St. Francis Institute of Technology, Mumbai-400 103

**Department Of Information Technology**

A.Y. 2024-2025

Class: TE-ITA/B, Semester: V

Subject: **Advanced DevOps Lab**

**Experiment – 1: To create a free tier AWS account.**

1. **Aim:** To understand benefits of cloud infrastructure and create a free tier AWS account.
2. **Objectives:** After study of this experiment, the student will be able to

* Understand basic cloud computing concepts
* Create a free tier AWS account.

1. **Lab objective mapped : ITL504.1:** To understand the fundamentals of Cloud Computing and be fully proficient with Cloud based DevOps solution deployment options to meet your business requirements.
2. **Prerequisite:** creating account on web app.
3. **Requirements:** Computer, Windows operating system, Internet Connection, web browser, credit card details.
4. **Pre-Experiment Theory:**

Cloud Computing definition:

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models.

Essential Characteristics:

On-demand self-service. A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.

Broad network access. Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g.,

mobile phones, tablets, laptops, and workstations).

Resource pooling. The provider’s computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, and network bandwidth.

Rapid elasticity. Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.

Measured service. Cloud systems automatically control and optimize resource use by leveraging a metering capability1 at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

Service Models:

Software as a Service (SaaS). The capability provided to the consumer is to use the provider’s applications running on a cloud infrastructure . The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited userspecific application configuration settings.

Platform as a Service (PaaS). The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider.3 The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.

Infrastructure as a Service (IaaS). The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls).

Deployment Models:

Private cloud. The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units). It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist

on or off premises.

Community cloud. The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises.

Public cloud. The cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider.

Hybrid cloud. The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds).

Cloud Computing Providers:

* AWS,
* Azure,
* Google Cloud

1. **Laboratory Exercise**
   * + 1. **Procedure:**

To create an AWS account:

1. Open the Amazon Web Services home page
2. Choose Create an AWS account.
3. Enter your account information, and then choose Verify email address. This will send a verification code to your specified email address.
4. Enter your verification code, and then choose Verify.
5. Enter a strong password for your root user, confirm it, and then choose Continue.
6. Choose Personal.
7. Enter your personal information.
8. Read and accept the AWS Customer Agreement
9. Choose Continue. At this point, you'll receive an email message to confirm that your AWS account is ready to use. You can sign in to your new account by using the email address and password you provided during sign up. However, you can't use any AWS services until you finish activating your account.
10. Enter the information about your payment method, and then choose Verify and Continue.
11. You can't proceed with the sign-up process until you add a valid payment method.
12. Enter your country or region code from the list, and then enter a phone number where you can be reached in the next few minutes.
13. Enter the code displayed in the CAPTCHA, and then submit.
14. When the automated system contacts you, enter the PIN you receive and then submit.
15. Select one of the available AWS Support plans.
16. Choose Complete sign up. A confirmation page appears that indicates that your account is being activated.
17. Check your email and spam folder for an email message that confirms your account was activated. Activation usually takes a few minutes but can sometimes take up to 24 hours.
18. After you receive the activation message, you have full access to all AWS services.

Walk through AWS home page:

AWS Management Console

In the image below, five controls are highlighted within the navigation bar:

1. Account information
2. Region selector
3. Service selector
4. Search box
5. AWS CloudShell



**8. Post-Experiments Exercise**

**A. Extended Theory:**

1. ACheck details about your account, including the address, contact information, billing settings, and more. Add a screenshot here.
2. Visit billing dashboard. Add a screenshot here.
3. Visit AWS global infrastructure- AWS Region menu.
4. Visit third highlighted menu the AWS service selector
5. Visit fifth highlighted item the AWS CloudShell icon.

**B. Questions:**

1. List advantages of Cloud Computing?(Soft copy)

2. Give some examples of AWS services to be used for Advanced DevOps?(Soft copy)

3. Write about AWS CloudShell(Write in Hand)

**C. Conclusion:**(Write in Hand)

1. Write what was performed in the experiment

2. Mention few applications of what was studied.

3. Write the significance of the studied topic

**D. References:**

1. NIST SP 800-145, The NIST Definition of Cloud Computing, Peter Mell

Timothy Grance.

