1. Describe IaaS.

Infrastructure as a Service (IaaS) is a form of cloud computing that provides virtualized computing resources over the Internet. IaaS is one of the three main categories of cloud services, alongside Software as a Service (SaaS) and Platform as a Service (PaaS). In an IaaS model, a third-party provider hosts hardware, software, servers, storage, and Other infrastructure components on behalf Of its users.

Key Features of IaaS:

Virtual Machines (VMs): Users can rent VMS for running their applications.
Scalability: Resources can be scaled up or down based on demand.
Pay-as-you-go Pricing: Users pay only for the resources they consume.
Automated Administrative Tasks: Including backup, security, and disaster recovery.
Multi-tenancy: Multiple users share the same infrastructure.

2. Compute and Storage Services in AWS and GCP.

Amazon Web Services (AWS)

Compute Services:

Amazon EC2 (Elastic Compute Cloud): Virtual Services in the cloud
Amazon ECS (Elastic Container Service): Run and manage docker containers.
AWS Lambda: Serverless computing service to run code without provisioning servers
Amazon EKS (Elastic Kubernetes Service): Managed Kubernetes service.
AWS Fargate: Serverless compute engine for containers

Global Cloud Platform (GCP)

Compute Services:

Google Compute Engine (GCE): Virtual machines running in Google's data centers.
Google Kubernetes Engine (GKE): Managed Kubernetes service.
Google Cloud Functions: Event-driven serverless compute service.
Google Cloud Run: Managed compute platform for running containerized applications

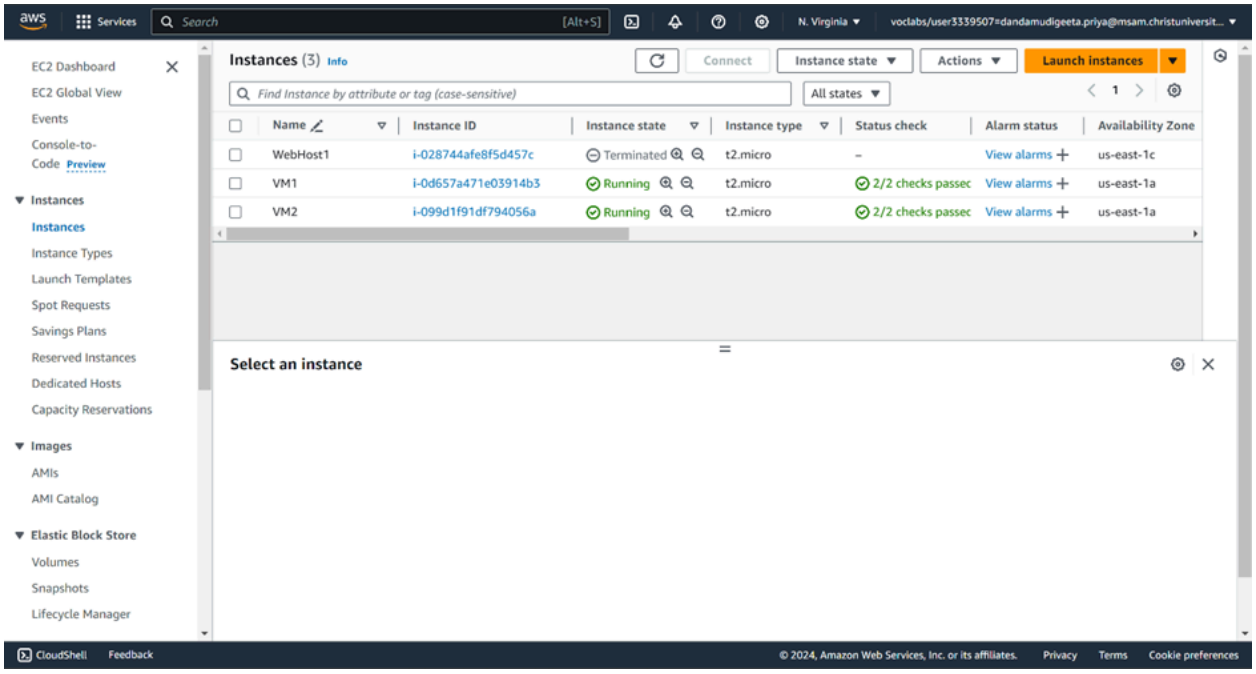
Storage Services:

Google Cloud Storage: Object storage service.
Persistent Disks: Block storage for use with GCE-
Filestore: Fully-managed file storage.
Google Cloud Storage Nearline and Coldline: Storage classes for data archiving and long-term backup-
Transfer Appliance: Hardware appliance for transferring large amounts of data to Google Cloud.

