Port Forwarder

COMP 8005 – Final Project - Testing

**Danny Lieu**

**Mike Zobac**

**Mar 27, 2018**

Table of Contents

[Port Forwarder 3](#_Toc509779936)

[Test outline 3](#_Toc509779937)

[Test case descriptions 3](#_Toc509779938)

[1. Test 1 3](#_Toc509779939)

[2. Test 2 5](#_Toc509779940)

[3. Test 3 6](#_Toc509779941)

[4. Test 4 7](#_Toc509779942)

[5. Test 5 8](#_Toc509779943)

# Port Forwarder

## Test outline

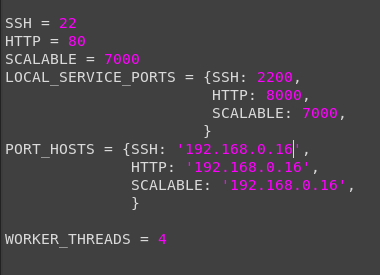
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Test Description | Tool Used | Expected Result | Pass/Fail |
| 1 | PortForwarder listens for incoming traffic from the given ports in the configuration file. | Netstat/Python | The program listen on the given port 2200, 8000, 7000. | Pass. See test case description for more detail. |
| 2 | PortForwarder successfully maps local port 2200 to port 22 on another machine. | SSH/Python | Log in to another machine through port 2200. | Pass. See test case description for more detail. |
| 3 | PortForwarder successfully maps local port 8000 to port 80 on another machine. | Web browser/Python | A default Apache web page shows when accessing through port 8000. | Pass. See test case description for more detail. |
| 4 | PortForwarder successfully maps local port 7000 to port 7000 on another machine. | Python client/server application | Client successfully connects to the server through PortForwarder. | Pass. See test case description for more detail. |
| 5 | PortForwarder successfully handle heavy throughput from multiple clients. | Python client/server application | Multiple clients able to connect/send data to the server. | Pass. See test case description for more detail. |

## Test case descriptions

### Test 1

PortForwarder listens for incoming traffic from the given ports in the configuration file.

The configuration file is listed as below:

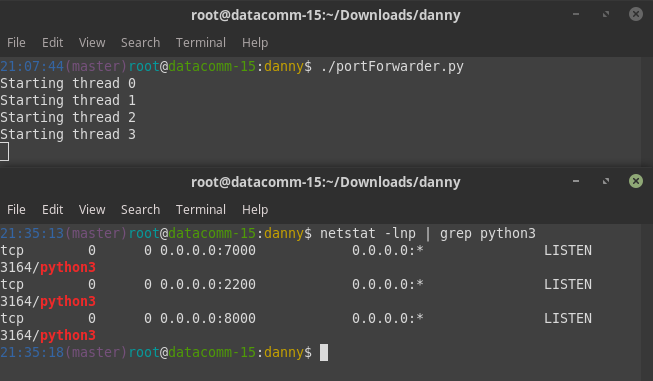


For simplicity, we only use 3 machines to run the test. Three machine IPs are:

* 192.168.0.14: the external machine
* 192.16.0.15: the machine running port forwarder application.
* 192.168.0.16: the internal machine which runs SSH service, Web server and the Epoll server.

Three ports that was used in the test are 22, 80, and 7000 maps respectively to port 2200, 8000, 7000 on the machine running port forwarder application.

After running the application, we use netstat to see if the app is listening for those ports:

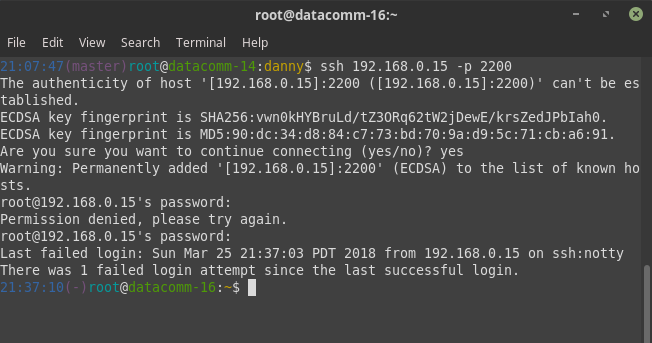


The application successfully listens for port 7000, 2200, 8000 as listed in the configuration file.

