**Capstone Project**

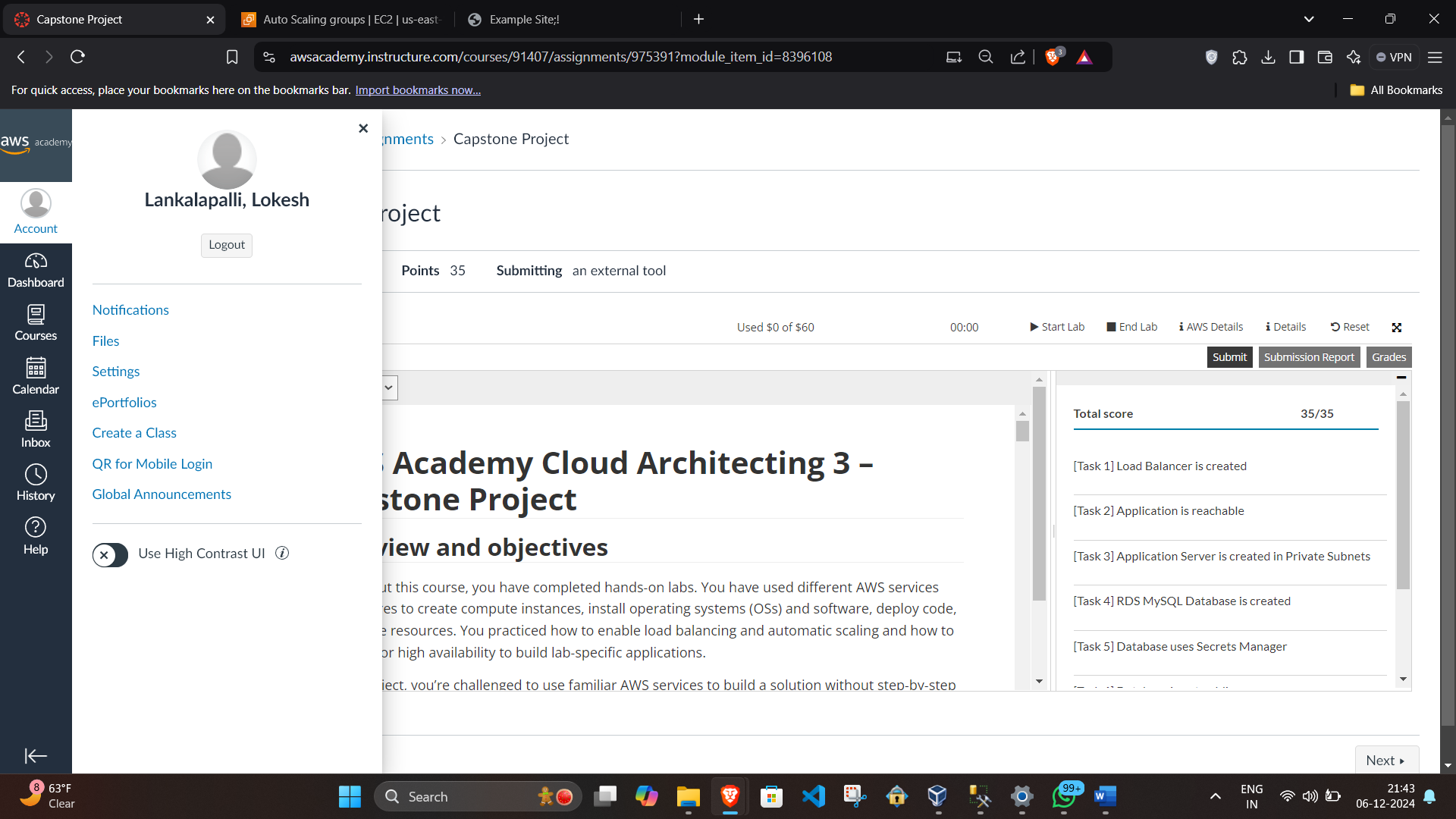
IFT 560: Cloud Architecture Info Tech

Student Name: Lokesh Lankalapalli

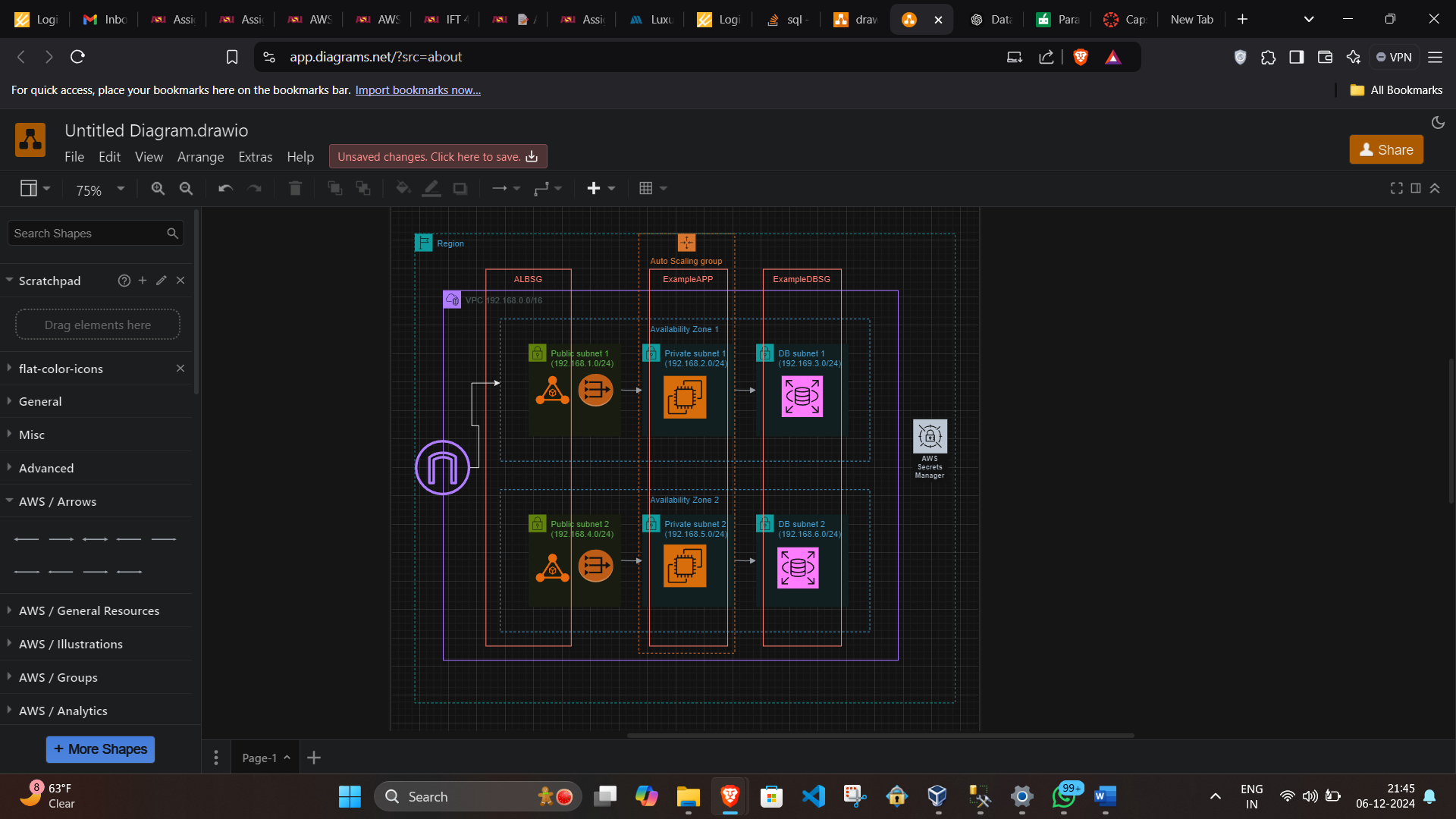
Professor Michael Walsh

Assignment Due Date: 12-06-2024

**Screenshot of the Grade:**

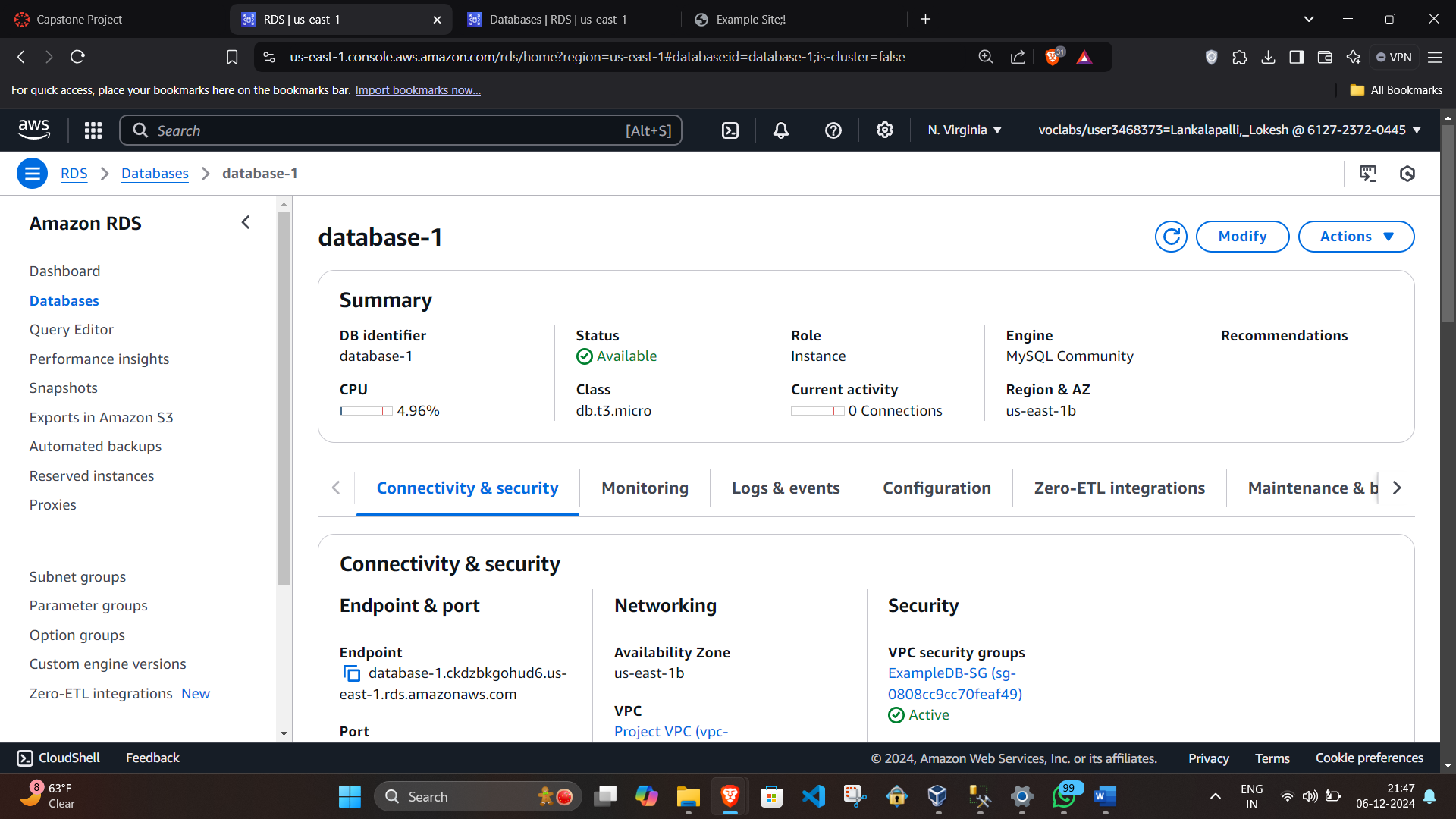
****

**Screenshot of the diagram:**

****

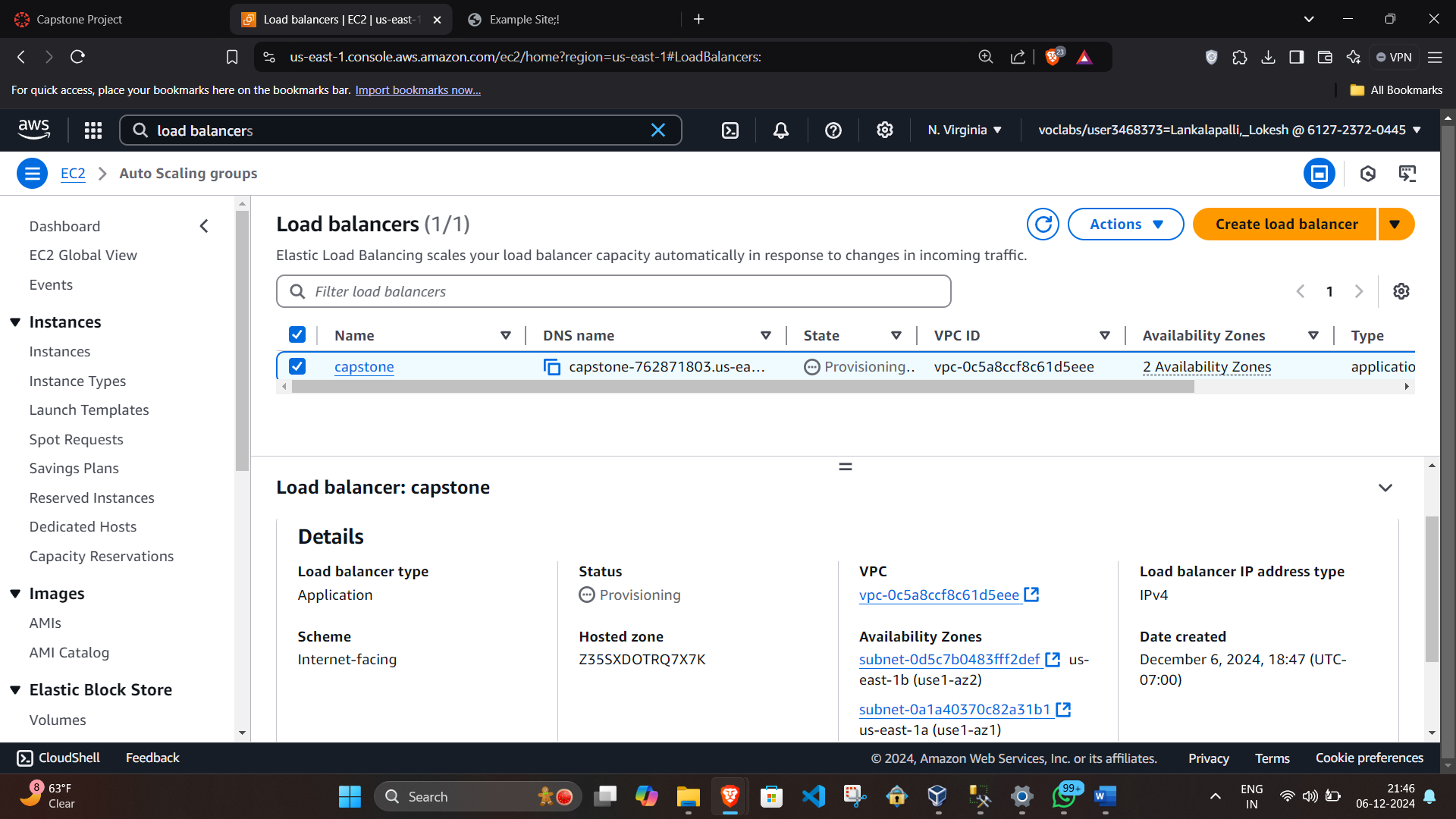
**Design Summary**

**Database Design**

****

To ensure a secure and scalable database backend, I utilized Amazon RDS MySQL. Amazon RDS was chosen for its managed nature, which simplifies maintenance tasks such as backups, patching, and scaling. The database was deployed within a private subnet, isolating it from direct internet access. Access was restricted through a dedicated security group that allowed connections solely from the private subnets hosting the application servers. To manage database credentials securely, AWS Secrets Manager was used, eliminating the need to hardcode sensitive details in the application. This approach enhanced both security and operational efficiency.

