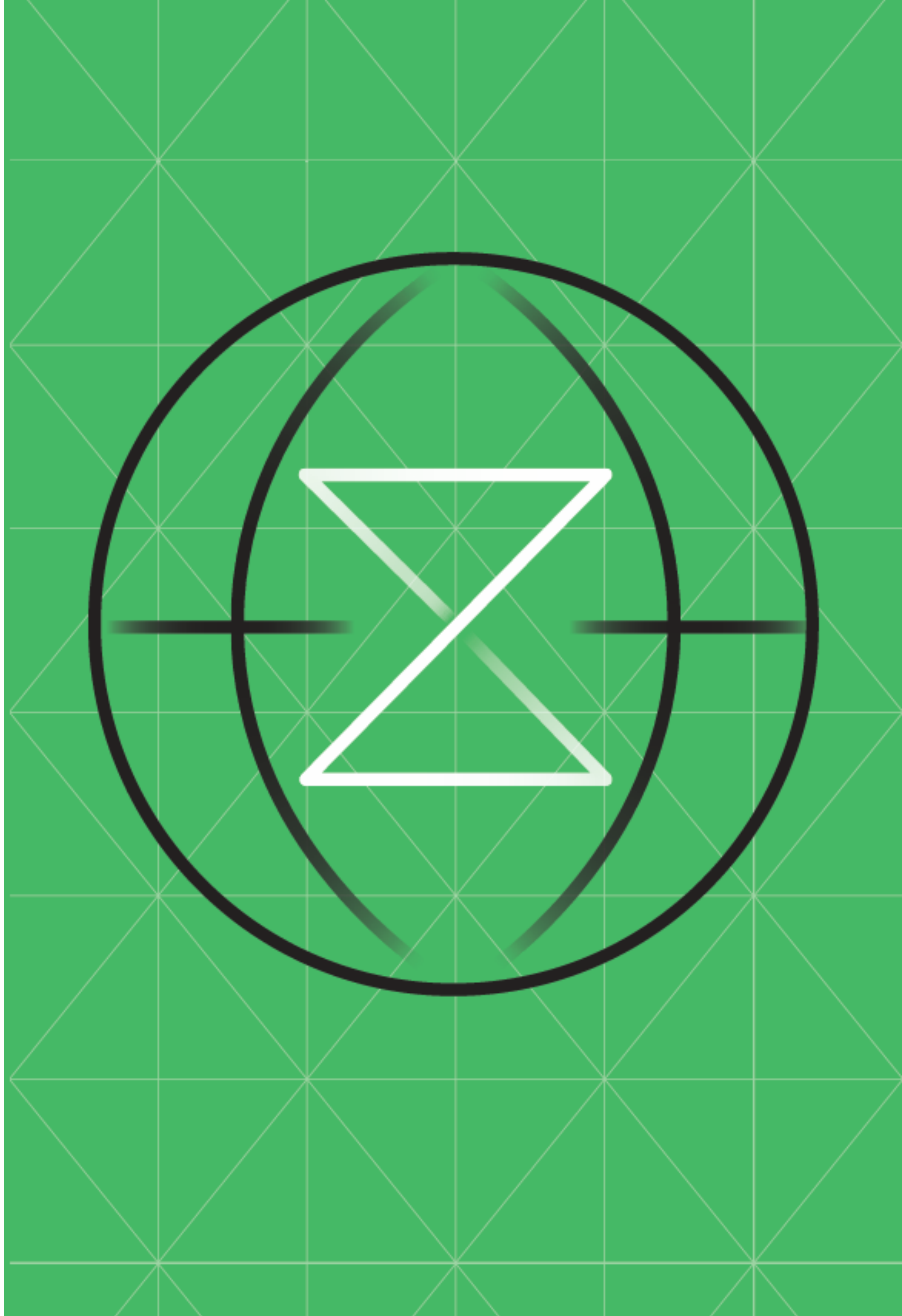
Extended

230807-1057

NODE1



Z Xplore

AN ODE TO NODE

The Challenge

During this challenge you will use Node.js to build a small web-server, accessible from the public internet.

You will set your own specific parameters to make sure your URL is unique.

Before You Begin

Make sure you have a basic understanding of Datasets, Dataset members and the usage of USS.

