**Intelligent EBS Volume Optimization Using Lambda, CloudWatch, SNS, DynamoDB & Step Functions**

**Project Objective:**

This project's objective was to implement a serverless automation pipeline to intelligently monitor Amazon EBS volumes, identify legacy gp2 volumes for conversion to the modern gp3 type, and execute the change. The resulting solution provides a full audit trail, real-time alerting, and scheduled execution to ensure ongoing cost and performance optimization.

### 1. Solution Architecture & Component Roles

The architecture is built on an event-driven, serverless model to ensure scalability, resilience, and cost-efficiency. The following AWS services were used:

* **Amazon EventBridge (CloudWatch Events):** Acts as the cron scheduler for the entire workflow. It is configured with a daily schedule to trigger the Step Functions state machine, initiating the optimization process without manual intervention.
* **AWS Step Functions:** Functions as the central orchestrator, managing the sequence of operations, data flow, and error handling. Its visual workflow and use of a Map state enable reliable, concurrent processing of multiple volumes.
* **AWS Lambda (EBS\_FilterAndLog):** This function contains the initial business logic. It scans all EBS volumes in the account, filtering for those that are type gp2 and have a specific tag (AutoConvert: true) to ensure only designated volumes are modified. It then logs the details of these eligible volumes to DynamoDB for auditing purposes.
* **Amazon DynamoDB:** Provides a persistent, serverless NoSQL database used as an audit trail. The EBSOptimizationLog table stores a record of every volume that was identified for conversion, including its ID, size, region, and a timestamp. This creates an immutable history of all automated actions.
* **AWS Lambda (EBS\_ModifyVolume):** This function performs the core action. It receives the details of a single volume from the Step Functions Map state and executes the ec2:ModifyVolume API call to convert its type from gp2 to gp3.
* **Amazon SNS (Simple Notification Service):** Facilitates real-time notifications. Once a volume modification is initiated, a message is published to an SNS topic. This message, containing the volume ID and the success or error status, is then sent to subscribed endpoints (e.g., an administrator's email) for immediate awareness.
* **AWS IAM (Identity and Access Management):** Provides the security foundation for the entire solution. Fine-grained IAM roles and policies were created to grant each component the principle of least privilege, ensuring they only have the exact permissions needed to perform their tasks.

### 2. Security Best Practices Implemented

Security was a core consideration throughout the project, addressed primarily through IAM:

* **Principle of Least Privilege:**
  + The **Lambda Execution Role** was granted a custom policy allowing only necessary actions: ec2:DescribeVolumes, ec2:ModifyVolume, dynamodb:PutItem on the specific EBSOptimizationLog table, and permissions to write logs to CloudWatch. It cannot perform any other actions.
  + The **Step Functions Execution Role** was granted permissions only to lambda:Invoke the specific functions in the workflow and sns:Publish to the designated topic.
* **Separation of Concerns:** Using two separate Lambda functions isolates the "read/scan" logic from the "write/modify" logic. This is a security best practice that limits the potential impact if one component were to be compromised.
* **Resource-Based Policies & Tagging:** The automation relies on a specific tag (AutoConvert: true) to operate. This acts as a manual safeguard, ensuring that the pipeline will never modify critical volumes that have not been explicitly approved for conversion by an engineer.

### 3. Real-World Challenges Simulated and Handled

This project effectively simulates and solves several common challenges faced by organizations operating in the cloud:

* **Challenge: Cloud Cost Creep & Inefficiency**
  + **How Handled:** The primary goal is to combat the unnecessary expense of running on older-generation infrastructure. By automating the upgrade from gp2 to the up-to-20%-cheaper gp3 volumes, the solution directly reduces the monthly AWS bill without requiring ongoing manual effort. This is a core practice of FinOps (Financial Operations).
* **Challenge: Manual Toil and Human Error**
  + **How Handled:** The serverless pipeline automates the tedious, slow, and error-prone manual process of scanning for and converting hundreds of volumes. This increases speed and reliability while freeing up valuable DevOps and Cloud Engineering time to focus on value-adding tasks.
* **Challenge: Lack of Auditing and Governance**
  + **How Handled:** To meet governance and regulatory requirements, the integration with DynamoDB provides a non-repudiable audit trail for every change made to the infrastructure. This log answers "who, what, and when" for every automated modification, while SNS notifications provide immediate visibility.
* **Challenge: Scalability**
  + **How Handled:** The solution is inherently scalable. Whether there are 10 volumes to convert or 10,000, the serverless architecture (Lambda and Step Functions) will scale automatically to handle the load without any changes to the underlying infrastructure. The use of a Step Functions Map state is key to this, as it processes each volume as a parallel, independent task.