A

Mini Project Report

on

TELEMETRY DATA INTEGRATION AND ANALYSIS FOR GENERAL MOTORS

Submitted to

Jawaharlal Nehru Technological University, Hyderabad

For the partial fulfilment of requirements for the award of the degree in

BACHELOR OF TECHNOLOGY

in

DATA SCIENCE (MINOR DEGREE)

SUBMITTED BY

P. AMULYA	**(21271A0405)**
K. BHAVANI	**(21271A0407)**
K. DEEKSHITHA	**(21271A0410)**
K. HARINI	**(21271A0415)**
K. NEERAJA	**(21271A0427)**

Under the Esteemed guidance of

Dr. R. JEGADEESAN

Prof..HOD Dept. of CSE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING (AI&ML)
JYOTHISHMATHI INSTITUTE OF TECHNOLOGY & SCIENCE
(Autonomous, NBA (CSE, ECE, EEE) and NAAC 'A' Grade)

(Approved by AICTE, New Delhi, Affiliated to JNTUH, Hyderabad) Nustulapur,
Karimnagar 505481, Telangana, India

2024-2025

JYOTHISHMATHI
INSTITUTE OF TECHNOLOGY AND SCIENCE
AUTONOMOUS

NAAC 'A' GRADE

NATIONAL BOARD OF ACCREDITATION
CSE . ECE . EEE

CERTIFICATE

This is to certify that the Mini Project Report entitled “**TELEMENTRY DATA INTEGRATION AND ANALYSIS FOR GENERAL MOTORS**” is being submitted by P.AMULYA (21271A0405),K.BHAVANI (21271A0407),K.DEEKSHITHA (21271A0410), K.HARINI (21271A0415), K.NEERAJA (21271A0427) in partial fulfilment of the requirements for the award of the Degree of **Bachelor of Technology** in **Data Science (Minor degree)** to the **Jyothishmathi Institute of Technology & Science,** Karimnagar, during academic year 2024-2025, is a bonafide work carried out by them under my guidance and supervision.

The results presented in this Project Work have been verified and are found to be satisfactory results embodied in this Project Work have not been submitted to any other University for the award of any other degree or diploma.

Project Guide	Head of the Department
Dr. R. JEGADEESAN	**Dr. R. JEGADEESAN**
Professor & HOD	Professor & HOD
Dept. of CSE	Dept. of CSE

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our advisor, **Dr. R. JEGADEESAN**, whose knowledge and guidance has motivated us to achieve goals we never thought possible. The time we have spent working under her supervision has truly been a pleasure.

The experience from this kind of work is great and will be useful to us in future We thank **Dr. R. JEGADEESAN, Professor & HOD** CSE Dept for his effort, kind cooperation, guidance and encouraging us to do this work and also for providing the facilities to carry out this work.

It is a great pleasure to convey our thanks to **Dr. T. ANIL KUMAR**, Principal of Jyothishmathi Institute of Technology & Science and the College Management for permitting us to undertake this project and providing excellent facilities to carry out our project work.

We thank all the **Faculty** members of the Department of Computer Science & Engineering for sharing their valuable knowledge with us We extend out thanks to the **Technical Staff** of the department for their valuable suggestions to technical problems Finally Special thanks to our parents for their support and encouragement throughout our life and this course Thanks to all our friends and well wishers for their constant support.

JYOTHISHMATHI
INSTITUTE OF TECHNOLOGY AND SCIENCE
AUTONOMOUS

NAAC
NATIONAL ASSESSENT AND ACCREDITATION COUNCIL
'A' GRADE

NBA
NATIONAL BOARD OF ACCREDITATION
CSE . ECE . EEE

DECLARATION

We hereby declare that the work which is being presented in this dissertation entitled, "**TELEMENTRY DATA INTEGRATION AND ANALYSIS FOR GENERAL MOTORS"** submitted towards the partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology in Data Science (Minor degree) Jyothishmathi Institute of Technology & Science,** Karimnagar is an authentic record of our own work carried out under the supervision of **Dr. R. JEGADEESAN, Professor & HOD, Department of CSE,** Jyothishmathi Institute of Technology and Science, Karimnagar.

To the best of our knowledge and belief, this project bears no resemblance to any report submitted to JNTUH or any other University for the award of any degree.

P. AMULYA	**(21271A0405)**
K. BHAVANI	**(21271A0407)**
K. DEEKSHITHA	**(21271A0410)**
K. HARINI	**(21271A0415)**
K. NEERAJA	**(21271A0427)**

Date:

Place:
Karimnagar

ABSTRACT

In an era marked by technological advancements, General Motors (GM) embarks on a transformative project aimed at harnessing the power of Internet of Things (IoT) data from vehicles to revolutionize their analytics capabilities. This groundbreaking initiative, in collaboration with a third-party provider, introduces a cutting-edge telemetry system designed to capture a myriad of critical parameters from GM vehicles. These parameters include temperature, speed, interior and exterior conditions, and vehicle turbulence. The primary objective of this project is to seamlessly integrate and analyze the wealth of telemetry data generated by GM vehicles, utilizing the robust capabilities of cloud computing platforms. The project unfolds in three key phases, each addressing crucial aspects of data migration, validation, and storage in the Azure Cloud environment.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER-1
INTRODUCTION

1.1. INTRODUCTION

In the dynamic landscape of automotive technology, General Motors (GM) is embarking on a groundbreaking project that underscores their commitment to innovation and digital transformation. This transformative initiative, developed in collaboration with a strategic third-party partner specializing in Internet of Things (IoT) and cloud integration, introduces a **s**tate-of-the-art telemetry **system** engineered to capture, transmit, and analyze a comprehensive array of vital parameters from GM vehicles. These parameters include but are not limited to engine temperature, vehicle speed, acceleration, cabin climate, vibration and turbulence metrics, driver inputs, fuel efficiency, GPS location data, and energy consumption patterns in electric vehicles.

Together, these metrics form the bedrock of a robust and intelligent data **ecosystem**, capable of powering real-time analytics, proactive decision-making, and enhanced service offerings.

The overarching goal of this project is to seamlessly integrate**,** process, and analyze telemetry data across millions of GM vehicles by leveraging the elastic computing capabilities of Microsoft Azure’s cloud infrastructure**.** The project is methodically structured into three pivotal phases:

1. **Data Migration** – involving the secure, high-fidelity transition of telemetry data from the existing AWS infrastructure to Azure;
2. **Data Validation & Conformance** – encompassing rigorous data cleansing, transformation, and format standardization;
3. **Data Storage & Analytics Enablement** – which focuses on building scalable, structured databases and analytics pipelines within Azure SQL and associated services.

Each phase is meticulously designed to address critical facets of data integrity, interoperability, security, and scalability collectively aiming to elevate GM's analytical and operational capabilities to new heights**.**

1.2. MOTIVATION

The motivation behind initiating this transformative telemetry analytics project at General Motors (GM) stems from a deep-rooted commitment to digital excellence and an ambition to lead the next era of intelligent mobility. Several interrelated motivations drive this endeavor to harness the vast power of IoT data via advanced telemetry systems:

- **Operational Efficiency**: Centralizing and automating telemetry data analysis facilitates streamlined decision-making, reduces latency in issue detection, and optimizes manufacturing and service operations.
- **Customer-Centric Innovation**: Real-time vehicle insights enable GM to offer personalized driving experiences, adaptive infotainment systems, and intelligent driver-assist technologies.
- **Proactive Maintenance**: Leveraging predictive analytics based on telemetry trends helps GM to preemptively address component failures and schedule service interventions, thereby minimizing vehicle downtime.
- **Safety Advancements**: Analysis of behaviour patterns, sensor anomalies, and environmental conditions enhances the development of collision avoidance, adaptive cruise control**, and** driver fatigue monitoring systems**.**
- **Data-Driven Decision-Making**: Unified and cleansed telemetry data empowers cross-functional teams from engineering to marketing—with actionable insights for strategic planning.
- **Competitive Edge**: Adopting telemetry analytics places GM at the forefront of connected vehicle ecosystems, paving the way for autonomous vehicle innovations and smart infrastructure integration.
- **Economic and Environmental Impact**: Fine-tuning vehicle performance and reducing fuel inefficiencies contribute to **l**ower emissions and cost savings for both the company and its customers.
- **Technological Adaptability**: Embracing cutting-edge telemetry and cloud-native analytics fosters agility and responsiveness in adapting to emerging technologies, industry regulations, and mobility-as-a-service (MaaS) trends.

This initiative is not merely a data modernization effort—it is a **s**trategic leap toward a fully connected, intelligent, and sustainable automotive future**.**

1.3. PROBLEM STATEMENT

Despite the strategic benefits of harnessing telemetry data, General Motors (GM) encounters a spectrum of challenges in its endeavor to build a unified, scalable, and intelligent telemetry analytics infrastructure.

- **Cloud Migration Complexity**: The current telemetry ecosystem is hosted on Amazon Web Services (AWS)**.** Migrating vast volumes of historical and real-time data to Microsoft Azure poses risks related to data loss, latency, and compatibility. Ensuring zero downtime and data fidelity during this transition is a core concern.

- **Data Integrity and Validation**: Post-migration, the telemetry data must undergo stringent validation to ensure schema compliance, especially with the target JSON-based ingestion formats. Issues like null values, inconsistent structures, and malformed records could compromise analytics reliability and machine learning model accuracy.
- **Schema Design and Optimization**: Developing an optimized schema for telemetry data in Azure SQL Cloud Database is a non-trivial task. The database must support time-series data, vehicle-specific identifiers, event triggers, and query performance at scale. Special consideration must be given **to** partitioning strategies, indexing, and metadata tagging for efficient retrieval.
- **Real-Time Analytics Enablement**: The current system lacks the agility to process telemetry streams in real time. Building a pipeline using Azure Event Hubs, Stream Analytics, and Power BI dashboards is essential to support instantaneous insights, anomaly detection, and automated decision workflows.
- **Security and Compliance**: Telemetry data often includes sensitive operational and behavioral information. Ensuring end-to-end encryption, role-based access controls (RBAC), and compliance with global data protection laws such as GDPR and CCPA is imperative.
- **Scalability and Future-Proofing**: The system must scale seamlessly with increasing telemetry data volumes from future vehicle models, including electric and autonomous vehicles. It must be designed with modularity and containerized microservices to support evolving analytics requirements.
- **Integration with Legacy Systems**: GM’s enterprise architecture includes multiple legacy platforms and tools. The telemetry platform must support API-based integration and data interoperability with CRM, ERP, and Vehicle Health Monitoring Systems.
- **Insight Derivation and Utilization**: Perhaps the most significant challenge is the translation of raw telemetry data into contextual, predictive, and prescriptive insights. This involves implementing machine learning models, AI-driven diagnostics, and intelligent alerting mechanisms to enable data-driven optimization of vehicle design, performance, and user satisfaction.

CHAPTER-2

LITERATURE REVIEW

The implementation of telemetry-based IoT analytics in the automotive sector is an evolving domain, backed by extensive research and industrial innovation. Several scholarly studies, industry white papers, and technological assessments provide critical insights into the use of telemetry data for enhancing operational intelligence, predictive maintenance, and connected vehicle ecosystems.

Telemetry Systems in Connected Vehicles

Telemetry systems are a fundamental component of modern **c**onnected vehicle architectures**,** as noted in the work of Zeadally et al. (2012)**,** who discuss the integration of vehicular networks for real-time data exchange. Their research underscores the role of Vehicle-to-Infrastructure (V2I) and Vehicle-to-Cloud (V2C) communication in achieving dynamic traffic management, emissions control, and autonomous navigation.

Further advancements in telemetry have been explored by Ghosh et al. (2018)**,** who emphasize the significance of cloud-connected vehicular telemetry in facilitating driver behavior analysis, accident reconstruction, and fleet management optimization. These studies support the need for real-time data ingestion and analytics pipelines that are both robust and scalable.

Cloud-Based Telemetry Data Management

The migration and management of telemetry data in cloud environments such **as** Microsoft Azure are well-documented in technical literature. Zhang et al. (2017) describe the challenges of data schema heterogeneity and propose methods for automated schema mapping and transformation using cloud-native tools. They advocate for the adoption of Azure Data Factory, Azure Data Lake, and Stream Analytics to handle the scale and velocity of IoT data streams.

Microsoft's own whitepaper, *"Designing Scalable IoT Solutions on Azure"* (2020), outlines architectural best practices for telemetry ingestion using Azure IoT Hub**,** Event Hubs, and Time Series Insights. This architecture is designed to meet enterprise-grade SLAs for latency, throughput, and data governance—aligning closely with GM's telemetry analytics goals.

Predictive Maintenance and Machine Learning Models

A seminal paper by Susto et al. (2015) delves into predictive maintenance using machine learning models trained on telemetry data. Their findings indicate that techniques such as Random Forests, Support Vector Machines, and Recurrent Neural Networks (RNNs) can forecast equipment failures and performance degradation with high accuracy. These methodologies are directly applicable to GM's use case for minimizing vehicle downtime and improving service scheduling.

Additionally, Abdulshaheed et al. (2019) demonstrate the use of sensor fusion and deep learning in real-time monitoring of vehicle health. Their research highlights the importance of clean, labeled, and well-structured data**,** reinforcing the necessity of rigorous validation and conformance protocols during the migration phase of GM's project.

Security and Data Governance

Security remains a cornerstone of any telemetry system. Papadimitratos et al. (2008) outline the threats associated with vehicular data transmission and propose encryption and authentication frameworks for protecting vehicular communication networks. More recent work by Zhou et al. (2020) investigates the use of blockchain and zero-trust architecture in securing cloud-based automotive systems.

