Assignment

Problem Statement : You will receive a string as input, potentially a mixture of upper and lower case, numbers, special characters etc. The task is to determine if the string contains at least one of each letter of the alphabet. Return true if all are found and false if not. Write it as a RESTful web service (no authentication necessary) in any language/framework you choose and document the service. Please describe how you would deploy this application into AWS, including which AWS services you would use, and what deployment method or tools.

1. **Code Implementation**

   - Developed a RESTful Web Service :
   - We created a simple Flask application that checks if a given string contains all letters of the English alphabet.
   - The service listens for POST requests at the `/check-alphabet` endpoint, accepts a JSON payload, and returns a JSON response indicating whether all letters are present.

```python
from flask import Flask, request, jsonify

app = Flask(__name__)

def contains_all_letters(input_string):
    found_letters = set()
    for char in input_string.lower():
        if 'a' <= char <= 'z':
            found_letters.add(char)
        if len(found_letters) == 26:
            return True
    return False

@app.route('/check-alphabet', methods=['POST'])
def check_alphabet():
    data = request.json
    if 'input_string' not in data:
        return jsonify({'error': 'input_string key is required'}), 400
```

```
    input_string = data['input_string']
    result = contains_all_letters(input_string)

    return jsonify({'contains_all_letters': result})

if __name__ == '__main__':
    app.run(debug=True)
```

2. **Testing the Functionality**
    - Created various test cases to validate the function's accuracy, checking for different combinations of characters, including uppercase, lowercase, numbers, and special characters.

3. **Creating the Deployment Environment**
    - Installed Flask :
    - We installed Flask using pip to run the application:

    ```bash
    pip install Flask
    ```

    - Prepared the Application for Deployment :
    - Created a virtual environment to manage dependencies.
    - Listed the required packages in a `requirements.txt` file.

4. **Setting Up AWS for Deployment**
    - AWS Account and CLI Configuration :
        - Set up an AWS account and configured the AWS CLI on the local machine using:

        ```bash
        aws configure
        ```

5. **Preparing for Elastic Beanstalk Deployment**
    - Elastic Beanstalk Configuration :
        - Created a directory named `.ebextensions` and added a configuration file (`python.config`) to specify the WSGI path:

        ```yaml
        ```

```
option_settings:
  aws:elasticbeanstalk:container:python:
    WSGIPath: app.py
```

6. Deploying to AWS Elastic Beanstalk

Initialize Elastic Beanstalk :

- Navigated to the project directory and initialized Elastic Beanstalk:

```bash
eb init -p python-3.8 Restful-api.py
```

- Create an Environment :
  - Created an Elastic Beanstalk environment:

```bash
eb create Restful-api
```

- Deploy the Application :
  - Deployed the application to the newly created environment:

```bash
eb deploy
```

- Accessing the Application :
  - Opened the deployed application in a web browser using:

```bash
eb open
```

8. Updating the Application

- For subsequent updates, changes were made to the code and redeployed using:

```bash
eb deploy
```