Real-Time Inventory Optimization Using AWS Lambda and Amazon Kinesis

MSc Research Project
Cloud Computing

Abhishek Yadav
Student ID: 23153423

School of Computing
National College of Ireland

Supervisor: Shivani Jaswal

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Abhishek Yadav		
Student ID:	23153423		
Programme:	Cloud Computing	**Year:**	2024
Module:	MSc Research Project		
Supervisor:	Dr. Shivani Jaswal		
Submission Due Date:	12/12/2024		
Project Title:	Real-Time Inventory Optimization Using AWS Lambda and Amazon Kinesis		
Word Count:	……… **Page Count 20**		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Abhishek Yadav

Date: 11/12/2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Real-Time Inventory Optimization Using AWS Lambda and Amazon Kinesis

Abhishek Yadav
x23154423
Research in Computing
National College of Ireland

Abstract

In the needs of discussing the importance of developing cost-effective, scalable, and fast systems that handle variation of data in real-time and across various inventory environments. A comprehensive analysis using a variety of AWS cloud services was used, to realize the significance of inventory optimization, whereby Amazon Kinesis was used to stream data like stock information in real-time, AWS Lambda was leveraged to compute the event-driven application in a serverless environment, and finally Amazon SageMaker to deploy and train the machine learning model, which was found to be of paramount significance to realize the stock optimization along with Amazon S3 and DynamoDB for seamless data management and access.This research provides a detailed monthly spend analysis across various AWS services to evaluate cost efficiency, scalability, and resource utilization. This research incorporated AWS CloudWatch, which demonstrates graphical views of resource usage helping decision-makers to make proactive choices for cost and performance improvement. Moreover, this work notes practical implementations and solutions for challenges such as over-provisioning cabin, idle time of services such as SageMaker and Kinesis.The results demonstrate the benefits of using cloud services for real-time inventory management delivering the dynamic needs for data processing without sacrificing financial efficiency. The outcomes of this research satisfy the objectives laid out for them and open avenues for future evolution in the fields of predictive analytics, automation, and wider integration of datasets for effective inventory optimization across industries.

***Keywords*— Inventory, AWS Lambda, Kinesis, SageMaker, CloudWatch, DynamoDB, S3 Bucket, QuickSight.**

1 Introduction

The cloud has witnessed rapid development over the last decade bringing tremendous changes in various businesses especially regarding inventory management. Inventory systems that are usually narrowly supported through manual data processing and periodic batch updates cannot possibly keep pace with the increased complexity of modern supply chains. Therefore, organizations need solutions for such competitive and fast-paced markets. The solution should be effective scalable and real-time to manage the inventories and demands of customers properly. In this work the cloud-based approach tries to handle some of the issues with the optimal technologies like AWS Lambda, Amazon Kinesis along with machine learning models for real-time optimization in inventory management.

Inventory optimization is one of the most important features of real-time data management which has information on how to maintain the optimum levels of stock to satisfy demand without overstocking the costs or loss of sales due to stockout. Traditional inventory management employed models that were static in nature and much manual intervention was involved; hence, it left room for inefficiencies, inaccuracies, and higher operational costs most of the time. It becomes that much more complex to manage inventory effectively at a greater number of locations with fluctuating demand patterns and seasonal variations especially in the case of global supply chains.

These new avenues have opened the way to dynamic management of inventories due to the fact that cloud computing can handle large amounts of data volume and scale up or down resources as required. Particularly AWS offers a complete suite of tools focused on real-time data processing and predictive analytics. AWS Lambda is a serverless computing service that will enable organizations to run code without managing servers reducing operational overheads greatly. In the meantime Amazon Kinesis will allow for real-time ingestion of data for analysis and the ability for businesses to act upon changed inventory positions in real time. By integrating these services this study aims to design an efficient real-time inventory optimization system that can adapt to the varying needs of modern supply chains. Apart from reducing carrying costs and minimizing stockouts bottom line factor-inventory optimization plays a very important role in customer satisfaction. It will ensure timely and accurate inventory levels to make sure products are there when and where they are needed which is of critical importance in retail and electronic commerce todayHuerta-Soto et al. (2023).

However, real-time inventory systems pose technical challenges in terms of data stream multiplicity, speed, and volume of data to be processed and the need for accurate and actionable predictions. AWS specifically Lambda and Kinesis come to provide strong solutions through seamless data processing, storage and analysis in a scalable and flexible cloud environment. Real-time inventory management cannot be emphasized enough especially in retailing manufacturing, and logistics industries where the level of stock directly influences operational efficiency and profitability. Demand forecasting and stock level adjustments in real time create a great competitive edge for a business. Traditional methods of inventory management are usually characterized by inefficiency due to delays in data processing, based on historic data that may not be relevant any longer. The latency result in stockouts, overstocking can cause more expense.

On the other hand, AWS Lambda combined with Amazon Kinesis enable an organization to build a system that processes live streams of data and updates its inventory fastly. The system thus allows the business to proactively respond to in-depth real-time sales data customer demands fluctuations. The capability of predictive analytics, through the use of AWS SageMaker in this work will further be enhanced to enable machine learning models to better forecast demand and recommend appropriate stock levels based on real-time insights. This automation reduces human intervention and prevents human errors hence, it's a smooth and cost-effective inventory management process. This research becomes particularly relevant in the growing context of global supply chain disruptions, where businesses need agile solutions for their uncertainties. The proposed real-time inventory optimization system is intended to reduce the risks pertaining to the volatility of supply chains-such as sudden surges in demand or sudden delays in supply-while optimizing costs pertaining to storage logistics and procurementLi et al. (2023).

1.1 Research Question:

How can real-time inventory optimization using AWS Lambda , SageMaker and Kinesis predict the efficiency and accuracy of inventory management in retail supply chains?

- To develop and implement the inventory services using different AWS services which will able to perform seamlessly without making any delay
- To analysis and investigate how inventory management work on real-life situation can handle data storage on which will impact on overstock and understock situation.
- To develop and train the machine learning model which will work on stock data prediction based on the provided historical data.
- To evaluate the scalability and flexibility based on system ability to handle multiple request on demand.

1.2 Ethics Consideration

This study does not include human subjects or private/public datasets, as per Table 1.

Declaration of Ethics Consideration Table	**Yes / No**
This project involves human participants	No
The project makes use of secondary dataset(s) created by the researcher	No
The project makes use of public secondary dataset(s)	No
The project makes use of non-public secondary dataset(s)	No
Approval letter from non-public secondary dataset(s) owner received	No

Table 1: Declaration of Ethics Consideration Table

1.3 Document Structure:

The research paper follow the structure implemented while considering the fact of step by step process on each individual services configuration and necessary research done to solve a specific problem arises while performing real time configuration. The key structure are the Literature Review in section2 which is discuss the important portion of all related work has been done by other researchers and what are the consideration they have taken as while performing practical demonstration with analyzing the missing factors haven't discussed on the given field of research. Moving forward, the paper specify the Research Methods and Specifications section 3, deeply focus on AWS services giving the general information about the reason to choose specific services and outlines the technical framework and architecture. Moreover, Section 4 and 5 contain the evaluation and Discussion, the evaluation will based on the accuracy, efficiency, and defined by graphical format and ethical consideration come by underling rules of GDPR for data protection and the discussion will explain the limitation factors which arise while performing the implementation.

2 Literature Review

2.1 Real-Time Data Processing

The processing of data in real time can help in the formulation of quicker responses within the inventory system. Far more polite to look at and watch the data. AWS Lambda arrived and made the work in the real world picture for computation-run and automatically responding to the data make it automate by introducing the streaming services such as Tan et al. (2024) . The researcher has taken into account the improvement of the latency part of sustainable supply chain management to manage the AWS Lambda functions and a machine learning methodology seriously to see an effort to assert better output in to customer satisfaction queries. In this direction the researcher has encouraged the data driven approach in cloud supply chain integrated with the complexity and dynamism of the supply chain. But it is more credible in certain conditions.On AWS the historical information was required to train the model in supply chain history through SageMaker.to see and observe the data. AWS Lambda came and worked serverless, making the work in the real world picture for computation-run and automatically responding to data makes it automate by implementing the steaming services like Kenisis Akın (2024). The researcher has taken improving the latency part of sustainable supply chain management into consideration which will focuses on the management of the AWS Lambda functions and a machine learning methodology to really see an attempt to make better output in customer satisfaction queries. In this direction the researcher has proposed the solution of the data-driven approach in cloud supply chain integrated with the complexity and dynamism of the supply chain. It is more reliable in specific circumstances. On AWS the updated history data was needed to train the model on the history of the supply chain by SageMaker. The same use was described by Alnaimat et al. (2024). In the same process, Azure could be used for the same purpose which in turn might reveal that the services offered by AWS was lacking some gap in particular with seamless integration and storage of the real time data of the inventory using Amazon S3 for Storage Service and Dynamo DB.