These services provide a robust and flexible infrastructure for a wide range of applications, from simple websites to complex data analytics.

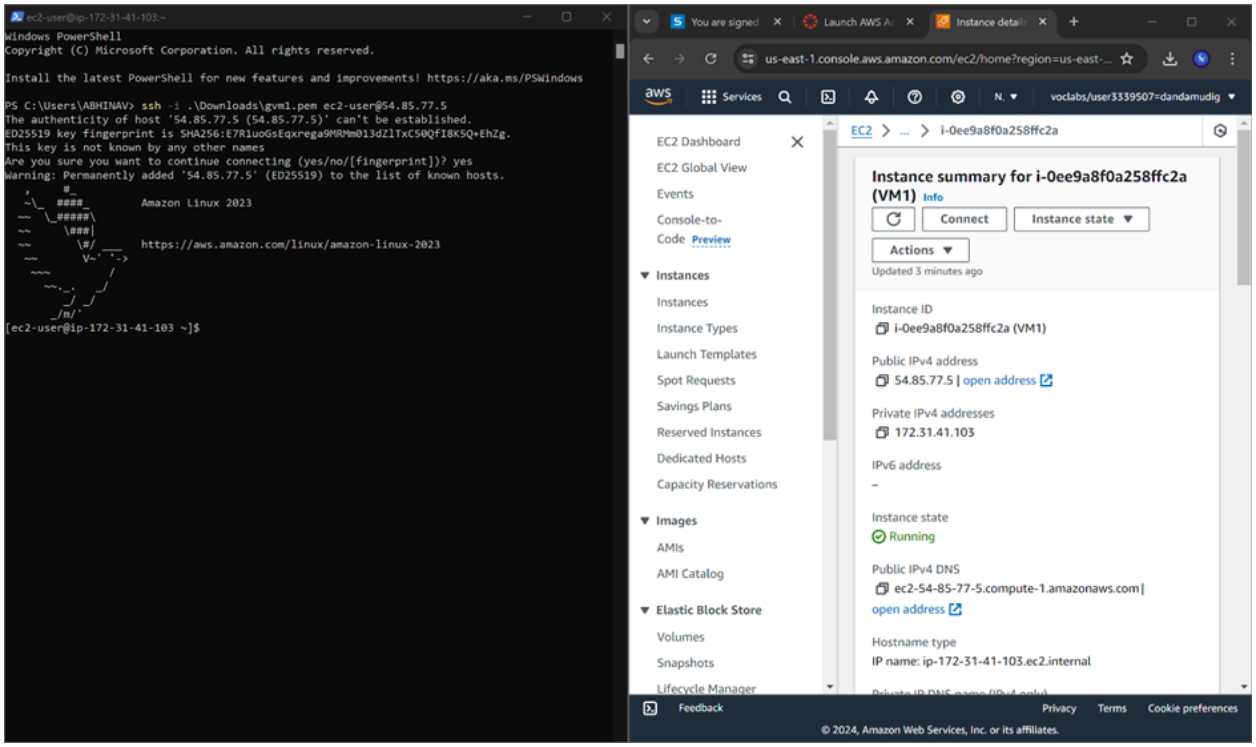
3. Creating 2 AWS EC2 Instances.

1: Creating the EC2 Instances.