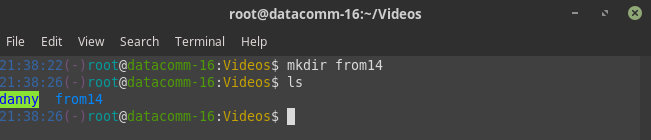
### Test 2

PortForwarder successfully maps local port 2200 to port 22 on another machine.

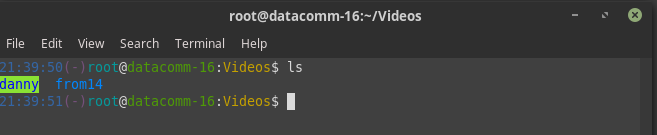
We use the external machine to log in into the internal one through the machine running port forwarder:



For testing purposes, we are going to make a directory on the internal machine named **from14:**



We are going to datacomm-16 to see if we have that directory:

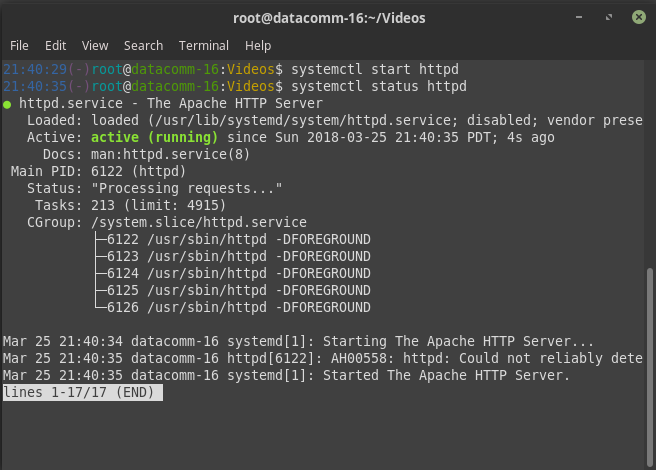


The internal machine which is **datacomm-14** successfully connects to **datacomm-16** through **192.168.0.15:2200**