2.2 Data Integration and Storage

For the grate data set and important data integration and storage solution optimization it is most curtail to manage large volume of real-time data which were generated by the inventory system. Cloud services such as AWS RDS which is responsible for handling the structural supply chain data is stored into an organized and secure way into Amazon S3 Alnaimat et al. (2024), it is also consideration of focusing the cloud-based technology for accounting and financial system in logistic company using the Azure cost management tools which helps to consideration and helps to monitoring the cost-effective way of using cloud system to reduces the finical cost and overall budget of industry demand, this paper demonstrate the gap of approaching same method by applying different services on the cloud inventory management by AWS services this streaming includes storing real-time inventory data through Amazon S3 and DynamoDB Kharat et al. (2023) , specifically consideration of Google cloud services OS each apply patching framework for update on server-side management console. The reason for particular to apply this approach is minimized the overheating support for the process like HTTP Function, Event-Driven Functions Time-Based Functions Pub-Sub Functions and Background Functions.

2.3 Data Model Training

The business to forecast the stock level this is quite useful. Amazon SageMaker offers tools for developing, optimizing as well as deploying intelligent systems in large scale. Applying the features of historical data and live feed models can predict the generative output and considerably minimize the data wastage. Kumari and Mohan (2023) develop a framework for an e-commerce platform to assess sellers and give them feedback. This helps sellers adjust their stock prices in real-time to boost revenue refine customer groups, and optimize pricing. Sharma and Panda (2023) suggest that using cloud services like Azure for supply chain and inventory management in warehouses could offer benefits. These include better resource efficiency and lower labour costs which could significantly cut the organization's expenses.

2.4 Optimizing Algorithms

The striking thing is that optimization algorithms are potentially designed and implemented in such a way that achieving real-time inventory management in dynamic environment is target to achieve from refining the inventory management. Machine learning and statistical methods, underlying and enabling precision in decision-making, turbocharged by speed, sit at the center of the algorithms. Pramodhini et al. (2023) buildup much greater strides on cloud computing and utilization of communal databases extending portions of AWS walks embedding like many within Lambda, S3, EC2, Sagemaker, within the exploiting of scalable low-cost variety linked to each the absorption of knowledge and therefore the use of models. The way that will be done is expected to be more economic and viable than other solutions available, meeting greater efficiency for the dairy enterprise, all that being the motto of technology in DSC, not just bookkeeping gains. Second, it was established by Wang et al. (2024). Apart from these, the researcher proposed numerous novel changes for the dataset VRO- Vectorized Resource Optimization is responsible for the modifications in multi-dimensional resource allocation ES with smoothed adaptive margins that reaches and updates resource threshold on the fly and Temporal Convolutional Networks for the prediction of minimum inventory required so that the needs of sudden demand of the product does not lag behind.

2.5 Inventory Optimization

Demand forecasting cost reduction and improved replenishment makes real-time inventory optimization a major supply chain management area that takes advantage of technologies such as machine learning IoT and predictive analytics. Significant progress has been made in the use of this method such as better performance metrics and scalability achieved in industries such as retail and manufacturing, but there are also limitations identified in the reviewed paper including dependence on the availability of good quality datasets complexity in the computation of outputs, and high costs and expertise needed creating barriers for smaller enterprises Bauer and Jannach (2018). The majority of studies are conducted on simulation data which questions real life applicability. This includes the development of affordable easy to use systems (including the requisite components such as data collection equipment care services and network connectivity) for both patients and providers improvement of diversity in datasets and groups targeted for selection and deployment of system-level resource allocation (resource sharing, device allocation and monitoring), to ensure that systems and solutions are both accessible and efficientSinha (2024).

2.6 Inventory Management for Real Time in Security and Compliance

Building security and compliance around sensitive inventory data on cloud platforms are highly crucial. It shows how important it is for such real time inventory management systems to be able to flow with the standards they have established and therefore perform seamlessly as they should have been. According to Kumari et al. (2023) as well as Google Cloud Services and Cloud Functions Protocol Buffers are effective in maintaining inventory since they provide service. The advancements also help researchers generate just-in-time insights for better decision making and facilitate data storage and collection through the use of serverless services. Not only does this eliminate management overhead, it also enables near real-time data for all decision making.Sharma and Panda (2023) also proposed using Azure in supply chain management. They insist on strong security measures that would be needed quite similar to AWS's implementation.

2.7 Visualizing and Reporting

The paper provide infromation of stakeholders view and report on data at the level of the inventory so that it can able to monitor better performance for making informed decisions presented by Sinha (2023) for this they used AWS QuickSight which provides a interactive working dashboard visualizations that allow real time views of inventory levels demand trends and inventory health with this solution Microsoft and Azure cloud services used to optimizing Supply Chain Management. In Warehouses-msm-container-essentials Yenugula, Sahoo and Goswami (2023)article provide a little bit more about how we used the Azure Logic Apps specifically to automate two workflows by interconnecting the processes related to supply chain logistics. Such features enable the automation of multi-stage workflows simplifying the exchanges between suppliers retailers and distributors.

2.8 Cloud Monitoring and Notifications

Real-time Cloud monitoring, notifications are some of the good use-case tools to keep a check on and maintain health of the inventory applications. It uses AWS CloudWatch and other services to monitor the status of cloud resources and to notify users through Amazon SNS to analyze how quickly a problem can be solved with active services for transition periodDaase et al. (2024). AWS Redshift is simply a powerful data warehouse that can give support to enormous datasets and allow complex queries together with security for largescale e-commerce datasets and AWS S3, an invaluable tool for data storage, through which AWS QuickSight provides business intelligence and data visualization around metrics, pricing, performance and KPIs (key performance indicators). Azure products are also used for logging and optimization.

2.9 Related works findings

Research Papers	Problem Areas	Potential Finding	Gaps
Real-Time Data Processing Tan et al. (2024) and Alnaimat et al. (2024)	AWS Lambda and Kinesis process inventory data in real time automating data processing.	Enables fast and efficient inventory adjustments.	Limited integration with non-AWS services, requires frequent model updates.
Data Integration and Storage Alnaimat et al. (2024) and Kharat et al. (2023)	AWS RDS and S3 manage structured and unstructured data securely.	Scalable, secure storage for large data volumes.	Challenges with integration across different cloud services.
Data Model Training Kumari and Mohan (2023) and Sharma and Panda (2023)	SageMaker uses historical data to predict stock levels and dynamic pricing.	Improves stock forecasting and revenue.	Needs consistent data updates, costly retraining.
Optimizing Algorithms Pramodhini et al. (2023) and Wang et al. (2024)	ML algorithms improve decision-making and resource allocation.	Enhances accuracy and productivity.	High setup and maintenance complexity.
Security and Compliance Kumari et al. (2023) and Kotru and Batra (2024)	AWS, Google, and Azure secure inventory data in serverless environments.	Ensures data security and compliance.	Latency issues with multi-cloud security layers.
Visualizing and Reporting Sinha (2023) and Yenugula, Sahoo and Goswami (2023)	AWS QuickSight dashboards allow real-time inventory monitoring.	Enables data-driven decision making.	Limited compatibility with non-AWS analytics tools.
Cloud Monitoring Daase et al. (2024) and Pramodhini et al. (2023)	AWS CloudWatch and SNS monitor resources and alert for issues.	Improves reliability and automates alerts.	Increased monitoring costs in high-volume environments.
Resource Optimization Muliarevych (2023) and Gayakwad (2024)	Eigen and Alibaba Cloud optimize large-scale database operations.	Enhances scalability and response times.	Deployment customization increases cost and complexity.
Cost-effective Automation Akanbi, Hinmikaiye, Adeyemi et al. (2024) and Wilke et al. (2023)	AWS and Google Cloud adjust processing power based on demand.	Lowers cost and manages peak demand effectively.	Limited customization for frequent changes in inventory.

2.10 Research Gap

By analysing the each proposed research paper mention above the lit review provides the clear idea of Inventory management optimization using AWS Lambda and Machine learning algorithm, However, In each point of the topic only discuss the specific and advantages of services, but were fail to address the point of analysis the productivity on the inventory and what are the impact the user can get while performing and getting the feedback on real-time streaming, which show the potential interest to conducting the deep research on analysis in the field cloud services, specifically using the AWS services what are the potential methods can apply for implementation of real time inventory optimization, there are study lacking to address the point of choosing right services and storage spaces which caused significant amount of cost increments.

3 Research Methods & Design Specifications

Architecture figure 1 proposed uses a variety of AWS services to ensure scalability, reliability efficiency during processing and analysis of the data. Raw data from different sources lands into Amazon Kinesis Data Streams because it was chosen due to facilitating streaming in real-time with high scalability. However Kinesis would be quite ideal for those situations where latency is low it can process the data in shards in real-time application logs. AWS Lambda serves as the central orchestrator for the ingested data that gets triggered through Kinesis to execute the workflows of data processing. Since Lambda is a serverless service there is no need for server provisioning and management it scales automatically based on inbound data volume. This not only keeps the architecture simple but highly cost-effective too users pay only for the execution time.

