	Pimpri Chinchwad Education Trust's **Pimpri Chinchwad College of Engineering (PCCoE)** (An Autonomous Institute) Affiliated to Savitribai Phule Pune University(SPPU) ISO 21001:2018 Certified by TUV	

Course: DevOps Laboratory	**Code:** BIT26VS01
Name: Amar Vaijinath Chavan	**PRN:** 124B2F001
Assignment 13: Explore the different services of AWS (EC2, S3, AWS Lambda, RDS, ELB, ECS) and create EC2 instance in AWS.	

Aim: To explore core Amazon Web Services (AWS) infrastructure and successfully provision an Ubuntu-based EC2 instance within the AWS cloud environment.

Objectives:

- To understand the utility and architecture of key AWS services.
- To navigate the AWS Management Console for cloud resource provisioning.
- To establish a secure remote connection to a cloud-based virtual server using SSH.

Theory:

1. **Amazon Elastic Compute Cloud (EC2)**
   EC2 provides scalable, resizable computing capacity in the form of virtual servers called instances. It eliminates the need to invest in hardware upfront, allowing for faster development and deployment of applications. Users can choose various configurations of CPU, memory, and storage to suit their specific application needs.

2. **Amazon Simple Storage Service (S3)**
   S3 is an object storage service offering industry-leading scalability, data availability, security, and performance. It is designed for 99.999999999% durability and stores data as objects within containers called "Buckets." It is widely used for backups, data archiving, and hosting static website content.

3. **AWS Lambda (Serverless Compute)**
   Lambda is a serverless, event-driven compute service that lets you run code for virtually any type of application or backend service without provisioning or managing servers. It automatically scales your application by running code in response to triggers such as changes in data in an S3 bucket or an HTTP request via API Gateway.

4. **Amazon Relational Database Service (RDS)**
   RDS is a managed service that makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while automating time-consuming administration tasks such as hardware provisioning, database setup, patching, and backups.

5. **Elastic Load Balancing (ELB)**
   ELB automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, and IP addresses. It increases the fault tolerance of your applications by ensuring that only healthy targets receive traffic and by scaling to handle changes in request patterns.

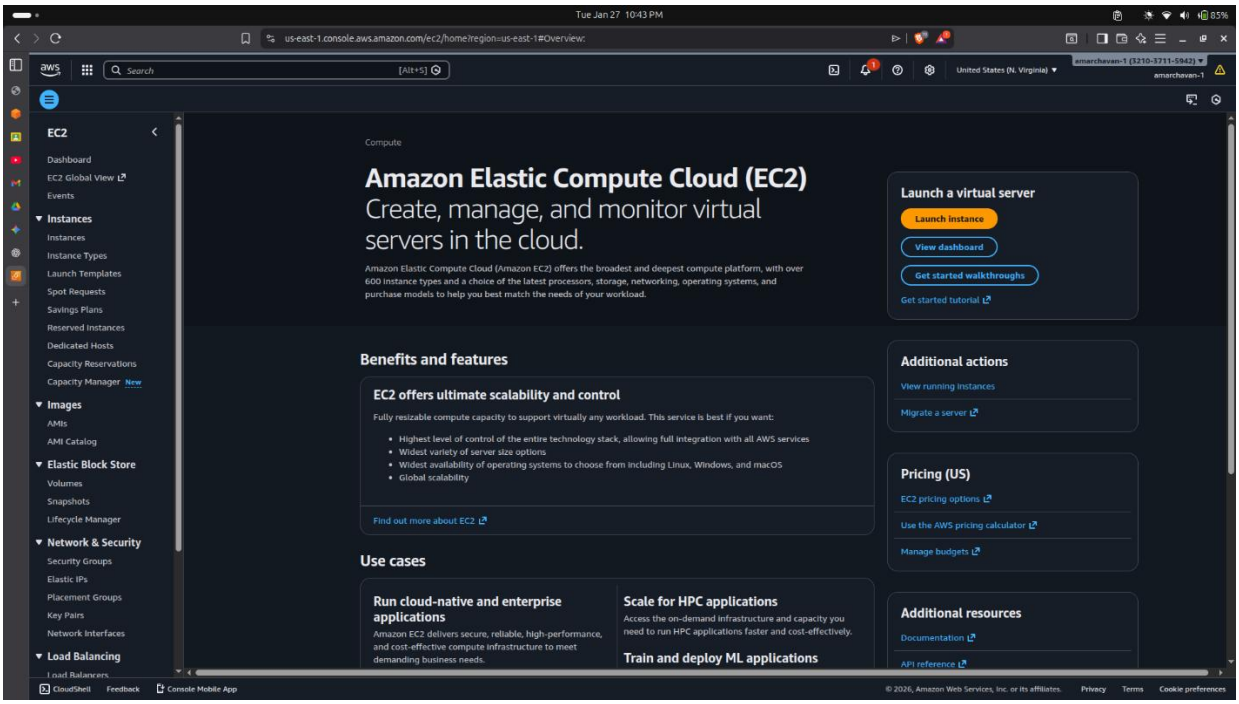
6. **Amazon Elastic Container Service (ECS)**
   ECS is a highly scalable, fast, container management service that makes it easy to run, stop, and manage Docker containers on a cluster. It integrates deeply with other AWS services such as IAM for security and ELB for traffic distribution, providing a robust platform for microservices.