Make sure you have set up your workstation correctly to connect to the IBM Z Xplore system with VSCode.

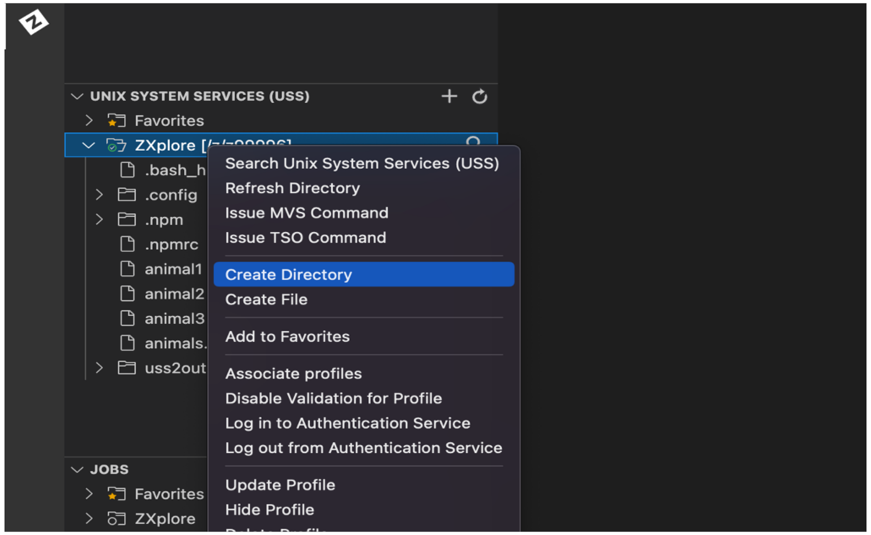
Investment

Steps	Duration
7	20 minutes

1 CREATE A NODE DIRECTORY

Set the filter on your VSCode **USS** view, to open your local user directory by entering: /z/zxxxxx. (As usual, make sure you replace “zxxxxx” with your own userid)

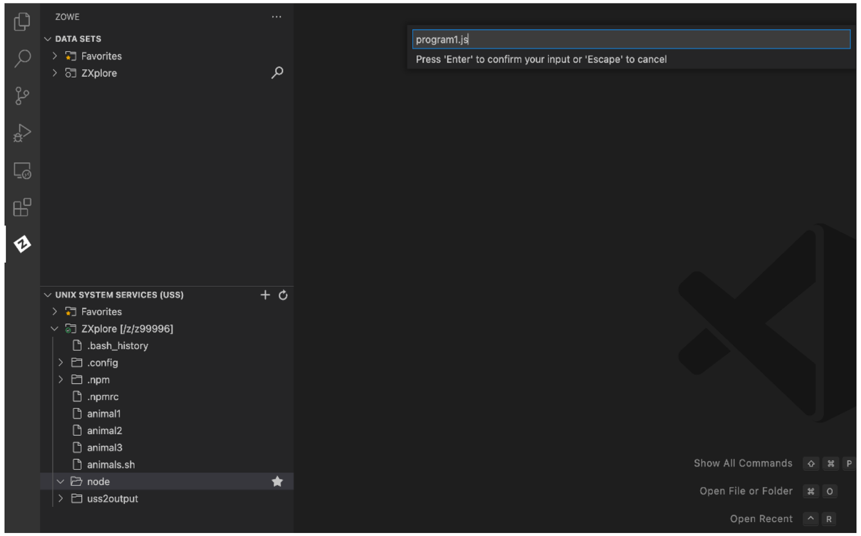
You will see files and directories from previous challenges.



Make sure your mouse is at the top-level directory, and right-click it to open the options menu. Select the option to “Create Directory” and name it “**node**”. You will be prompted to select a location for your new directory.

Double-check that it says /z/zxxxxx

2 CREATE A NODE FILE



program1.js

Press 'Enter' to confirm your input or 'Escape' to cancel

Right-click on the new directory and create a new file called **program1.js**, making sure to set the ".js" at the end of the filename to make sure it will be identified as a Node.js program.