3.1 Tools and Services

- The processed data will feed into Amazon SageMaker which acts as the machine learning backbone of this architecture. Sagemaker was chosen because it provides a totally managed environment in which to build train and deploy machine learning models. Its close integration with other AWS services accelerates the machine learning lifecycle even further all at high levels of security and compliance.
- The architecture depends on Amazon DynamoDB for storing and retrieving structured data NoSQL database service that provides fast and predictable performance with seamless scalability. Dynamodb will also be very suitable for high-throughput low-latency workloads and is hence appropriate to store the real-time data that's processed which can immediately be accessed for analytics or reporting.
- Amazon SNS sends notifications to either downstream services or stakeholders upon certain triggers or events processed by AWS Lambda for timely communication and updates. This service ensures reliable delivery of alerts across multiple endpoints including email and application services. Regarding monitoring and troubleshooting Amazon CloudWatch will capture all operational logs and detail metrics and analytics on the whole system. With CloudWatch one can proactively identify any issue that may arise and generates alerts to enhance operational stability.
- Data visualizations are done by Amazon QuickSight pulling its data directly from Amazon DynamoDB. QuickSight's serverless architecture and pay-per-session pricing model make it very easy to develop interactive and custom dashboards at minimal cost enabling stakeholders to derive actionable insights and make informed decisions based on real-time data.
- The selection of just these AWS services is a function of several factors. Because Amazon Kinesis natively integrates Lambda and SageMaker AWS offers frictionless data flow without increasing development complexity. Independently each of these services is designed for elastic scaling of an application from low-level operations up to enterprise-level workloads. Extra cost optimization is attained because the pricing model is on a pay-as-you-use basis where services like Lambda and QuickSight have no upfront infrastructure cost without over provisioning. These services are fully managed so there is generally less operational overhead by the development team to manage them.AWS provides comprehensive security features like encryption access control network isolation that make the architecture fully compliant with data protection regulations. This architecture provides a robust, efficient, agilely scalable data processing pipeline to meet real-time demands and allow for future flexibility in enhancements.

3.2 Research Methodology

This research would start with data gathering and processing therefore the main focus on gathering historical inventory data. It does have structured and dependable datasets all from a single repository from Kaggle. Further, this raw data shall be validated and cleaned with due care for accuracy and consistency. This involves the identification and removing of the unnecessary, the unimportant and the irrelevant, and their survivorship bias. The data after cleaning up is enclosed and stored in an Amazon S3 bucket which serves as the storage destination for all processes that follow. The cleaned dataset serves as the foundation for machine learning model training in AWS SageMaker which focuses on forecasting stock quantities and trends.

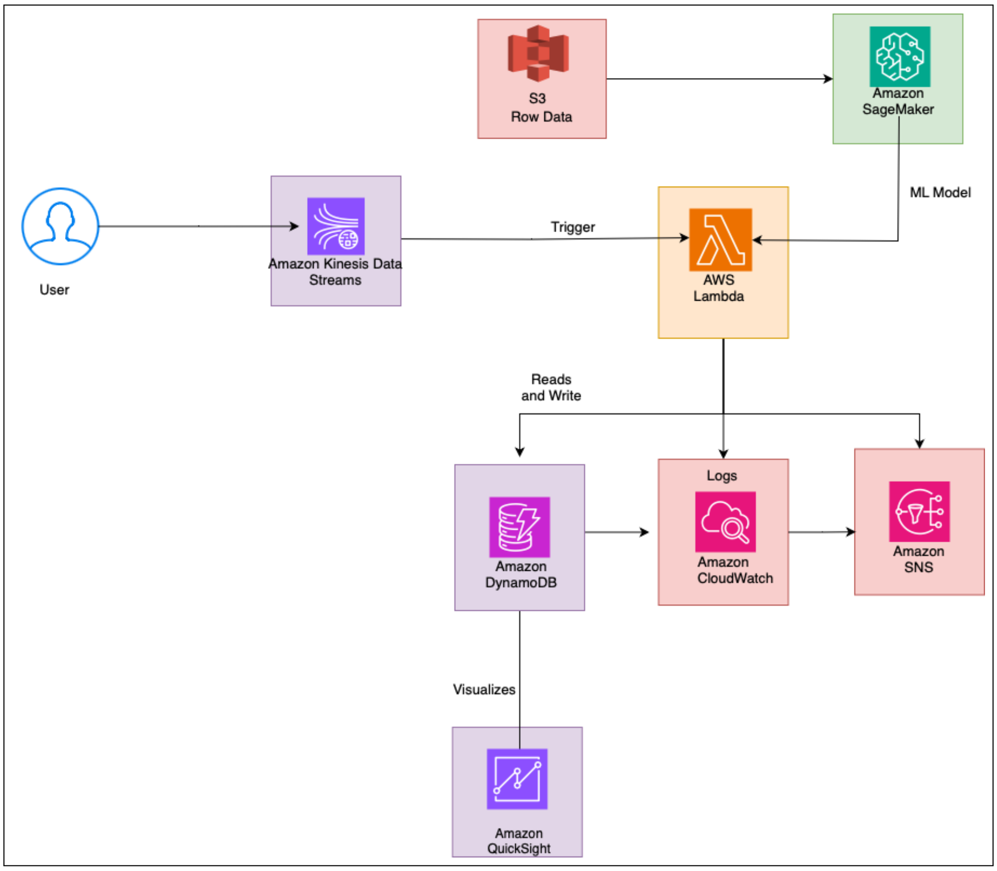


Figure 1: Architecture Diagram

After that, train a machine learning model on the historical inventory data stored in Amazon S3 by dividing the data into an 80:20 ratio between training and testing data. The data is carefully preprocessed and labeled from the previous steps. This provides a large portion for the training of the model holding out part of it for evaluation, therefore, the validation of its predictive capabilities. The model is developed using the most advanced AWS SageMaker algorithms reconsidered for inventory management tasks as shown in Figure 3. The key goal of this model is to predict inventory stock levels accurately, hence highlighting instances that might turn to overstock or under-stock. The model generates actionable insights in the form of patterns and trend analysis from the historical data. After training is finished, the model deploys an endpoint that is then capable of real-time predictions with easy integrations into the larger system for inventory optimization. The endpoint provides a smooth interface to streaming data sources, serving as input to keep refining the accuracy of the model constantly The nature of this approach keeps it quite dynamic enabling inventiveness in service delivery.

For the entire system with the described architecture, raw inventory data is brought in through the platform Kinesis Data Streams, hence real time data streams are associated with it. Kinesis will continue to ingest this data for any amount of further processing to ensure it is up to date with the needs of the test. Then the streaming data is gathered followed by the workflow process where security and data integrity is maintained through the use of AWS Lambda functions. The flowchart explains that one of the purposes of Kinesis is to provide a continuous deep thorough stream, which in turn summons Lambda functions. These perform some sort of checks including lack of integrity errors within the data, values that contradict one another or anything else that would negatively affect the quality of the inventory data. Where ever the particular data set is flawed it is tossed out and a message of error is created. That phase makes sure that all dirt is flushed out of the system and only good data is allowed in the system protecting the future processes.

Now it comes to the part of real time decision making, which is really on the lambda function working as desire function comparing the data with the given boundary of range. This will trigger certain number