7. **Key Infrastructure Components (AMI, Security Groups, and Key Pairs)**
   An **Amazon Machine Image (AMI)** is a template containing the software configuration (OS, application server) required to launch an instance. **Security Groups** act as virtual firewalls to control inbound and outbound traffic. **Key Pairs** utilize public-key cryptography to secure the login process; the user holds the private key (.pem), and AWS stores the public key.

Practical Procedure / Steps:

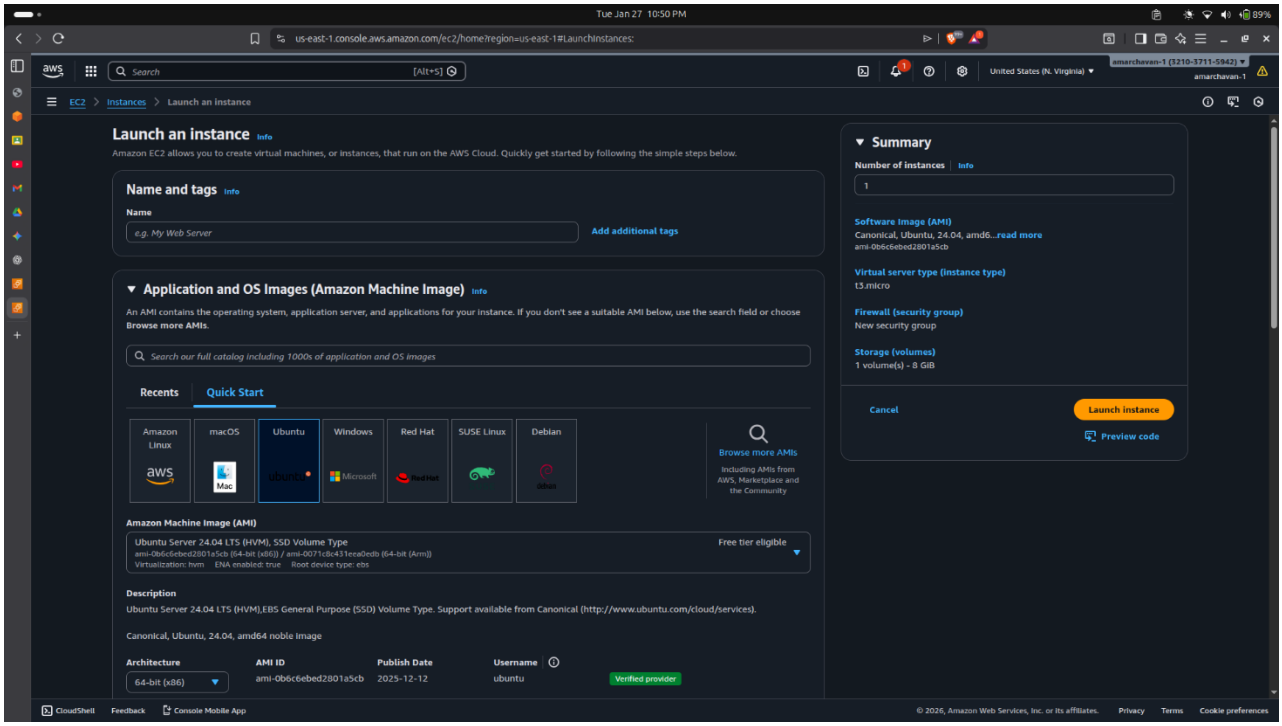
Step 1: AWS Console Navigation

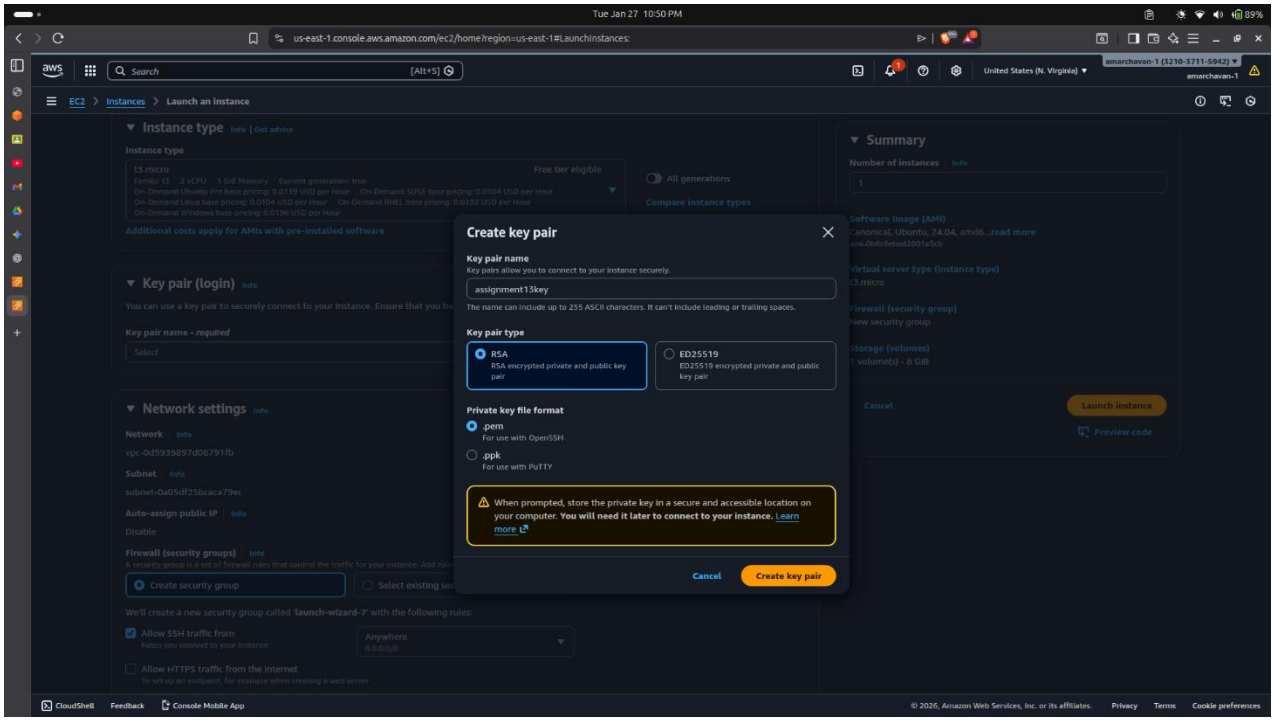
- Log in to the AWS Management Console.
- Search for and select the **EC2** service to open the EC2 Dashboard.