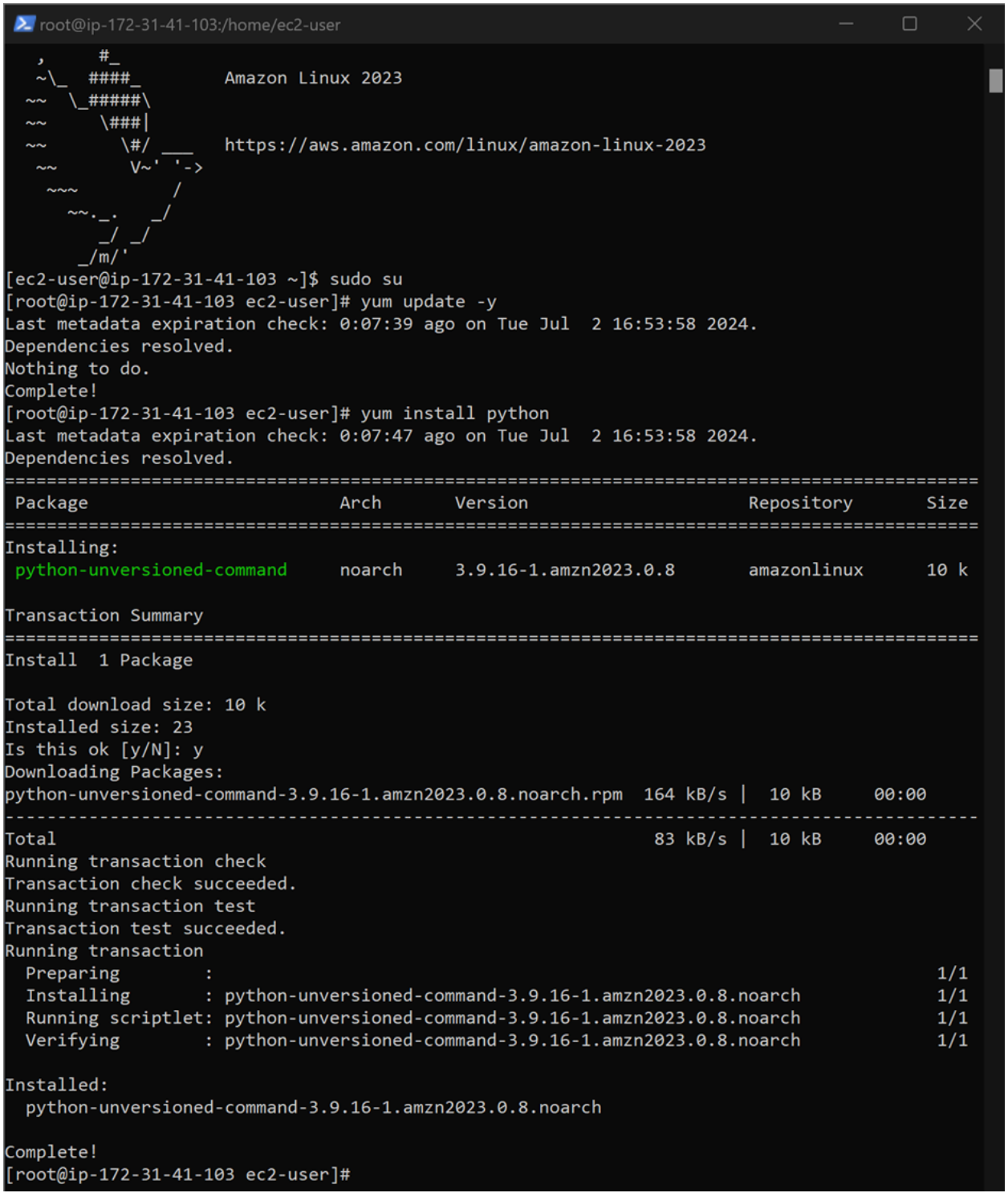
2. Copy the Public IPv4 address and use the Windows SSH Client to connect to the instance VM1 using this command *ssh -i* .

\Downloads\gvm1.pem ec2-user@"ip-address"