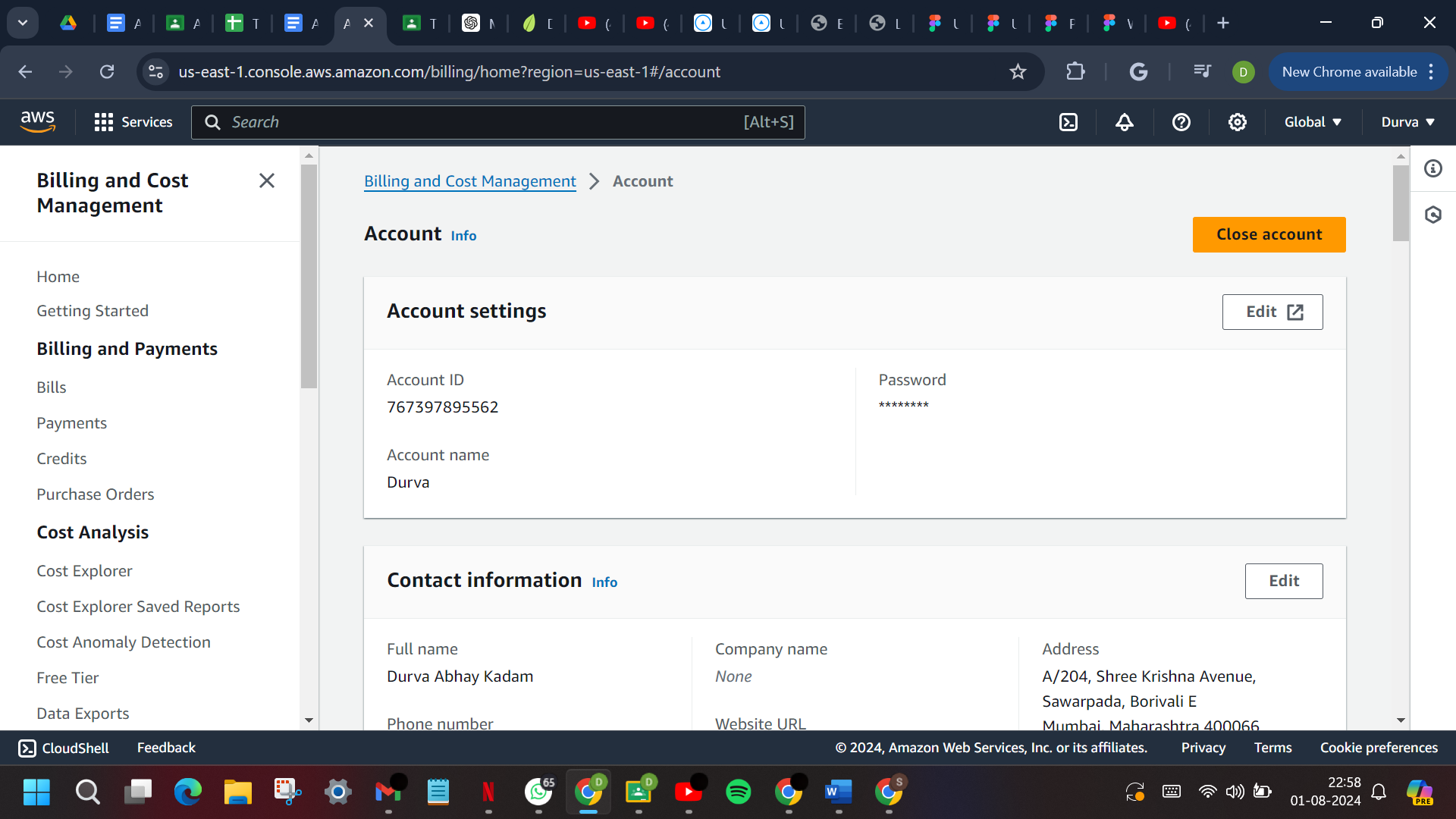
2. <https://aws.amazon.com/>

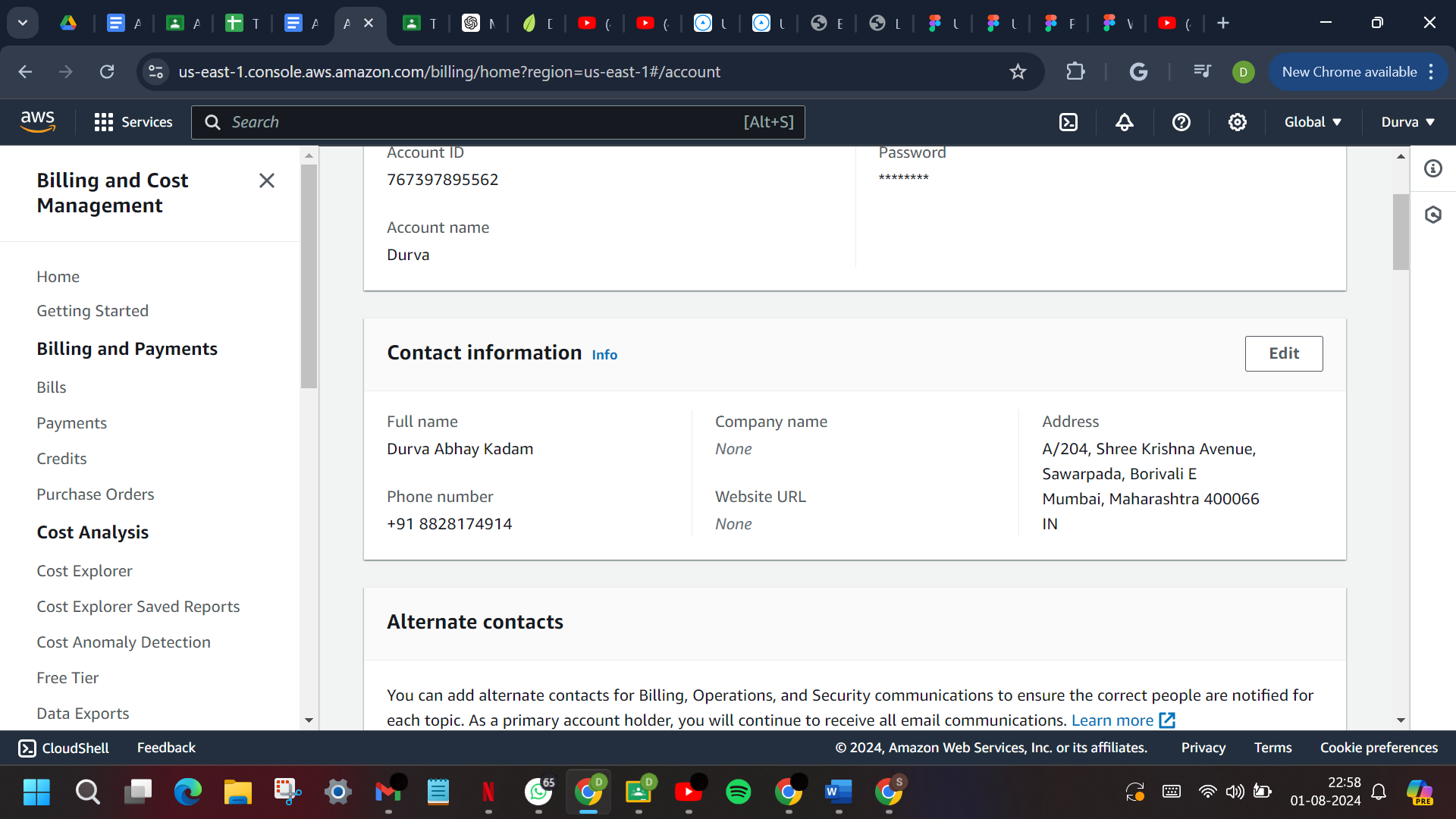
**—-------------------------------------------------------------------------------------------------**