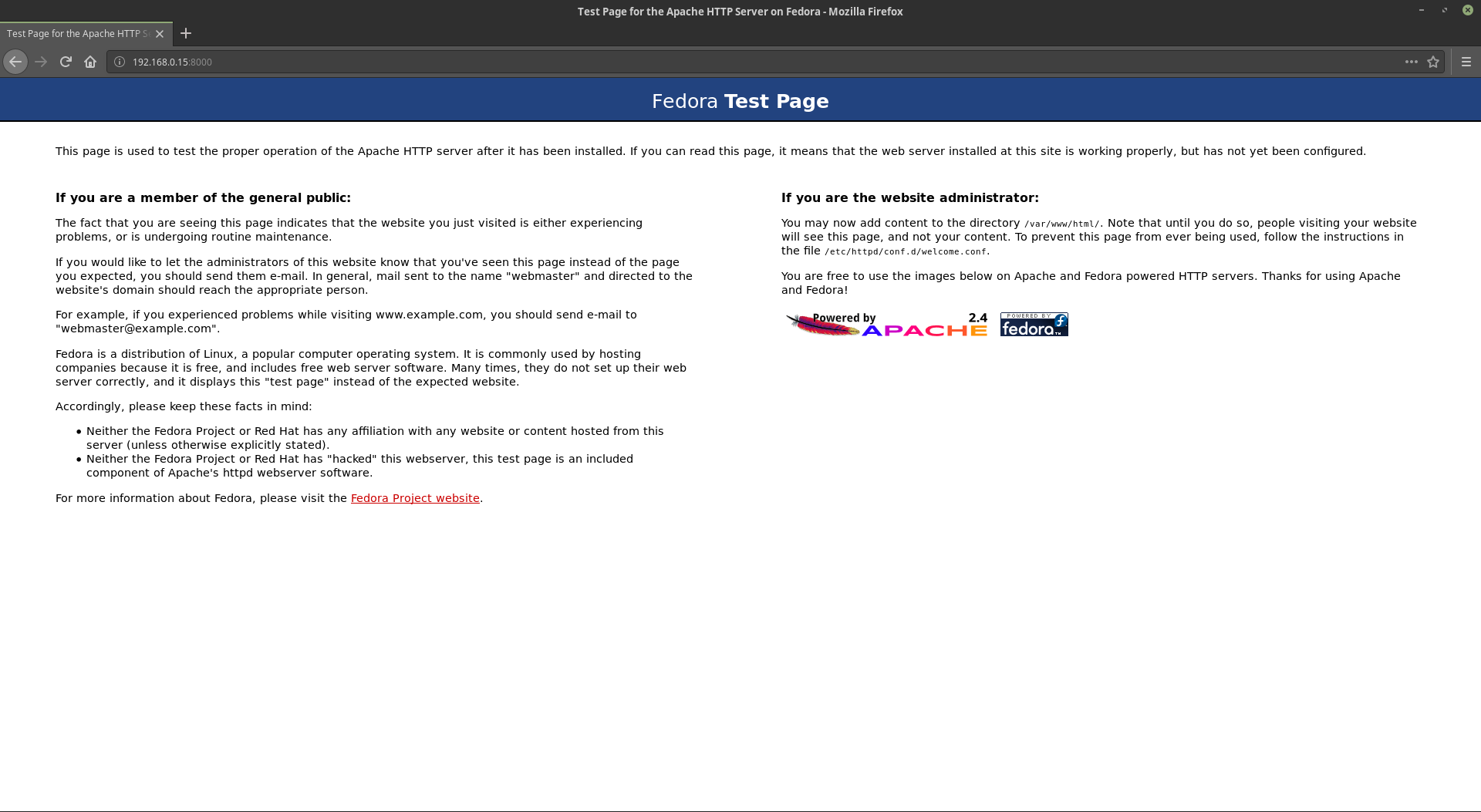
### Test 3

PortForwarder successfully maps local port 8000 to port 80 on another machine.

On datacomm-16, we will start running an Apache server:



After that, we try to access the web server through Firefox to see if we can load the Apache default web page:

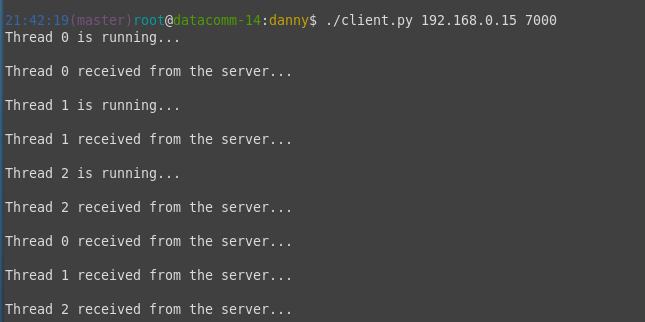


As the page shows, we successfully connect to the web server through port 8000 on the port forwarder.

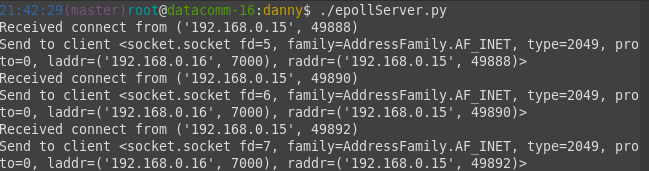
### Test 4

PortForwarder successfully maps local port 7000 to port 7000 on another machine.

For this test, we are going to use the custom client and server application which will run on port 7000. The client will try to connect to the server through port 7000 of the port forwarder:



On the server, we need to see if it is able to receive connection from the external machine:



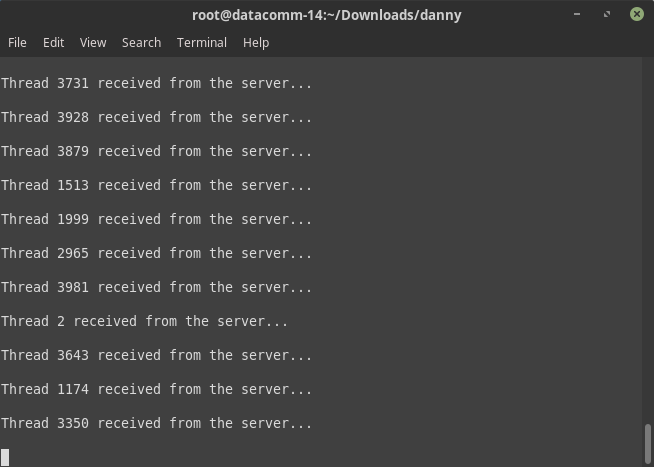
The server shown that it received connections from 192.168.0.15 which was coming from the port forwarder.