From an enterprise compliance perspective**,** data protection laws **such as** GDPR (General Data Protection Regulation) and CCPA (California Consumer Privacy Act) **are extensively studied. Articles from journals such as *IEEE Access* and *ACM Computing Surveys* consistently recommend the integration of** privacy-preserving data mining (PPDM) **techniques and** fine-grained access control mechanisms **both of which are critical for GM's Azure-based telemetry framework.**

Scalability and Performance

A key consideration in literature is the **scalability of telemetry platforms**. Research by **Xu et al. (2019)** evaluates distributed processing systems like **Apache Kafka**, **Apache Spark**, and **Azure Synapse Analytics**, identifying patterns for handling petabyte-scale streaming data. Their findings justify the modular design of telemetry systems with decoupled ingestion, processing, and storage layers to support elastic scaling.

The use of serverless computing**,** particularly Azure Functions and Azure Logic Apps, is also advocated in cloud architecture reviews for reducing operational overhead and improving response times in event-driven systems.

CHAPTER-3

EXISTING & PROPOSED SYSTEM

3.1 EXISTING SYSTEM

The current telemetry data management system at General Motors (GM) faces several limitations and structural inefficiencies that hinder the organization's ability to fully harness the potential of its vast vehicle-generated data. These issues impact data quality, analytical insight, operational speed, and compliance with security and governance standards.

Limited or No Automated Data Validation:

The existing system lacks a standardized, automated pipeline for validating incoming telemetry data. This leads to inconsistent data formats, missing values, and erroneous readings. Manual or ad hoc validation processes are error-prone and fail to scale with the rapidly increasing data volume generated by connected vehicles.

No Structured Analytics or Relational Storage Tailored for Querying Large Volumes of telemetrydata:

Telemetry data is stored in loosely structured formats, making it difficult to perform optimized, high-performance queries. This prevents analysts and data scientists from efficiently extracting insights or correlating multiple telemetry dimensions for advanced modeling.

No Integrated Real-Time ETL (Extract, Transform, Load) Pipeline:

The current system does not support real-time ETL operations. As a result, data ingestion and transformation are delayed, and real-time analytics or near-real-time alerting are not feasible. This severely limits GM's ability to react swiftly to issues such as mechanical anomalies or safety concerns.

Unclear or Unsecured Handling of Sensitive Data:

Existing data management processes lack clearly defined protocols for securing telemetry data during transmission, storage, and access. This poses a risk of data breaches or non-compliance with privacy regulations such as GDPR and CCPA.

Analytics Not Centralized or Fully Operationalized:

Analytical processes across GM's systems are fragmented, often performed in silos using different tools and data sources. There is no centralized data lake or dashboard environment where telemetry data is harmonized and visualized for enterprise-wide use. This restricts collaboration and slows down decision-making.

Problem statement for existing system:

1. **Data Privacy Issues**: Unauthorized data collection and sharing without explicit user consent has led to regulatory actions and loss of trust.
2. **Legacy System Integration**: Disconnected OT and IT systems hinder real-time data processing and seamless analytics.
3. **Poor Data Quality**: Inaccurate, incomplete, and inconsistent telemetry data reduces the effectiveness of analysis.
4. **Lack of Standardization**: Inconsistent data formats and sources complicate integration and processing.
5. **Resistance to Technology Adoption**: Internal reluctance to adopt new solutions slows modernization efforts.
6. **Scalability Challenges**: Existing infrastructure struggles to handle large volumes of real-time vehicle data efficiently.

3.2. PROPOSED SYSTEM

To overcome the limitations of the existing system, the proposed architecture introduces a robust, scalable, and cloud-native telemetry analytics platform built on Microsoft Azure. This system is designed to support end-to-end telemetry processing, from ingestion and validation to secure storage, advanced analytics, and real-time operational integration.

Automated and Secure Data Validation Pipeline:

The proposed system integrates Azure Data Factory, Azure Functions, and Data Bricks-based services to automatically validate incoming telemetry data. It performs format checks, schema conformity verification, null-value handling, and error logging. This ensures that only high-quality, structured data proceeds through the analytics pipeline.

Structured and Query-Optimized Storage in Azure SQL:

A relational data model specifically designed for telemetry data is implemented in Azure SQL Database. Tables are partitioned and indexed to support efficient queries across time-series data, vehicle IDs, and sensor types. This enables rapid data exploration and supports integration with tools like Power BI, Synapse Analytics, and machine learning platforms.

Use of Modern Azure Services (Function Apps, Data Factory, Key Vault):

The system is built using a microservices architecture, with Azure Function Apps handling event-driven tasks, Azure Data Factory orchestrating data flows, and Azure Key Vault managing secrets and credentials securely. This architecture promotes automation, modularity, and compliance with security best practices.

Better Scalability, Monitoring, and Integration with Existing GM Analytics Tools:

The proposed infrastructure is inherently scalable using Azure's elastic compute model, ensuring high availability and performance even with growing telemetry volumes. Azure Monitor, Application Insights, and Log Analytics are integrated for real-time observability, diagnostics, and performance tracking. Additionally, APIs are built to allow seamless integration with GM's existing analytics dashboards, BI tools, and vehicle diagnostics systems.

Foundation for Future Enhancements:

The system is designed to be future-proof, serving as a foundation for next-generation features such as:

- Real-time analytics pipelines using Azure Stream Analytics or Kafka-based ingestion.
- Advanced visualizations and dashboards using Power BI Embedded.
- Predictive and prescriptive maintenance models using Azure Machine Learning.
- Multi-source data integration, combining telemetry with CRM, ERP, and manufacturing systems for enterprise-level intelligence.
- Support for edge computing and vehicle-side analytics for latency-sensitive applications.

This proposed system not only resolves the shortcomings of the existing architecture but also empowers GM to **transition into a data-first, insight-driven organization**. By leveraging modern cloud technologies and adhering to data governance best practices, GM will be well-positioned to **lead the industry in connected mobility, predictive diagnostics, and customer-centric vehicle innovation**.

Advantages of Proposed System:

1. **Enhanced Data Accuracy and Standardization**: Unified data formats and improved validation ensure more reliable and consistent analytics across all systems.

2. **Real-Time Insights**: Advanced analytics and cloud-based processing enable faster decision-making for vehicle diagnostics, maintenance, and performance optimization.
3. **Improved System Integration**: Seamless integration of OT and IT systems eliminates data silos, allowing for a more connected and automated workflow.
4. **Scalable Infrastructure**: The new system is designed to efficiently manage and process high volumes of data, supporting GM's growing vehicle fleets and future expansion.
5. **Better Data Privacy and Compliance**: Built-in privacy controls and transparent user consent mechanisms help GM comply with regulations and rebuild customer trust.
6. **Operational Efficiency and Cost Savings**: Predictive maintenance, optimized fleet management, and reduced downtime lead to lower operational costs and improved productivity.

CHAPTER-4

4.1 MACHINE LEARNING APPLICATIONS FOR TELEMETRY DATA

As General Motors (GM) continues to innovate with connected vehicle ecosystems, the integration of machine learning (ML) models into telemetry data pipelines becomes increasingly essential for unlocking predictive and prescriptive insights. One of the most impactful applications of ML is **predictive maintenance**, where models trained on historical sensor data can forecast potential component failures before they occur. For example, by analyzing trends in temperature fluctuations, vibration intensity, or braking patterns, ML algorithms can identify early signs of engine wear or battery degradation, thereby minimizing unexpected breakdowns and extending vehicle lifespan.

Another significant application is the **classification of driver behavior**, which is particularly relevant for fleet management and insurance customization. ML models can process telemetry features like acceleration, deceleration, cornering, and speed patterns to categorize driving styles into profiles such as aggressive, cautious, or distracted. These classifications can be used to tailor user feedback, enhance safety training, or provide insurance companies with risk-adjusted driving scores.

In addition, telemetry data enables **intelligent maintenance scheduling** through ML-driven recommendations. By analyzing usage patterns, terrain data, and historical service records, algorithms can dynamically propose optimized maintenance windows that align with actual wear-and-tear instead of rigid time-based schedules. This results in more efficient resource utilization and an improved customer experience.

Platforms such as **Azure Machine Learning** and **Azure Databricks** offer robust support for building and deploying these models at scale. Azure ML provides managed environments for training and deploying models, versioning, and automating retraining workflows, while Databricks enables distributed data processing and model development using Spark-based frameworks like MLlib, TensorFlow, or Scikit-learn. Together, these tools empower GM to transform raw telemetry data into actionable insights, paving the way for smarter, safer, and more adaptive vehicles.

4.2 DATA GOVERNANCE AND COMPLIANCE

In the era of connected vehicles and large-scale telemetry collection, **data governance and regulatory compliance** play a critical role in ensuring that General Motors (GM) manages its vast telemetry datasets responsibly and ethically. To meet global data privacy laws such as the **General**

Data Protection Regulation (GDPR) in Europe and the **California Consumer Privacy Act (CCPA)** in the United States, GM has adopted stringent data governance frameworks that prioritize **transparency, data minimization, and consent-based processing**. Personally identifiable information (PII) is either pseudonymized or anonymized before analytics, ensuring that data subjects' rights are preserved throughout the telemetry pipeline.

GM enforces a robust **data lifecycle management** policy to control the retention, archival, and deletion of telemetry data. This policy defines retention periods for different categories of data such as critical diagnostic logs, sensor readings, or driver interaction records ensuring data is not stored longer than necessary. Automated **data expiration and purging mechanisms** are implemented through Azure Data Factory and Azure SQL stored procedures to align with legal and operational requirements, thereby optimizing storage costs while remaining compliant with retention mandates.

To ensure accountability and operational transparency, GM leverages **Azure Purview** for data cataloging and **compliance auditing**, enabling fine-grained visibility into data lineage, sensitivity classification, and access histories. Purview automates the discovery and classification of telemetry data assets, allowing data stewards to monitor usage and enforce data handling policies. Additionally, **Azure Sentinel**, a cloud-native Security Information and Event Management (SIEM) solution, is used to implement real-time **security monitoring and threat detection** across the telemetry data infrastructure. Sentinel helps in identifying anomalous behavior, unauthorized access attempts, or policy violations, supporting both incident response and compliance reporting.

Together, these governance mechanisms ensure that GM's telemetry data infrastructure is **secure, auditable, privacy-compliant, and aligned with international standards**, thereby fostering trust among stakeholders and positioning GM as a responsible innovator in the automotive IoT domain.

4.3 DIGITAL TWIN INTEGRATION

One of the most forward-looking applications of telemetry data in the automotive domain is the creation of **digital twins** virtual replicas of physical vehicles that mirror their real-time behavior and condition. At General Motors (GM), telemetry data serves as the foundational input to develop dynamic digital twins of vehicles, enabling real-time monitoring, diagnostics, simulation, and predictive analysis. By continuously streaming parameters such as engine temperature, battery voltage, tire pressure, vehicle speed, and component stress levels, a digital twin can reflect the exact state of a physical vehicle at any given moment.

Microsoft's **Azure Digital Twins** service offers a scalable and powerful platform to model and simulate these virtual representations. Using the **Digital Twins Definition Language (DTDL)**, GM can model vehicle components, their relationships, and their telemetry-driven state changes within a graph-based digital environment. For example, each vehicle’s powertrain, suspension system, and onboard computer can be modeled as interconnected entities, responding to real-world telemetry feeds in real time.

This integration enables **"what-if" simulations**, where engineers can model how a vehicle might respond under various conditions such as extreme temperatures, rough terrains, or component failures without needing to physically test every scenario. It also empowers **predictive diagnostics** by identifying early warning signals in the digital twin before an issue manifests in the physical vehicle. Additionally, insights from digital twins can be fed back into product design, allowing iterative enhancements based on simulated performance data.

Moreover, when combined with other Azure services like **Azure IOT Hub**, **Time Series Insights**, and **Machine Learning**, Azure Digital Twins allows GM to build a fully intelligent system where **real-time data, historical trends, and predictive analytics converge**. This ecosystem not only supports operations and R&D but also opens doors for advanced customer experiences such as personalized maintenance alerts or in-app driving tips based on the digital twin’s analysis.

In essence, the implementation of digital twin technology through Azure Digital Twins represents a transformative leap for GM in operational intelligence, vehicle innovation, and predictive maintenance, aligning with the broader goals of smart mobility and Industry 4.0.

4.4 EDGE COMPUTING CONSIDERATIONS

As the volume and velocity of telemetry data from connected vehicles continue to grow, General Motors (GM) must carefully evaluate the balance between **edge computing and centralized cloud processing**. While cloud platforms like Azure offer virtually unlimited scalability, advanced analytics, and centralized data management, they can introduce latency and bandwidth constraints especially for vehicles operating in areas with intermittent or limited connectivity. In such contexts, **edge computing emerges as a strategic solution** to complement cloud infrastructure by enabling local, low-latency data processing directly on or near the vehicle.

The **Azure IoT Edge** platform is particularly suited for this purpose, allowing GM to deploy AI models, business logic, and filtering mechanisms directly onto onboard computing devices within vehicles or edge gateways. By leveraging IoT Edge, telemetry data such as GPS coordinates, engine diagnostics, or brake status can be **pre-processed locally** to perform tasks such as anomaly detection, real-time threshold monitoring, or even triggering local alerts, without needing to transmit every data point to the cloud. Only relevant, aggregated, or exception-based data is forwarded to the Azure cloud, optimizing both **network efficiency and cloud storage costs**.

This hybrid architecture provides significant **trade-offs**: cloud computing excels at large-scale historical analysis, machine learning model training, and centralized visualization, while edge computing enables **real-time responsiveness, reduced latency, and offline resilience**. For example, a vehicle could detect overheating in real-time and trigger immediate mitigation actions using edge logic, while detailed post-event analytics are later conducted in Azure.

