

Sai M. Valluru

Professor Bo Sheng

CS446

10/20/23

## Homework 1 GCP Testing

**Step 1:** As mentioned in the instructions, I created a VM on GCP with the configured firewall rules to allow incoming TCP traffic to the port number (9999) the program will use.

### VM instances

Filter Enter property name or value								
<input type="checkbox"/>	Status	Name ↑	Zone	Recommendations	In use by	Internal IP	External IP	Connect
<input type="checkbox"/>	✓	<a href="#">instance-1</a>	us-central1-c			10.128.0.2 ( <a href="#">nic0</a> )	34.171.131.237 ( <a href="#">nic0</a> )	SSH ▾

### VPC firewall rules

Firewall rules control incoming or outgoing traffic to an instance. By default, incoming traffic from outside your network is blocked. [Learn more](#)

Note: App Engine firewalls are managed in the [App Engine Firewall rules section](#).

SMTP port 25 disallowed in this project. <a href="#">Learn more</a>											
REFRESH CONFIGURE LOGS DELETE											
Filter Enter property name or value											
<input type="checkbox"/>	Name	Type	Targets	Filters	Protocols / ports	Action	Priority	Network ↑	Logs	Hit count	Last hit
<input type="checkbox"/>	<a href="#">default-allow-icmp</a>	Ingress	Apply to all	IP ranges: 0.0.0.0/0	icmp	Allow	65534	<a href="#">default</a>	Off	—	—
<input type="checkbox"/>	<a href="#">default-allow-internal</a>	Ingress	Apply to all	IP ranges: 10.128.0.0/9	tcp:0-65535 udp:0-65535 icmp	Allow	65534	<a href="#">default</a>	Off	—	—
<input type="checkbox"/>	<a href="#">default-allow-rdp</a>	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:3389	Allow	65534	<a href="#">default</a>	Off	—	—
<input type="checkbox"/>	<a href="#">default-allow-ssh</a>	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:22	Allow	65534	<a href="#">default</a>	Off	—	—
<input type="checkbox"/>	<a href="#">tcp-9999-allow</a>	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:9999	Allow	65534	<a href="#">default</a>	Off	—	—

Step 2: I then transferred the necessary files of my proxy server program to my VM and compiled HW1Server.java in the VM, which will act as the server. Similarly, I also compiled my HW1Client.java on my local machine, which will act as the client.

```
srisai_valluru@instance-1:~/ProxyServerProject/src$ pwd
/home/srisai_valluru/ProxyServerProject/src
srisai_valluru@instance-1:~/ProxyServerProject/src$ javac server/HW1Server.java
srisai_valluru@instance-1:~/ProxyServerProject/src$
```

```
~/Desktop/ProxyServerProject/src (1.006s)  
javac client/HW1Client.java
```

Step 3: I ran HW1Server.java on my VM and HW1Client on my local machine with the following inputs respectively. Once the client connection was established, I typed in the required command to fetch the requested file and download it. Once the download was complete, I used the command “ls” to check if my downloaded file, “Summary.html” was present in my directory.

- IP address of my VM: 34.171.131.237

```
srisai_valluru@instance-1:~/ProxyServerProject/src$ java server/HW1Server 9999  
Client connection established: /73.16.36.73  
□
```

```
~/Desktop/ProxyServerProject/src (22.405s)  
java client/HW1Client 34.171.131.237 9999  
GET http://info.cern.ch/hypertext/WWW/Summary.html  
  
~/Desktop/ProxyServerProject/src (0.034s)  
ls  
Summary.html      client      server  
  
~/Desktop/ProxyServerProject/src
```