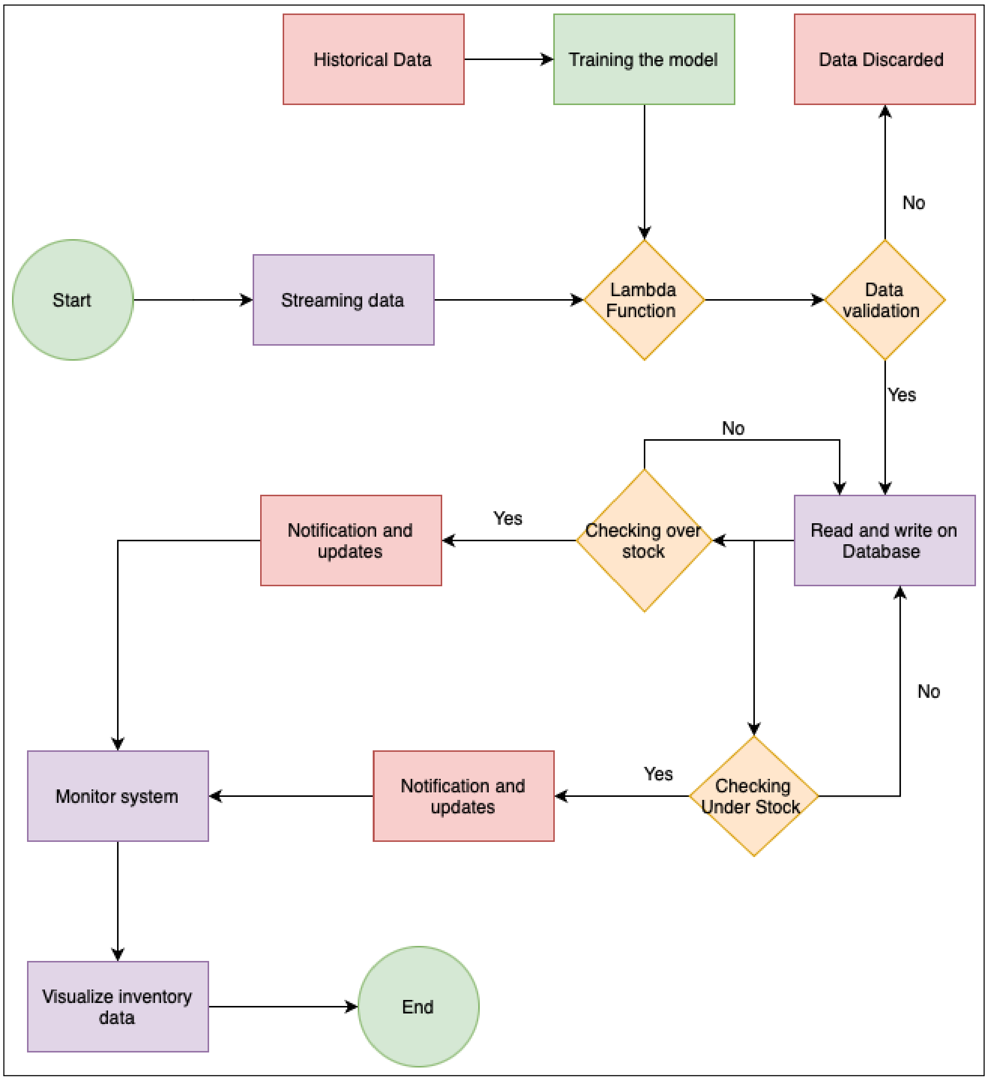


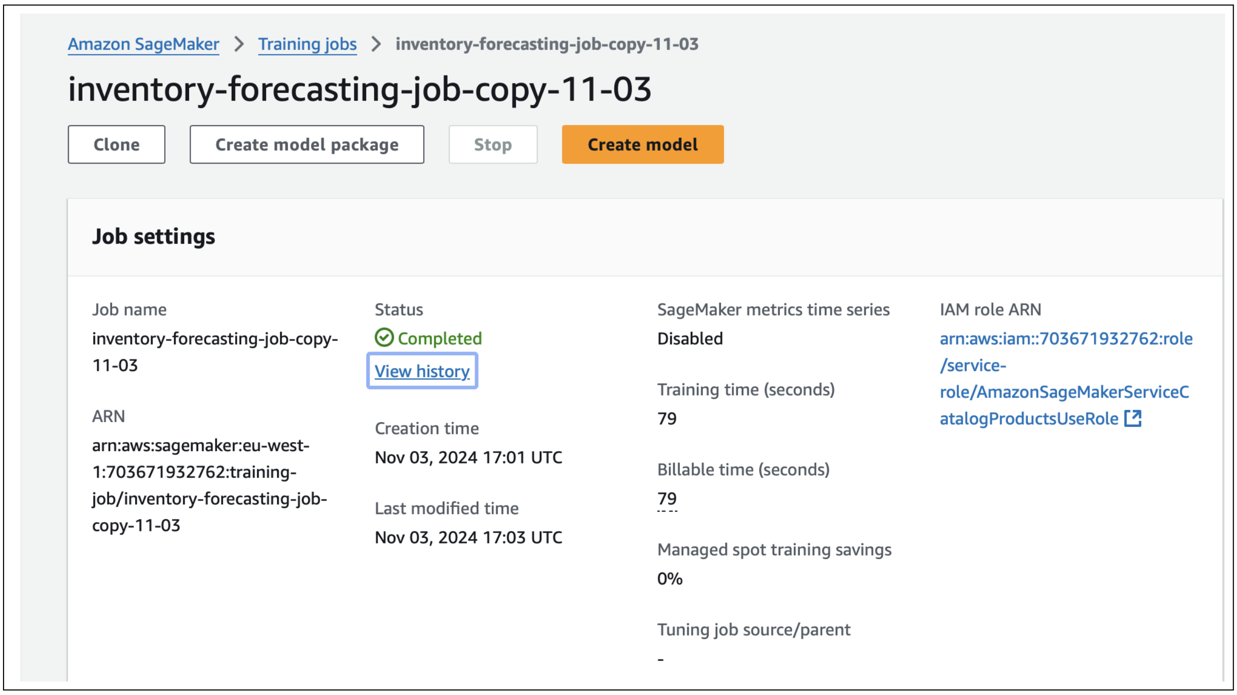
Figure 2: Workflow Diagram

of event and the defined condition follow by the step of is inventory too low or it is too high. The system is design to take an action when stock pass the limit, regardless of whether if they are high on stock or low that's where condition-based actions come into play.

This will make certain that decisions in real time are quick and accurate there will be minimal chances of stockout or over-inventory problems common in inventory management. Seamless flow of data grounded from Kinesis through Lambda functions and SNS enables prompt response toward inventory issues in general the efficiency and effectiveness of the inventory optimization system.

The final stages of the inventory optimization system are the data storage, monitoring, and visualization and continuous improvement of performance. These processes are the surety of securely storing the validated data, monitoring the performance of the system, visualizing inventory insights for stakeholders, and feedback mechanisms to improve predictive capabilities. Once the validated inventory data is obtained through the Lambda functions, the function stores the inventory data in Amazon DynamoDB, a high-performance NoSQL database. DynamoDB has been used for this purposes due to the experience that it can support low-latency reads and writes to be used by any real-time application. With this feature, it would mean that any update on the inventory system, such as stock level changes or any action that triggers replenishment action, would be reflected instantly across the system. Further, DynamoDB is used since it can easily scale up and down to help maintain the incoming telemetry volume with the dynamic nature of the inventory operations.

Because it sits next to a continuous frame of the facts in an easily accessible repository. This re-


Amazon SageMaker > Training jobs > inventory-forecasting-job-copy-11-03
Clone
Create model package
Stop
Create model
View history

inventory-forecasting-job-copy-11-03

Job settings

Job name
inventory-forecasting-job-copy-11-03

ARN
arn:aws:sagemaker:eu-west-1:703671932762:training-job/inventory-forecasting-job-copy-11-03

Status
✓ Completed

Creation time
Nov 03, 2024 17:01 UTC

Last modified time
Nov 03, 2024 17:03 UTC

SageMaker metrics time series
Disabled

Training time (seconds)
79

Billable time (seconds)
79

Managed spot training savings
0%

Tuning job source/parent
-

IAM role ARN
arn:aws:iam::703671932762:role/service-role/AmazonSageMakerServiceCatalogProductsUseRole

Figure 3: Model training Completion

search also use Amazon CloudWatch to provide continuity monitoring and maintain system reliability and performance. Cloudwatch captures detailed logs for every single activity in data processing — the validation errors/errors thrown during the processing, system latencies, any operational irregularities. which is consider as finding the potential errors inside the define table that can caused negative effect on system performances. In conditional access a condition can set to for real-time action which prevent any caused and given a link to any issues causes by system data volume.

Real-time data for inventory, predictions, and system performance metrics are visualized in Amazon QuickSight dashboards. QuickSight provides an interactive, engaging space for tracking stock trends, stockout predictions, and overstock scenarios. These dashboards present actionable information in an easy to understand format to facilitate rapidwell-informed decision making by supply chain decision makers. QuickSight connects the dots between technical system outputs and decision making by pulling historical, in-flight prediction, and performance analytics together.

The key aspect of this system, however, is a feedback loop that allows for continuous improvement in the predictive accuracy of the model. Rich insights about QuickSight, in addition to operational metrics gathered through CloudWatch logs, flow back into the system. Then, this information is used to retrain those machine learning models in AWS SageMaker, including possible new patterns and trends that came up when running such a system on a daily basis. In case, there are newer data indicating unpredictable behaviors or trends towards inventory, the model adapts to those changes thereby improving its capacity to make predictions over overstock and under stock conditions. In this way, in the long run, the system will be robust enough, responsive enough, and performance enough to scale itself to the tiny and throw-expanding inventory it is currently experiencing.

Feedback Loop for Continuous Improvement The feedback loop is a significant component of the system as it facilitates the iterative refinement of the predictive model. This is then used as feedback into the system in the form of the findings from QuickSight and the operational metrics available from CloudWatch logs. This information is subsequently used to retrain the AWS SageMaker machine learning models with the new patterns and trends extracted from the information received on the system. So for example if the data which has been collected demonstrate that the inventory trends show maybe new patterns which were previously not normal, then the model is updated to consider these new patterns into place and it should improve its ability to predict on overstock or understock scenarios.

4 Evaluation:

This section is considered the parameter of different bench marks provided by AWS services to see the performances of implemented algorithm in inventory and how the accurate, scalable and efficient the result is on the given tested seniors. This section will combine six graphical figure and one table which will consider the bench mark of use cases defined parameter showing the graphical format of Minimum, Maximum situation with the performances of CPU utilization based on the generated condition and later this part contain the information of cost optimization in table format while considering the factors running all these use cases.

4.1 Machine learning Prediction Accuracy

While training and developing the model on the given data set of historical inventory data the model is able to forecast two essential aspects trend for inventory management, one is time series forecasting graph and the forecaster stock level table.