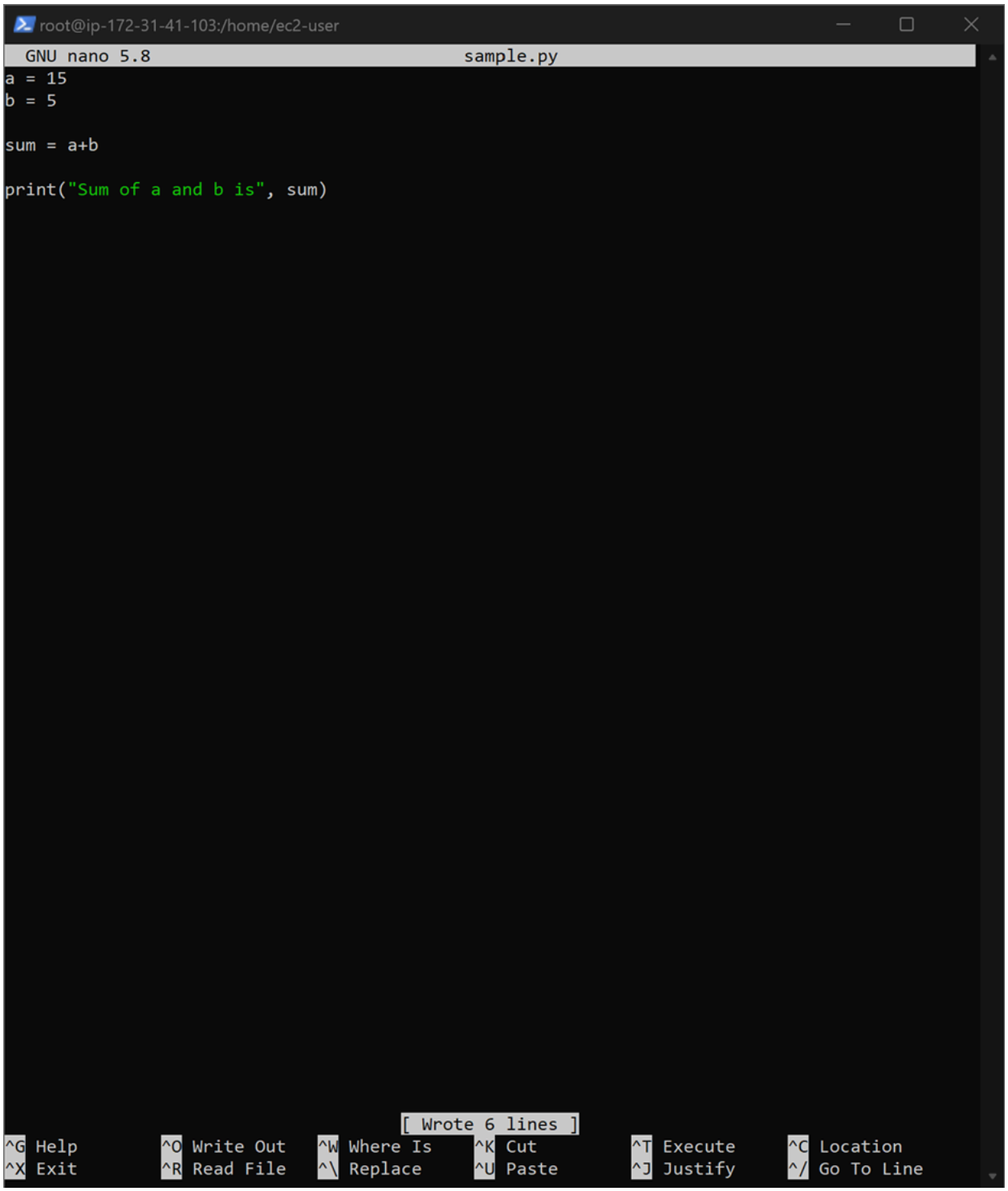


It's standard procedure after this use basic linux commands to install python and run a simple python program

