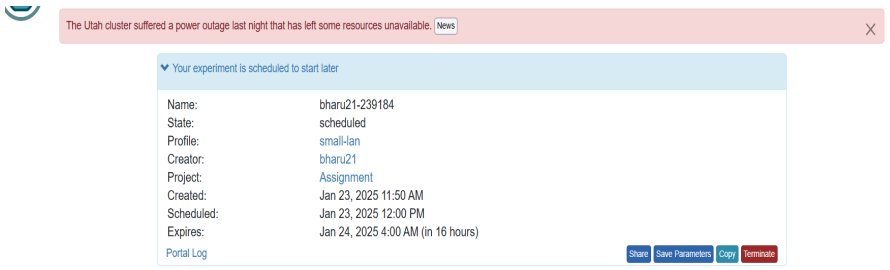
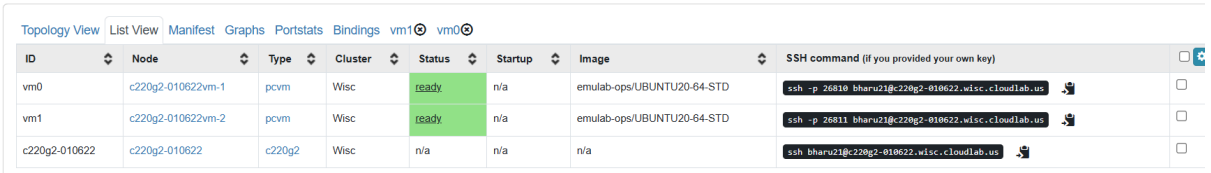
Assignment - 1: Creating VMS and Socket programming

Name: Bhargavi Rengarajan

Student ID: 862467935

1. **Initial Setup:** Creating VMS in cloud lab
    - Created two Virtual Machines (VMs) using **VirtualBox** or **CloudLab**.
    - Configured them to be on the same network (NAT/Bridged mode) so they can communicate.
    - Assigned one VM to act as the **server** and the other as the **client**.





2. **Server Side output**

Created a server program Node.js to:

- Bind to a specific IP address and port (e.g., `0.0.0.0:8080`).
- Wait for incoming connections from clients.
- Send the file (`mydata.txt`) to the client.

i) **Server Program**

```
const net = require('net');
const fs = require('fs');

const PORT = 8080;

// Create the server
const server = net.createServer((socket) => {
    console.log('Client connected.');

    // Open the file to read
    const fileStream = fs.createReadStream('mydata.txt');

    // Pipe the file stream to the socket
    fileStream.pipe(socket);

    fileStream.on('end', () => {
        console.log('File sent successfully.');
        socket.end();
    });
    socket.on('close', () => {
        console.log('Connection closed.');
    });

    socket.on('error', (err) => {
        console.error('Socket error:', err.message);
    });
});

// Start the server
server.listen(PORT, () => {
    console.log(`Server is listening on port ${PORT}`);
});
```

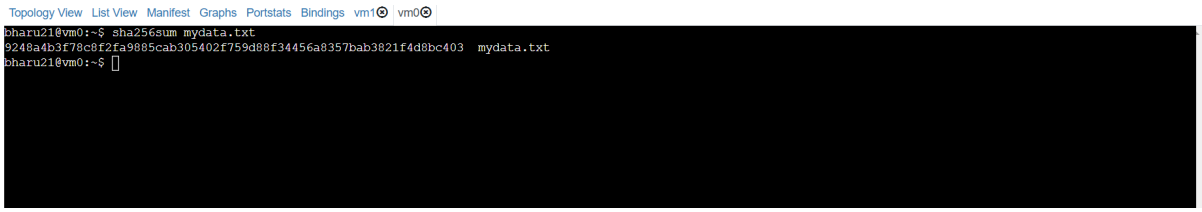
i) <u>Application running:</u>

- Ran the server program (`server.js`) on the Server VM.
- Observed the following logs:

- Confirmation of the server listening on `PORT 8080`.
- Logs showing when a client connects and when the file transfer is complete.

```
bharu21@vm0:~$ sha256sum mydata.txt
9248a4b3f78c8f2fa9885cab305402f759d88f34456a8357bab3821f4d8bc403  mydata.txt
bharu21@vm0:~$ node server
Server is listening on port 8080
Client connected.
File sent successfully.
Connection closed.
```

II) Checksum (Server):



The checksum value
9248a4b3f78c8f2fa9885cab305402f759d88f34456a8357bab3821f4d8bc403

3. **Client output:**
   Created a client program to:

   - Connect to the server using its IP address and port (`172.17.34.1:8080`).
   - Receive the file sent by the server.
   - Save the file as `mydata_client_copy.txt` on the client VM's local filesystem.

i) Client program

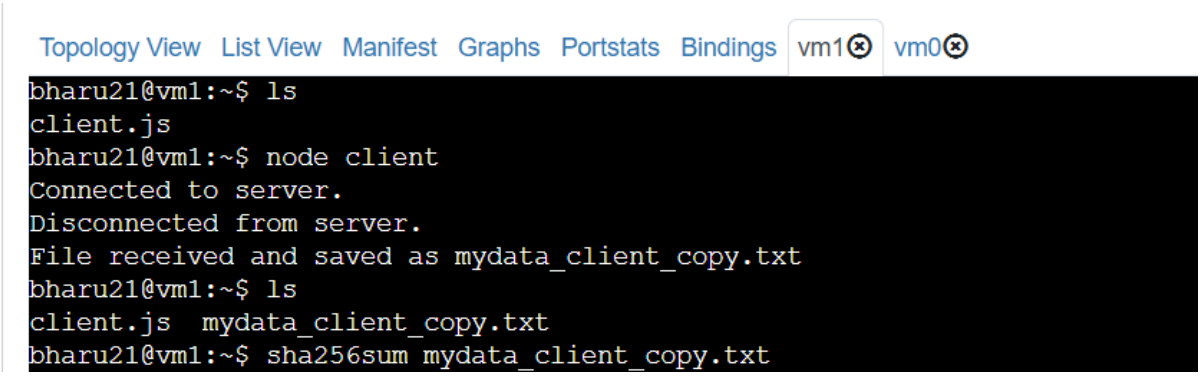
```
const net = require('net');
const fs = require('fs');

const PORT = 8080;
const SERVER_IP = '172.17.34.1'; // Replace with the IP address of VM0

// Create a socket to connect to the server
const client = net.createConnection({ port: PORT, host: SERVER_IP }, ()
=> {
```

```
    console.log('Connected to server.');

    // Open a writable stream to save the received file
    const fileStream = fs.createWriteStream('mydata_client_copy.txt');

    // Pipe the socket data into the writable stream
    client.pipe(fileStream);

    fileStream.on('finish', () => {
        console.log('File received and saved as mydata_client_copy.txt');
        client.end();
    });
});

client.on('error', (err) => {
    console.error('Connection error:', err.message);
});

client.on('end', () => {
    console.log('Disconnected from server.');
});
```

ii) Application running:

Ran the client program (`client.js`) on the Client VM.
Observed the following logs:

- Confirmation of connection to the server.
- Logs showing the file was received and saved as `mydata_client_copy.txt`.
- Disconnection from the server after the transfer.



iii) Checksum (Client):

Checksum:
9248a4b3f78c8f2fa9885cab305402f759d88f34456a8357bab3821f4d8bc403

RESULT:

The client and server gives the same check sum. Hence the give program executes and complies with the assignment.