**High Availability and Load Balancing**



I put in place an Application Load Balancer (ALB) to effectively manage user traffic and offer fault tolerance. To act as the gateway for inbound queries, the ALB was set up throughout public subnets. High availability and failure resilience were guaranteed by dividing traffic across application servers spread across many Availability Zones. In order to allow the ALB to divert traffic in the event that a server became unhealthy, health checks were set up to monitor server status. Additionally, the ALB was essential in connecting with an Auto Scaling group that optimized cost and performance by dynamically adjusting the number of servers based on demand.

**Auto Scaling and Application Servers**

EC2 instances that were started by an Auto Scaling group housed the PHP application. This team made use of a launch template that was already set up and had the required application code and software stack. To improve security, instances were placed inside private subnets, and NAT Gateways were utilized to provide outgoing internet connection for patches and updates. In order to dynamically modify the number of instances, the Auto Scaling group used a target tracking strategy, scaling down during periods of low demand and scaling up during periods of high traffic. This kept expenses under control and guaranteed steady performance.

**Migration of Data**

One crucial step was moving data from the EC2 instance's old database to the new RDS MySQL database. Using common tools, the lab's SQL dump file was imported into the RDS database. The database credentials were safely obtained via Secrets Manager, ensuring security all the way through. This migration made sure that the new, managed database configuration has all the data needed for the application.

**Security Procedures**

Throughout the project, security was a top priority. The architecture reduced vulnerability to possible attacks by placing the database and application servers on private subnets. IAM roles were used to securely control access, and the public subnet's ALB served as the sole point of entry for users. Additionally, by following the least privilege principle and guaranteeing strong credential security, AWS Secrets Manager offered safe storage for private information.

**Validation and Testing**

By using the load balancer's DNS name to access the web application, the finished architecture was examined. Data from the RDS database was queried to confirm the PHP application's functionality. By tracking traffic distribution via the ALB and analyzing Auto Scaling behavior under simulated load, performance and scalability were assessed. These tests verified that the solution satiated the project's security, scalability, and high availability requirements.  
  
In accordance with architectural best practices, our strategy made optimal use of important AWS services to create a reliable and secure solution for hosting the MySQL database and PHP application. For the website of the social research organization, the resulting architecture offers a scalable, high-performing, and secure environment.