In future implementations, GM could also utilize Azure IoT Edge modules to run **containerized machine learning models** trained in Azure Machine Learning and deployed at the edge for dynamic updates. This allows for **continuous learning and adaptation** as models are periodically refined in the cloud and redistributed to edge devices. Overall, the integration of Azure IoT Edge empowers GM to achieve an **intelligent, distributed computing architecture** that enhances vehicle autonomy, operational efficiency, and system scalability in a cost-effective and resilient manner.

4.5 REAL TIME DASHBOARDING AND VISUALIZATION

An essential phase of any telemetry data pipeline is the **visualization of insights** in a form that is accessible, actionable, and tailored to various stakeholders. At General Motors (GM), the telemetry analytics platform culminates in robust real-time dashboards that translate complex vehicle data into intuitive visual formats. These dashboards enable data-driven decision-making across multiple departments, including engineering, technical support, product development, and executive leadership.

One of the primary tools employed for data visualization is **Microsoft Power BI**, a powerful business intelligence platform that seamlessly integrates with **Azure SQL Database**, **Azure Data Lake Gen2**, and **Azure Synapse Analytics**. Power BI enables the creation of dynamic, interactive dashboards that allow users to drill down into specific telemetry metrics such as vehicle speed trends, battery health over time, regional performance comparisons, and incident heat maps. Engineers can use these dashboards to identify performance bottlenecks, detect outliers in component behavior, and analyze historical sensor data for product refinement.

In parallel, **Azure Synapse Analytics** plays a critical role in aggregating, transforming, and querying massive datasets at scale. As a unified analytics platform, Synapse can ingest and process telemetry data from Azure Data Lake Gen2, run complex SQL and Spark queries, and feed pre-processed results directly into Power BI dashboards. For example, Synapse pipelines can perform batch processing to calculate average fuel efficiency across different regions or generate anomaly scores based on predictive models. These outputs are then visualized in near real-time using Power BI, providing insights that are both deep and responsive.

From a **technical support perspective**, real-time dashboards allow teams to **monitor the health and behavior of vehicles in the field**, enabling proactive intervention in the event of performance anomalies or sensor failures. Alerts can be triggered when critical thresholds are breached such as engine temperature spikes or repeated error codes allowing the support team to escalate issues promptly and dispatch appropriate resources.

For **executive stakeholders**, high-level summary dashboards provide **KPI-driven insights** such as fleet-level uptime, predictive maintenance impact, cost savings, and safety improvement trends. These dashboards often include **custom reports**, interactive charts, and time-series visualizations that help executives assess the ROI of the telemetry integration initiative and identify strategic opportunities for innovation or expansion.

By combining the advanced processing capabilities of Azure Synapse with the user-friendly and real-time visualization power of Power BI, GM is able to **close the loop between raw telemetry data and impactful business decisions**. This visualization layer ensures that telemetry insights are not confined to data scientists or engineers but are democratized across the organization, fostering a culture of evidence-based decision-making and continuous improvement.

CHAPTER-5
OVERVIEW OF TECHNOLOGIES

5.1 AMAZON S3

Amazon S3 (Simple Storage Service) is a highly scalable, durable, and secure object storage service provided by Amazon Web Services (AWS). Designed to store and retrieve any amount of data from anywhere over the internet, S3 serves as a foundational component for a wide range of cloud-native applications, including data lakes, backup and recovery, archiving, analytics, and content distribution.

In the context of General Motors' telemetry data architecture, Amazon S3 functions as the initial ingestion and storage layer, collecting raw telemetry data streamed from connected vehicles. This data, typically in JSON format, includes various real-time parameters such as vehicle speed, temperature, engine diagnostics, and sensor statuses. The storage structure in S3 is organized hierarchically often by year, month, and day to facilitate efficient data partitioning, retrieval, and lifecycle management.

S3's object-based storage model is particularly well-suited for IoT telemetry data because it supports virtually infinite scalability and allows metadata tagging for categorization and quick searchability. Each object is stored in a "bucket" and can be uniquely identified by a key, enabling systematic management of large volumes of structured and unstructured data.

A key feature of Amazon S3 is its integration with AWS Identity and Access Management (IAM) and bucket policies, which provide fine-grained control over access to stored objects, ensuring secure storage. S3 also supports versioning, encryption (at rest and in transit), cross-region replication, and lifecycle policies, making it a robust solution for mission-critical data retention and disaster recovery strategies.

Moreover, S3 integrates seamlessly with other AWS services such as AWS Lambda, Amazon Athena, and AWS Glue, supporting serverless processing, querying, and ETL operations directly over the stored data. In GM's telemetry pipeline, S3 acts as the source system, from which Azure Data Factory later extracts data to initiate migration into Azure-based services for further validation and analytics.

5.2 DATA LAKE GEN 2

Azure Data Lake Storage Gen2 is Microsoft Azure’s enterprise-grade data lake solution built specifically to address the needs of big data analytics by combining the scalability and low-cost capabilities of Azure Blob Storage with the advanced features of Azure Data Lake Storage Gen1, such as hierarchical namespaces and fine-grained access control.

Designed to handle petabyte-scale volumes of structured, semi-structured, and unstructured data, Data Lake Gen2 is ideal for storing telemetry data generated from IoT-enabled vehicles in a centralized, accessible, and secure manner. It supports a hierarchical file system that enables directory-based organization, allowing General Motors (GM) to logically arrange telemetry files based on parameters like date, vehicle ID, or event type. This not only improves manageability but also boosts performance for file operations such as listing directories and accessing nested files.

In GM’s telemetry pipeline, Data Lake Gen2 serves as the central landing and staging area once the raw JSON data is migrated from AWS S3 using Azure Data Factory. Upon landing in the designated folder, data is then validated using Azure Function Apps and routed to appropriate subdirectories (e.g., "staging", "rejected", "processed") for further processing or analysis. This tiered structure supports modular ETL pipelines and simplifies downstream analytics.

One of the key benefits of Data Lake Gen2 is native integration with Azure services such as Azure Synapse Analytics, Azure Databricks, Azure HDInsight, and Power BI, making it an ideal foundation for building a full-stack analytics solution. Additionally, it supports POSIX-compliant access control lists (ACLs), enabling secure and role-based data access, which is essential for enforcing data governance in compliance with enterprise security standards and regulatory frameworks like GDPR.

Furthermore, Data Lake Gen2 supports high-throughput data ingestion, parallel processing, and batch or real-time analytics, making it highly scalable for GM’s future needs as telemetry data volumes grow. Features like lifecycle management policies, encryption at rest, and integration with Azure Key Vault for key management enhance the platform’s reliability, security, and cost-effectiveness.

5.3 AZURE DATA FACTORY

Azure Data Factory (ADF) is a powerful cloud-based data integration service offered by Microsoft Azure. It is designed to enable users to seamlessly create, schedule, and orchestrate data workflows or pipelines that can move and transform data across a wide range of on-premises and cloud-based data sources. ADF supports both structured and unstructured data and can connect with various storage solutions such as Azure Blob Storage, Azure SQL Database, Azure Data Lake, and even non-Microsoft platforms like Amazon S3 or Google Cloud Storage. One of its key strengths is the ability to perform Extract, Transform, and Load (ETL) operations, which are crucial for preparing data for analytics and business intelligence. Users can leverage a visual interface to design complex workflows or write custom logic using data flows and activities. Additionally, Azure Data Factory integrates well with other Azure services such as Azure Synapse Analytics, Azure Databricks, and Azure Machine Learning, making it an essential tool for building scalable, automated, and enterprise-grade data integration solutions. With features like triggers for scheduling, monitoring capabilities, and data lineage tracking, ADF simplifies the management of big data and hybrid data movement scenarios, ensuring reliability, security, and compliance in data processing pipelines.

5.4 AZURE KEY VAULT

Azure Key Vault is a cloud-based service offered by Microsoft Azure that is designed to securely store and manage sensitive information such as secrets, encryption keys, passwords, API keys, and digital certificates. It plays a critical role in enhancing the security and compliance of cloud applications by providing a centralized and secure location for managing cryptographic keys and secrets. Azure Key Vault enables developers and IT administrators to control access to sensitive data using fine-grained access policies and integrates seamlessly with Azure Active Directory for authentication and authorization. It supports hardware security modules (HSMs) for added protection of cryptographic keys, ensuring that data is encrypted both at rest and in transit. By isolating sensitive data from application code, Azure Key Vault reduces the risk of accidental exposure or unauthorized access. Additionally, it offers capabilities such as automated key rotation, secure backup and recovery, audit logging, and integration with other Azure services like Azure Functions, Azure App Service, and Azure Kubernetes Service (AKS). These features make it easier to manage secrets in DevOps pipelines, secure application configurations, and comply with industry regulations and organizational security standards.

5.5 AZURE FUNCTION APP

Azure Function App is a serverless compute service provided by Microsoft Azure that enables developers to build and deploy lightweight, event-driven functions without the need to manage underlying infrastructure. This service abstracts away the complexities of server provisioning and scaling, allowing developers to focus solely on writing code that responds to events such as HTTP requests, database changes, message queue activity, scheduled tasks, and more. Azure Function Apps support a wide range of programming languages, including JavaScript, Python, and PowerShell, offering flexibility and ease of use across different development environments.

One of the key advantages of Azure Function Apps is their scalability functions automatically scale based on the workload, ensuring optimal performance and cost-efficiency. This makes them an ideal choice for building microservices, processing data streams, integrating with other Azure and third-party services, and automating repetitive or time-sensitive tasks. Developers can also set up durable functions to manage long-running processes and workflows. Additionally, Azure Function Apps integrate seamlessly with Azure Logic Apps, Azure Event Grid, Azure Service Bus, and other Azure services, enabling robust and scalable application architectures. With built-in support for monitoring, logging, and continuous deployment, Azure Function App simplifies the development lifecycle and accelerates time to market for cloud-native applications.

5.6 AZURE SQL DATABASE SERVER

Azure SQL Database Server is a fully managed relational database platform offered by Microsoft Azure, built on the robust SQL Server engine. It is specifically designed to support the development and deployment of modern, scalable, and secure cloud-based applications without the overhead of managing physical hardware, software updates, or database maintenance tasks. As a Platform as a Service (PaaS), Azure SQL Database Server handles critical operational aspects such as provisioning, patching, backups, and performance tuning automatically, allowing developers and database administrators to focus more on application development and less on infrastructure management.

This service offers high availability and disaster recovery out of the box through features like geo-replication and automated failover, ensuring business continuity even in the face of unexpected outages. Additionally, it provides advanced security capabilities such as data encryption at rest and in transit, threat detection, and integration with Azure Active Directory for access control and identity management.

Performance can be optimized through intelligent tuning, indexing recommendations, and built-in machine learning features that adapt to application usage patterns. Azure SQL Database Server also supports elastic pools, enabling the management of multiple databases with shared resources, which is ideal for SaaS applications with variable workloads. Furthermore, it allows seamless integration with other Azure services like Azure Data Factory, Power BI, and Azure Logic Apps, making it a key component in building end-to-end data-driven applications and enterprise solutions.

The technologies mentioned form a robust and integrated ecosystem that supports comprehensive data storage, integration, and processing capabilities across both AWS and Azure cloud platforms. At the foundation of this ecosystem are **Amazon S3** and **Azure Data Lake Storage Gen2**, which offer highly scalable, durable, and secure storage solutions for handling vast amounts of structured and unstructured data. These storage systems are designed to accommodate a wide range of data types and formats, making them ideal for data ingestion, archival, and analytics.

To enable seamless movement and transformation of data across these storage environments, **Azure Data Factory** plays a pivotal role. It serves as a powerful cloud-based ETL (Extract, Transform, Load) and data integration service, allowing organizations to create, schedule, and manage complex data workflows. With support for numerous data sources, both cloud-based and on-premises, Azure Data Factory ensures that data can flow efficiently between systems such as Amazon S3, Azure Data Lake, and Azure SQL Database.

Security is a critical aspect of any data ecosystem, and **Azure Key Vault** addresses this need by providing a centralized and secure mechanism for managing sensitive information such as API keys, secrets, certificates, and encryption keys. This ensures that credentials and confidential configurations are protected throughout the data lifecycle. In addition, **Azure Function App** enhances the architecture by offering serverless computing capabilities, enabling users to run lightweight, event-triggered functions without the need to provision or manage infrastructure.

Finally, **Azure SQL Database** provides a fully managed, intelligent relational database service that is optimized for high performance, scalability, and reliability. It is well-suited for storing processed and structured data that supports downstream analytics, business applications, and reporting tools. By leveraging these technologies together, organizations can build highly scalable, secure, and automated cloud-based data solutions that are capable of supporting modern data engineering, analytics, and application development needs across multi-cloud environments.

CHAPTER-6

HIGH LEVEL DESIGN AND PIPELINE

6.1 HIGH LEVEL DESIGN

Telemetry Data Integration and Analysis for General Motors (GM) involves building a scalable, secure, and intelligent data pipeline that enables ingestion, processing, storage, analysis, and visualization of vehicle telemetry data. Here's a structured high-level architecture. The following diagram illustrates high level design implementation of our project.

