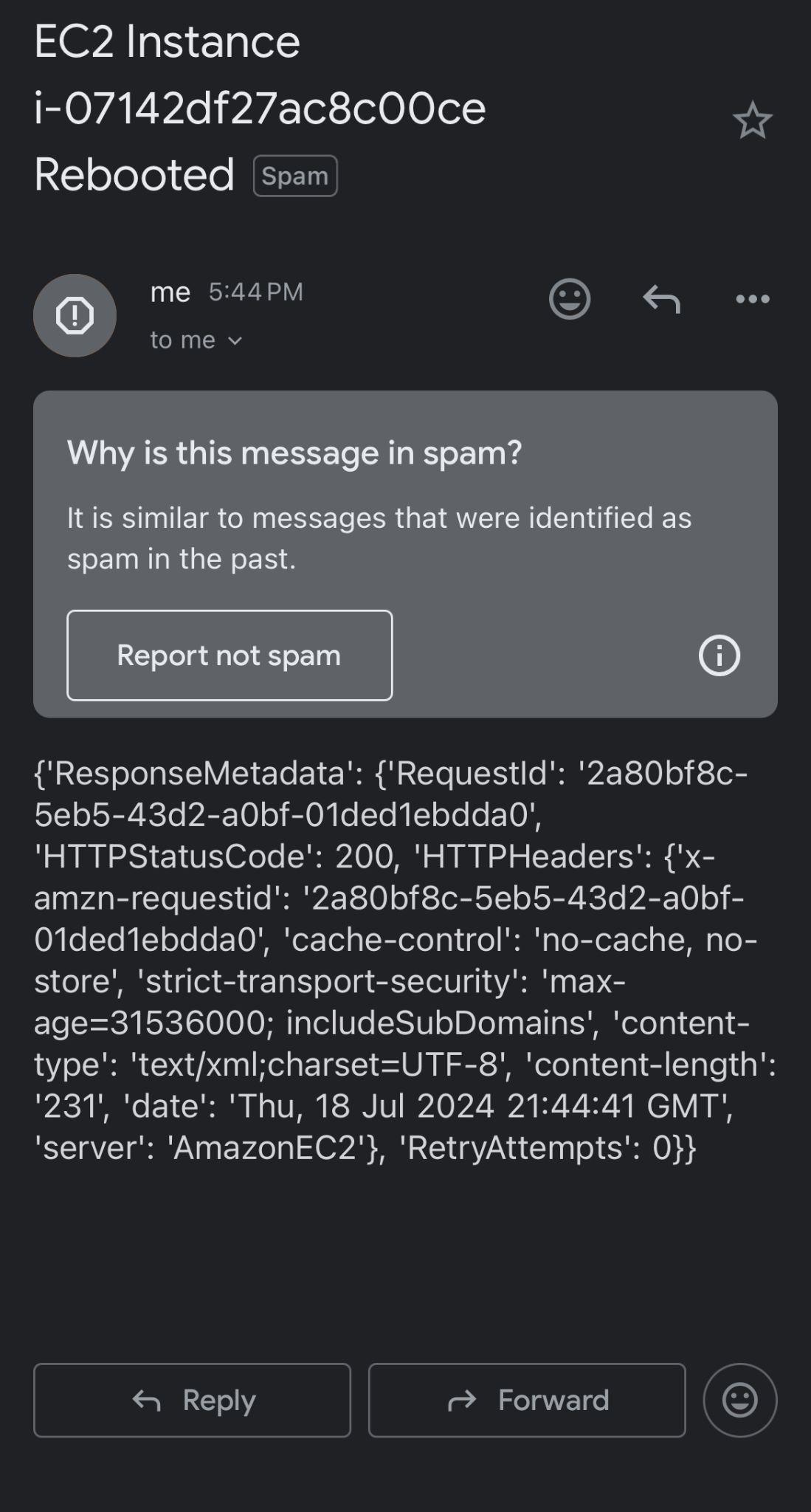
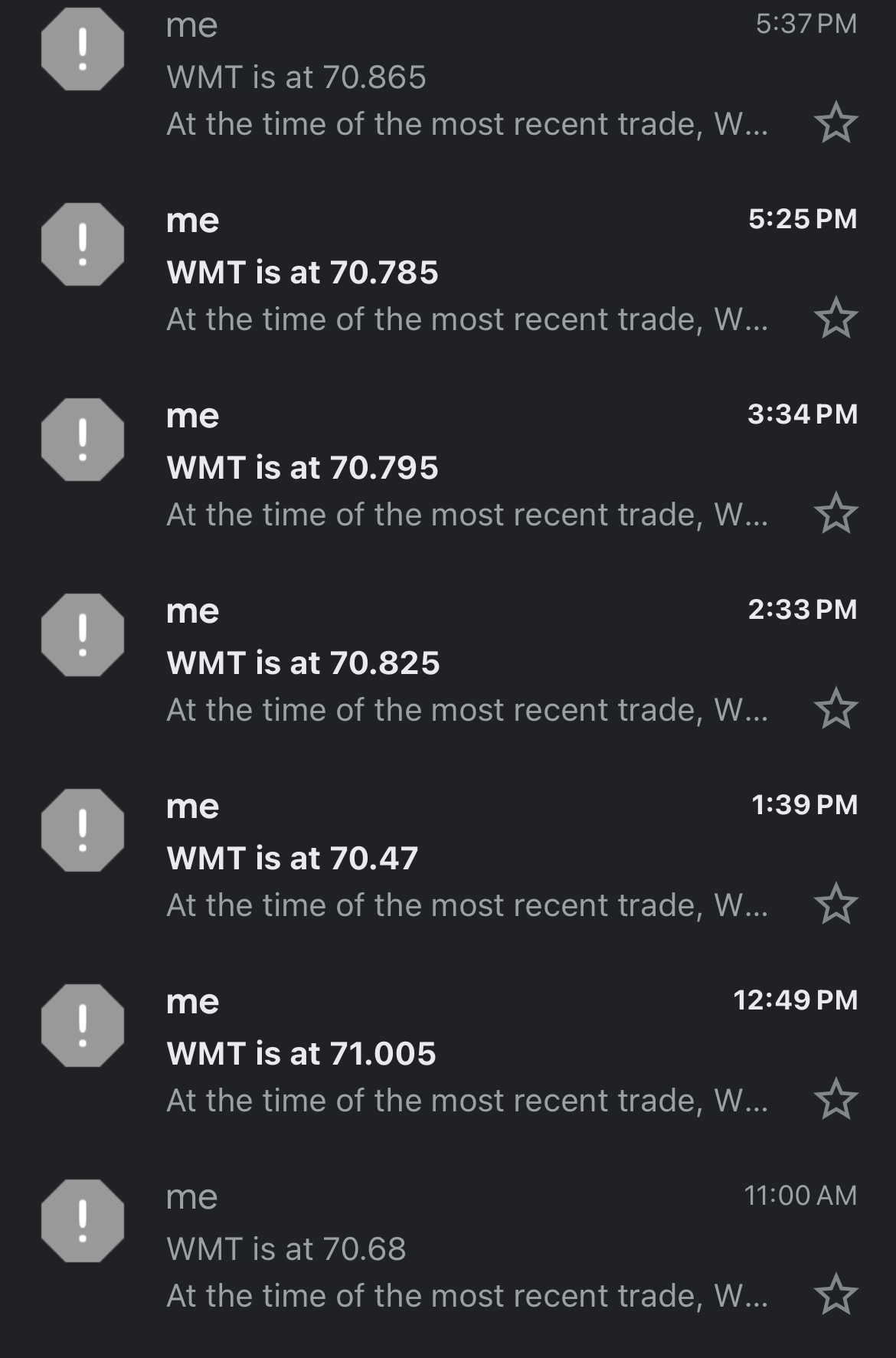
Team 1: Andrea Ayala, Miguel Garcia, Patrick Grodach, Patty Kingery, Bryce Williams