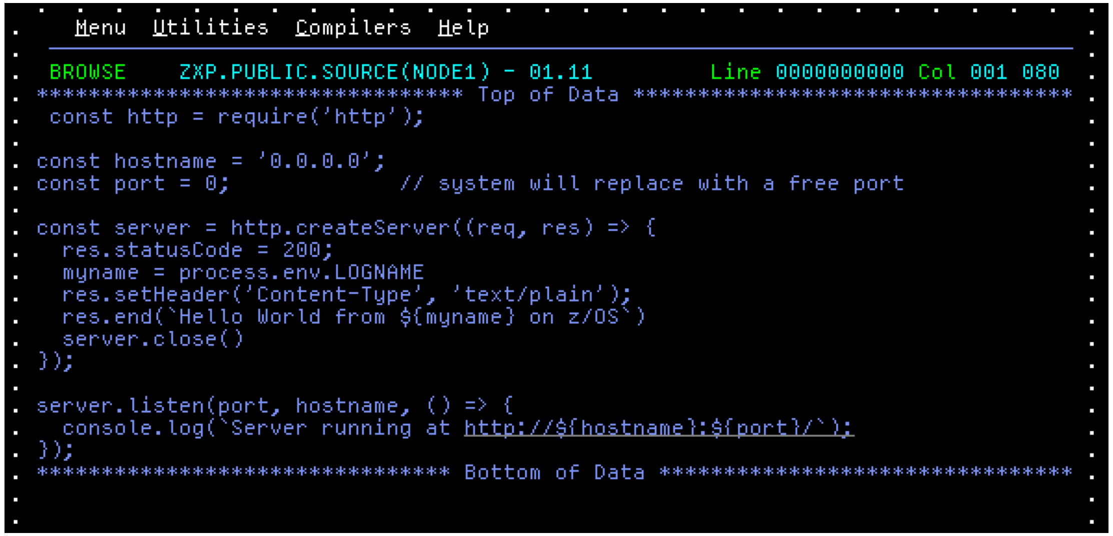
Now locate the example code provided to help you get started.

Open the DATA SETS navigation panel in VSCode and make sure your filter is set to include **zxp.public.***

Locate the **ZXP.PUBLIC.SOURCE** dataset and open the **NODE1**

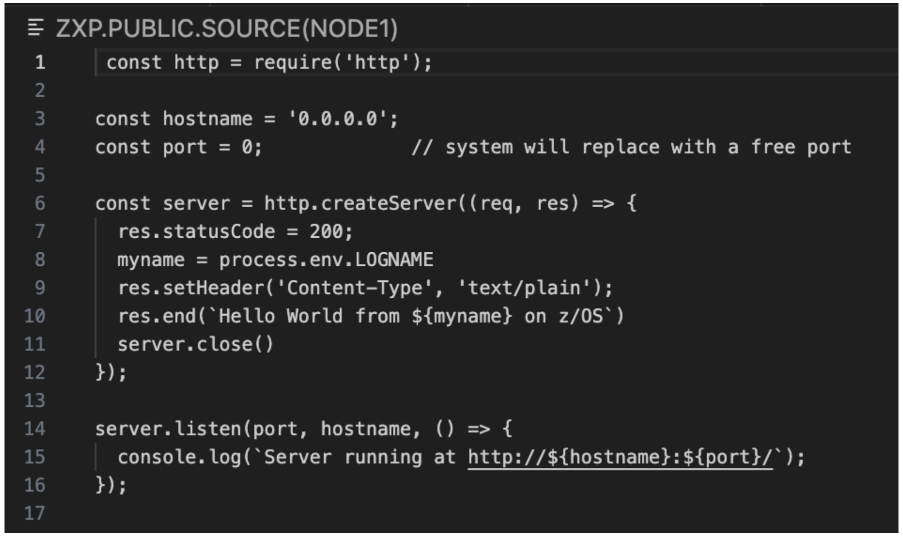
Click to open and view the source code contents.