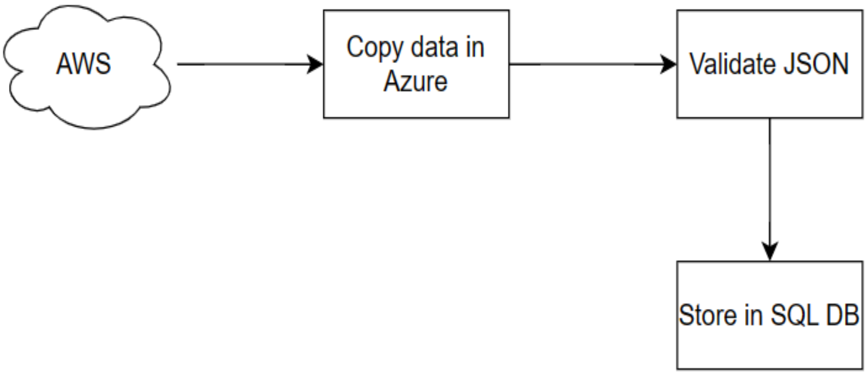


Figure 6.1: Cloud data ingestion and processing pipeline

The high-level design for telemetry data integration and analysis at General Motors (GM) involves building a secure, scalable, and intelligent data platform that enables the collection, processing, analysis, and visualization of vehicle-generated telemetry data. This system begins at the vehicle level, where data is generated by onboard sensors, control units (ECUs), and connected systems such as infotainment units and mobile applications. These data points include metrics like GPS location, engine status, speed, braking patterns, battery health, and diagnostic trouble codes. Once collected, the data is transmitted securely through in-vehicle edge gateways or directly via mobile networks to a cloud-based ingestion layer using protocols such as MQTT or HTTP, managed by an IoT gateway platform like AWS IoT Core or Azure IoT Hub. This ingestion layer uses message brokers such as Apache Kafka or AWS Kinesis to handle high-throughput, real-time data streams, enabling efficient routing and buffering.

Once ingested, telemetry data is stored in a centralized data lake using scalable cloud storage solutions such as Amazon S3 or Azure Data Lake, often organized in cost-effective tiers (hot, warm, cold) based on usage patterns. Data is stored in optimized formats like Parquet or Avro, and governed using a metadata catalog for discoverability. The data processing layer includes both real-time and batch processing capabilities using tools like Apache Flink and Spark, allowing GM to perform real-time anomaly detection (e.g., geofence breaches or engine faults) as well as long-term trend analysis (e.g., fuel efficiency over time or fleet performance). Cleaned and enriched data is then passed to an analytics layer where machine learning models are developed, trained, and deployed using frameworks like TensorFlow, Scikit-learn, or PyTorch. These models support use cases such as predictive maintenance, driver behavior profiling, and route optimization.

Insights generated from the data are shared with stakeholders via business intelligence tools like Power BI, Grafana, or Tableau, through intuitive dashboards and alerts. To ensure compliance with data privacy regulations (such as GDPR and CCPA), the system includes features like data anonymization, access controls, and audit logging. The entire architecture is built on a microservices-based foundation to ensure modularity and scalability, supported by CI/CD pipelines for efficient updates and integrated monitoring tools for system reliability. This holistic architecture empowers GM to leverage telemetry data for smarter decision-making, enhanced vehicle performance, and improved customer experiences.

6.2 PIPELINE

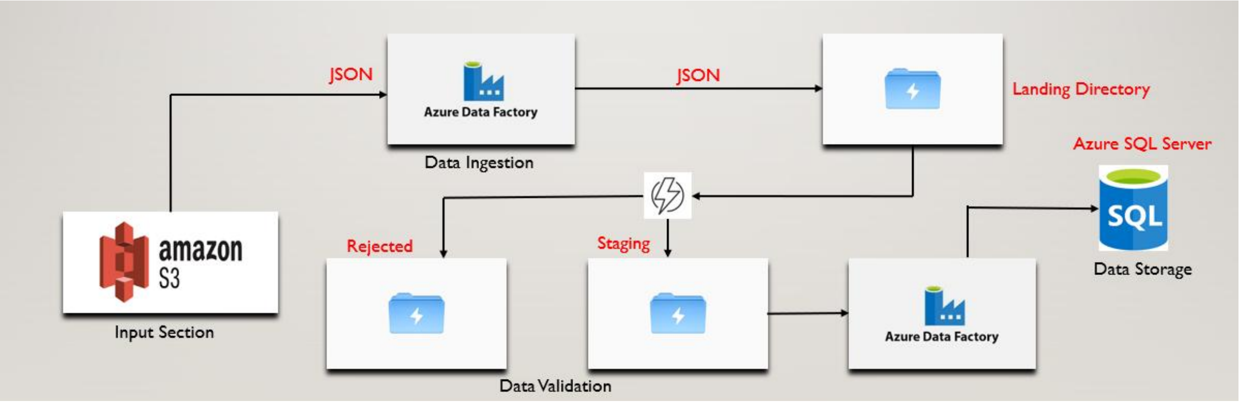


Figure 6.2 : Azure Data Pipeline

The architecture for seamlessly transferring telemetry data from AWS S3 storage to Azure Cloud involves a structured and efficient flow to ensure data integrity and validation. The process begins with the input source being the AWS S3 storage bucket where telemetry data in JSON format is stored, originating from IoT devices.

i. AWS S3 Bucket Setup: Before initiating the data transfer, we create an S3 bucket in the AWS account. Where IOT data of Vehicles is got loaded into on a day-to-day basis into the S3 bucket with a hierarchical structure (S3 bucket > Year > Month > Day), mimicking the organization of telemetry data from connected vehicles.

ii. Data Migration with Azure Data Factory: The first step is to move data from the AWS S3 storage bucket to Azure Cloud. Azure Data Factory, a robust data integration service, is employed for this task. The telemetry data is transferred to the landing folder in Azure, marking the initial point of data reception.

iii. Azure Data Lake Storage (ADLS) Setup: We create a Azure Data Lake Storage (ADLS) account. Within the ADLS account, a landing folder is established, serving as the starting point for telemetry data within the Azure environment.

iv. Data Validation using Azure Functions: Upon arrival in the landing folder, the telemetry data undergoes validation for JSON format compliance. Azure Functions, configured with a storage-based trigger, automatically verify whether the incoming data contains valid JSON. If successful, the data is moved to the staging folder; otherwise, it is redirected to the rejected folder for tracking failed instances.

v. Moving Validated Data to Azure SQL Database: The validated telemetry data in the staging folder is then moved to an Azure SQL Database. Again, Azure Data Factory is leveraged to ensure a seamless transfer of data from files to the SQL database. This step marks the completion of the data journey, providing a centralized and structured storage solution for further analytics.

Implementation With Respect To Technology

The project pipeline can be divided into three main stages: Ingestion, ETL (Extract, Transform, Load), and Visualization. Each stage plays a crucial role in processing telemetry data from connected vehicles and deriving meaningful insights. Here's an overview of how the pipeline flows through these stages:

1.Ingestion:

Source Ingestion to Amazon S3: Telemetry data from connected vehicles is ingested from Amazon S3, where it is initially stored. Before initiating the data transfer, we create an S3 bucket in the AWS account. Where IOT data of Vehicles is got loaded into on a day-to-day basis into the S3 bucket with a hierarchical structure (S3 bucket > Year > Month > Day), mimicking the organization of telemetry data from connected vehicles.

Azure Data Factory (ADF) Ingestion: Azure Data Factory is employed to seamlessly transfer telemetry data from Amazon S3 to Azure Data Lake Storage Gen2. This marks the initial step in the Azure environment, utilizing ADF for efficient data movement. For this we have created a new Azure Data Factory and define data movement and transformation workflows within the ADF. Then we proceeded with Linked Services Implementation and created a Linked Service for AWS S3 in ADF, providing necessary credentials to connect to the "iot-raw-data" bucket that is available in S3 Bucket. Then we created another Linked Service for Azure SQL Database, supplying connection details.

- Datasets Implementation: Then we defined a couple of datasets in ADF for both the source (AWS S3) and destination (Azure SQL Database) to store the data that is in staging folder. Configured dataset properties, including file format, column mapping, and partitioning.
- Pipelines Implementation: Develop pipelines within ADF to orchestrate data movement. We utilized activities like Copy Data for efficient movement between S3, landing, staging, and Azure SQL Database. Triggers Implementation: Established triggers within ADF to automate pipeline execution. Set up these triggers to a schedule for daily pipeline runs to ensure synchronized data movement.

1.ETL (Extract, Transform, Load):

JSON Data Validation with Azure Function App: Azure Blob trigger function is triggered upon the arrival of data in the landing folder, validates the JSON format of incoming files. If validation is successful, the data is moved to the staging folder; otherwise, it is directed to the rejected folder. Azure Function Setup Implementation: Created an Azure Function App and we had to choose the appropriate runtime and trigger type based on our requirements. Blob Trigger Function Implementation: Developed an Azure Function with a Blob Trigger to execute when a file lands in the landing folder. The code for Blob trigger as follows:

```
module.exports = async function (context, myBlob) {
context.log("JavaScript blob trigger function processed blob \n Blob:");
context.log("********Azure Function Started********");
     var result =true;
    try{
      context.log(myBlob.toString());
      JSON.parse(myBlob.toString().trim().replace('\n', ' '));
    }catch(exception){
      context.log(exception);
      result =false;
```

```
          }
          if(result){
            context.bindings.stagingFolder = myBlob.toString();
            context.log("********File Copied to Staging Folder Successfully********");
          } else{

            context.bindings.rejectedFolder = myBlob.toString();
            context.log("********Inavlid JSON File Copied to Rejected Folder Successfully********");
          }

    context.log("*******Azure Function Ended Successfully*******");

      };
```

We utilized Azure Storage to interact with the landing folder and perform necessary actions. JSON Validation: In the above function, implement logic to validate if the incoming file contains valid JSON. If valid, move the file to the staging folder; otherwise, move it to the reject folder.

Azure SQL Database for ETL: Azure Data Factory orchestrates the ETL process, extracting validated telemetry data from the staging folder and loading it into the Azure SQL Database. This process involves transforming the data to match the desired schema and structure for efficient storage and querying.

Secure Credential Management using Azure Key Vault: Key vault is used for securely storing and managing sensitive credentials and secrets, such as access keys, ensuring secure connections throughout the ETL process.

- Key Vault Setup Implementation: In the Azure Portal, we created a new Azure Key Vault to securely store sensitive information, such as database credentials, within the Key Vault.
- Secrets Implementation: One of the important tasks in key vault is to add secrets in Azure Key Vault containing database credentials (username, password) and configured.
- Link Service Configuration Implementation: To configure Link Service in ADF, we configure the Azure SQL Database link service to reference secrets stored in Azure Key Vault. Enhance security by avoiding the direct exposure of sensitive information.

1. Data Querying and Analysis:

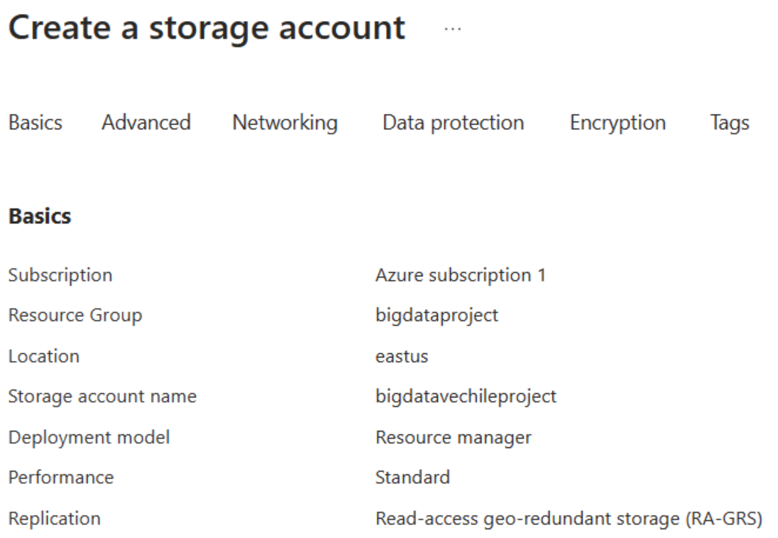
Using Azure SQL Database for Analysis: With the telemetry data securely stored in Azure SQL Database, users can leverage SQL queries and analytics tools to extract meaningful insights. This allows for in-depth analysis of vehicle performance, driver behavior, and other relevant metrics.

- Database and Table Creation Implementation: Create Azure SQL Database, in the Azure Portal, create an Azure SQL Database. Design tables within the database to match the schema of the IoT data.
- Then implemented Azure SQL Database Link Service where we created a linked service within ADF, create a link service to connect to Azure SQL Database. Configured the dataset to read from the staging folder and write to the Azure SQL Database. Then Designed a pipeline to ensure the pipeline is triggered to move data from the staging folder to the Azure SQL Database when needed. Monitor and optimize pipeline execution for performance.
- By meticulously following these detailed steps, the project establishes a robust end-to-end data pipeline that seamlessly moves, validates, and stores IoT data from an S3 bucket to an Azure SQL Database. The integration of various Azure services ensures reliability, security, and efficiency throughout the process.

The project pipeline, spanning Ingestion, ETL, and Visualization, ensures a systematic and secure flow of telemetry data from its origin in Amazon S3 to its destination in Azure SQL Database. By leveraging Azure services such as Data Factory, Function App, Key Vault, and SQL Database, the pipeline orchestrates a seamless journey, allowing for efficient data processing, validation, and visualization for informed decision-making within the connected vehicle ecosystem.

