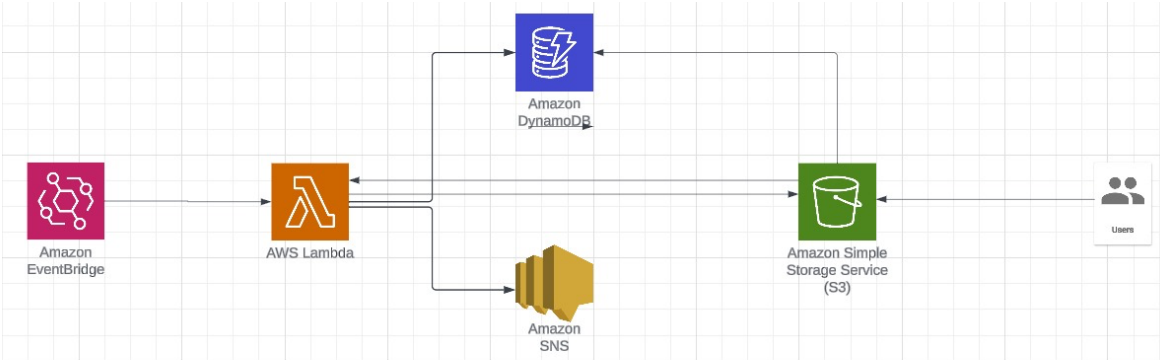
Automated Price Drop Notification for iPad on Amazon.com

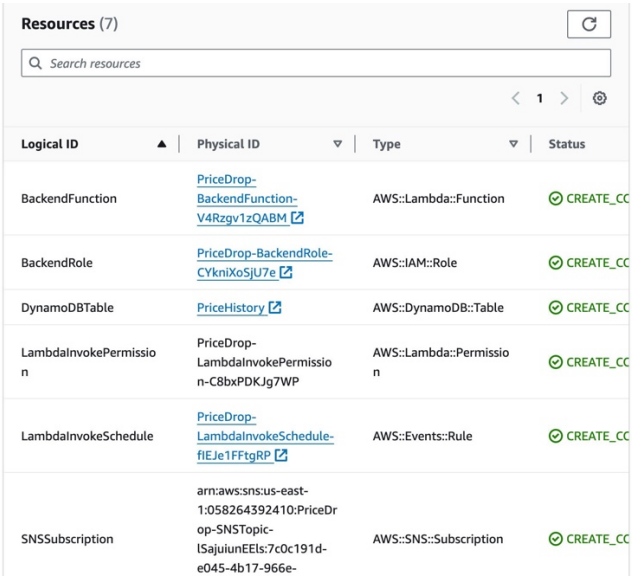
Step-by-Step Implementation Report



System Architecture Overview

1. Setting up AWS Infrastructure

Utilized AWS CloudFormation templates to provision necessary services automatically:
- Created an S3 bucket to store Lambda function code.
- Set up DynamoDB table for storing scraped data.
- Configured AWS SNS for email notifications.
- Scheduled Lambda invocations using CloudWatch Events.



Resources (7)

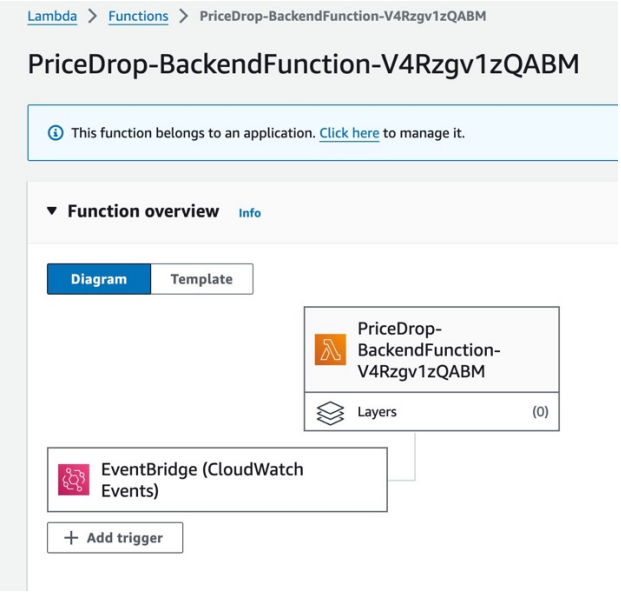
Search resources

1

Logical ID	Physical ID	Type	Status
BackendFunction	PriceDrop-BackendFunction-V4Rzgv1zQABM	AWS::Lambda::Function	CREATE_CC
BackendRole	PriceDrop-BackendRole-CYkniXoSjU7e	AWS::IAM::Role	CREATE_CC
DynamoDBTable	PriceHistory	AWS::DynamoDB::Table	CREATE_CC
LambdaInvokePermission	PriceDrop-LambdaInvokePermission-C8bxPDKJg7WP	AWS::Lambda::Permission	CREATE_CC
LambdaInvokeSchedule	PriceDrop-LambdaInvokeSchedule-fIEJe1FFtgRP	AWS::Events::Rule	CREATE_CC
SNSSubscription	arn:aws:sns:us-east-1:058264392410:PriceDrop-SNSTopic-lSajuiunEEls:7c0c191d-e045-4b17-966e-	AWS::SNS::Subscription	CREATE_CC