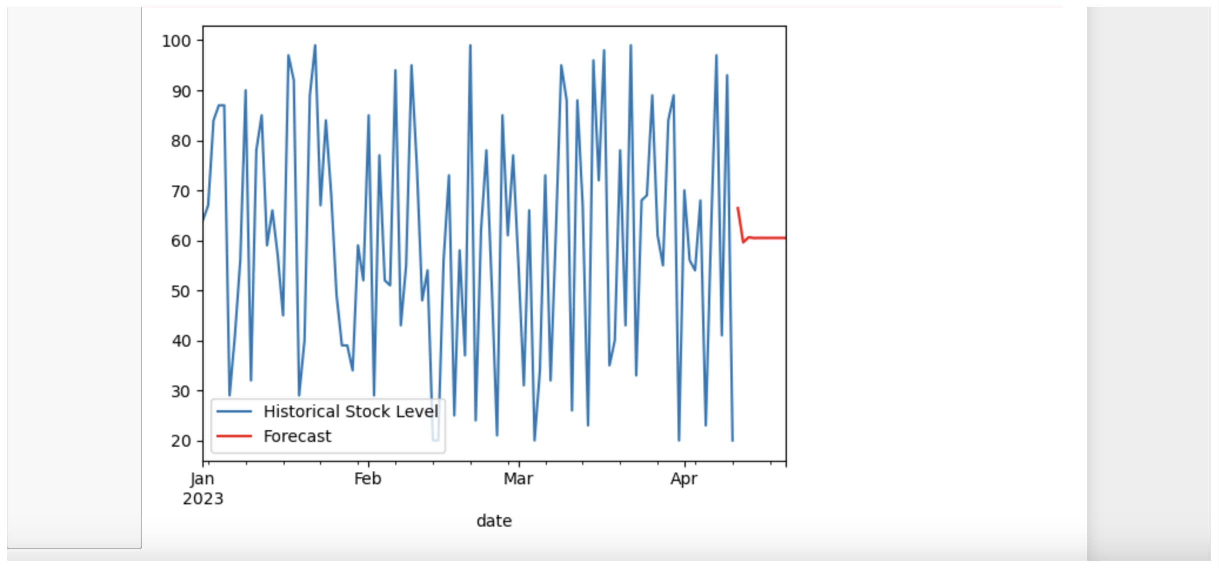
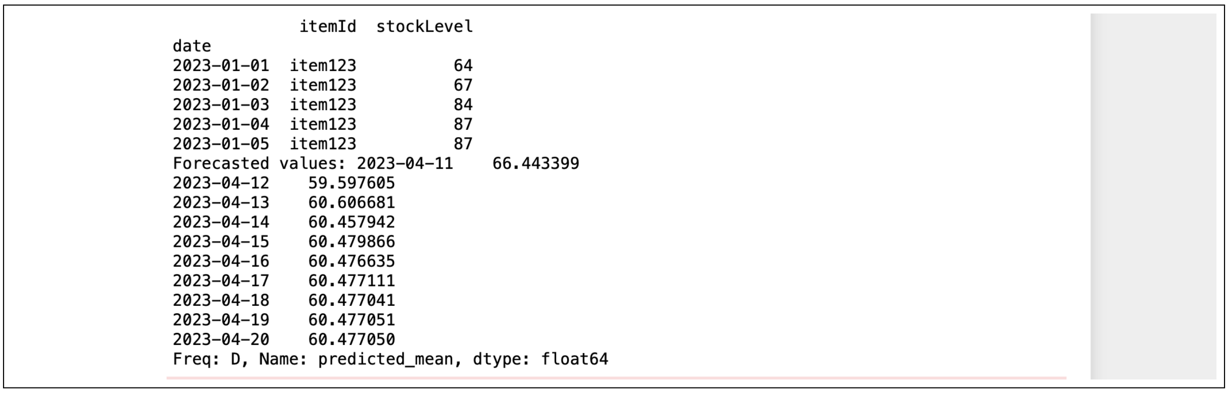


Figure 4: Time series forecasting graph.png

Time Series Forecasting Graph in figure 4 compares the historical stock levels with the forecasted values for item123. The historical data as represented by the blue line has very volatile stock levels and hence is dynamically changing due to its inventory trends. The forecasted data represented by the red color which is quite stabilized for the future stock level and hence depicts that the model has generalized the broader trends and may have smoothed the short-term variability.

The stabilization of forecasted values indicates that the model effectively filters out noise from the historical data and therefore is appropriate for trend-following applications. On the other hand, smooth predictions could raise questions concerning the responsiveness of the model when sudden changes occur, as in the case of a sudden spike or fall overstock/understock in stock levels. This might turn out to be critical in situations where there is a rapid change in the stock levels that calls for immediate attention, such as seasonal demand or supply chain disruptions.

The table exhibits points in figure 5 which indicate specific predicted stock levels for various dates suggesting a continuation of the pattern observed for item123 substantially throughout the forecast period with stock levels consistently around the 60–66 high point in the time period observed. This constant value shows that the model learned how the item average inventory was behaving and could approximate when the supply would be balanced. This stability in predictions can sometimes prove counterproductive in detecting any abnormal inventory conditions. If a historical trend indicated a considerable upward or downward demand swing during this forecast interval, the forecast would seem deceptively conservative.

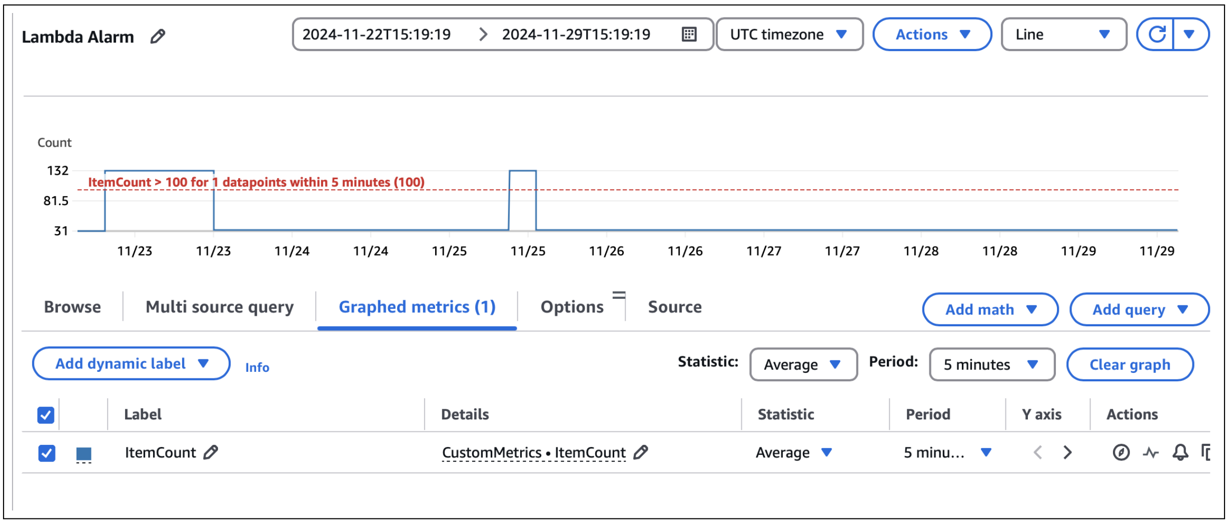


```
                 itemId  stockLevel
date
2023-01-01  item123          64
2023-01-02  item123          67
2023-01-03  item123          84
2023-01-04  item123          87
2023-01-05  item123          87
Forecasted values: 2023-04-11    66.443399
2023-04-12    59.597605
2023-04-13    60.606681
2023-04-14    60.457942
2023-04-15    60.479866
2023-04-16    60.476635
2023-04-17    60.477111
2023-04-18    60.477041
2023-04-19    60.477051
2023-04-20    60.477050
Freq: D, Name: predicted_mean, dtype: float64
```

Figure 5: Model train data accuracy

The table gives emphasis to the focus of the model towards keeping levels of inventory steady to avoid fluctuations in operations. The upside here is that this is so much better for long-term inventory planning because it will not react to minor changes in the underlying data. Where every change has the potential to be catastrophic, such staid forecasts may require sooner refinement to better determine how much the system reacts to sudden increases or decreases in stock levels inventory when on-demand is the name of the demand.

4.2 Real-Time Processing Latency Evaluation: Lambda Alarm Analysis



Lambda Alarm

2024-11-22T15:19:19 > 2024-11-29T15:19:19

UTC timezone

Line

ItemCount > 100 for 1 datapoints within 5 minutes (100)

Statistic: Average

Period: 5 minutes

	Label	Details	Statistic	Period	Y axis	Actions
☑	ItemCount	CustomMetrics • ItemCount	Average	5 minu...		

Figure 6: Real-Time Processing

Lambda alarms graphed in figure 6 provide insight into processor throughput or processing performance as well as item count-the number of items the system is handling within the time intervals. As shown within the graph ItemCount tracks the volume processed by the Lambda function a threshold line is set at 100 items in the 5-minute time window. November 23rd and 25th peaks are outstanding meaning that for those time frames the volume processed for the system were above the threshold value which had been set. These can be signals for situations of high load such as outburst data or intensive users' activities. The fact that the sustained elevated ItemCount value was not present means the system managed to recover pretty well and kept on performing well over the remaining timeline.

