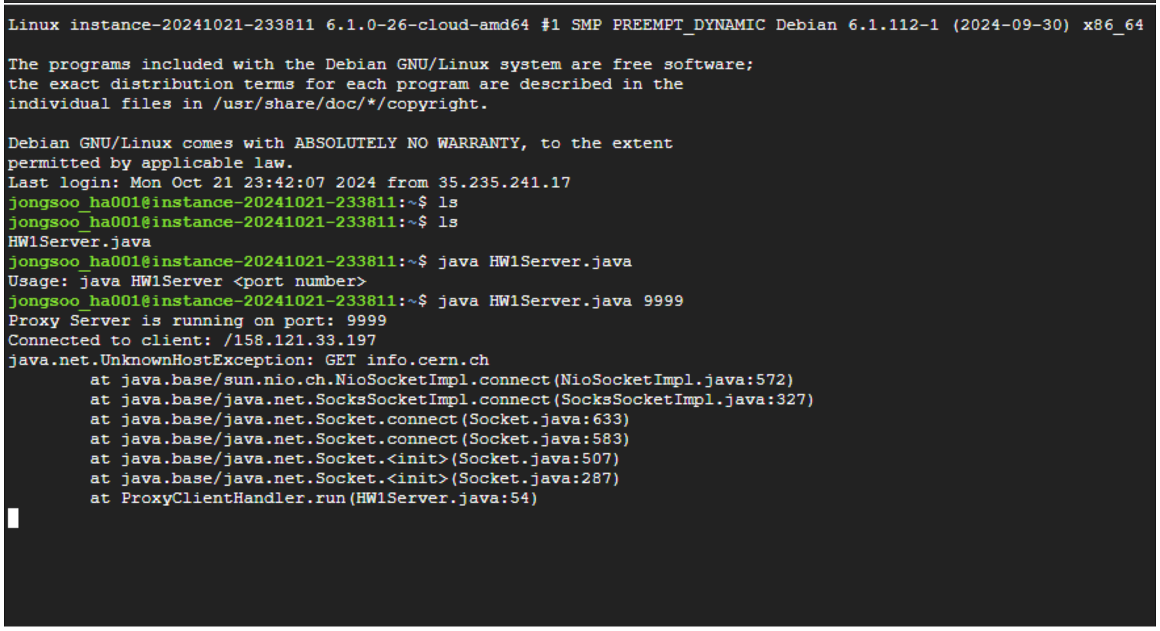
```
PS C:\Users\user\Desktop\cs446\hw1> java .\HW1Client.java 35.224.132.69 9999
Enter the GET command (e.g., GET example.com/index.html):
GET info.cern.ch/hypertext/WWW/Summary.html
Downloading Summary.html...
Download complete. File saved as: Summary.html
PS C:\Users\user\Desktop\cs446\hw1>
```



```
PS C:\Users\user\Desktop\cs446\hw1>
Enter the GET command (e.g., GET example.com/index.html):
GET info.cern.ch/hypertext/WWW/Summary.html
Downloading Summary.html...
Download complete. File saved as: Summary.html
PS C:\Users\user\Desktop\cs446\hw1> ls


-a----        10/21/2024   4:58 PM           2485 HW1Client.java
-a----        10/21/2024   4:58 PM           3485 HW1Server.java
-a----        10/21/2024   4:29 PM           1660 MultiThreadEchoServer.java
-a----        10/21/2024   7:49 PM              0 Summary.html


PS C:\Users\user\Desktop\cs446\hw1>
```



```
Linux instance-20241021-233811 6.1.0-26-cloud-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.112-1 (2024-09-30) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Oct 21 23:42:07 2024 from 35.235.241.17
jongsoo_ha001@instance-20241021-233811:~$ ls
jongsoo_ha001@instance-20241021-233811:~$ ls
HW1Server.java
jongsoo_ha001@instance-20241021-233811:~$ java HW1Server.java
Usage: java HW1Server <port number>
jongsoo_ha001@instance-20241021-233811:~$ java HW1Server.java 9999
Proxy Server is running on port: 9999
Connected to client: /158.121.33.197
java.net.UnknownHostException: GET info.cern.ch
        at java.base/sun.nio.ch.NioSocketImpl.connect(NioSocketImpl.java:572)
        at java.base/java.net.SocksSocketImpl.connect(SocksSocketImpl.java:327)
        at java.base/java.net.Socket.connect(Socket.java:633)
        at java.base/java.net.Socket.connect(Socket.java:583)
        at java.base/java.net.Socket.<init>(Socket.java:507)
        at java.base/java.net.Socket.<init>(Socket.java:287)
        at ProxyClientHandler.run(HW1Server.java:54)
```