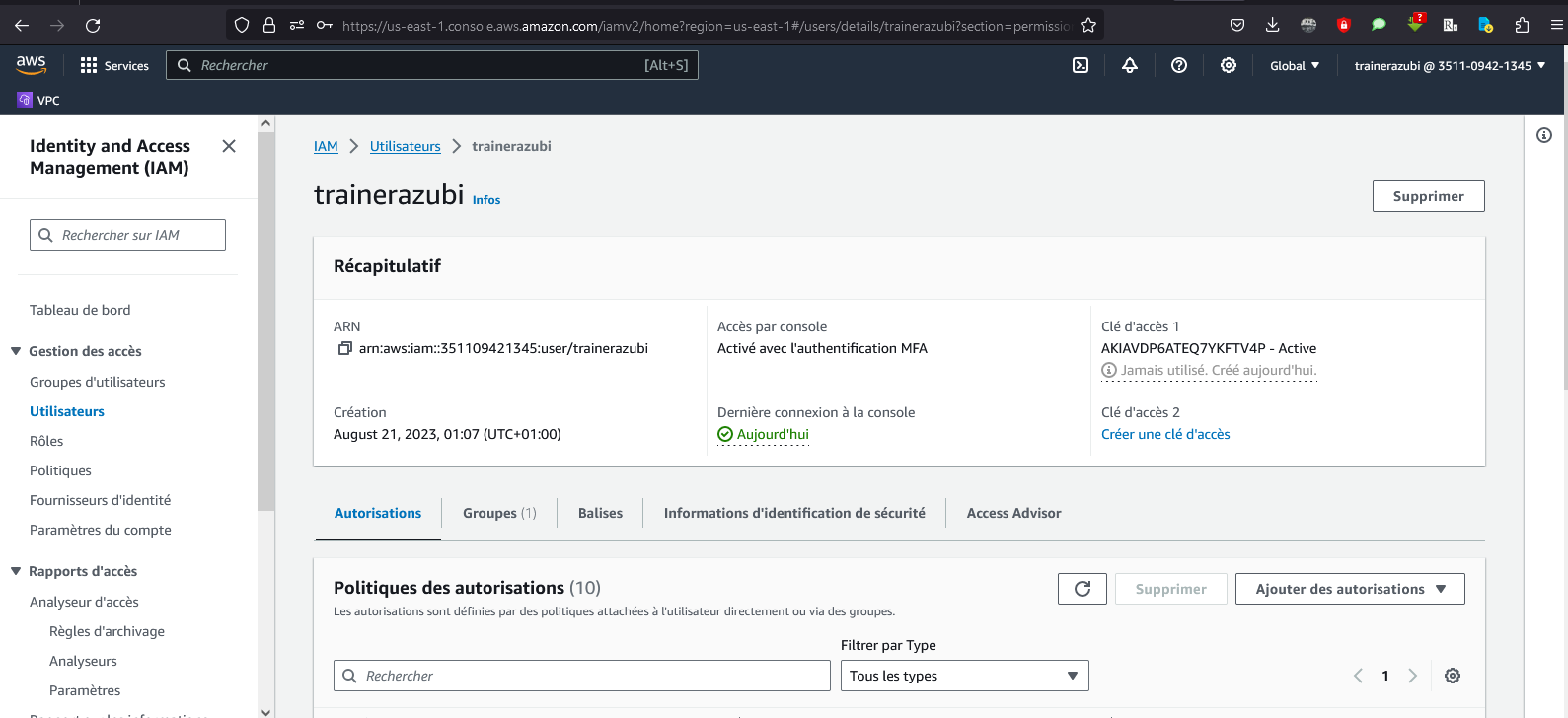
**Project:** MIGRATION OF A SAMPLE DATABASE TO AWS

1. **Sign up for an AWS account:**

This is the starting point, and it's crucial to have an AWS account to access AWS services.

* IAM
* Users
* Add user
* access to console
* IAM user
* custom password, not reset
* Attach existing policies
* not tags
* return to user or login
* select user
* security credentials
* CLI, I understand
* create acccess key



1. **Introduction to AWS:**

**EC2 (Elastic Compute Cloud)**: For deploying virtual instances.

**S3 (Simple Storage Service)**: For data storage.

**RDS (Relational Database Service)**: For managing relational databases.

**DynamoDB:** For NoSQL databases.

**AWS DMS (Database Migration Service):** The key tool in our project for migration.

**Concrete Example:**

Here is a concrete example of navigating the AWS console after logging in:

You open your browser and go to aws.amazon.com.

You click on "Sign in to the AWS Console" and enter your authentication information.

Once logged in, you see the AWS Console dashboard with shortcuts to various services.

You use the search bar to find the "RDS" service and access the RDS console for database management.

You explore other services like EC2 and S3 to understand how they work.