Step 4: Lastly, I used “cat” to verify if the downloaded file in my directory was accurate.

```
~/Desktop/ProxyServerProject/src (0.043s)
cat Summary.html
<TITLE>Summary -- /WWW</TITLE>
<NEXTID 7>
<H1>WorldWideWeb - Summary</H1>The <A NAME=6 HREF=TheProject.html>WWW</A> project merges the techniques of information retrieval and
hypertext to make an easy but powerful global information system.<P>
The project is based on the philosophy that much academic information
should be freely available to anyone. It aims to allow information
sharing within internationally dispersed teams, and the dissemination
of information by support groups. Originally aimed at the High Energy
Physics community, it has spread to other areas and attracted much
interest in user support, resource discovery and collaborative work
areas.
<H3>Reader view</H3>The WWW world consists of documents, and links. Indexes are special
documents which, rather than being read, may be searched. The result
of such a search is another ("virtual") document containing links
to the documents found. A simple protocol ("<A NAME=3 HREF=Protocols/HTTP/AsImplemented.html> HTTP</A> ") is used to allow
a browser program to request a keyword search by a remote information
server.<P>
The web contains documents in many formats. Those documents which
are hypertext, (real or virtual) contain links to other documents,
or places within documents. All documents, whether real, virtual or
indexes, look similar to the reader and are contained within the same
addressing scheme.<P>
To follow a link, a reader clicks with a mouse (or types in a number
if he or she has no mouse). To search and index, a reader gives keywords
(or other search criteria). These are the only operations necessary
to access the entire world of data.
<H3>Information provider view</H3>The WWW browsers can access many existing data systems via existing
protocols (FTP, NNTP) or via HTTP and a gateway. In this way, the
critical mass of data is quickly exceeded, and the increasing use
of the system by readers and information suppliers encourage each
other.<P>
Making a web is as simple as writing a few <A NAME=1 HREF=MarkUp/MarkUp.html>SGML</A> files which point
to your existing data. Making it public involves running the FTP or
HTTP <A NAME=2 HREF=Daemon/Overview.html>daemon</A> , and making at least one link into your web from another.
In fact, any file available by anonymous FTP can be immediately linked
into a web. The very small start-up effort is designed to allow small
contributions. At the other end of the scale, large information providers
may provide an HTTP server with full text or keyword indexing. This
may allow access to a large existing database without changing the
way that database is managed. Such gateways have already been made
into Digital's VMS/Help, Technical University of Graz's "Hyper-G",
and Thinking Machine's "W.A.I.S." systems.<P>
The WWW model gets over the frustrating incompatibilities of data
format between suppliers and reader by allowing negotiation of format
between a smart browser and a smart server. This should provide a
basis for extension into multimedia, and allow those who share application
standards to make full use of them across the web.<P>
This summary does not describe the many exciting possibilities opened
up by the WWW project, such as efficient document caching, the reduction
of redundant out-of-date copies, and the use of knowledge daemons.
There is more information in the online project documentation, including
some background on hypertext and many technical notes.
<H3>Try it</H3>You can try the simple <A NAME=4 HREF=LineMode/Browser.html>line mode browser</A> by telnetting to info.cern.ch
(no user or password. From UK JANET, use the
<a href=JANETAccess.html>Gateway</a>).
You can also find out more about WWW in this
way. This is the least sophisticated browser -- remember that the
window-oriented ones are much smarter!<P>
It is much more efficient to install a browser on your own machine.
The line mode browser is currently available in source form by anonymous
FTP from node info.cern.ch [currently 128.141.201.74] as
<XMP> /pub/www/src/WWWLineMode_v.vv.tar.Z.

</XMP>(v.vv is the version number - take the latest.) <P>
Also available is a <A NAME=5 HREF=NeXT/WorldWideWeb.html>hypertext editor</A> for the NeXT (WWWNeXTStepEditor_v.vv.tar.Z),
the ViolaWWW browser for X11, and a skeleton server daemon (WWWDaemon_v.vv.tar.Z).
<XMP>
</XMP>Documentation is readable using www. A plain text version of the installation
instructions is included in the tar file! Printable (postscript)
documentation and articles are in /pub/www/doc<P>

-----
<ADDRESS><A NAME=0 HREF=../WWW/disclaimer.html>Tim BL</A>
</ADDRESS>
```

```
~/Desktop/ProxyServerProject/src
```