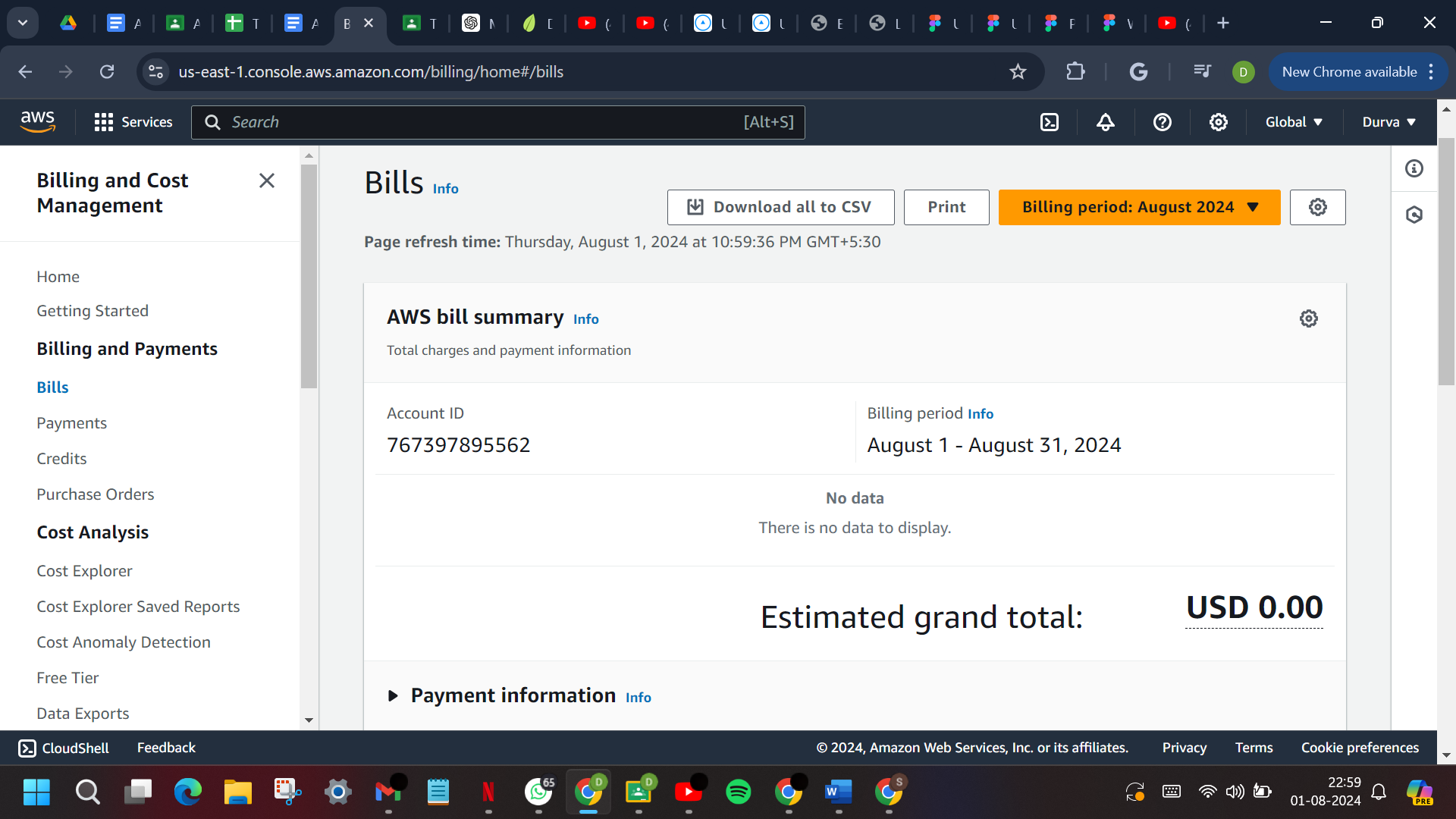
**8. Post-Experiments Exercise**

**A. Extended Theory:**

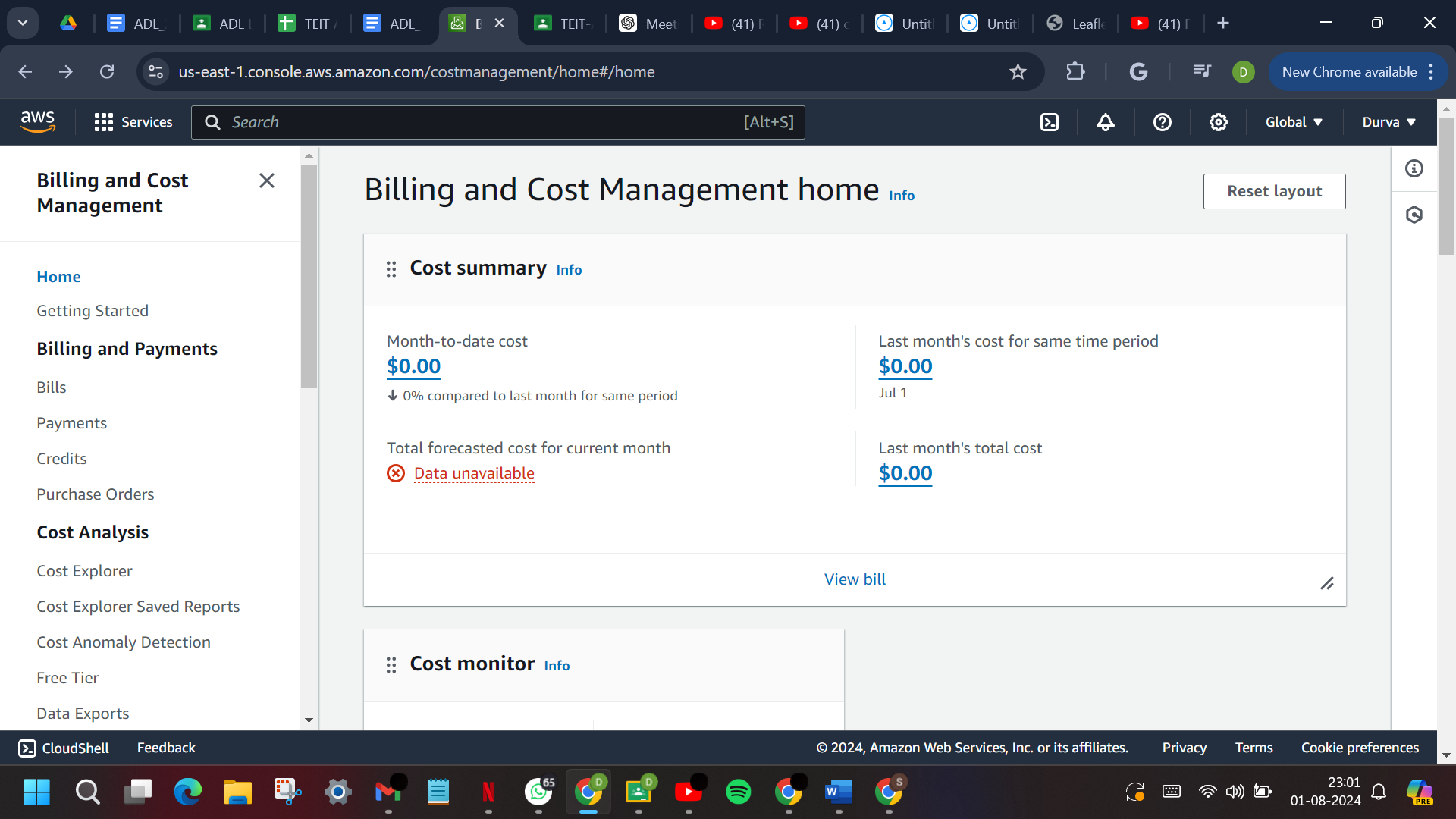
1. Check details about your account, including the address, contact information, billing settings, and more.



