**Description**

**Overview:**

**Develop a system that automatically manages the lifecycle of a web application hosted on  EC2 instances, monitors its health, and reacts to changes in traffic by scaling resources.  Furthermore, administrators receive notifications regarding the infrastructure's health and scaling events.**

**Detailed Breakdown:**

**1. Web Application Deployment:**

 - Use `boto3` to:

 - Create an S3 bucket to store your web application's static files.

 - Launch an EC2 instance and configure it as a web server (e.g., Apache, Nginx).  - Deploy the web application onto the EC2 instance.

**2. Load Balancing with ELB:**

 - Deploy an Application Load Balancer (ALB) using `boto3`.

 - Register the EC2 instance(s) with the ALB.

**3. Auto Scaling Group (ASG) Configuration:**

- Using `boto3`, create an ASG with the deployed EC2 instance as a template.

 - Configure scaling policies to scale in/out based on metrics like CPU utilization or network traffic.

**4. SNS Notifications:**

 - Set up different SNS topics for different alerts (e.g., health issues, scaling events, high traffic).

 - Integrate SNS with Lambda so that administrators receive SMS or email notifications.

**5. Infrastructure Automation:**

 - Create a single script using `boto3` that:

 - Deploys the entire infrastructure.

 - Updates any component as required.

 - Tears down everything when the application is no longer needed.

**6. Optional Enhancement – Dynamic Content Handling:**

- Store user-generated content or uploads on S3.

 - When a user uploads content to the web application, it gets temporarily stored on the  EC2 instance. A background process (or another Lambda function) can move this to the S3  bucket and update the application's database to point to the content's new location on S3.

**Objectives:**

- Gain a comprehensive understanding of key AWS services and their integration. - Understand the lifecycle of a dynamic web application and its infrastructure.

- Learn how to automate infrastructure deployment and management tasks using boto3. - Experience with real-time monitoring and alerting systems.

**Submission Instructions:**

To submit your assignment, please follow these guidelines:

- Ensure that your assignment is fully completed.

- Push your code to a GitHub repository.

- Share the repository link by including it in a text, Word, or PDF file format.

Submit the file/text containing the repository link via Vlearn.

**Read Less**

keyboard\_arrow\_up

**Start Date**

6 Sep, 2024, 12:00 AM

**Due Date**

26 Sep, 2024, 11:59 PM