**Architecture Flow**

1. Users sign up or log in via API Gateway → Lambda (signup/login) → DynamoDB (Users Table).
2. The scraper Lambda fetches news every hour via Step Functions → feeds into Kinesis.
3. The consumer Lambda reads from Kinesis → processes and stores results in DynamoDB (NewsArticles Table).
4. API Gateway exposes a GET /getnews endpoint → triggers getNewsAPI Lambda → fetches from DynamoDB.
5. CloudWatch logs and monitors Lambda/Step Function performance.
6. SNS sends email alerts if the Step Function fails.

**Services and Purpose**

**1. AWS Lambda (Serverless compute)**

* news\_scraper: Scrapes news articles. Scheduled every hour via Step Function + EventBridge.
* consumer: Consumes Kinesis stream records and stores processed news into DynamoDB.
* get\_news\_api: Serves GET /getnews request via API Gateway.
* signup & login: Handle user authentication. Interact with the Users table in DynamoDB.

**2. API Gateway (v2) (REST endpoints)**

* Creates a single API with 3 routes:
  + GET /getnews → get\_news\_api Lambda
  + POST /signup → signup Lambda
  + POST /login → login Lambda
* Enables CORS for frontend integration.

**3. DynamoDB (NoSQL database)**

* Users table:
  + Stores user details with id as the primary key.
* NewsArticles table:
  + Stores articles scraped from the web.
  + article\_id is the partition key, publishedAt is the sort key.

**4. Kinesis Stream (Real-time ingestion)**

* Stream named news-stream.
* Used to stream news from the scraper → consumer Lambda.
* consumer Lambda is triggered by this stream using an event\_source\_mapping.

**5. Step Functions (Orchestration with retries)**

* A single-state state machine that:
  + Invokes the news\_scraper Lambda.
  + Retries up to 2 times with a 5-minute delay if it fails.
* Scheduled via EventBridge every 1 hour.

**6. EventBridge (Scheduling)**

* Triggers the Step Function NewsScraperStateMachine every hour using a rate(1 hour) expression. (I might change the time so before writing it in the report just ask me)

1. **CloudWatch (Monitoring & Logs)**

* Logs Lambda execution per function.
* Logs Step Function execution with a dedicated log group.
* Monitors ExecutionsFailed metric for the Step Function.

1. **SNS (Alerting)**

* Topic step-function-failure-alerts created.
* Email alert sent to suryanshsrivastava22@gmail.com if Step Function fails (after retries).

1. **Secrets Manager (🔐 Secure API key storage)**

* Stores OpenAI API key under secret name apikeys.
* Allows secure referencing of keys in Lambda if needed (though not explicitly shown in environment blocks yet).