1. **AWS Well-Architected Framework:**

Understand the five pillars of the AWS Well-Architected Framework:

**Operational Excellence:** Optimize operations to deliver a reliable and efficient service.

**Security:** Implement robust security measures to protect sensitive data.

**Reliability:** Ensure the system operates reliably and without interruption.

**Performance Efficiency:** Optimize performance to meet application needs.

**Cost Optimization:** Reduce unnecessary costs while maintaining optimal performance.

1. **AWS Infrastructure Overview:**

Detailed Description: This step aims to familiarize you with the fundamentals of AWS infrastructure, including regions, Availability Zones, and Edge Locations, as well as to understand the key role of AWS Database Migration Service (DMS) in our project. Here is a detailed guide with configurations, commands, and outcomes:

1. **Understanding Key Concepts**:

AWS Regions: AWS is distributed worldwide into geographical regions. Each region is an independent geographical area with multiple data centers. Availability Zones (AZs): Each AWS region consists of multiple Availability Zones. AZs are distinct and isolated data centers from each other. Edge Locations: These are AWS endpoints for content distribution via Amazon CloudFront, AWS's content delivery network (CDN) service.

1. **Using the AWS Console**:

Sign in to the AWS console. Use the search bar to access the AWS DMS console. You can perform the search by typing "DMS" in the bar.

1. **Exploring AWS DMS**:

Once in the AWS DMS console, explore the available features and options. For example, you can configure database replication, replication instances, and monitor the status of replication tasks.

1. **Understanding the Role of AWS DMS in Migration**:

AWS DMS is the primary tool we will use for migrating the database. It enables continuous replication of source data to the target database on AWS.

**Outcome**:

After following these steps, you will have a basic understanding of AWS infrastructure, including how regions, Availability Zones, and Edge Locations are structured. You will also be familiar with the AWS DMS console, which plays a central role in our database migration project.

**Concrete Example**: Imagine you are exploring the AWS DMS console:

You access the AWS console. In the search bar, you type "DMS" and click on the result to access the DMS console. You can see options to create a new replication instance and configure migration tasks. By exploring replication instances, you can check the real-time data replication status between the source and target.

This example illustrates how you can use the AWS console to understand the operation of AWS DMS, which will be crucial for our database migration project.

1. **Planning and Discovery:**

Detailed Description: This step is crucial for laying the foundation of database migration. It involves defining your migration goals, assessing your source database, and preparing a draft of your web application. Here is a detailed guide with configurations, commands, and outcomes:

1. Define Migration Goals:

Identify the reasons for migration: You are migrating the database to reduce costs, enhance performance, or for other specific reasons. Establish clear objectives: For example, reduce the database downtime duration during migration.

1. Perform a Database Assessment:

Evaluate the source database: Analyze the database structure, data volume, schema complexity, and identify application dependencies. Identify potential issues: Spot potential incompatibilities between the source database and the target database engine on AWS.

1. Develop a Simple Web Application:

Create a demonstration web application using common technologies like HTML, CSS, and JavaScript. This application will serve as a testbed for data migration and connectivity with the target database on AWS.

1. Security and Compliance Objectives:

Identify security and compliance requirements: If your organization has strict security or compliance rules, ensure that you incorporate them into the migration planning.

**Outcome:**

After following these steps, you will have a clear understanding of your migration goals, the characteristics of your source database, and a demonstration web application ready for testing.