Step 2: Launching the EC2 Instance

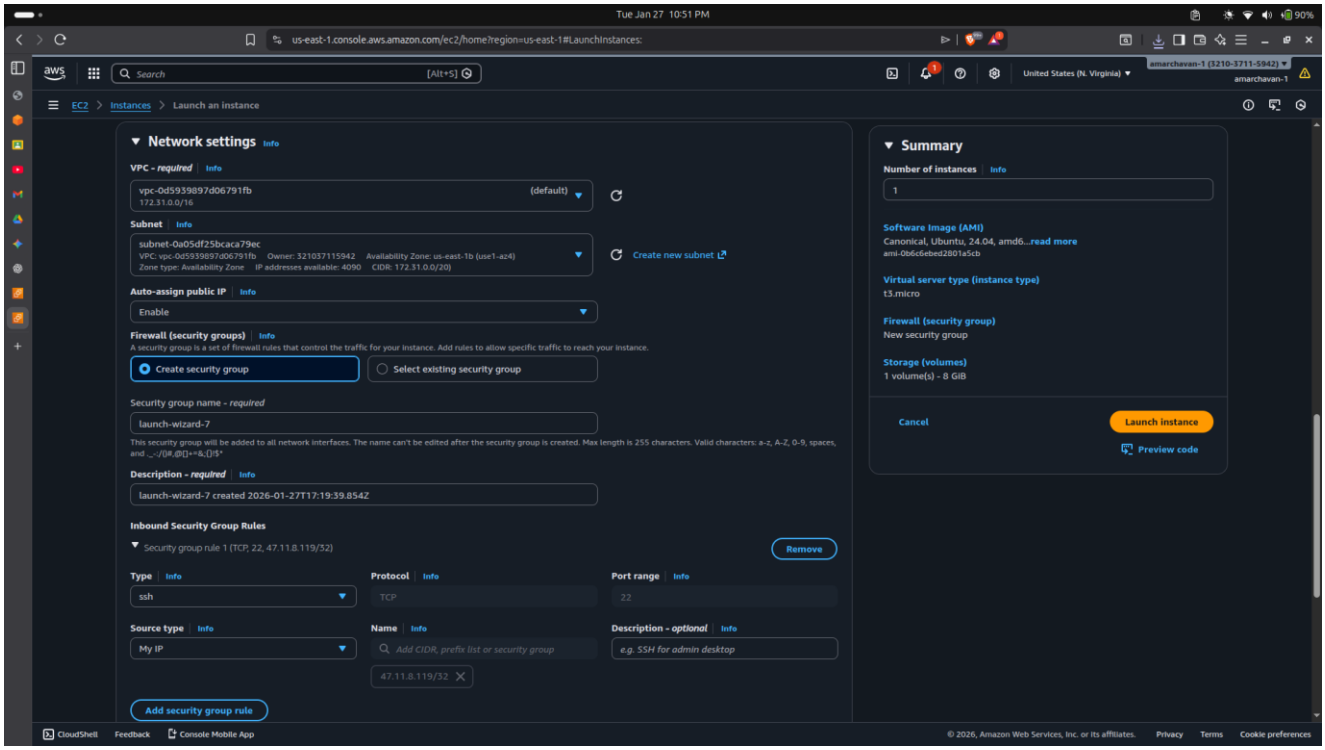
1. **Name and Tags:** Enter a name for the instance, such as Assignment13-Server.
2. **Application and OS Images (AMI):** Select **Ubuntu** and choose the **Ubuntu Server 24.04 LTS (HVM)** 64-bit (x86) image.
3. **Instance Type:** Select t3.micro, which is part of the AWS Free Tier.
4. **Key Pair (login):** Select "Create new key pair." Name it assignment13key, select **RSA**, and download the .pem file.