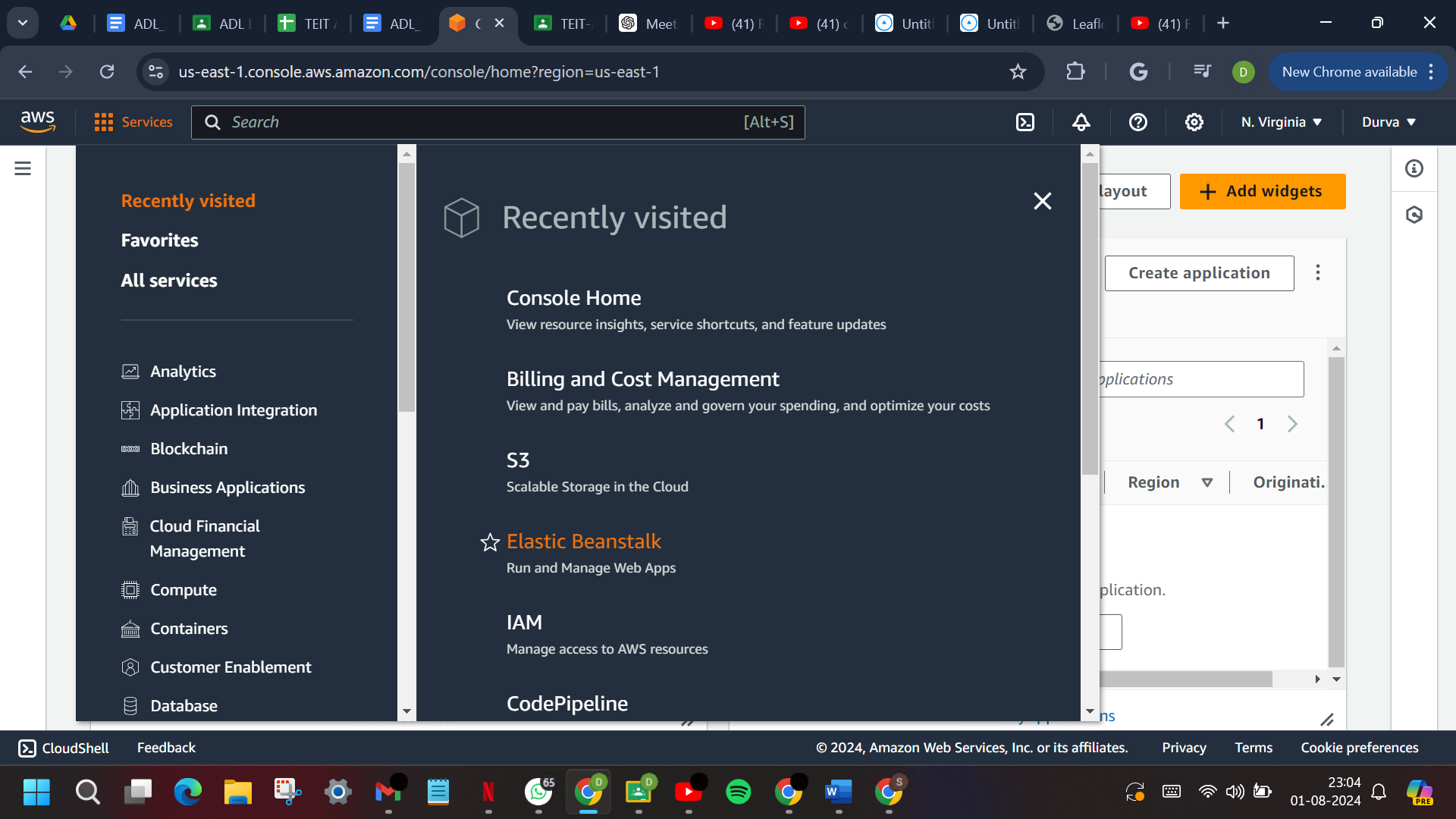




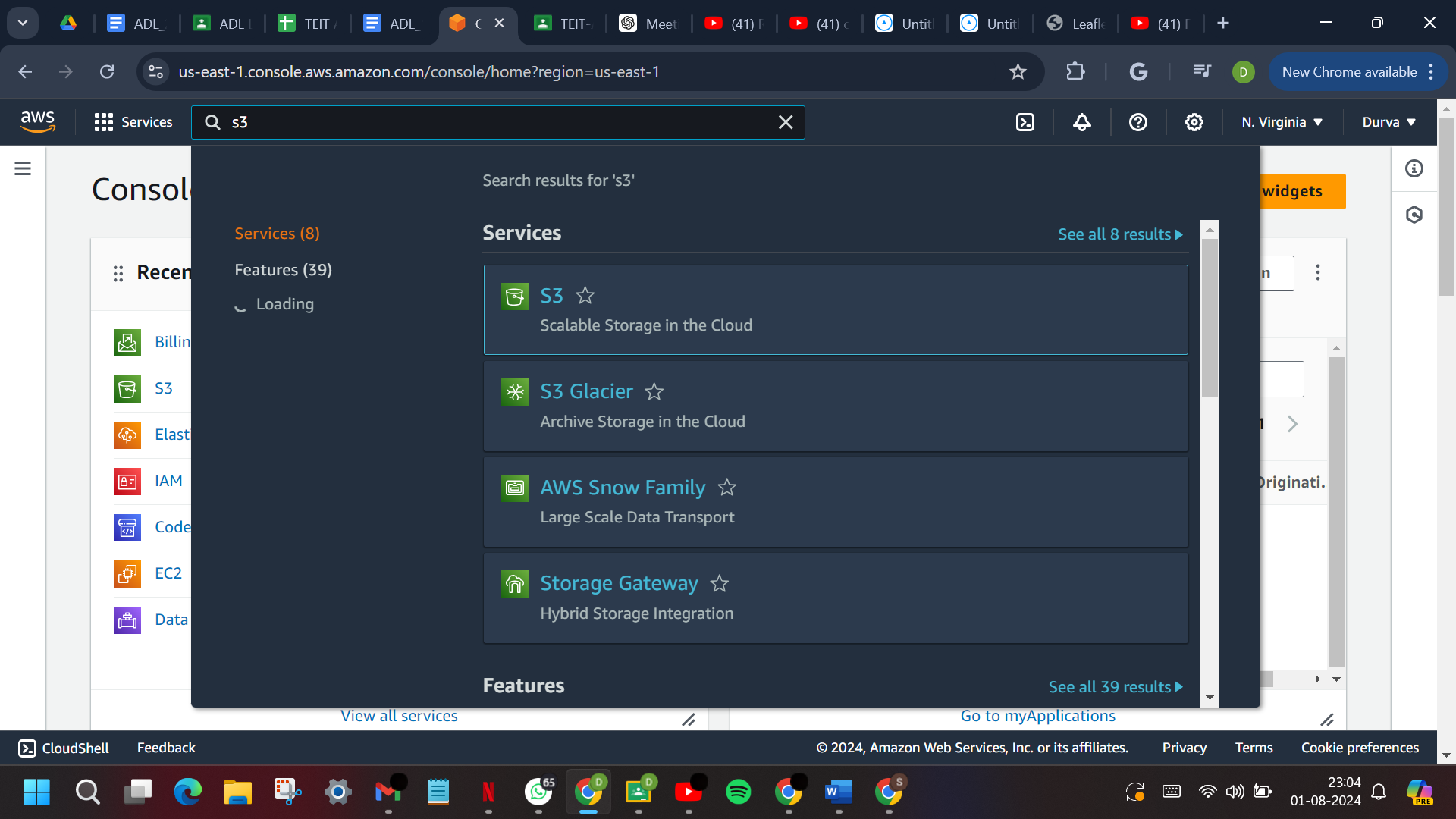
1. Visit billing dashboard. Add