CHAPTER-7

RESULTS AND ANALYSIS

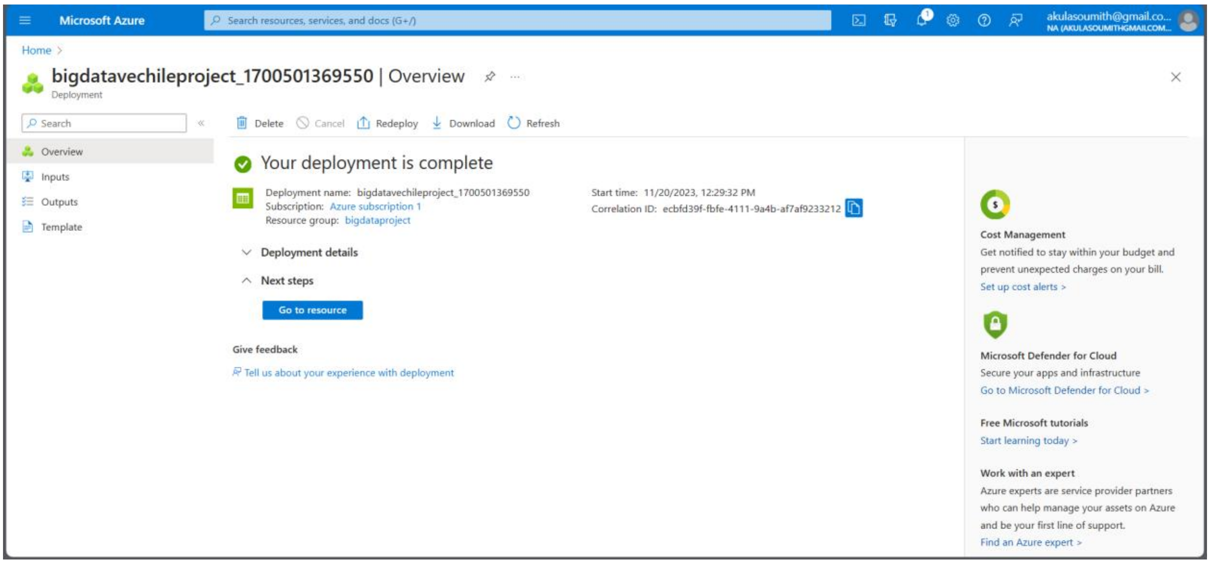


Create a storage account ···

Basics

Subscription	Azure subscription 1
Resource Group	bigdataproject
Location	eastus
Storage account name	bigdatavechileproject
Deployment model	Resource manager
Performance	Standard
Replication	Read-access geo-redundant storage (RA-GRS)

Figure 7.1 : Azure storage account creation



bigdatavechileproject_1700501369550 | Overview

Deployment

Your deployment is complete

Deployment name: bigdatavechileproject_1700501369550
Subscription: Azure subscription 1
Resource group: bigdataproject

Start time: 11/20/2023, 12:29:32 PM
Correlation ID: ecbfd39f-fbfe-4111-9a4b-af7af9233212

Deployment details

Next steps

Give feedback

Tell us about your experience with deployment

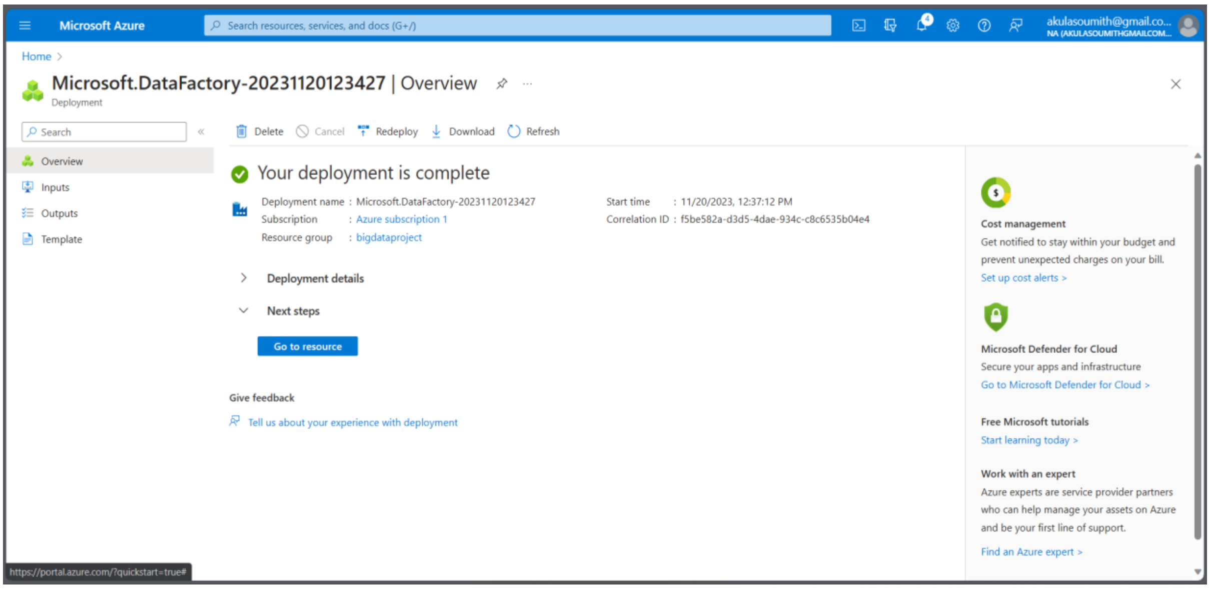
Cost Management
Get notified to stay within your budget and prevent unexpected charges on your bill.
Set up cost alerts >

Microsoft Defender for Cloud
Secure your apps and infrastructure
Go to Microsoft Defender for Cloud >

Free Microsoft tutorials
Start learning today >

Work with an expert
Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support.
Find an Azure expert >

Figure 7.2 : Azure resource deployment complete



Microsoft.DataFactory-20231120123427 | Overview

Deployment

Your deployment is complete

Deployment name : Microsoft.DataFactory-20231120123427
Subscription : Azure subscription 1
Resource group : bigdataproject
Start time : 11/20/2023, 12:37:12 PM
Correlation ID : f5be582a-d3d5-4dae-934c-c8c6535b04e4

Deployment details

Next steps

Give feedback

Tell us about your experience with deployment

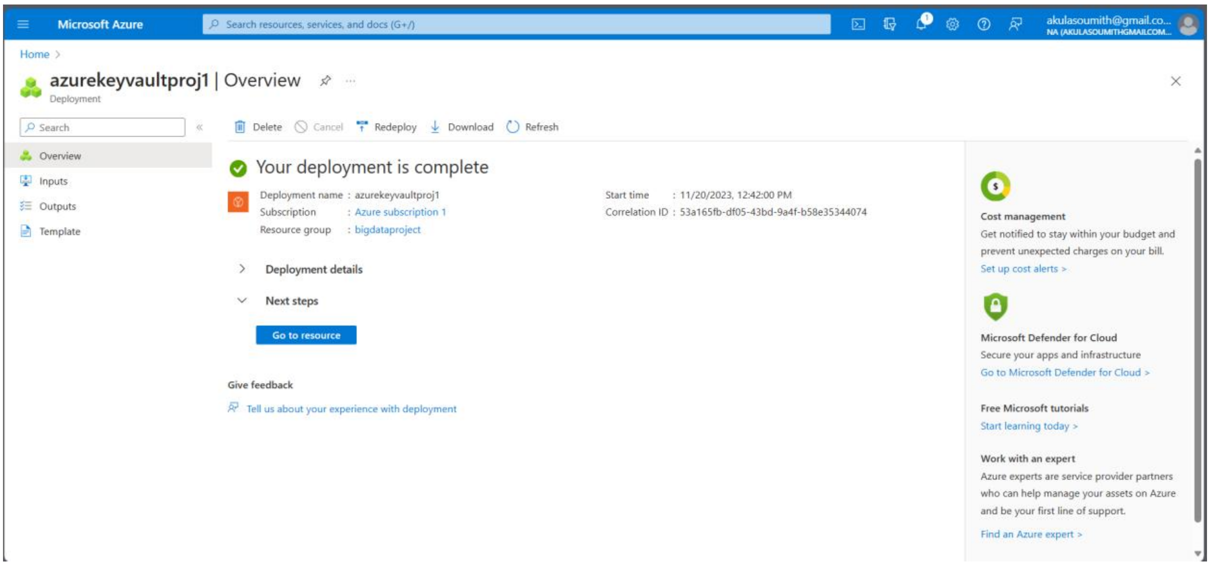
Cost management
Get notified to stay within your budget and prevent unexpected charges on your bill.
Set up cost alerts >

Microsoft Defender for Cloud
Secure your apps and infrastructure
Go to Microsoft Defender for Cloud >

Free Microsoft tutorials
Start learning today >

Work with an expert
Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support.
Find an Azure expert >

Figure 7.3 : Azure data factory deployment complete



azurekeyvaultproj1 | Overview

Deployment

Your deployment is complete

Deployment name : azurekeyvaultproj1
Subscription : Azure subscription 1
Resource group : bigdataproject
Start time : 11/20/2023, 12:42:00 PM
Correlation ID : 53a165fb-df05-43bd-9a4f-b58e35344074

Deployment details

Next steps

Give feedback

Tell us about your experience with deployment

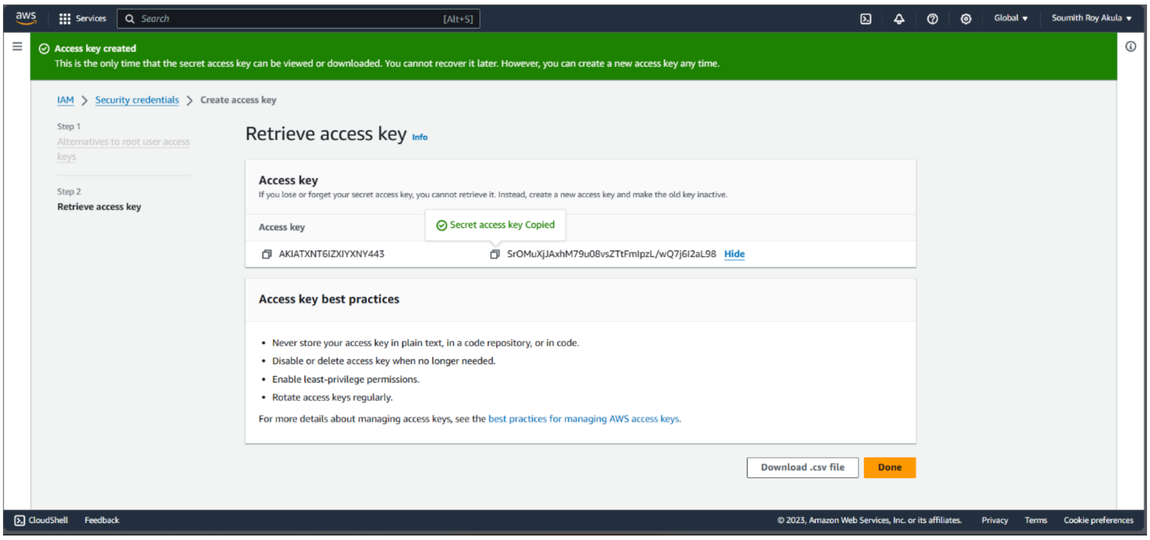
Cost management
Get notified to stay within your budget and prevent unexpected charges on your bill.
Set up cost alerts >

Microsoft Defender for Cloud
Secure your apps and infrastructure
Go to Microsoft Defender for Cloud >

Free Microsoft tutorials
Start learning today >

Work with an expert
Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support.
Find an Azure expert >

Figure 7.4 : Azure key vault deployment complete



Access key created

This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

Retrieve access key

Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Secret access key Copied

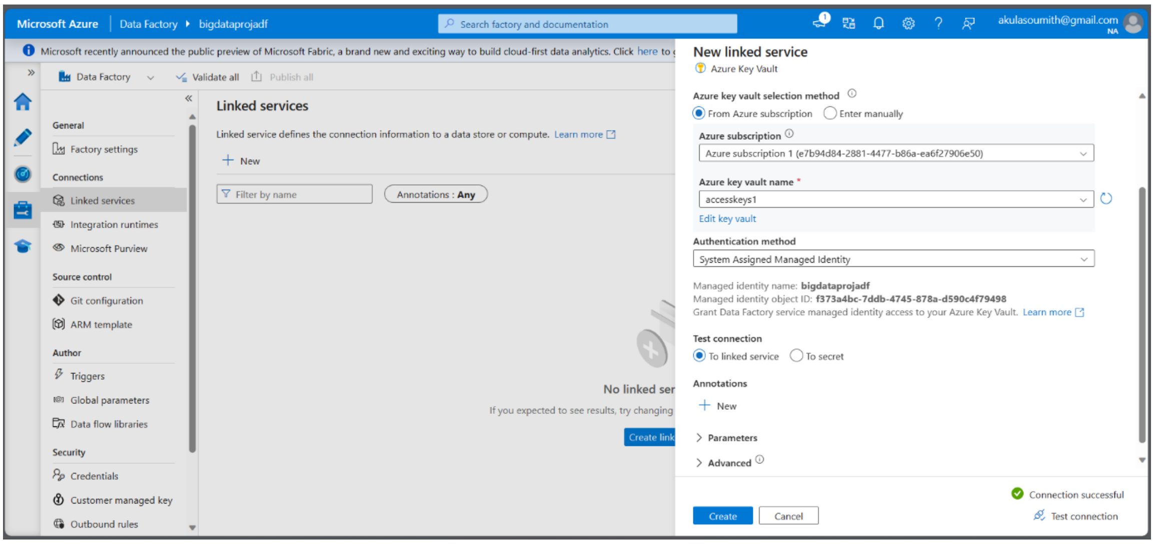
Access key	
AKIATXNT6IZXIYXNY443	SrOMuXjJAxhM79u08vsZTtFmlpzL/wQ7j6I2aL98

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the best practices for managing AWS access keys.