NODE1|230807-1057



```
 BROWSE    ZXP.PUBLIC.SOURCE(NODE1) - 01.11          Line 0000000000 Col 001 080
********************************* Top of Data **********************************
 const http = require('http');

const hostname = '0.0.0.0';
const port = 0;             // system will replace with a free port

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  myname = process.env.LOGNAME
  res.setHeader('Content-Type', 'text/plain');
  res.end(`Hello World from ${myname} on z/OS`)
  server.close()
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
******************************** Bottom of Data ********************************
```

or



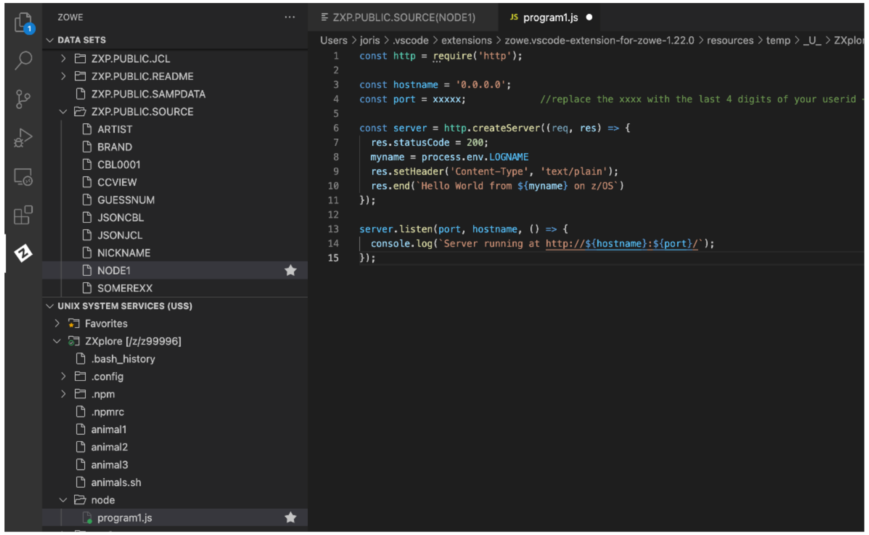
```
const http = require('http');

const hostname = '0.0.0.0';
const port = 0;             // system will replace with a free port

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  myname = process.env.LOGNAME
  res.setHeader('Content-Type', 'text/plain');
  res.end(`Hello World from ${myname} on z/OS`)
  server.close()
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});

```

Now select all the text and press *Control+c* or *Cmd+c* to copy the text, depending on your local workstation operating system.

3 CREATE FIRST NODE PROGRAM

Return to your **program1.js** file and click to open it.

Use *Control+v* or *Cmd+v* to paste the content from NODE1.



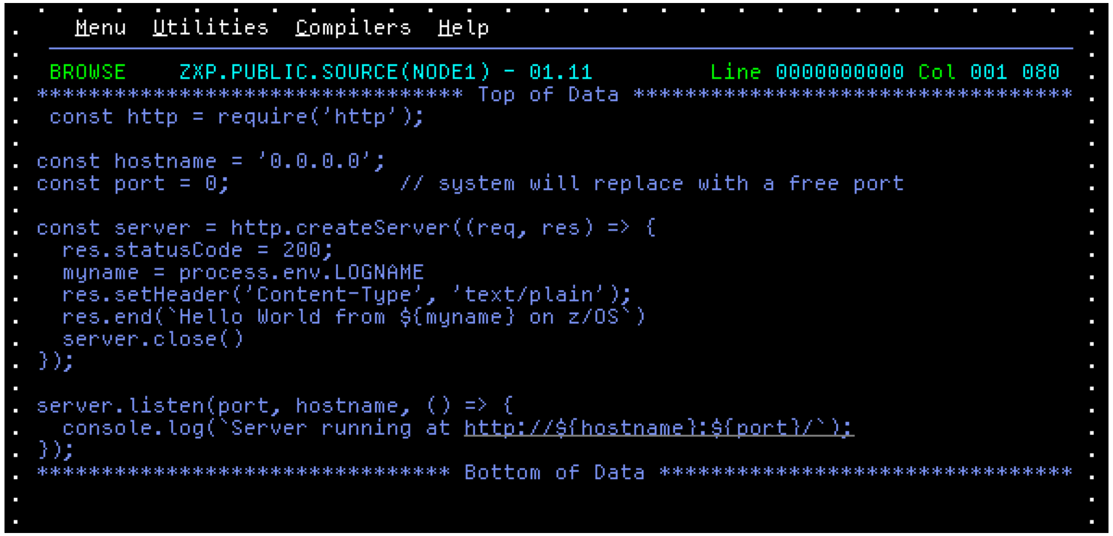
```
const http = require('http');

const hostname = '0.0.0.0';
const port = xxxxx;             //replace the xxxx with the last 4 digits of your userid

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  myname = process.env.LOGNAME
  res.setHeader('Content-Type', 'text/plain');
  res.end(`Hello World from ${myname} on z/OS`)
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

You may be prompted by VSCode to install a Node.js plugin which helps you in formatting Node.js code.

Feel free to accept and install this – the syntax highlighting will help with understand the node.js javascript programming language.

4 OPEN THE SERVER PORT

Investigate the code and see that line 4 has a port definition.