a screenshot here.



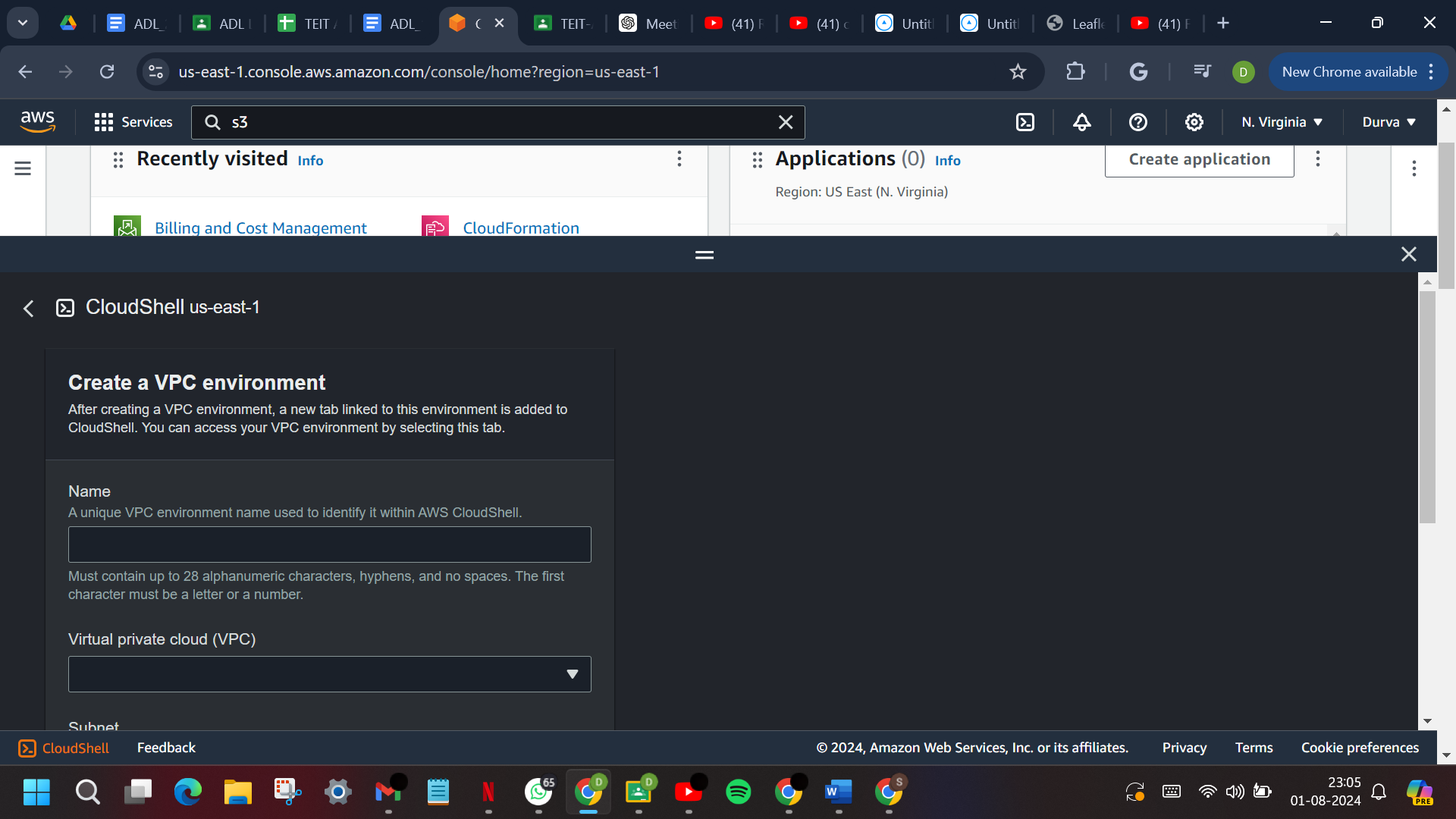
1. Visit AWS global infrastructure- AWS Region menu



