Software Developer Assignment Brief

Position: AWS Security Developer

Duration: Confirm back when you can complete

Objective

This assignment assesses your ability to:
- Design and implement backend APIs using Python and Flask
- Work with AWS CloudFormation and Subnet resources using Boto3
- Manipulate and transform JSON-based information
- Apply best practices in API development
- Demonstrate your ability to learn and apply new skills quickly in a realistic project scenario

Skills Being Evaluated

- Python & Flask (RESTful APIs)
- JSON parsing and transformation
- Boto3 usage (CloudFormation and networking)
- Understanding of AWS resources (Public/Private Subnets)

Pre-requisites

Please ensure the following before starting:
1. Create a Public Subnet in your AWS Free Tier account.
2. You may do this via the AWS Console, but make sure to import the subnet and related resources into a CloudFormation stack (use 'import resources to stack' if needed).

Assignment Tasks

1. Get CloudFormation Template – REST API

- Implement a GET endpoint in Flask that:
- Accepts the CloudFormation stack name as a parameter
- Uses Boto3 to retrieve the stack's template (JSON format) - Returns the template in the response

2. Update Subnet from Public to Private – REST API

- Implement a PUT endpoint in Flask that:
  - Modifies the template JSON to convert the subnet from Public to Private
- Returns the modified template as a response

3. Create and Monitor CloudFormation ChangeSet – REST API

- Implement a POST endpoint in Flask that:
- Accepts stack name and template JSON
- Creates a ChangeSet using Boto3 for the given stack
- Waits (polls) for completion of ChangeSet creation - Returns:

- Success message with ChangeSet ID and status - Or, error message if creation fails

Expected Output

- Functional Flask app with 3 endpoints
- Proper error handling and status codes
- Clean, readable, and modular code

Evaluation Criteria

- - Correctness and completeness of each API
- - Quality of code, error handling, and modularity
- - Communication via README and/or docstrings

Submission

Please share:
- GitHub repo link
- Any screenshots or logs as proof of successful API interaction with AWS
- README with:
    - Setup instructions
    - Brief explanation of each API
    - Example curl or Postman calls

Key Inputs:

1) Public vs Private subnet:

the change from public to private covers different aspects. Please cover the following attributes in your program,

In AWS, a public subnet is identified by having a route to an Internet Gateway (IGW) in its route table, while a private subnet lacks this route. Essentially, if a subnet's route table contains a route to an IGW (0.0.0.0/0 or ::/0), it's considered public, allowing resources within to communicate directly with the internet. Conversely, if the route table doesn't have a route to the IGW, it's a private subnet.

Key Differences:

Public Subnet:
Resources can directly access the internet and other AWS services, and resources on the internet can access them.
Private Subnet:
Resources cannot directly access the internet. To access the internet, they typically use a NAT Gateway (Network Address Translation) located in a public subnet.

Please follow this link to learn what needs to be done to create Public route https://000058.awsstudygroup.com/2-prerequiste/2.1-createec2/2.1.2-createpublicsubnet/

2. Convert yaml template string into JSON format, not just wrap it in JSON body in GET endpoint

3. Create changeset with completely JSON template (including the original portion returned in yaml) in PUT endpoint