# **Conclusion:**

1. In this assignment I was able to understand how to install and configure a web server in a remote VM and access the servers home page via its IP address. This helped me to learn the apache2 installation via apt-get which was quite intuitive. Also configuring a firewall to allow access to the port via command line for non-production environment. Finally, I was able to reboot the machine and restart server via command line itself.

2. In this assignment I scraped the apache2 home page through its IP Address using python’s beautiful soup package and parse it for various components with the page. I was able to read the title header through HTML element, page header and section titles using HTML tag along with the CSS class selector. The beautiful soup library has all required features to seamlessly parse the page and extract the relevant information. I was able to easily loop though the section headers and print all required values with less code. This helped me to understand the process to parse a webpage through code and gather necessary information.

3. In this assignment I connected to the VM using python’s paramiko package and SSH through port 22 using the VM’s IP Address, username, and password. I was able to invoke a shell session and send Linux commands which was executed in the VM. In addition to that I received the data back from VM and checked for any errors. This helped me to learn the steps to connect via SSH from host machine through code and execute commands remotely in VM

4. In this assignment I remotely checked the open ports for VM and using the inbuilt socket library by scanning the ports in VM. I connected to remote VM post and printed whether its open or close. In addition to that I specifically marked the SSH (22) and HTTP (80). This helped me to understand the port scanning feature and also the performance impact of it due to the time taken to read port.

5. In this assignment I repurposed the approach of assignment 3 to invoke a shell session and execute Linux commands in VM. Based on that I remotely executed commands on VM to install curl library and create folders with the hierarchy specified. This helped me to appreciate the facility to executing the commands via shell script in VM and can install any required libraries and creating files and folders.

6. In this assignment through Terraform’s Infrastructure as a code, I created Custom VPC, subnet and VM machine in AWS. I learnt various items in this section. Firstly, I logged in to my AWS account and enabled Multi Factor Authentication. Post that I enabled the access keys for my account which provided me secret key, token. Then I installed aws-cli to configure the default region and credential in my machine. Then I installed Terraform and created scripts in Terraform. I explored the variables feature of the terraform to configured my region, tag names, AMI, and instance type. Later I added the scripts in Terraform to create VPC and subnet to use VPC’s id to add it to the network. Then I configured the VM to use the subnet id to get it added to the subnet. I explored the Terraforms format command and validate comment. I understood the purpose of Terraform’s init and plan command and evaluated the results for the aws resources. Finally, I used Terraform’s Apply command to create all the resources and verified the same in AWS. I also explored the Terraforms state file to see the contents of it and compared the same in AWS to see how its mapped. Once I verified everything, I used Terraforms destroy to delete all the resources in AWS. This assignment helped me to understand the end-to-end process of Infrastructure as code and how easy it is to use Terraform to create, manage and delete resources instead of doing it manually or through aws-cli.

# **GitHub Repository link:**