Figure 7.5 : AWS retrieve access key



Microsoft recently announced the public preview of Microsoft Fabric, a brand new and exciting way to build cloud-first data analytics. Click here to

Linked services

Linked service defines the connection information to a data store or compute. Learn more

No linked ser

If you expected to see results, try changing

New linked service

Azure Key Vault

Azure key vault selection method: ● From Azure subscription ○ Enter manually

Azure subscription: Azure subscription 1 (e7b94d84-2881-4477-b86a-ea6f27906e50)

Azure key vault name *: accesskeys1

Edit key vault

Authentication method: System Assigned Managed Identity

Managed identity name: **bigdataprojadf**

Managed identity object ID: **f373a4bc-7ddb-4745-878a-d590c4f79498**

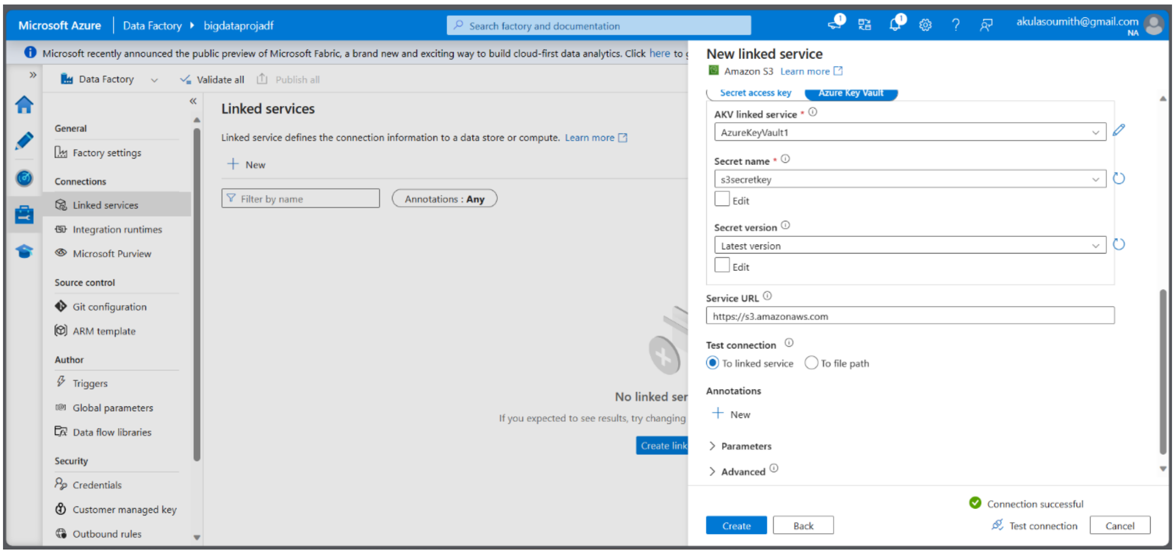
Grant Data Factory service managed identity access to your Azure Key Vault. Learn more

Test connection: ● To linked service ○ To secret

Annotations

Connection successful

Figure 7.6 : Azure key vault linked service



Microsoft recently announced the public preview of Microsoft Fabric, a brand new and exciting way to build cloud-first data analytics. Click here to

Linked services

Linked service defines the connection information to a data store or compute. Learn more

No linked ser

If you expected to see results, try changing

New linked service

Amazon S3 Learn more

AKV linked service *: AzureKeyVault1

Secret name *: s3secretkey

□ Edit

Secret version: Latest version

□ Edit

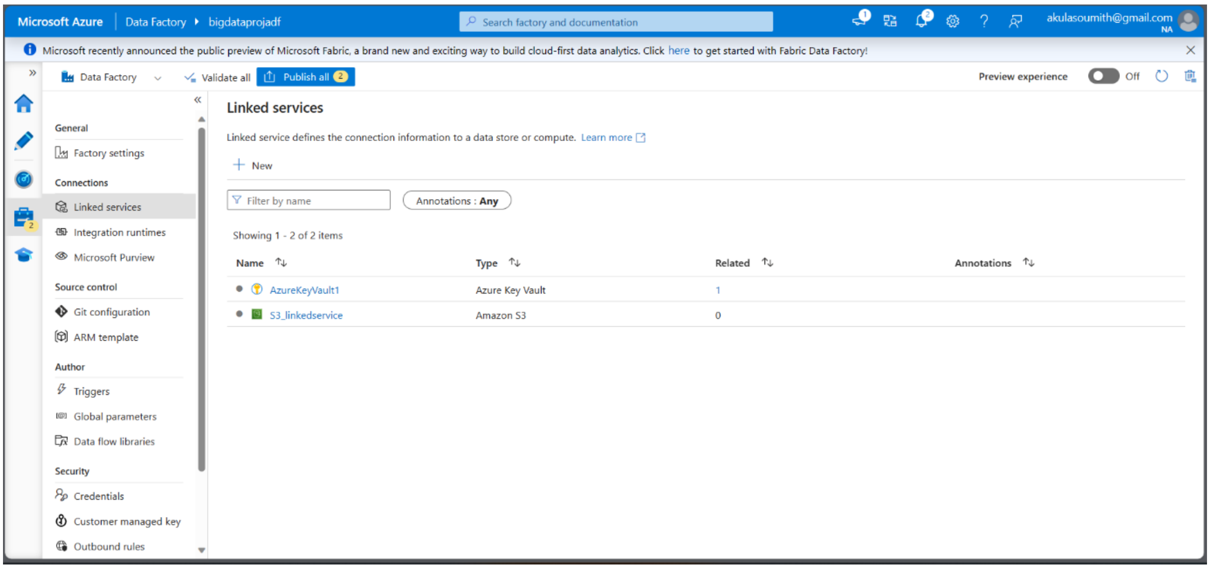
Service URL: https://s3.amazonaws.com

Test connection: ● To linked service ○ To file path

Annotations

Connection successful

Figure 7.7 : Amazon S3 linked service



Microsoft recently announced the public preview of Microsoft Fabric, a brand new and exciting way to build cloud-first data analytics. Click here to get started with Fabric Data Factory!

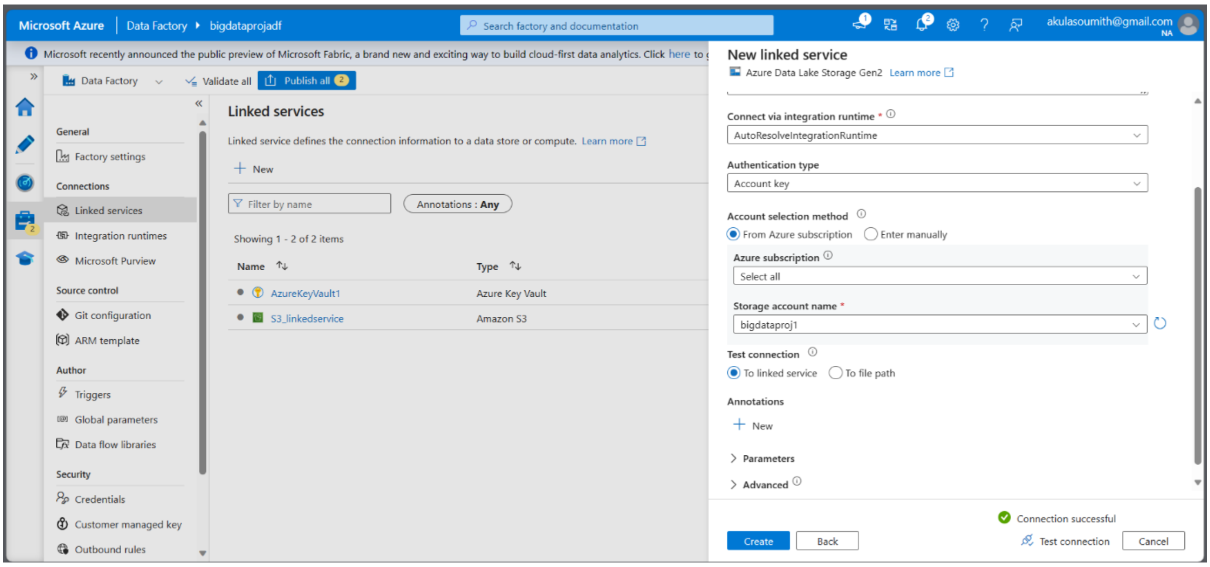
Linked services

Linked service defines the connection information to a data store or compute. Learn more

Showing 1 - 2 of 2 items

Name	Type	Related	Annotations
AzureKeyVault1	Azure Key Vault	1	
S3_linkedservice	Amazon S3	0	

Figure 7.8 : Linked services list



Microsoft recently announced the public preview of Microsoft Fabric, a brand new and exciting way to build cloud-first data analytics. Click here to

Linked services

Linked service defines the connection information to a data store or compute. Learn more

Showing 1 - 2 of 2 items

Name	Type
AzureKeyVault1	Azure Key Vault
S3_linkedservice	Amazon S3

New linked service

Azure Data Lake Storage Gen2 Learn more

Connect via integration runtime *: AutoResolveIntegrationRuntime

Authentication type: Account key

Account selection method: ● From Azure subscription ○ Enter manually

Azure subscription: Select all

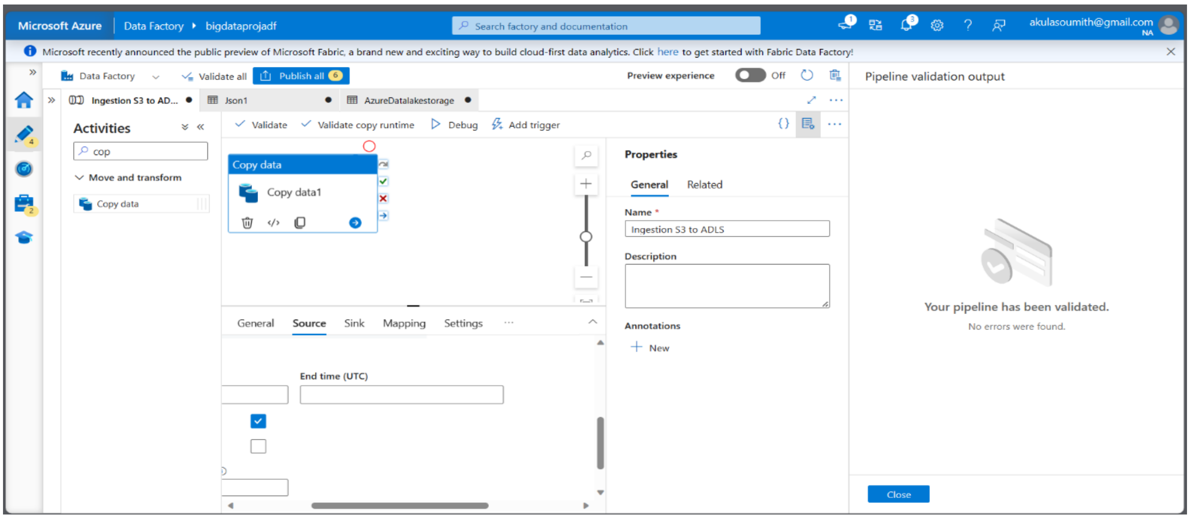
Storage account name *: bigdataproj1

Test connection: ● To linked service ○ To file path

Annotations

Connection successful

Figure 7.9 : ADLS Gen2 linked service



Microsoft recently announced the public preview of Microsoft Fabric, a brand new and exciting way to build cloud-first data analytics. Click here to get started with Fabric Data Factory!

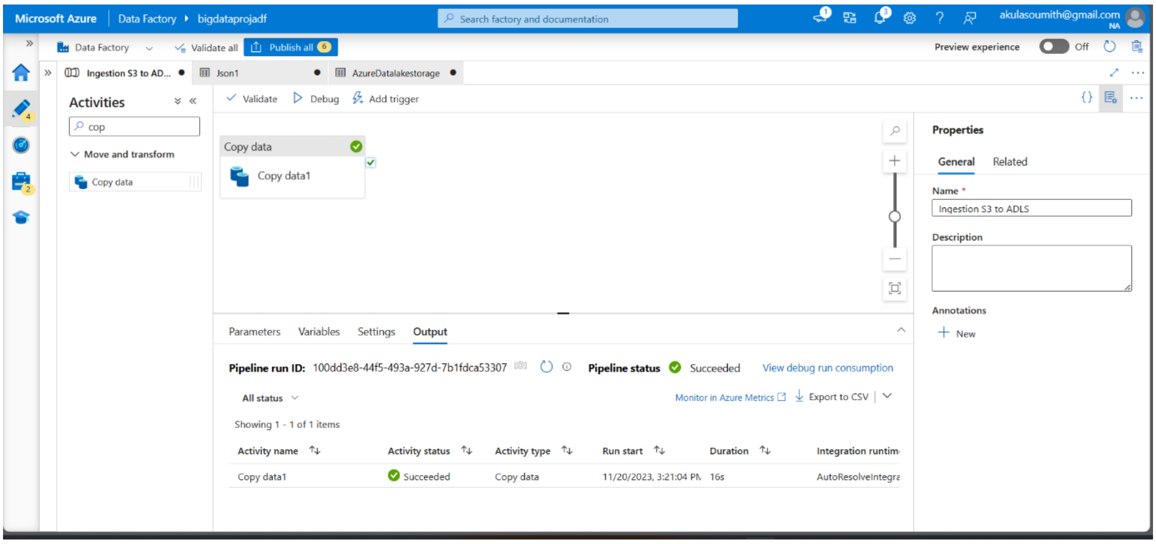
Name * Ingestion S3 to ADLS

Description

Your pipeline has been validated.

No errors were found.