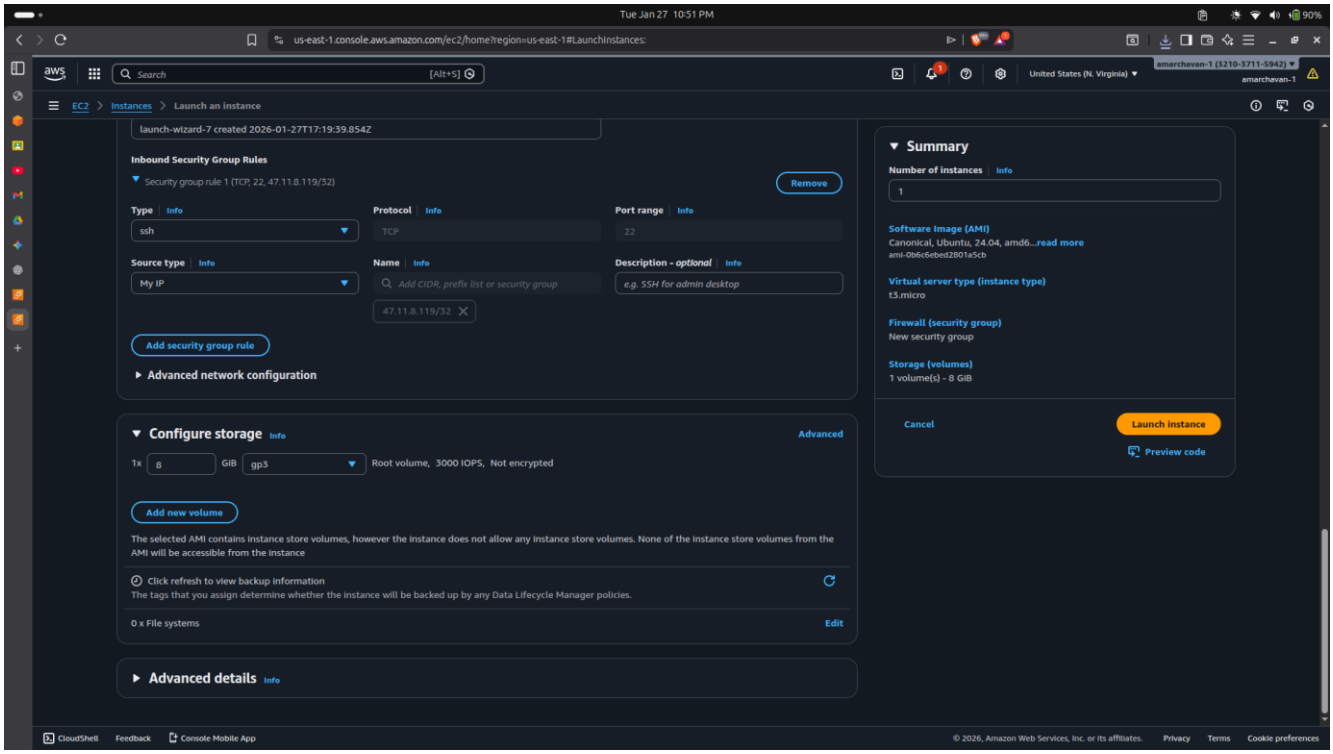




Step 3: Network and Storage Configuration

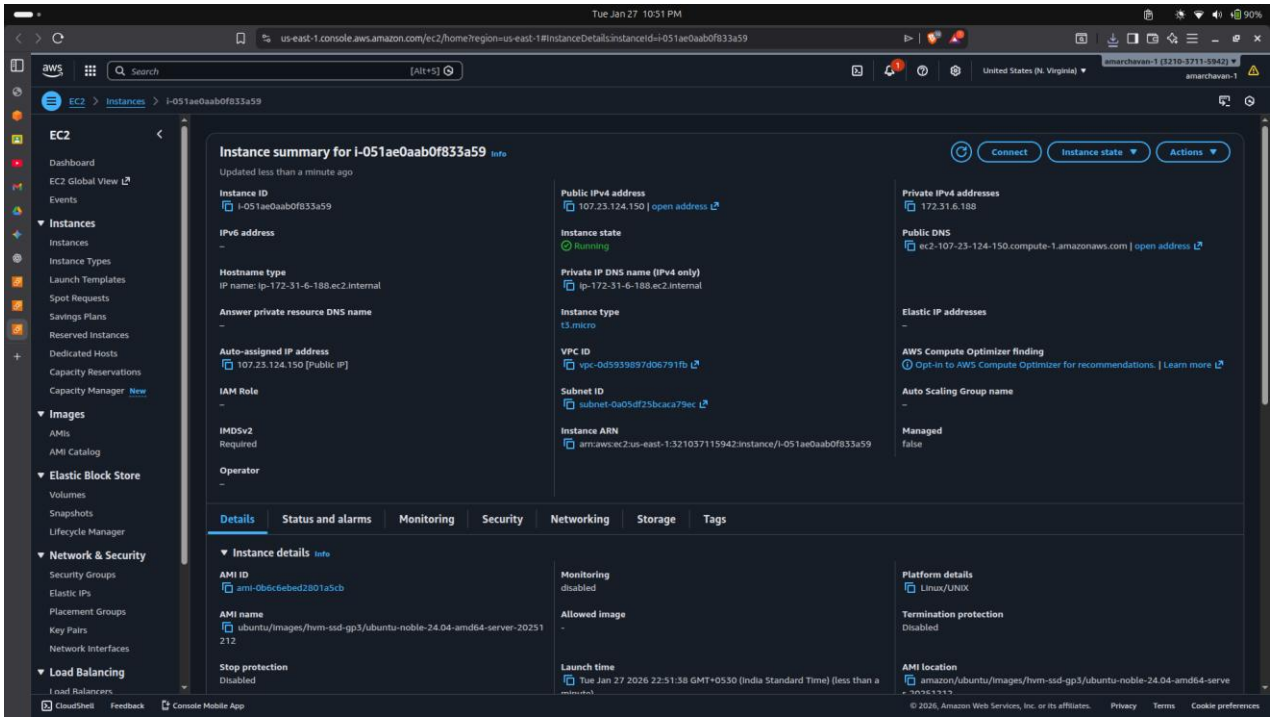
1. **Network Settings:** Ensure a VPC and Subnet are selected. Enable "Auto-assign public IP" to allow external access.
2. **Firewall (Security Groups):** Create a new security group. Add a rule to allow **SSH** traffic on port 22 from **My IP** for secure access.
3. **Configure Storage:** Maintain the default 8 GiB of gp3 (General Purpose SSD) storage.