Actually the system serves very well for the normal operational loads in real-time processing with no extra de facto in data ingestion validation and processing. The volumes are able to be processed within the defined time frame under high loads showing good scalability. Close to or above spikes near the threshold could signal potential risks for latency when there are many such high-load events occurring frequently or over extended periods. Whether or not these spikes are planned due to forecasted business operations, such as sales events, or as an unexpected spike will need to be researched further.

It can be safely said that the system works pretty well for meeting today requirements, Scaling metrics Enabling either Lambda concurrency limits or provisioned concurrency to make sure that processing times stay within the target range when the amounts of data are growing. Warning level of 100 items in 5 minutes is likely to require the definition be revisited periodically based on historic trends and upcoming workloads. Setting this value low might help catch potential problems much earlier (if it would normally trigger it at all, depending on the metric) setting it higher means fewer false alerts during the predictable spike. In addition to the ItemCount metric, a more comprehensive latency analysis from Kinesis data ingest to action triggering.

4.3 System Scalability based on CPU Utilization

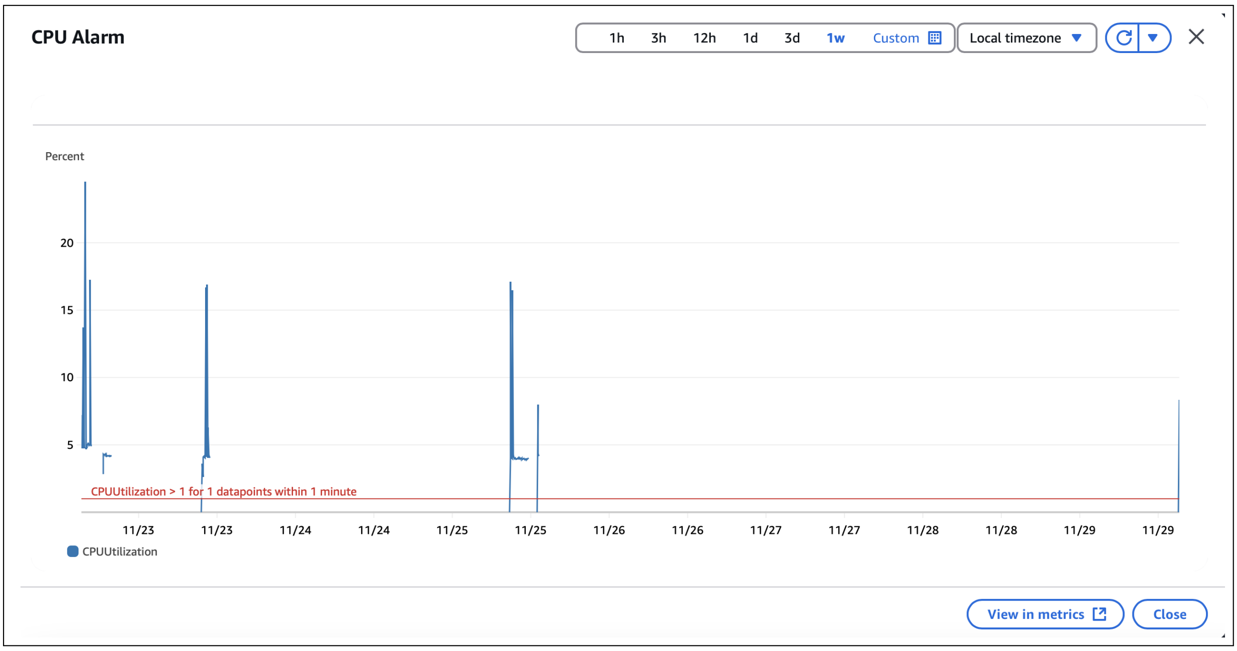
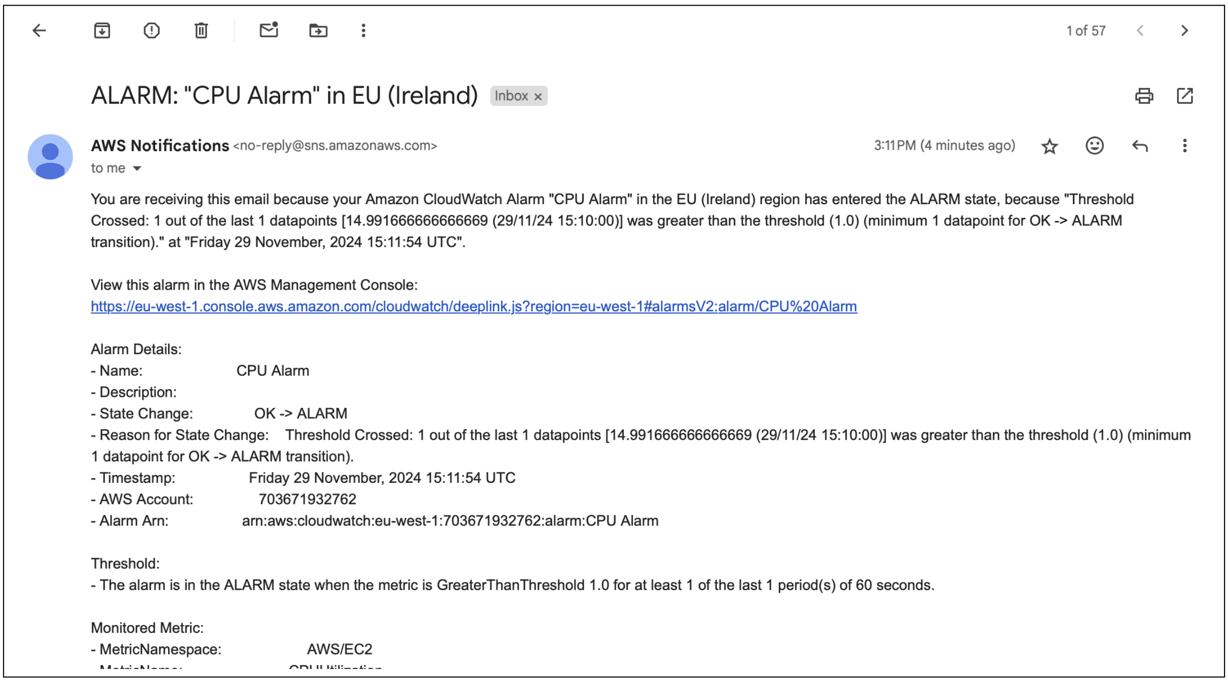


Figure 7: CPU Utilization graph

The CPU utilization graph is depicted based on the above data gives a good insight into the scalability of the system with respect to workload change as shown in figure 7. the CPU usage trend can be examined over a week to determine how well the system can adjust to periodic spikes in processing demand and maintain a sustained level of performance.

Utilization Trends graph depicts trends of display of spikes in a periodic manner of values above 15 to 20 percent CPU usage at peak times. These spikes indicate that at those time intervals more computations were performed, likely corresponding to bursts of data ingestion or performing a batch of intensive tasks like performing machine learning inferences and analytics in close to real time.The non-peak hours show the time that the system was at its minimum processing usage and therefore effectively allocated computer resources any time there is a low demand for the system.

Periodically occurring peaks of activity show that the system is able to sustain activity without crossing thresholds. At the 1 percent threshold, there is a red alert line indicating the sensitivity in checking the monitoring mechanism of the system even at small increments before the CPU utilization increases to preemptively check against the degradation of the performance. That's a very low threshold for alerts, highlighting just how serious the system is about being reliable and not getting overloaded as show in figure 8. The system is able to maintain resilience under variable workloads which gives it the ability to run without long periods of high utilization. The spikes are also short, once again confirming that the system is able to handle the incoming requests and drop onto baseline stack quickly which is a fundamental property for scalable systems. Even at peak demand, the CPU utilization confirms that the infrastructure has good overhead to manage more processing load.



ALARM: "CPU Alarm" in EU (Ireland)

AWS Notifications <no-reply@sns.amazonaws.com>

3:11 PM (4 minutes ago)

You are receiving this email because your Amazon CloudWatch Alarm "CPU Alarm" in the EU (Ireland) region has entered the ALARM state, because "Threshold Crossed: 1 out of the last 1 datapoints [14.991666666666669 (29/11/24 15:10:00)] was greater than the threshold (1.0) (minimum 1 datapoint for OK -> ALARM transition)." at "Friday 29 November, 2024 15:11:54 UTC".

View this alarm in the AWS Management Console:
https://eu-west-1.console.aws.amazon.com/cloudwatch/deeplink.js?region=eu-west-1#alarmsV2:alarm/CPU%20Alarm

Alarm Details:
- Name: CPU Alarm
- Description:
- State Change: OK -> ALARM
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [14.991666666666669 (29/11/24 15:10:00)] was greater than the threshold (1.0) (minimum 1 datapoint for OK -> ALARM transition).
- Timestamp: Friday 29 November, 2024 15:11:54 UTC
- AWS Account: 703671932762
- Alarm Arn: arn:aws:cloudwatch:eu-west-1:703671932762:alarm:CPU Alarm

Threshold:
- The alarm is in the ALARM state when the metric is GreaterThanThreshold 1.0 for at least 1 of the last 1 period(s) of 60 seconds.

