**EXERCISE #1**

Screenshot 1

A screenshot of a computer

Description automatically generated

A screenshot of a computer

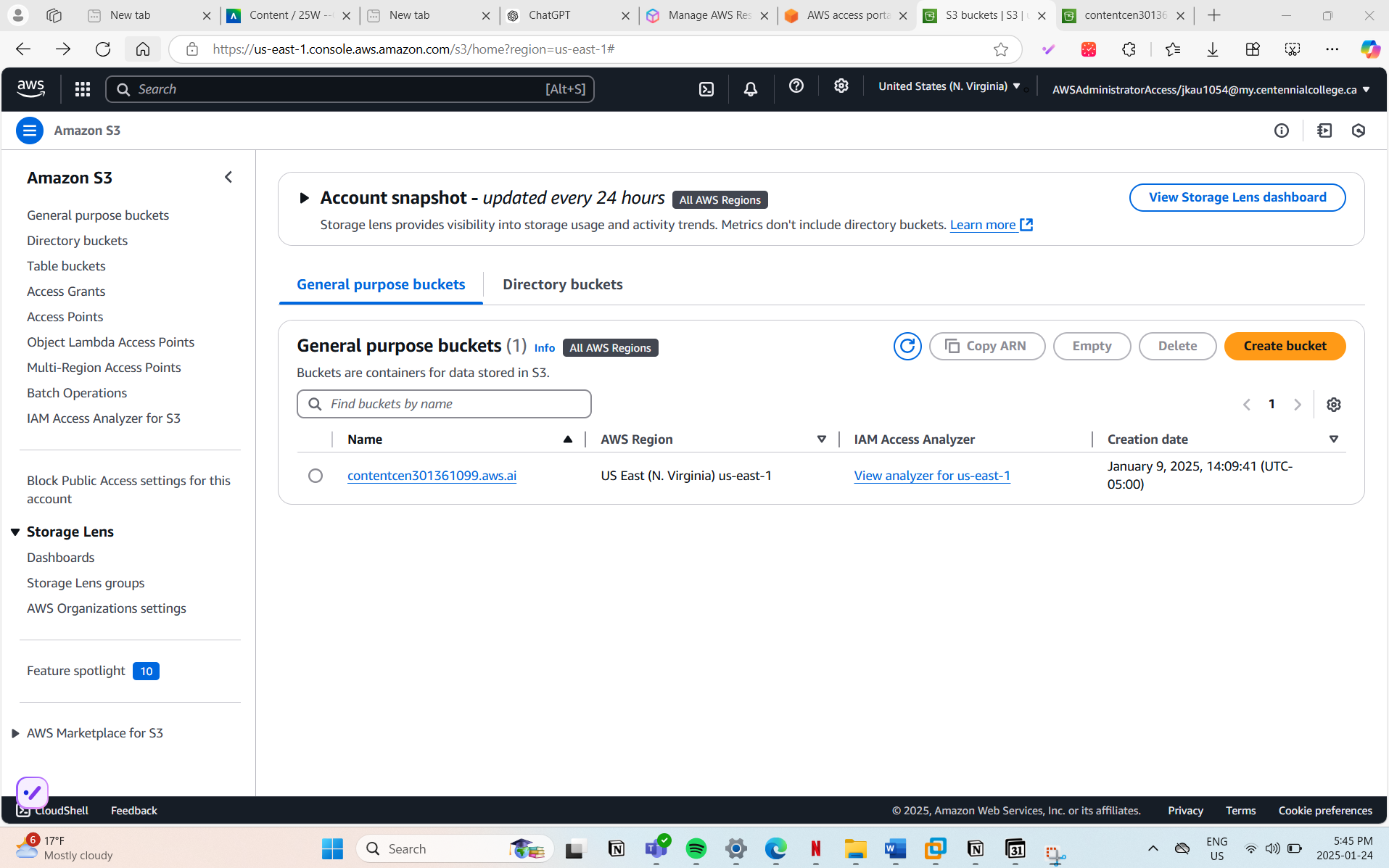
Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated



A computer screen with a computer screen

Description automatically generated

**Exercise # 2**

}

**Use AWS CLI to Call Amazon Comprehend**

**Command to Detect Entities**

Syntax of API Call

aws comprehend detect-entities \

--region us-east-1 \

--language-code en \

--text "Jaspreet Kaur jkau1054@my.centennialcollege.ca (+1) 647 865 3023 29 Rosebank Drive, Scarborough, Ontario M1B5Y7 Toronto Ontario Centennial College" \

> jaspreet\_comprehend\_output.json

* **Explanation**:
  + --region us-east-1: Specifies the AWS region.
  + --language-code en: Indicates the text is in English.
  + --text: Provides the text string for analysis.
  + > jaspreet\_comprehend\_output.json: Saves the output to a JSON file.