Figure 7.10 : Pipeline validation output



Name * Ingestion S3 to ADLS

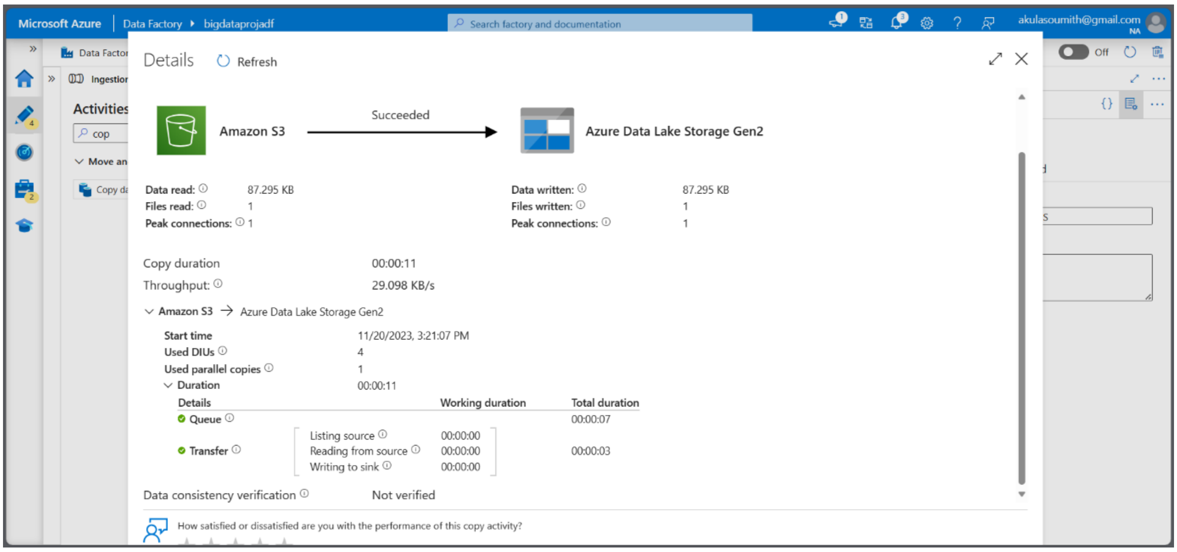
Description

Pipeline run ID: 100dd3e8-44f5-493a-927d-7b1fdca53307 Pipeline status ✓ Succeeded View debug run consumption

Showing 1 - 1 of 1 items

Activity name	Activity status	Activity type	Run start	Duration	Integration runtim
Copy data1	✓ Succeeded	Copy data	11/20/2023, 3:21:04 PM	16s	AutoResolveIntegra

Figure 7.11 : S3 to ADLS copy activity output


Microsoft Azure
Data Factory ▸ bigdataprojadf
Search factory and documentation
akulasoumith@gmail.com
Details
Refresh
Amazon S3
Succeeded
Azure Data Lake Storage Gen2

Data read:	87.295 KB	Data written:	87.295 KB
Files read:	1	Files written:	1
Peak connections:	1	Peak connections:	1

Copy duration 00:00:11

Throughput: 29.098 KB/s

Amazon S3 → Azure Data Lake Storage Gen2

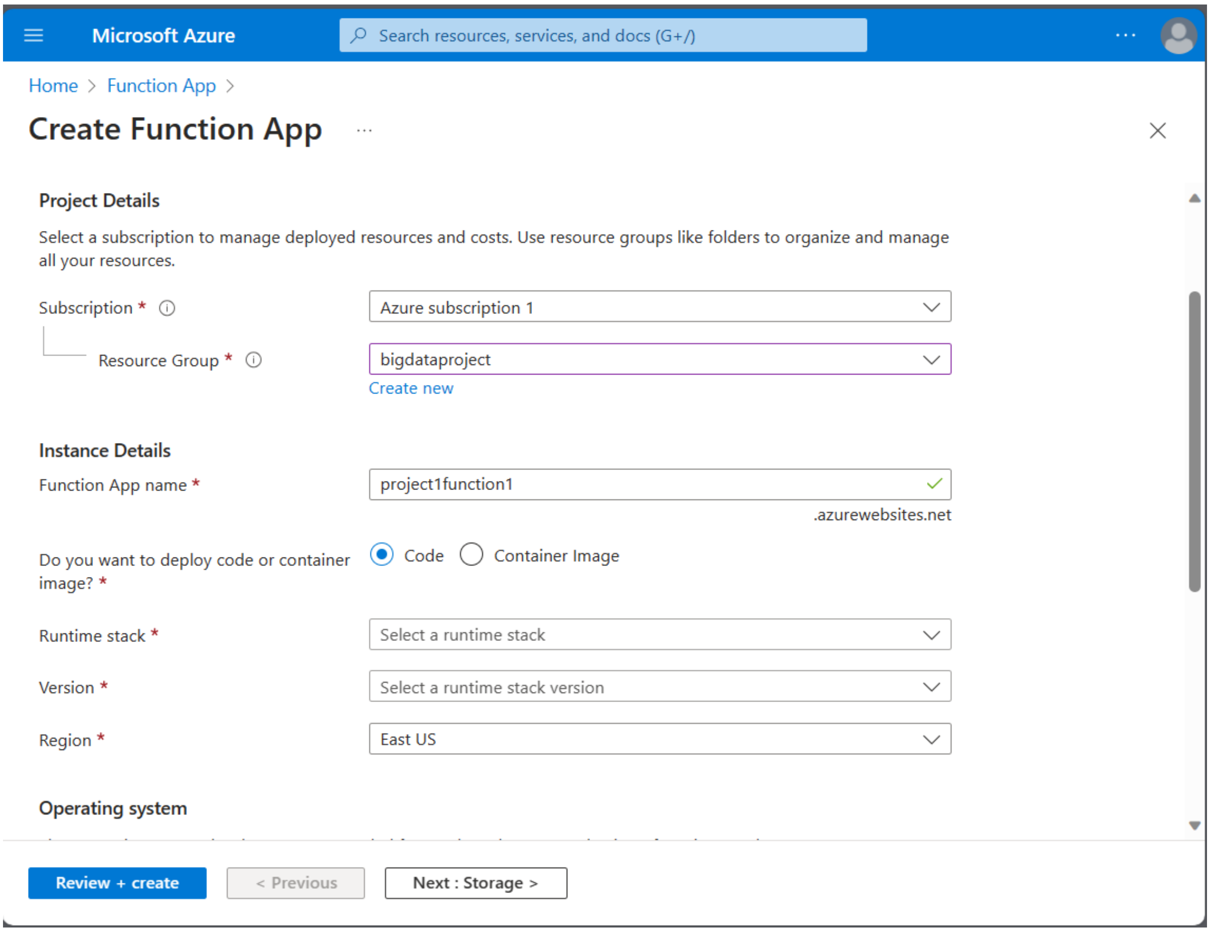
Start time	11/20/2023, 3:21:07 PM
Used DIUs	4
Used parallel copies	1
Duration	00:00:11

Details		Working duration	Total duration
✓ Queue			00:00:07
✓ Transfer	Listing source	00:00:00	00:00:03
	Reading from source	00:00:00	
	Writing to sink	00:00:00	

Data consistency verification Not verified

How satisfied or dissatisfied are you with the performance of this copy activity?

Figure 7.12 : S3 to ADLS copy activity output details



Create Function App

Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * : Azure subscription 1

Resource Group * : bigdataproject

Create new

Instance Details

Function App name * : project1function1 .azurewebsites.net

Do you want to deploy code or container image? * ● Code ○ Container Image

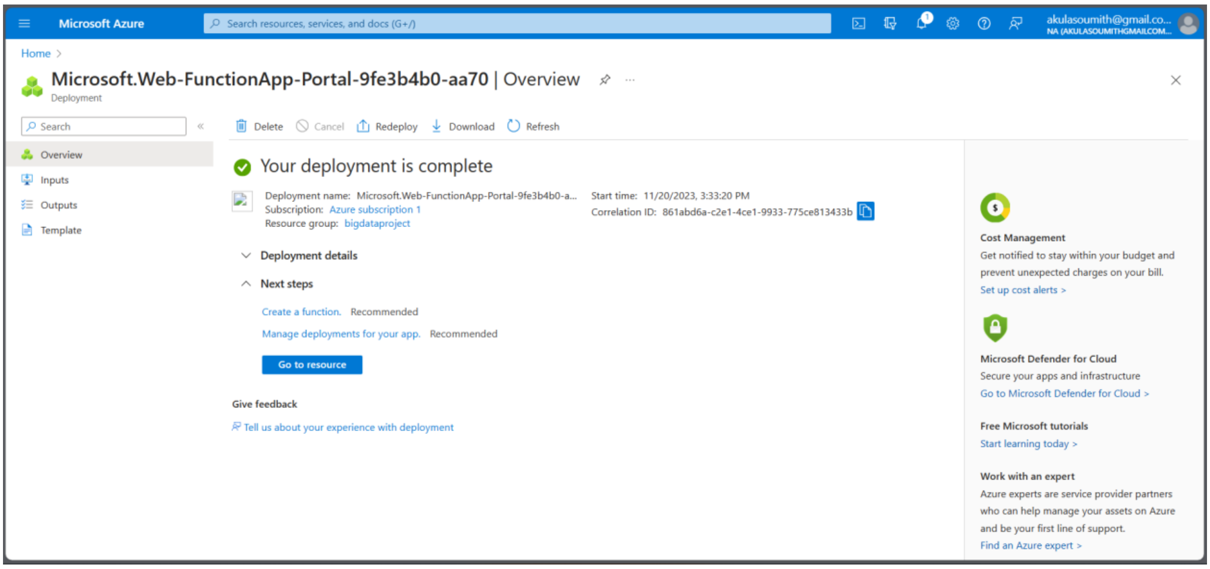
Runtime stack * : Select a runtime stack

Version * : Select a runtime stack version

Region * : East US

Operating system

Figure 7.13 : Azure function app creation



Microsoft.Web-FunctionApp-Portal-9fe3b4b0-aa70 | Overview

Deployment

Your deployment is complete

Deployment name: Microsoft.Web-FunctionApp-Portal-9fe3b4b0-a...

Subscription: Azure subscription 1

Resource group: bigdataproject

Start time: 11/20/2023, 3:33:20 PM

Correlation ID: 861abd6a-c2e1-4ce1-9933-775ce813433b

Deployment details

Next steps

Create a function. Recommended

Manage deployments for your app. Recommended

Give feedback

Tell us about your experience with deployment

Cost Management

Get notified to stay within your budget and prevent unexpected charges on your bill.

Set up cost alerts >

Microsoft Defender for Cloud

Secure your apps and infrastructure

Go to Microsoft Defender for Cloud >

Free Microsoft tutorials

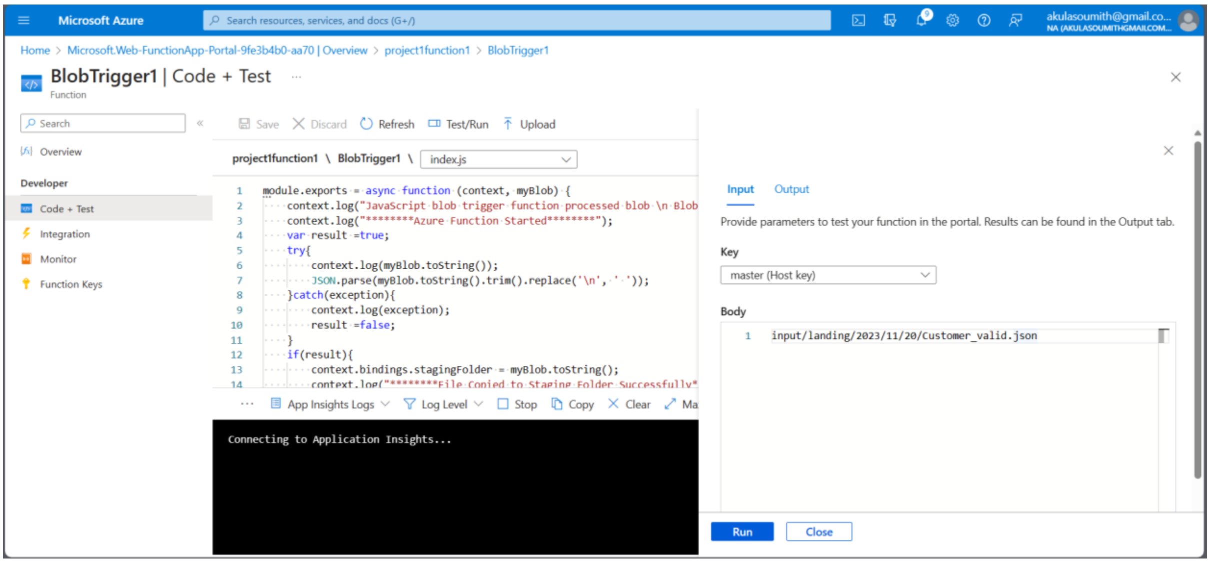
Start learning today >

Work with an expert

Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support.