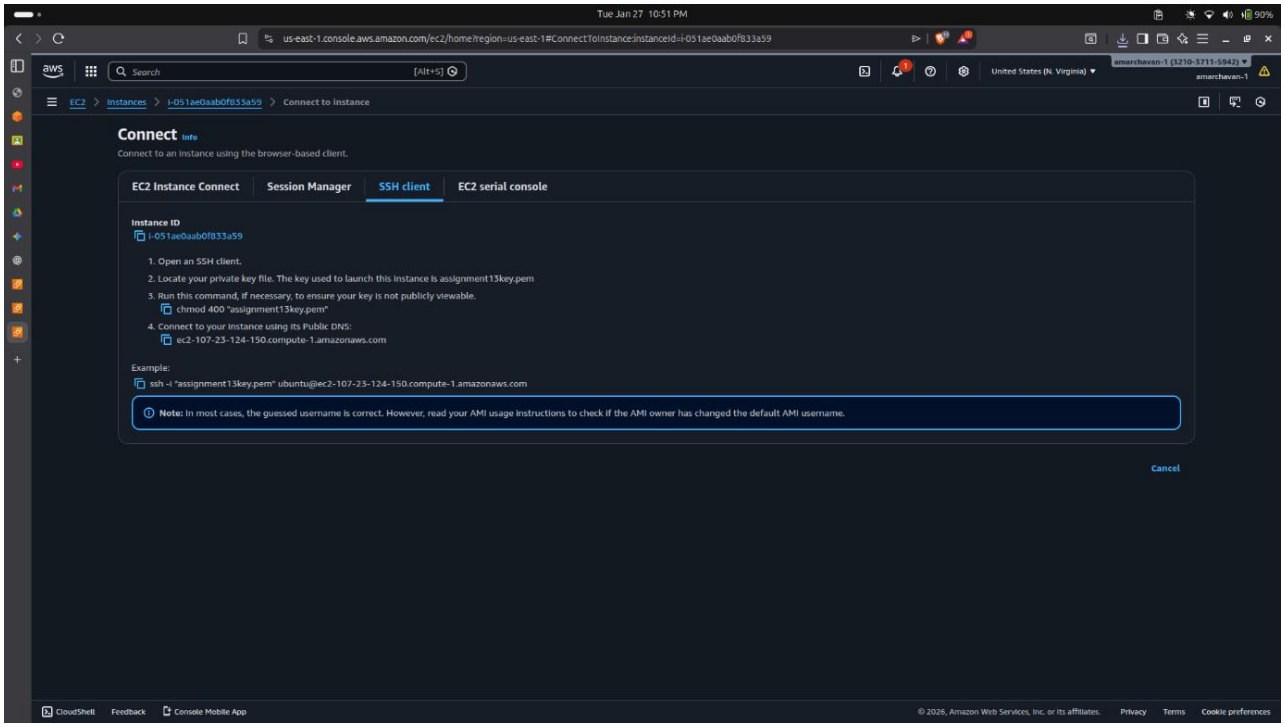


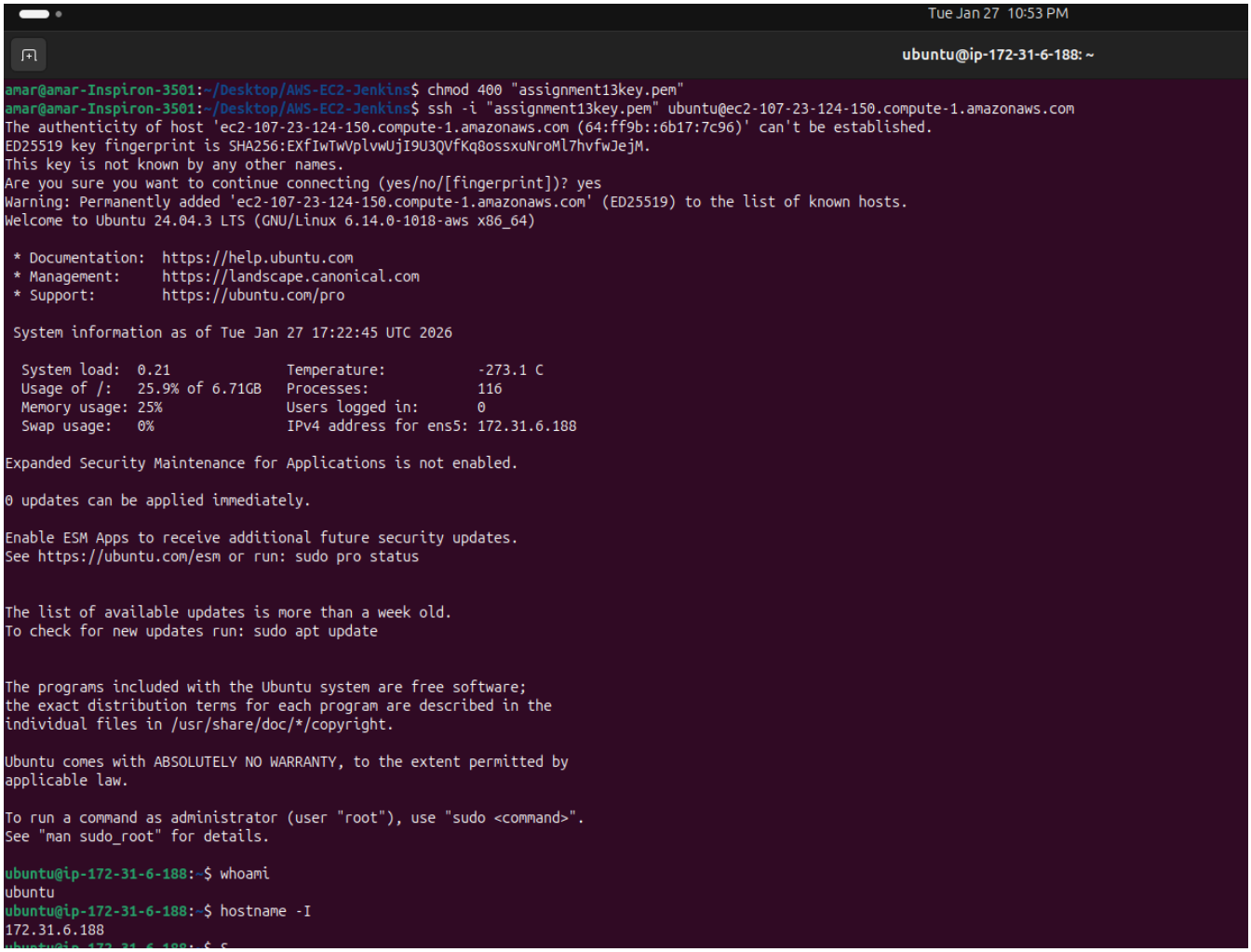


Step 4: Establishing SSH Connection

1. Open the terminal on your local Ubuntu machine.
2. Navigate to the directory containing your .pem file and secure the file permissions: chmod 400 "assignment13key.pem"
3. Connect to the instance using its Public IPv4 address: ssh -i "assignment13key.pem" ubuntu@ec2-107-23-124-150.compute-1.amazonaws.com







Conclusion

The successful completion of Assignment 13 demonstrates the efficient orchestration of cloud infrastructure using the AWS Management Console. By exploring a diverse suite of services—including EC2, S3, RDS, Lambda, ELB, and ECS—I gained an understanding of the decoupled and highly scalable nature of modern cloud environments. The practical execution involved the end-to-end provisioning of a virtual Ubuntu 24.04 server, covering critical configuration steps such as AMI selection, security group firewall rules, and key-pair management. Establishing a secure SSH connection to the cloud instance proved the ability to remotely manage Linux workloads in a production-ready environment.