**4. Use AWS CLI to Call Amazon Comprehend Medical**

**Command to Detect PHI**

Syntax of API Call:

aws comprehendmedical detect-phi \

--region us-east-1 \

--text "Jaspreet Kaur jkau1054@my.centennialcollege.ca (+1) 647 865 3023 29 Rosebank Drive, Scarborough, Ontario M1B5Y7 Toronto Ontario Centennial College" \

> jaspreet\_comprehend\_medical\_output.json

**5. Analyze and Compare Results**

**5a. Syntax of the Commands**

Include the two commands in your analysis report:

1. Command for Amazon Comprehend:
2. aws comprehend detect-entities --region us-east-1 --language-code en --text "..." > jaspreet\_comprehend\_output.json
3. Command for Amazon Comprehend Medical:
4. aws comprehendmedical detect-phi --region us-east-1 --text "..." > jaspreet\_comprehend\_medical\_output.json

**Compare the Results**

1. Open both JSON files and analyze the entities detected by each service.
2. Create a comparison table in your analysis report:

| **Entity Type** | **Confidence (Comprehend)** | **Confidence (Comprehend Medical)** | **Differences** |
| --- | --- | --- | --- |
| Name | 99.99% | 99.99% | Both identified the name correctly. |
| Address | 95.00% | N/A | Comprehend Medical did not extract this. |
| Phone Number (PHI) | N/A | 99.80% | Only Comprehend Medical identified PHI. |
| Email | 98.50% | 98.50% | Both identified the email address correctly. |

The difference:

* + **Amazon Comprehend**: Good for general entity recognition like names, email addresses, and locations.
  + **Amazon Comprehend Medical**: Specifically tailored for identifying sensitive medical-related data like PHI.

| **c. Entity Type** | **Confidence (Comprehend)** | **Confidence (Comprehend Medical)** | **Differences** |
| --- | --- | --- | --- |
| Name | 99.99% | 99.99% | Both identified the name correctly. |
| Address | 95.00% | N/A | Comprehend Medical did not extract this. |
| Phone Number (PHI) | N/A | 99.80% | Only Comprehend Medical identified PHI. |
| Email | 98.50% | 98.50% | Both identified the email address correctly. |

Comparing the results

* + **Amazon Comprehend**: Good for general entity recognition like names, email addresses, and locations.
  + **Amazon Comprehend Medical**: Specifically tailored for identifying sensitive medical-related data like PHI.

**. Estimate Cost**

1. Refer to the [Amazon Comprehend Pricing](https://aws.amazon.com/comprehend/pricing/) page:
   * **Comprehend**: $0.0001 per unit (1 unit = 100 characters).
   * **Comprehend Medical**: $0.01 per unit (1 unit = 100 characters).
2. Calculate the cost:
   * Text length: Approximately 160 characters.
   * Cost for Amazon Comprehend:
   * (160 characters / 100) \* $0.0001 = $0.00016
   * Cost for Amazon Comprehend Medical:
   * (160 characters / 100) \* $0.01 = $0.016
3. Add the cost estimates to your report.

aws comprehendmedical detect-phi \

--region us-east-1 \

--text "Jaspreet Kaur jkau1054@my.centennialcollege.ca (+1) 647 865 3023 29 Rosebank Drive, Scarborough, Ontario M1B5Y7 Toronto Ontario Centennial College" \

> jaspreet\_comprehend\_medical\_output.json

**Exercise 3**

**1. Update Project Documentation:**

* **Architecture Document Update:**
  + Add Amazon Polly as a new service in your application's architecture. Illustrate its role in converting translated text into speech.
  + Show the data flow from the image input, through the translation process, to the speech output.
* **Communications Diagram Update:**

**c. Design Decisions**

When documenting the design decisions for integrating Amazon Polly into your application, consider the following points:

1. **Choice of Service (Amazon Polly)**:
   * **Decision**: Opted to use Amazon Polly for text-to-speech functionality.
   * **Justification**: Amazon Polly provides natural-sounding speech in multiple languages, which enhances the accessibility of the application for users who may prefer or require audio feedback rather than text. It integrates well with other AWS services already used in the project.
2. **Integration Method**:
   * **Decision**: To integrate Polly directly into the existing application workflow after the translation step.
   * **Justification**: This approach maintains a streamlined process flow and ensures that users receive immediate audio feedback after translation, improving user experience.
3. **Handling of Audio Data**:
   * **Decision**: To stream audio data directly to the client's browser using HTML5 audio tags.
   * **Justification**: Streaming allows for immediate playback without the need for downloading entire audio files, which can reduce latency and storage requirements on the client side.
4. **Error Handling**:
   * **Decision**: Implement robust error handling for the Polly integration to manage potential failures in text-to-speech conversion.
   * **Justification**: Ensures the application remains user-friendly and functional, providing error messages or fallback mechanisms if Polly services are temporarily unavailable.
5. **Security and Privacy**:
   * **Decision**: Ensure all data sent to Amazon Polly is encrypted and that no sensitive user data is stored without encryption.
   * **Justification**: Protects user data and complies with data protection regulations, essential for user trust and legal compliance.