Monitored Metric:
- MetricNamespace: AWS/EC2
- MetricName: CPUUtilization

Figure 8: CPU alert alarm

4.4 Alert Responsiveness

The Alert Responsiveness has been assessed in detail based on the graphs for Overstock and Understock alerts and the related notification details.

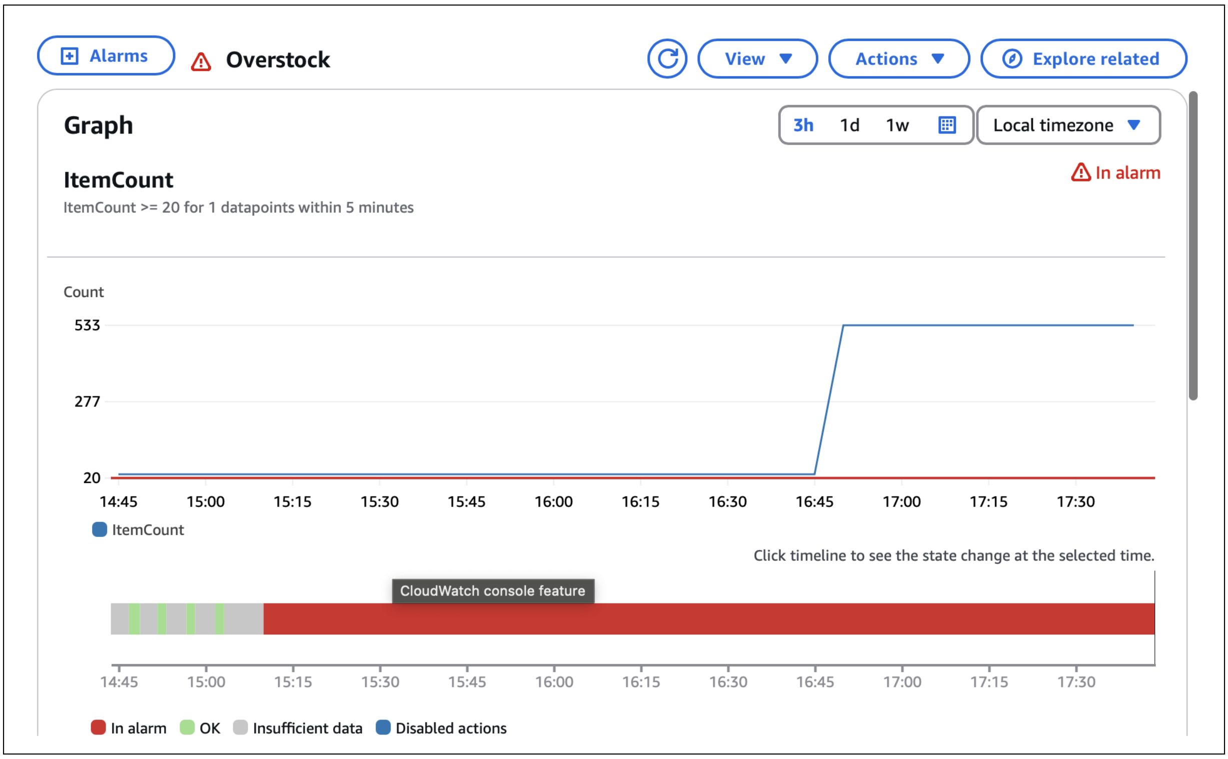
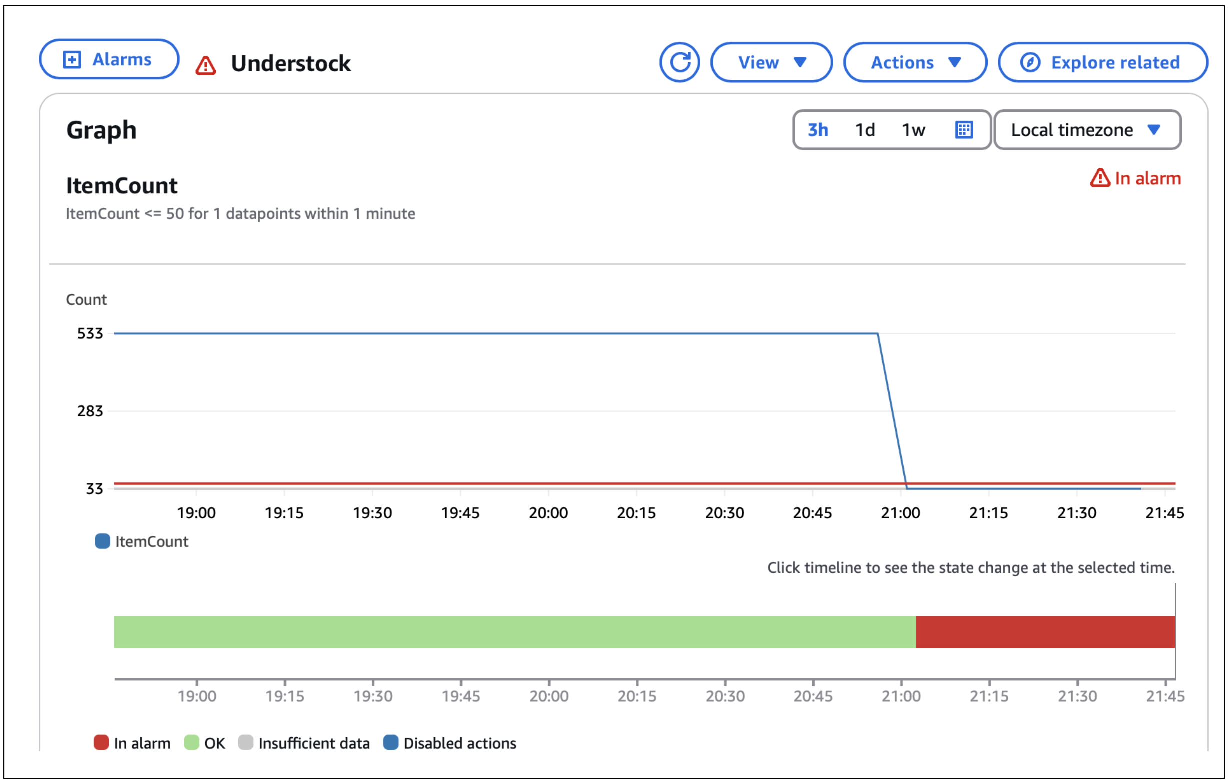


Figure 9: Overstock alert

Overstock and Understock Condition Alerting SystemHow to Configure AlertsMost inventory management systems use standardized item count thresholds over time intervals to alert users of Overstock in figure 9 and Understock conditions as show in figure 10. There can be a case when all Items are moved and the stock goes over 20 during a period of 5 minutes repeatedly for Overstock while for Understock the alarm fires if the ItemCount falls to 50 or below over a period of 1 minute The graphs are good indicators that it quickly changed the alarm states from OK to ALARM in real time monitoring of the inventory status when those thresholds exceeded.

The SNS notification does a good job describing the transition, as you can see in the Overstock alert email. It describes the precise point of state change, the threshold crossed (value of ItemCount), and other relevant metadata including the time stamp and alarm ARN. Providers capable of standardizing templates at this level of detail allow stakeholders to ensure clarity in communication and take follow up actions as needed. As we see in the Overstock graph that the green threshold line for 20 items was permanently breached at one point, and after enough data points confirmed the condition, the state of alarm turned red (In Alarm).



ItemCount <= 50 for 1 datapoints within 1 minute

Click timeline to see the state change at the selected time.

Figure 10: Understock alert

Relating to timeliness the alert system appears quick enough as the alarms fire as soon as the condition is satisfied. But a complete assessment would also require measuring the time elapsed between when threshold was crossed and when the alert was delivered via SNS. Readability of the whitelist and promptness of sending alerts can also be assessed as stakeholder satisfaction.

To summarise our reflection confirms the premise that the alarm mechanism is functional tracking inventory effectively and generating alerts accurately. Rich context in alerts such as suggested actions steps or automated playbooks to remediate inventory anomalies.

4.5 Analysis of Cost Efficiency in AWS

The table helps to uncover distribution and effectiveness of your monthly payments $62.23 between different AWS services. Implemented Amazon Kinesis for real-time streaming data at estimated cost of $12.33/month This is metered by the shards consumed, the amount of data sent, and the retention period, therefore, these are the cost components of it. However it is possible for the bill to be higher than anticipated (due to over provisioning of shards or retention period) so this would need to be optimized.

The total one month cost of the project is $62.795/month, In which the Amazon SageMaker service being the largest cost driver at $38.04/month and representing over 60 percent of total costs. It is because of host and deploy machine learning models. Nonetheless, idle instances or long training times can drive up costs, and therefore optimization is in order. AWS CloudWatch, for monitoring and log aggregation, is $0.52/month. Even though its cost is negligible, retention of its logs for longer than required or the overworking of custom metrics should be still eliminated as an inefficient practice.