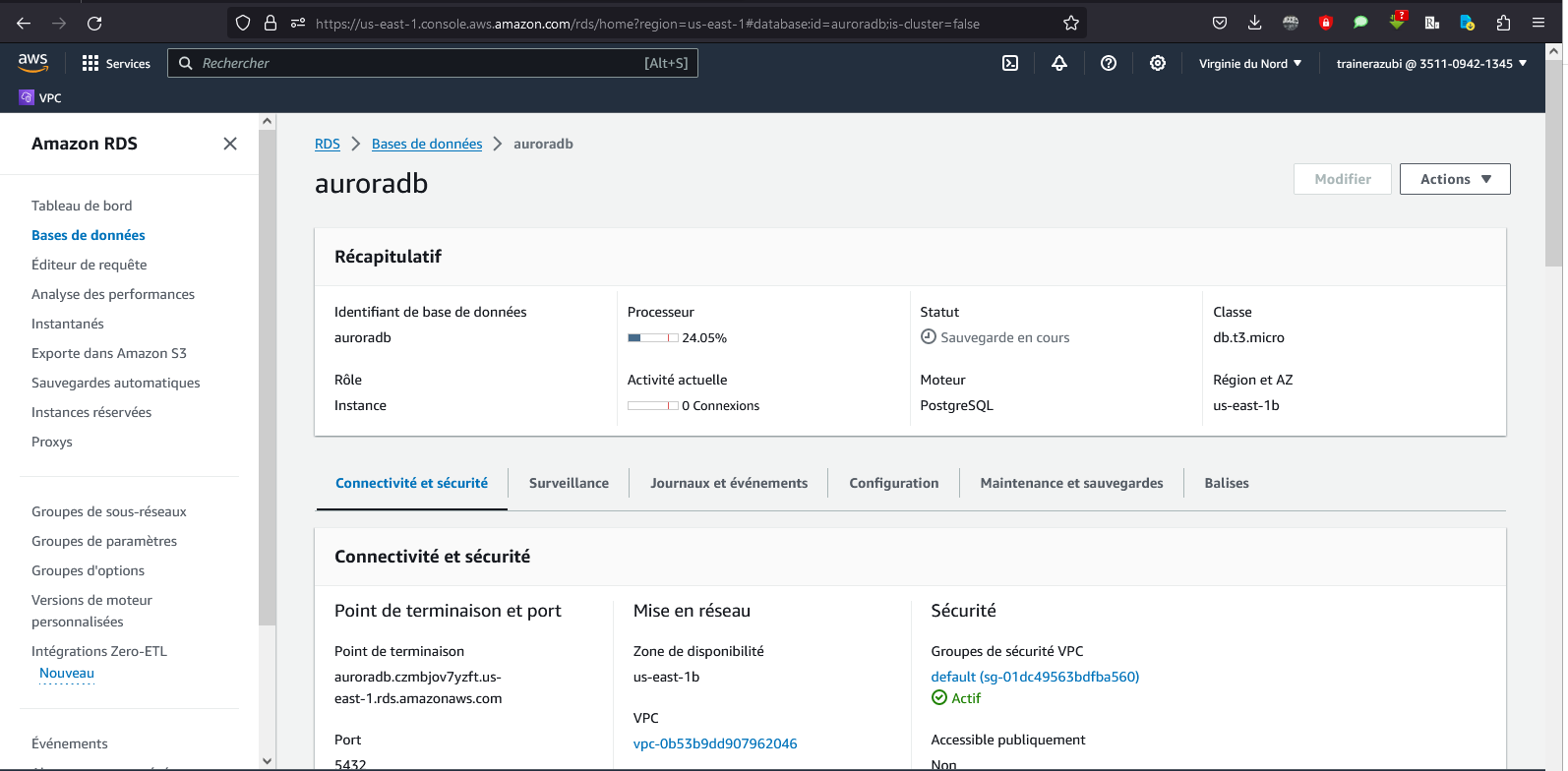
**Concrete Example**:

Let's take the example of defining migration goals:

* You identify that the primary goal of the migration is to reduce the operational costs of the database.
* You establish a specific goal to reduce costs by at least 20% compared to the current infrastructure.
* You also set a goal for minimal database availability during migration, which should not exceed 2 hours of downtime.

This example demonstrates how you can define clear objectives for your database migration project, which will guide the entire migration process.

1. **Selecting the Target Database on AWS**:



**Detailed Description:**

In this step, you will choose the appropriate AWS database service for your migration project. There are several options, including Amazon RDS, Amazon Aurora, Amazon DynamoDB, and more. Here is a detailed guide with configurations, commands, and outcomes:

1. **Evaluate Your Project Requirements**:

Understand the specific requirements of your project, such as the type of database, capacity, performance, availability, and scalability. Consider compatibility, security, and compliance requirements.

1. **Explore AWS Database Services:**

Use the AWS console or AWS documentation to explore available database services, including: Amazon RDS (Relational Database Service): For relational databases compatible with MySQL, PostgreSQL, Oracle, SQL Server, etc. Amazon Aurora: For highly-performant and scalable MySQL or PostgreSQL databases. Amazon DynamoDB: For a fully managed NoSQL database. Other services like Amazon Redshift for data analytics.

1. **Compare Options:**

Compare the features of each service, such as performance, availability, scalability, costs, and ease of management. Use tools like the AWS Cost Calculator to estimate the costs associated with each option.

1. **Select the Appropriate Service**:

Based on your evaluation, select the AWS database service that best meets your project's needs. Configure initial service settings, such as instance size, storage type, security, and backup options.

**Outcome**:

At the end of this step, you will have chosen the most suitable AWS database service for your migration project. You will be ready to configure this service to accommodate your database.

**Concrete Example**: Let's assume you choose Amazon RDS as the target database service for your migration project:

You assessed that your existing database is a MySQL database.

You compared Amazon RDS with other options and found that it offers good MySQL compatibility, high availability, and backup and restore features.

You estimated the costs associated with Amazon RDS using the AWS Cost Calculator and determined that it fits within your budget.

1. **Designing the Target Database Architecture**:
2. **Data Migration**:
3. **Architecture**: