EduVault: Secure Serverless Assignment Submission & Grading Portal

Abstract

Educational institutions often struggle with secure, scalable methods for students to submit large digital assets. Traditional local servers often crash during peak "deadline" hours. **EduVault** is a cloud-native submission portal that ensures high availability and automated workflow management.

The system uses an **Amazon EC2** instance to host a student-facing frontend where users can log in and upload assignments. These files are moved directly to **Amazon S3**, which acts as a durable, encrypted repository. Upon upload, an S3 Event Notification triggers **AWS Lambda** to categorize the file and update a record in **Amazon DynamoDB**, marking the student's submission as "Received." To close the feedback loop, **Amazon SNS** sends a confirmation receipt to the student. This architecture ensures that even if thousands of students upload simultaneously, the serverless backend scales automatically to handle the metadata and notification processing.

1. Clear Goals & Objectives

- **Scalable Asset Handling:** Decouple file storage from the web server to ensure the website remains responsive regardless of file sizes.
- **Automated Confirmation:** Provide students with immediate, verifiable proof of submission through automated messaging.
- **Centralized Record Keeping:** Maintain a real-time ledger of submissions that instructors can query via a NoSQL database.
- **Data Integrity:** Use S3 versioning and IAM policies to ensure students can only upload to their own folders and cannot delete submitted work.

2. Architectural Component Breakdown

Service	Role in EduVault	Why it matters
Amazon S3	The Vault	Provides 99.999999999% durability for student files, far exceeding local hardware.
AWS Lambda	The Registrar	Automatically processes file metadata (size, type, timestamp) as soon as a file hits S3.

Amazon EC2	The Frontend	Hosts the user authentication and the "Upload" interface for the students.
Amazon DynamoDB	The Gradebook Index	Tracks student IDs, file links, and submission times for easy teacher access.
Amazon SNS	The Receipt Issuer	Sends a "Submission Successful" email containing a unique tracking ID to the student.

3. Implementation Roadmap (What to do)

- **Step 1: Frontend Deployment (EC2):** Set up a t2.micro instance with a simple PHP or Python upload form.
- **Step 2: Storage Policy (S3):** Create a bucket and enable "Event Notifications" to trigger a Lambda function on s3:ObjectCreated:*.
- **Step 3: Metadata Automation (Lambda):** Code the Lambda to extract the filename and student ID from the S3 metadata and write it to DynamoDB.
- **Step 4: Closing the Loop (SNS):** Subscribe student email addresses to an SNS topic. Configure Lambda to publish a message to this topic upon successful DB write.

4. Impact: The Cloud Value Proposition

EduVault demonstrates **Operational Excellence** by automating the administrative task of confirming receipts. By leveraging the **Shared Responsibility Model**, the institution relies on AWS for the security of the stored files (S3) and the database (DynamoDB), focusing solely on the student experience.