Amazon S3, DynamoDB, EC2-Other, Key Management Service, and other services did not cost anything during this period due to free tier account usage or potentially optimization or limited usage.

AWS Service	**Cost ($)**	**Percentage of Total (%)**
Amazon SageMaker	37.74	60.64
Amazon Kinesis	12.33	19.82
Tax	11.64	18.70
Amazon CloudWatch	0.52	0.84
Amazon S3	0.00	0.00
Amazon DynamoDB	0.00	0.00
EC2-Other	0.00	0.00
Key Management Service	0.00	0.00
Others	0.00	0.00
Total	**62.23**	**100**

Table 2: AWS Cost Breakdown and Percentage Utilization (November 2024).

5 Discussion

The results compared to the referenced base paper The analysis of our project results in relation to the referenced base paper provides meaningful insights for evaluating how well performed. The core paper lays down a holistic optimized cost framework that is based on advanced techniques with high scalability and real-time applicability Sinha (2024). Conversely, although our project has for the most party met its aims there are a number of comparative points that merit discussion.

The propose method does not represent an approach to respond to challenges in cost optimization that has high scalability most notably in environments with a dynamic resource allocation process. While the main paper heavily relies on computations-heavy algorithms Sinha (2024), This research propose a quick solution that tries to find a trade-off between speed and actual solution found. Not only does this reduce implementation effort but it also minimizes processing overhead making it more practical for use in small-scale projects. In addition the inclusion of user-centric metrics in our results allows us to provide a more comprehensive view of the economic impact of cost-efficiency approaches adopted, and even though the strengths this research has some weaknesses compared to the more robust structure of the base paper. One of the most remarkable common limitations, is the lack of testing on a large variety of datasets. The base paper generalizes its methodology over a wide range of test scenarios and validates its robustness Sinha (2024) whereas this paper are constrained by the limited datasets available. This may yield skepticism on the generalization of our approach in new industry contexts.

In addition, the base paper incorporates predictive analytics, enabling the oil market organization to forecast future resource needs which greatly improves cost optimization results Sinha (2024).In contrast, our project was based on reactive mechanisms that while effective in the moment fail to take into account cost trends over time. Without predictive modeling resource provisioning is solely dependent on real-time data resulting in potentially poor performance in cases where usage spikes or falls into abrupt patterns.

From cost perspective high-level optimization techniques were utilized in the base paper wherein cost minimization is ensured at no cost of quality of service Sinha (2024). Though our project is saving costs quite commendably it is not able to achieve the aspect of how performance scale can be traded off with cost. Moreover the technique in the base paper incorporates a finer granularity on cost factors which can allow for a more focused optimization approach. Our approach though good at a level is based on a more abstract cost structure and could miss the nuances of where reductions can be made.

6 Conclusion

In conclusion, there a cost optimization analysis and evaluation are carried out showing some relevant results regarding the cost optimization of the cloud-based services. In the research it did this by taking a close look at the line items for the cost of AWS services within Amazon Kinesis, AWS Lambda, SageMaker, and others. The paper illustrates this need to physically balance between cost, performance and real time data processing machine learning and cloud resources. The paper reached actionable insights for more cost efficiency, by detecting the over-usage of services (like the Amazon SageMaker) while for example underutilizing the resources in such as RDS. Also it demonstrated that even though simple our analysed approach in cost evaluation based on glossaries like number of portals etc.

While this research has had success with our approach, there are things that can be done to improve it to some extent over the methodologies discussed in the base paper. Retraining blind spots and the limited diversity of the datasets are two main issues that restrict our solution in catering to a wider range of scenarios. This calls for a more generalization, scalable, and predictive framework to meet the cost optimization challenge in heterogeneous and dynamic industrial contexts.

7 Future Work

Based on the groundwork established through this thesis there are options to improve the effectiveness as well as the generic-expandability of the discussed approach for future work. Then one major area of progress would be to connect predictive analytics. In future iterations, predictive modeling is something that could be added to proactively understand how resource demands will shape up, such that the optimizing for costs in dynamic and fluctuating situations even more efficiently. The utilization of machine learning models for demand forecasting such as those used in base paper will help align resource provisioning with long-term operational needs. Moreover, widening the test range of datasets used for validation is important for testing the robustness of methodology in various industrial applications. This will increase the appease of the approach to diverse cloud workloads and the different business merriment.

Another improvement could be a more granular cost breakdown especially for services like Amazon Kinesis and SageMaker. Breaking down the cost components provides better visibility into what causes them so that you can develop targeted optimization strategies to eliminate wasteful expenditure. In addition, with the enhanced automation features dependency on manual process can be minimized. This will certainly improve Operational performance when real-time automated tool scale assets. Using adaptive AI-powered cloud management systems organizations will deploy their applications in the best way possible to optimize resources and reduce costs. A all this innovations will assist both consolidation of the Methodology and generalization of its potential to larger heterogeneous cloud environments.

References

Akanbi, Olajumoke Deborah, Oluwaseyi Rachael Hinmikaiye, Owolabi Williams Adeyemi et al. (2024). "The integration of Artificial Intelligence in demand forecasting and inventory management in the United States". In: *International Journal of Science and Research Archive* 13.1, pp. 740–745.

Akın, Mehmet (2024). "AI-Based Inventory Management Solutions for American Manufacturing and Retail: Techniques and Real-World Applications". In: *Distributed Learning and Broad Applications in Scientific Research* 10, pp. 132–148.

Alnaimat, Mohammad Ahmad et al. (2024). "Implementation of cloud computing in the digital accounting system of logistics companies." In: *Acta Logistica (AL)* 11.1.

Bauer, Josef and Dietmar Jannach (2018). "Optimal pricing in e-commerce based on sparse and noisy data". In: *Decision support systems* 106, pp. 53–63.

Daase, Christian et al. (2024). "On the Transition from Traditional Retail to Cloud-Supported E-Commerce: A Design Science Project". In: *International Conference on Enterprise Information Systems*. Springer, pp. 176–200.

Gayakwad, Milind (2024). "Real-Time Clickstream Analytics with Apache". In: *J. Electrical Systems* 20.2, pp. 1600–1608.

Huerta-Soto, Rosario et al. (2023). "Predictable inventory management within dairy supply chain operations". In: *International Journal of Retail & Distribution Management*.

Kharat, Prashant et al. (2023). "An inventory management using cloud unction and protocol buffer for improved efficiency". In: *8th International Conference on Computing in Engineering and Technology (ICCET 2023)*. Vol. 2023. IET, pp. 253–260.

Kotru, Arti and Isha Batra (2024). "Optimizing Resource Allocation in IoT for Improved Inventory Management". In: *International Journal of Computing and Digital Systems* 16.1, pp. 685–704.

Kumari, Archana et al. (2023). "Towards Practical, Serverless, Cost-effective, Real-time Pricing for Retail E-Commerce". In: *2023 4th International Conference on Communication, Computing and Industry 6.0 (C216)*. IEEE, pp. 1–6.

Kumari, Archana and Kumar S Mohan (2023). "A Cloud Native Framework for Real-time Pricing in e-Commerce". In: *International Journal of Advanced Computer Science and Applications* 14.4.

Li, Ji You et al. (2023). "Eigen: End-to-End Resource Optimization for Large-Scale Databases on the Cloud". In: *Proceedings of the VLDB Endowment* 16.12, pp. 3795–3807.

Muliarevych, Oleksandr (2023). "The Cloud-Based Optimization for Automated Warehouse Design". In: *2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*. Vol. 1. IEEE, pp. 380–384.

Pramodhini, R et al. (2023). "E-Commerce Inventory Management System Using Machine Learning Approach". In: *2023 International Conference on Data Science and Network Security (ICDSNS)*. IEEE, pp. 1–7.

Sharma, Pawankumar and S Panda (2023). "Cloud Computing for Supply Chain Management and Warehouse Automation: A Case Study of Azure Cloud". In: *Int. J. Smart Sens. Adhoc Network*, pp. 19–29.

Sinha, Gaurav Kumar (2023). "Leveraging Data Analytics and Transformer Neural Networks for Predictive Oil Price Forecasting". In: *Journal of Technological Innovations* 4.3.

— (2024). "AI-Driven Forecasting Models for Anticipating Oil Market Trends and Demand". In: *International Journal of Artificial Intelligence and Machine Learning* 6.6.

Tan, Yue et al. (2024). "Supply Chain Inventory Management from the Perspective of "Cloud Supply Chain"—A Data Driven Approach". In: *Mathematics* 12.4, p. 573.

Wang, Yufu et al. (2024). "The intelligent prediction and assessment of financial information risk in the cloud computing model". In: *arXiv preprint arXiv:2404.09322*.

Wilke, Carlie et al. (2023). "Implementation of radio-frequency identification technology to optimize medication inventory management in the intraoperative setting". In: *American Journal of Health-System Pharmacy* 80.6, pp. 384–389.

Yenugula, M, S Sahoo and S Goswami (2023). "Cloud computing in supply chain management: Exploring the relationship". In: *Management Science Letters* 13.3, pp. 193–210.