**Methodology:**

AWS offers many services. In this lab, we are demonstrating that we do not need to create a server to run code in AWS. Our first goal was to build a Lambda function to check a stock price for a given ticker every hour, and send you an email with the updated price. Our second goal was to set up another Lambda function to monitor a website that we had previously set up, and reboot the server if that web server is not working.

For the first goal, we needed to set up an account at Tingo from where to get the stock price and to get an API key token to communicate between our Lambda function and Tingo. Then we set up a service policy in our AWS environment to enable the ability to send emails. We had to register our email account to become a verified identity allowed to send emails that will work with our Lambda function. Then we created the Lambda function with code provided to us and we amended the code with our specific stock ticker, email address and API token so the function can fetch the stock price. We tested the Lambda function and then setup timing to trigger when to execute the Lambda function every hour for when the market is open. Each hour, we received an email with the stock price.

Below are the screenshots of the stockChecker function emails showing times they were received and the subjects. You can clearly see that the price of the stock was updated with each email.



For the second goal of this lab, we built a Lambda function that monitors a website, and if the website is not accessible, the policy has full access to reboot the web server. This involves choosing a unique word that is on your website. The function will check to see if that key word is present, and if it is not, the function will reboot the server. We set the reboot capability because that is often a quick resolution for when sites are down. We also retained the email policy so that if the site is down and the server gets rebooted, there is an email sent to the person. We tested that canary function by stopping our web server manually in AWS. Below is a screen shot of the email:

**Analysis:**

1. **You created a Lambda function for your organization that processes incoming data. Each invocation requires 475ms of computing time using 128MB of memory and 512 MB of storage. Your data stream will invoke this function 50 million times per day. Assuming that your account is not eligible for the AWS Free Tier, what will be the monthly cost for a 30-day month in the Northern Virginia region? Show your work.**

Lambda Function Pricing (not eligible for AWS Free Tier) for a 30-day Month in N. Virginia Region:

Each invocation (computing time) - 475ms

Memory – 128 MB - $0.0000000021 per ms

Function Invoked frequency – 50 million/day

Requests: $0.20 per 1M requests

Duration: $0.0000166667 for every GB-second (for the first 6 Billion of GB-second)

Total requests a month - 50M \* 30(days) = 1,500,000,000

Total computing time = 475ms \* 1,500,000,000 \* 0.0001(ms/s) = 712,500,000 seconds

128MB = 0.125GB (1GB = 1024MB)

0.125 GB \* 712,500,000 seconds = 89,062,500 GB-second

Compute Charge = 89,062,500 GB-second \* $0.0000166667 = $1,484.3779

Requests Cost: 1,500 \* $0.20 = $300

Storage Cost = $0

**Total Cost:**

Compute Cost + Requests Cost

= $1,484.38 + $300

**= $1,784.38**

1. **Think about a company that you’re familiar with, preferably one where you have worked or have other internal knowledge. Describe the company’s work briefly and then provide three examples of how a similar company could use serverless computing and why serverless is a good choice for that type of scenario.**

For this question, we will analyze a public accounting firm and the potential benefits from serverless computing. The primary operational activity that we will be analyzing is the preparation of tax returns during the busy months of March and September. Information is provided by clients regarding a business’s activity during the prior year, and the firm will calculate taxable income and report the activity in a standardized manner on a US Income tax return that is due to the federal government.

One potential benefit relates to the opportunity to scale compute resources, particularly given the volatile and seasonal nature of the workload of a public accounting firm. Lambda can automatically scale during the busy months as workloads and data processing requirements increase, allowing the firm to handle varying volumes of data processing without devoting time to server maintenance.

Another use case relates to potential alerts and notifications. Lambda functions that monitor impending deadlines and provide key alerts to clients at specified intervals ahead of these deadlines can help ensure the right information is getting in the door at the right time to ensure a quality deliverable. For example, Lambda functions can be set up to remind clients to submit relevant information x days before a statutory deadline.

Lastly, Lambda functions can be used for client data ingestion/transformation purposes. Key data cleaning tasks can be performed upon receipt of certain client data so that information can be processed quickly by client service teams and delivered to the appropriate recipients in an automated manner.

1. **Use ChatGPT (https://chat.openai.com/) to comment on both the canary function and the stock picker function. Copy and paste the ChatGPT-generated code comments into your response to this prompt, and describe how the comments affected your understanding of the code. How could you use a tool like ChatGPT in the workforce?**

ChatGPT-generated comments for canary function:

# Consider implementing retries for HTTP requests to handle intermittent connection issues.

# Ensure permissions for EC2 operations are restricted to necessary actions only.

# Utilize CloudWatch metrics for monitoring Lambda function performance and execution frequency.

ChatGPT's recommendations significantly specify understanding and efficiency in serverless application development. By suggesting retries for HTTP requests, ChatGPT ensures reliability in checking website statuses amidst potential connection issues. Furthermore, ChatGPT monitors Lambda function performance, offering insights for optimizing execution. Adopting ChatGPT for code comments in professional settings accelerates the implementation of best practices and optimizations, fostering the creation of robust, secure, and efficient serverless applications. This approach promotes proactive maintenance and adherence to business standards.

ChatGPT-generated comments for stock picker function:

# Ensure proper error handling in case the API call fails.

# Consider adding logging to track historical data and debug potential issues.

# The use of boto3.client('ses') for email sending is appropriate here.

The impact of ChatGPT's code comments helps error handling to ensure validity, enables logging for performance analysis and debugging, and validates the use of AWS services like boto3.client('ses') for effective email sending. In the workforce, leveraging ChatGPT for code reviews elevates code quality by validating practices and suggesting optimizations for adherence to best practices. Moreover, it helps continuous learning offering specific insights that increase understanding and proficiency in serverless computing and AWS service integration.