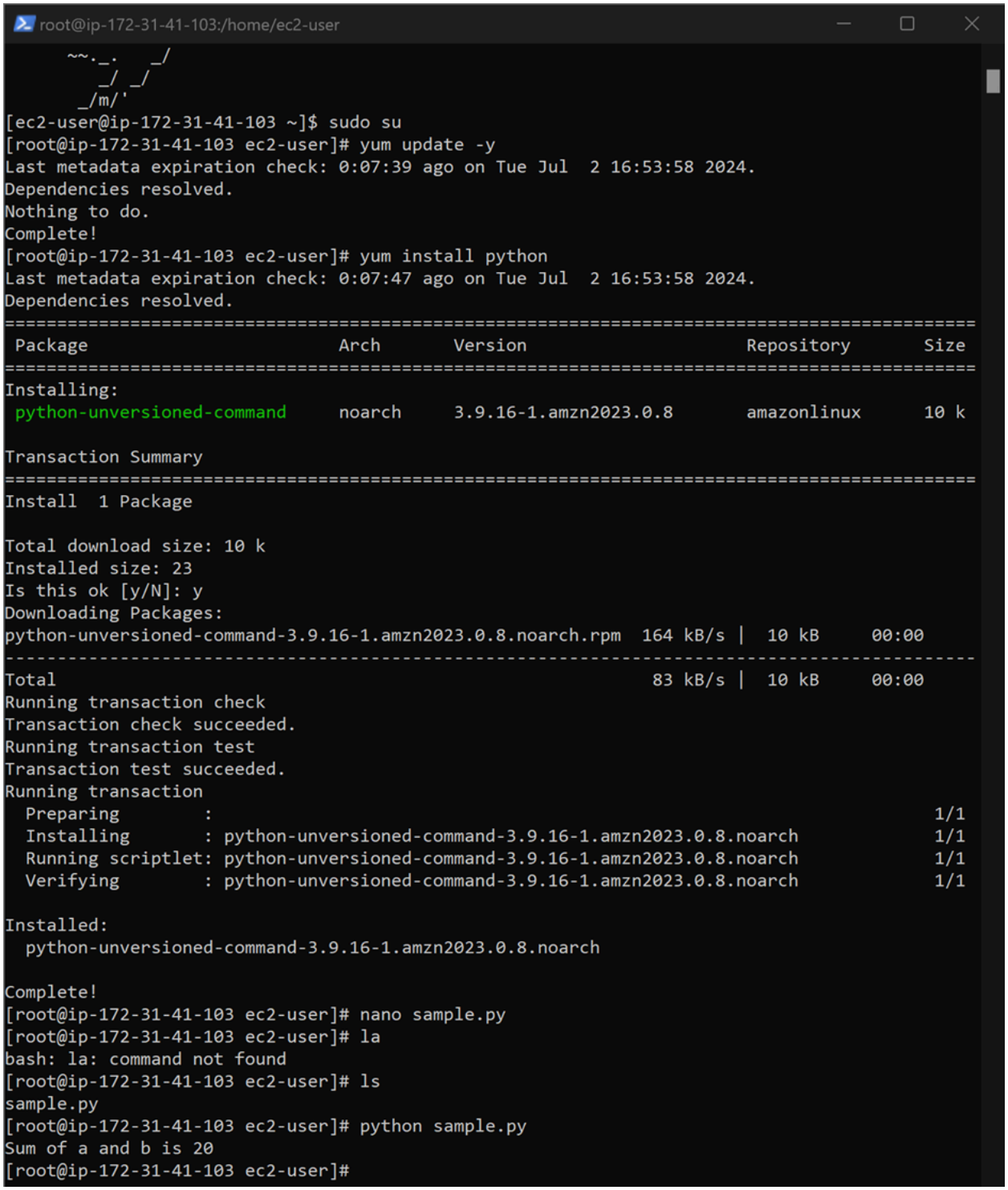
Installing python


root@ip-172-31-41-103:/home/ec2-user
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023
[ec2-user@ip-172-31-41-103 ~]$ sudo su
[root@ip-172-31-41-103 ec2-user]# yum update -y
Last metadata expiration check: 0:07:39 ago on Tue Jul 2 16:53:58 2024.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-41-103 ec2-user]# yum install python
Last metadata expiration check: 0:07:47 ago on Tue Jul 2 16:53:58 2024.
Dependencies resolved.
Package Arch Version Repository Size
Installing:
python-unversioned-command noarch 3.9.16-1.amzn2023.0.8 amazonlinux 10 k
Transaction Summary
Install 1 Package
Total download size: 10 k
Installed size: 23
Is this ok [y/N]: y
Downloading Packages:
python-unversioned-command-3.9.16-1.amzn2023.0.8.noarch.rpm 164 kB/s | 10 kB 00:00
Total 83 kB/s | 10 kB 00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing : 1/1
Installing : python-unversioned-command-3.9.16-1.amzn2023.0.8.noarch 1/1
Running scriptlet: python-unversioned-command-3.9.16-1.amzn2023.0.8.noarch 1/1
Verifying : python-unversioned-command-3.9.16-1.amzn2023.0.8.noarch 1/1
Installed:
python-unversioned-command-3.9.16-1.amzn2023.0.8.noarch
Complete!
[root@ip-172-31-41-103 ec2-user]#