The port forwarder successfully forward incoming traffic to the internal machine.

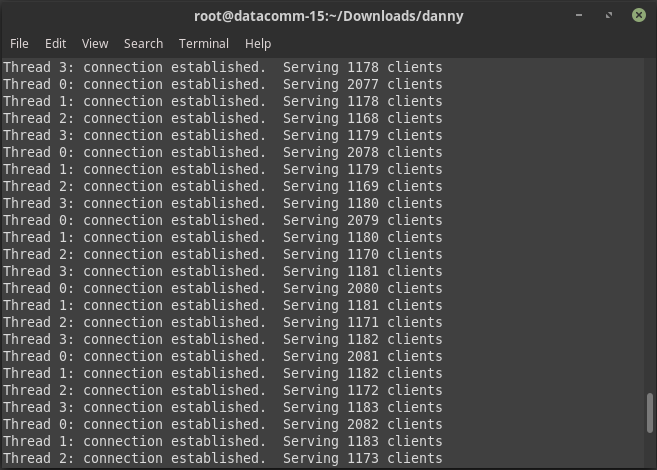
### Test 5

PortForwarder successfully handle heavy throughput from multiple clients.

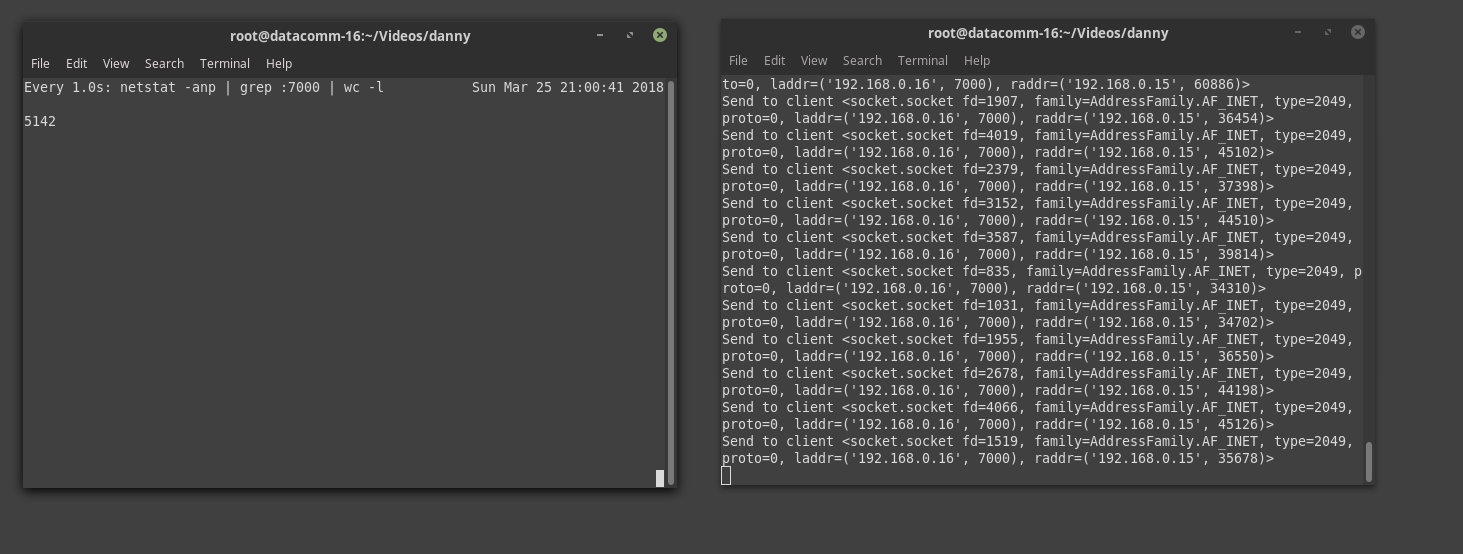
To test the scalability of both the port forwarder and server, we will run multiple clients and try to connect to the server through the port forwarder.



The port forwarder will also print out the number of clients that have been connected and forwarded to the server:



On the server side, we will see how many established connections that have been maintained from the external machine:



The server is capable of sending and receiving data with 5142 established connections. For the simplicity of testing, we are not going to go any further that this number.

As the test shown, the port forwarder is able to handle heavy throughput from multiple clients.