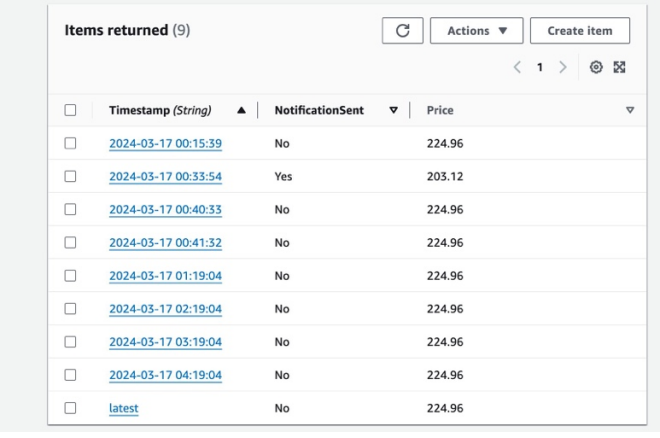
2. Web Scraping and Backend Logic

- Developed a Python script for web scraping Amazon.com to fetch iPad prices.
- Integrated the script into AWS Lambda function for periodic execution.
- Implemented backend logic in Python to handle data retrieval from DynamoDB, price comparison, and notification triggering.



3. DynamoDB Setup

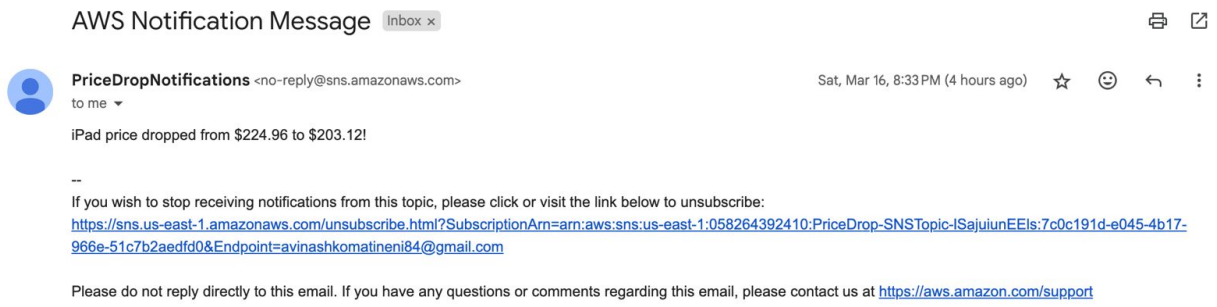
- Designed DynamoDB table schema to efficiently store scraped iPad prices and timestamps.
- Enabled seamless integration with Lambda functions for data storage and retrieval.



4. AWS SNS Configuration

- Configured AWS SNS to send email notifications upon detecting a price drop.

- Linked Lambda functions to publish notifications to SNS when a price drop occurred.

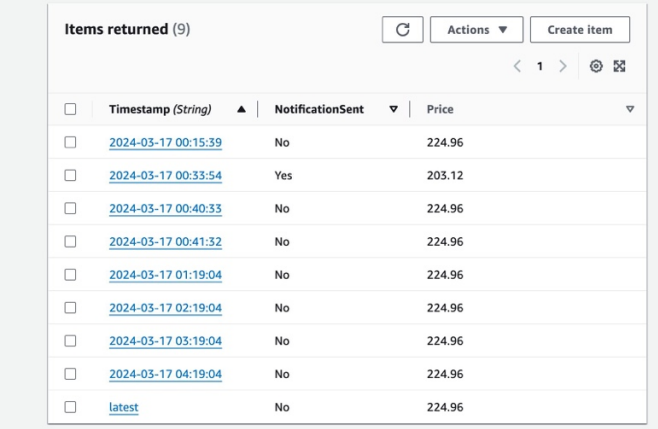


5. CloudWatch Events for Scheduled Invocations

- Scheduled regular invocations of the scraper Lambda function using CloudWatch Events.
- Ensured timely and periodic scraping of Amazon.com for up-to-date price information.

6. Testing and Deployment

- Conducted comprehensive testing to ensure the functionality, reliability, and scalability of the system.
- Deployed Lambda functions and associated AWS services using CloudFormation templates for automated deployment.



Logging info

To Configure and Use the Application:

To set up and use the application:
- Deploy the `cloudFormation.yaml` script using AWS CloudFormation to automatically create all the necessary services.
- Upload `lambda_function.zip` to an S3 bucket.

- EventBridge (CloudWatch Events) will automatically trigger the Lambda function at predefined intervals.

That's it! The application will now periodically scrape Amazon.com for iPad prices and send notifications when a price drop is detected.

Additional Features Implemented: Real-World Deployment on AWS

In addition to the core functionality outlined in the project requirements, the implementation of the automated price drop notification system for iPads on Amazon.com was extended to include deployment on AWS in a real-world scenario. This involved setting up the entire project infrastructure on AWS cloud services, ensuring scalability, reliability, and ease of maintenance.

Challenges Faced

Integrating multiple AWS services required careful configuration and testing to ensure seamless communication between components. I thoroughly tested the interactions between Lambda functions, DynamoDB, SNS, and CloudWatch Events to ensure reliability.

Conclusion

The implementation of the Automated Price Drop Notification System for iPad on Amazon.com involved setting up AWS infrastructure using CloudFormation templates, developing web scraping functionality and backend logic in Python, configuring DynamoDB for data storage, integrating AWS SNS for email notifications, scheduling Lambda invocations with CloudWatch Events, and conducting testing and deployment. Through this step-by-step implementation, users can stay informed about iPad price changes on Amazon.com and make informed purchasing decisions effectively.