```
 BROWSE    ZXP.PUBLIC.SOURCE(NODE1) - 01.11          Line 0000000000 Col 001 080
******************************** Top of Data **********************************
 const http = require('http');

const hostname = '0.0.0.0';
const port = 0;              // system will replace with a free port

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  myname = process.env.LOGNAME
  res.setHeader('Content-Type', 'text/plain');
  res.end(`Hello World from ${myname} on z/OS`)
  server.close()
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
******************************* Bottom of Data ********************************
```

This setting tells the operating system to assign a free TCP port that your application can use - this will usually be between 1025 and 65535.

This will be the port number you use to connect to your server application in the next steps.

The program will create a mini web-server application which will listen for an incoming request to the assigned port, send a response message, and shutdown.

You won't know which port will be used until your program is executed.

To know which port has been assigned (*each time you execute your program*), you will need to make a change to the program at line 15;

before – ${port}
    this assumes the port requested is the same port used by the server

```
after - ${server.address().port}
    this will show the actual port being used by the server
```

5 READY TO RUN

You now need to connect to the IBM Z Xplore system with a SSH session. This will be done by issuing the following command:

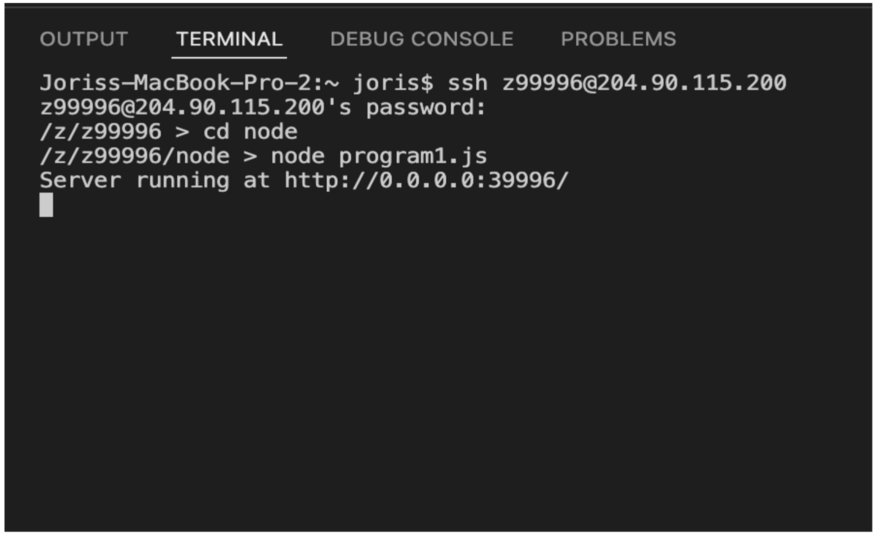
```
ssh zxxxxx@204.90.115.200
```

You will be prompted for your password - enter your IBM Z Xplore Z systems password.

By default, it will open your own home directory inside USS. This is the same directory that you set earlier in the USS section in your VSCode navigation view.

Change your current working directory to the **node** directory, and start the node application:

```
node program1.js
```



```
Joriss-MacBook-Pro-2:~ joris$ ssh z99996@204.90.115.200
z99996@204.90.115.200's password:
/z/z99996 > cd node
/z/z99996/node > node program1.js
Server running at http://0.0.0.0:39996/
```

You should see a message like:

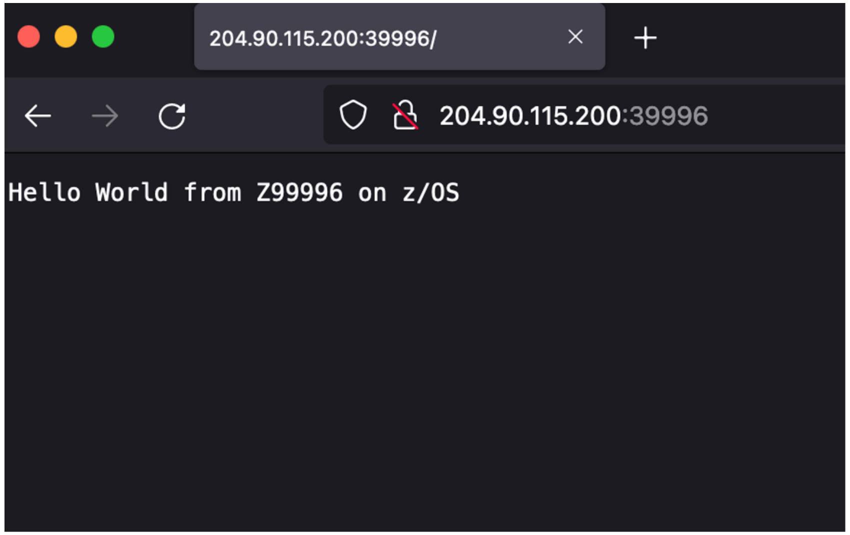
```
Server running at http://0.0.0.0:2573/
```

Note the port number displayed - you will need this to connect to the web-server.

6 IS IT WORKING?

The 0.0.0.0 in the server address is displayed because your application “binds” to the default IP interface for the IBM Z Xplore system.

From the internet, you will connect to the **204.90.115.200** address to reach your program.



```
Hello World from Z99996 on z/OS
```

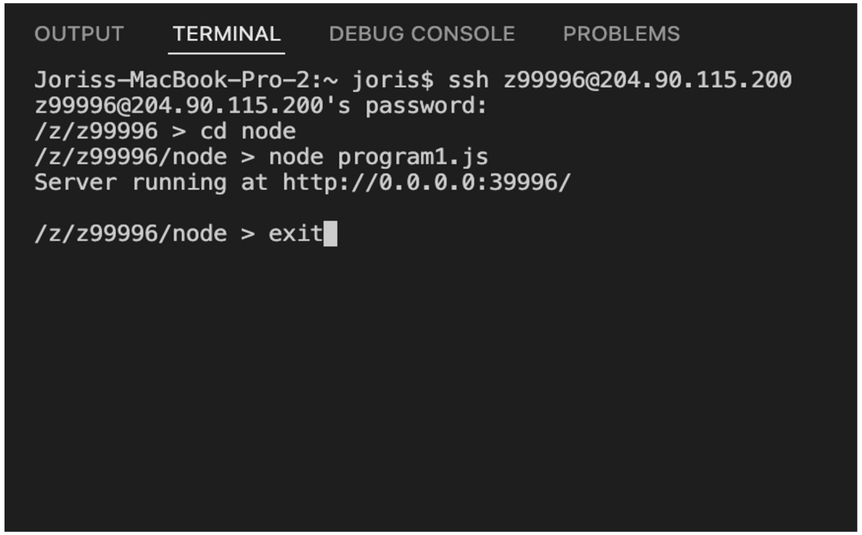
In order to check if your first web-server is running properly, you need to use a browser to connect to your web-server.

Type in the IBM Z Xplore IP address followed by “:” and your port number to navigate to **your** web-server.

This will open the website running in your Node.js web-server and should look similar to the screenshot above.

7 STOP YOUR SERVER

To validate you have successfully completed this challenge, start running your node application, and then submit the **CHKNODE1** job from **ZXP.PUBLIC.JCL**



```
Joriss-MacBook-Pro-2:~ joris$ ssh z99996@204.90.115.200
z99996@204.90.115.200's password:
/z/z99996 > cd node
/z/z99996/node > node program1.js
Server running at http://0.0.0.0:39996/

/z/z99996/node > exit
```

The validation process should cause your application to end; if your application is not active when you submit CHKNODE1, the validation will fail.

Type exit and press enter to end your SSH session.

Nice job - let’s recap	**Next up …**
You just built your own web-server, responding as a simple website showing some text. You have customized the URL to use a random free port number and used a terminal session to connect to IBM Z Xplore and run your program. This the kind of thing that launched the internet!	Find some other cool challenges on the IBM Z Xplore Extended channel

NODE1|230807-1057