Making a python file and saving it



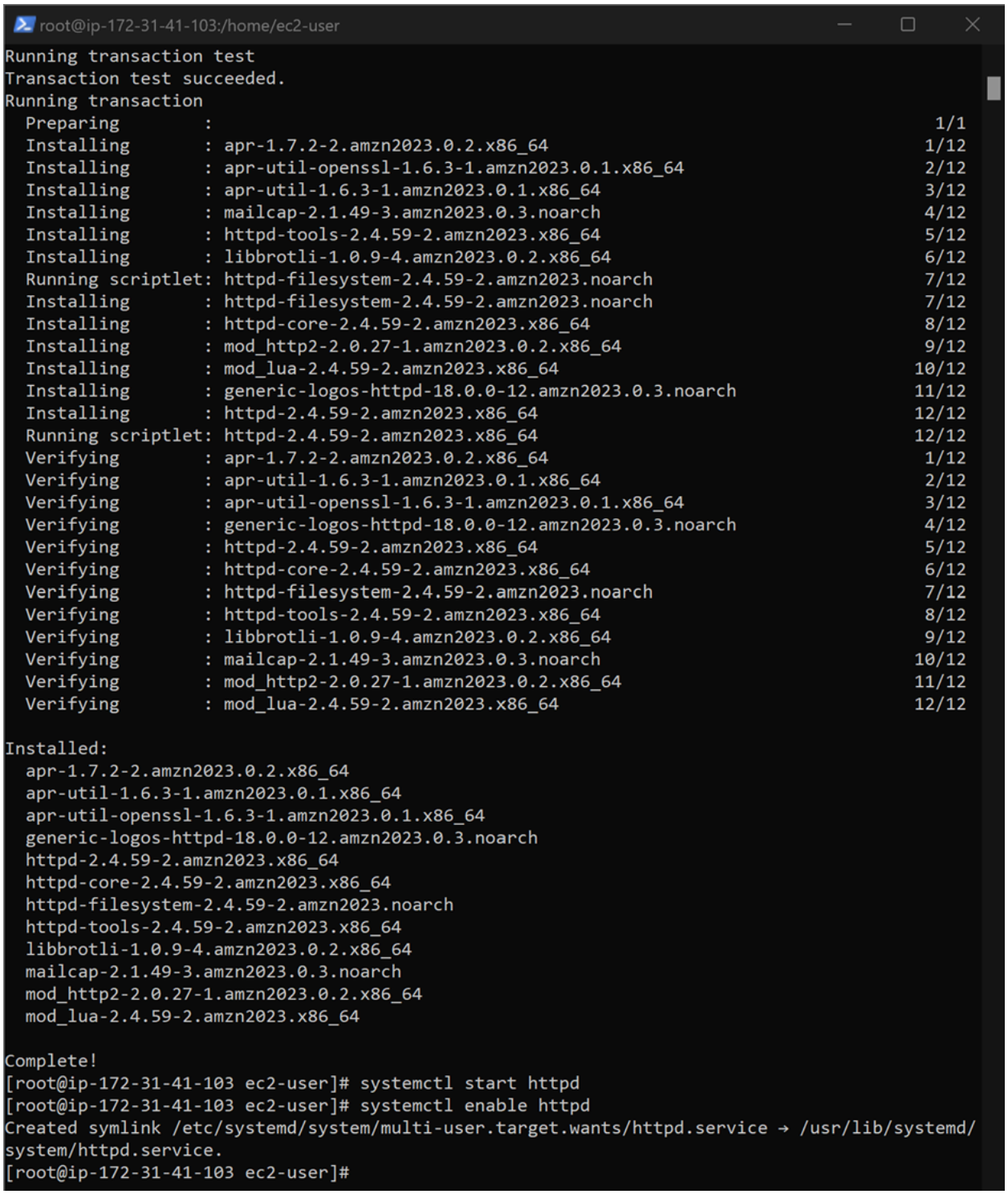
Listing the files and running the python file



Installing Apache and enabling it.

Start the server using: systemctl start httpd

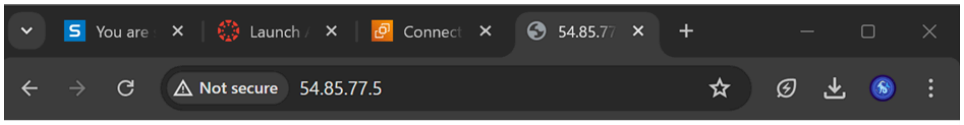
Enable the server using: systemctl enable httpd



Writing the html code.

4. Configure a Webserver on ‘Regno_EC2_VM2’ Instance and host your organization's website (Static Website) and provide access only to your machine.

Displaying the webpage: go to the instance page and copy the public IPv4 address and copy paste it into a new tab and the web page should load.



Hello. This is Christ University