1. Visit third highlighted menu the AWS service selector



1. Visit fifth highlighted item the AWS CloudShell icon.



**B. Questions:**

**1. List advantages of Cloud Computing?(Soft copy)**

Cloud computing is the delivery of computing services including servers, storage, databases, networking, software, and more. It offers scalable resources, on-demand availability, and flexible payment models, enabling businesses to access technology services without managing physical infrastructure.

**Advantages of Cloud Computing :**

1. **Cost Efficiency**: Reduces capital expenditure and operational expenses.
2. **Scalability**: Easily scales resources up or down based on demand.
3. **Accessibility**: Access applications and data from anywhere with an internet connection.
4. **Flexibility**: Offers a range of services, including IaaS, PaaS, and SaaS.
5. **Disaster Recovery and Backup**: Provides easy data backup and disaster recovery solutions.
6. **Performance and Reliability**: Ensures services are available with minimal downtime through redundant infrastructure.
7. **Automatic Updates**: Automatically updates software and services with the latest features and security patches.
8. **Collaboration**: Allows multiple users to work on documents and projects simultaneously.
9. **Security**: Implements strong security measures and offers compliance with various industry standards and regulations.
10. **Environmental Benefits**: Reduces energy use and promotes sustainable IT practices.

**2. Give some examples of AWS services to be used for Advanced DevOps?**

1. **AWS CodePipeline:** Facilitates CI/CD by automating the build, test, and deployment phases, ensuring rapid and reliable updates.
2. **AWS CodeBuild**: A managed service that compiles source code, executes tests, and generates deployable software packages.
3. **AWS CodeDeploy**: Automates code deployment across various environments, including EC2 instances and on-premises servers.
4. **AWS CloudFormation**: Allows you to manage and provision AWS resources through infrastructure as code, using declarative template files.
5. **AWS Elastic Beanstalk**: Simplifies the process of deploying, scaling, and managing web applications in multiple programming languages.
6. **AWS OpsWorks:** Uses Chef and Puppet for automating server configurations and deployments.
7. **Amazon CloudWatch:** Offers comprehensive monitoring and observability, providing insights to optimize performance and respond to system changes.
8. **AWS Lambda**: Enables serverless computing by running code without the need to manage servers, ideal for scalable applications.
9. **AWS ECS and EKS:** Managed services for running, managing, and scaling containerized applications with ease.
10. **AWS Systems Manager:** Centralizes operational data from multiple AWS services, automating tasks like patching and inventory collection.

These services enhance automation, scalability, and efficiency in managing the software development lifecycle, from code development to deployment and monitoring.