Find an Azure expert >

Figure 7.14 : Azure function app deployment complete



Microsoft Azure
Search resources, services, and docs (G+/)
akulasoumith@gmail.co...
NA (AKULASOUMITHGMAILCOM...
Home > Microsoft.Web-FunctionApp-Portal-9fe3b4b0-aa70 | Overview > project1function1 > BlobTrigger1
BlobTrigger1 | Code + Test
Function
Search
Overview
Developer
Code + Test
Integration
Monitor
Function Keys
Save
Discard
Refresh
Test/Run
Upload
project1function1 \ BlobTrigger1 \ index.js
App Insights Logs
Log Level
Stop
Copy
Clear
Input
Output
Run
Close

```
module.exports = async function (context, myBlob) {
    context.log("JavaScript blob trigger function processed blob \n Blob
    context.log("********Azure Function Started********");
    var result =true;
    try{
        context.log(myBlob.toString());
        JSON.parse(myBlob.toString().trim().replace('\n', ' '));
    }catch(exception){
        context.log(exception);
        result =false;
    }
    if(result){
        context.bindings.stagingFolder = myBlob.toString();
        context.log("********File Copied to Staging Folder Successfully*
```

```
Connecting to Application Insights...
```

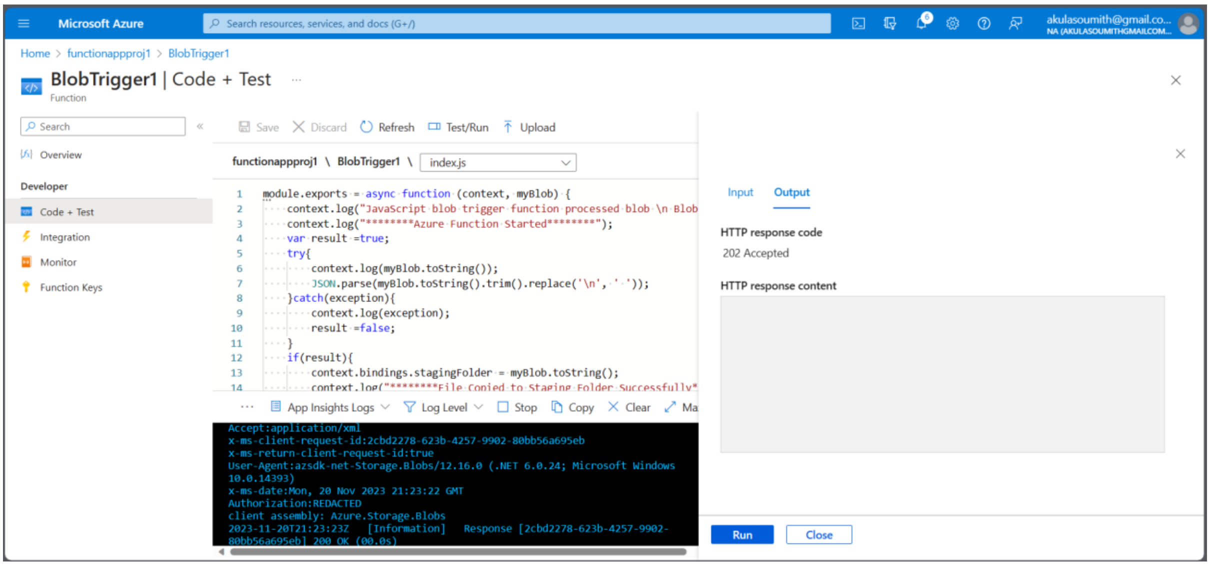
Provide parameters to test your function in the portal. Results can be found in the Output tab.

Key: master (Host key)

Body:

```
input/landing/2023/11/20/Customer_valid.json
```

Figure 7.15 : Blob trigger code and test



```
module.exports = async function (context, myBlob) {
    context.log("JavaScript blob trigger function processed blob \n Blob
    context.log("********Azure Function Started********");
    var result =true;
    try{
        context.log(myBlob.toString());
        JSON.parse(myBlob.toString().trim().replace('\n', ' '));
    }catch(exception){
        context.log(exception);
        result =false;
    }
    if(result){
        context.bindings.stagingFolder = myBlob.toString();
        context.log("********File Copied to Staging Folder Successfully*
```

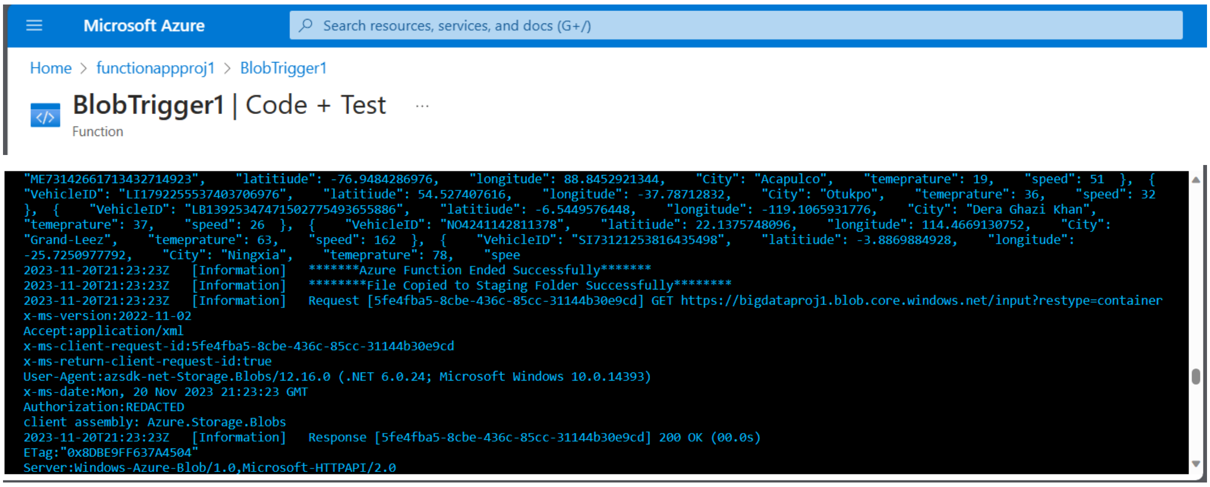
```
Accept:application/xml
x-ms-client-request-id:2cbd2278-623b-4257-9902-80bb56a695eb
x-ms-return-client-request-id:true
User-Agent:azsdk-net-Storage.Blobs/12.16.0 (.NET 6.0.24; Microsoft Windows
10.0.14393)
x-ms-date:Mon, 20 Nov 2023 21:23:22 GMT
Authorization:REDACTED
client assembly: Azure.Storage.Blobs
2023-11-20T21:23:23Z   [Information]   Response [2cbd2278-623b-4257-9902-
80bb56a695eb] 200 OK (00.0s)
```

HTTP response code

202 Accepted

HTTP response content

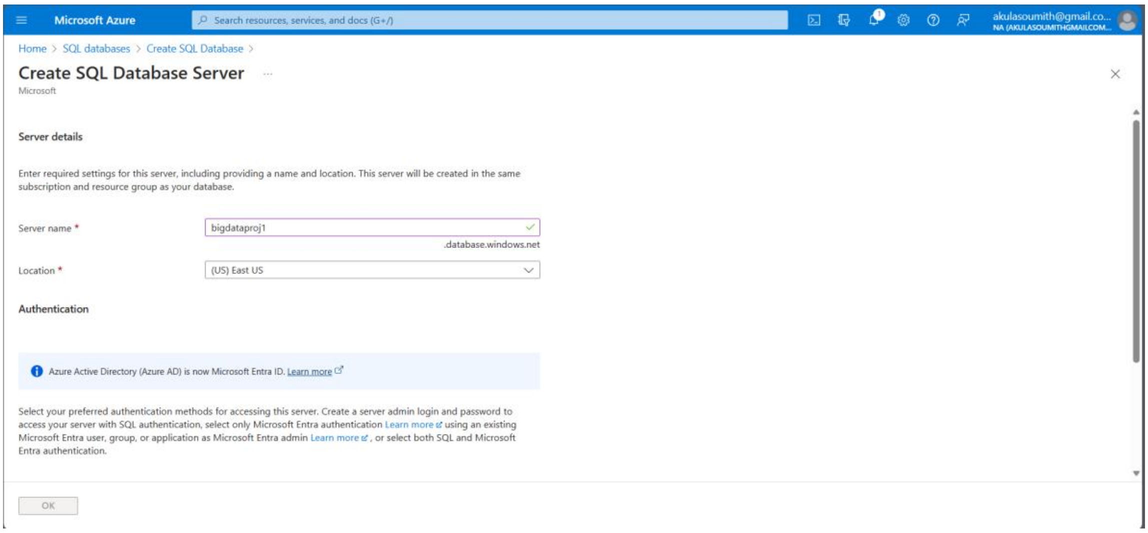
Figure 7.16 : Blob trigger test with HTTP 202



Microsoft Azure
Search resources, services, and docs (G+/)
Home > functionappproj1 > BlobTrigger1
BlobTrigger1 | Code + Test
Function

```
"ME73142661713432714923",    "latitiude": -76.9484286976,    "longitude": 88.8452921344,    "City": "Acapulco",    "temeprature": 19,    "speed": 51  },  {
"VehicleID": "LI1792255537403706976",    "latitiude": 54.527407616,    "longitude": -37.78712832,    "City": "Otukpo",    "temeprature": 36,    "speed": 32
},  {    "VehicleID": "LB13925347471502775493655886",    "latitiude": -6.5449576448,    "longitude": -119.1065931776,    "City": "Dera Ghazi Khan",
"temeprature": 37,    "speed": 26  },  {    "VehicleID": "NO4241142811378",    "latitiude": 22.1375748096,    "longitude": 114.4669130752,    "City":
"Grand-Leez",    "temeprature": 63,    "speed": 162  },  {    "VehicleID": "SI73121253816435498",    "latitiude": -3.8869884928,    "longitude":
-25.7250977792,    "City": "Ningxia",    "temeprature": 78,    "spee
2023-11-20T21:23:23Z   [Information]   *******Azure Function Ended Successfully*******
2023-11-20T21:23:23Z   [Information]   ********File Copied to Staging Folder Successfully********
2023-11-20T21:23:23Z   [Information]   Request [5fe4fba5-8cbe-436c-85cc-31144b30e9cd] GET https://bigdataproj1.blob.core.windows.net/input?restype=container
x-ms-version:2022-11-02
Accept:application/xml
x-ms-client-request-id:5fe4fba5-8cbe-436c-85cc-31144b30e9cd
x-ms-return-client-request-id:true
User-Agent:azsdk-net-Storage.Blobs/12.16.0 (.NET 6.0.24; Microsoft Windows 10.0.14393)
x-ms-date:Mon, 20 Nov 2023 21:23:23 GMT
Authorization:REDACTED
client assembly: Azure.Storage.Blobs
2023-11-20T21:23:23Z   [Information]   Response [5fe4fba5-8cbe-436c-85cc-31144b30e9cd] 200 OK (00.0s)
ETag:"0x8DBE9FF637A4504"
Server:Windows-Azure-Blob/1.0,Microsoft-HTTPAPI/2.0
```

Figure 7.17 : Blog trigger output log



Microsoft Azure
Search resources, services, and docs (G+/)
akulasoumith@gmail.co...
NA (AKULASOUMITHGMAILCOM...
Home > SQL databases > Create SQL Database >

Create SQL Database Server

Microsoft

Server details

Enter required settings for this server, including providing a name and location. This server will be created in the same subscription and resource group as your database.

Server name * bigdataproj1 .database.windows.net

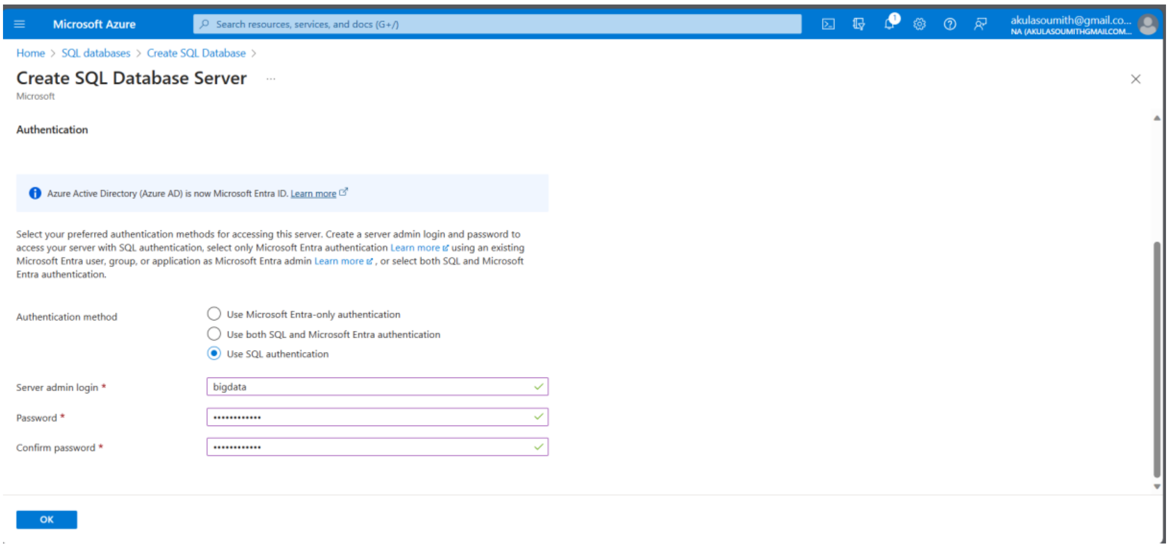
Location * (US) East US

Authentication

Azure Active Directory (Azure AD) is now Microsoft Entra ID. Learn more

Select your preferred authentication methods for accessing this server. Create a server admin login and password to access your server with SQL authentication, select only Microsoft Entra authentication Learn more using an existing Microsoft Entra user, group, or application as Microsoft Entra admin Learn more , or select both SQL and Microsoft Entra authentication.

Figure 7.18 : Azure database server creation



Microsoft Azure
Search resources, services, and docs (G+/)
akulasoumith@gmail.co...
NA (AKULASOUMITHGMAILCOM...
Home > SQL databases > Create SQL Database >

Create SQL Database Server

Microsoft

Authentication

Azure Active Directory (Azure AD) is now Microsoft Entra ID. Learn more

Select your preferred authentication methods for accessing this server. Create a server admin login and password to access your server with SQL authentication, select only Microsoft Entra authentication Learn more using an existing Microsoft Entra user, group, or application as Microsoft Entra admin Learn more , or select both SQL and Microsoft Entra authentication.

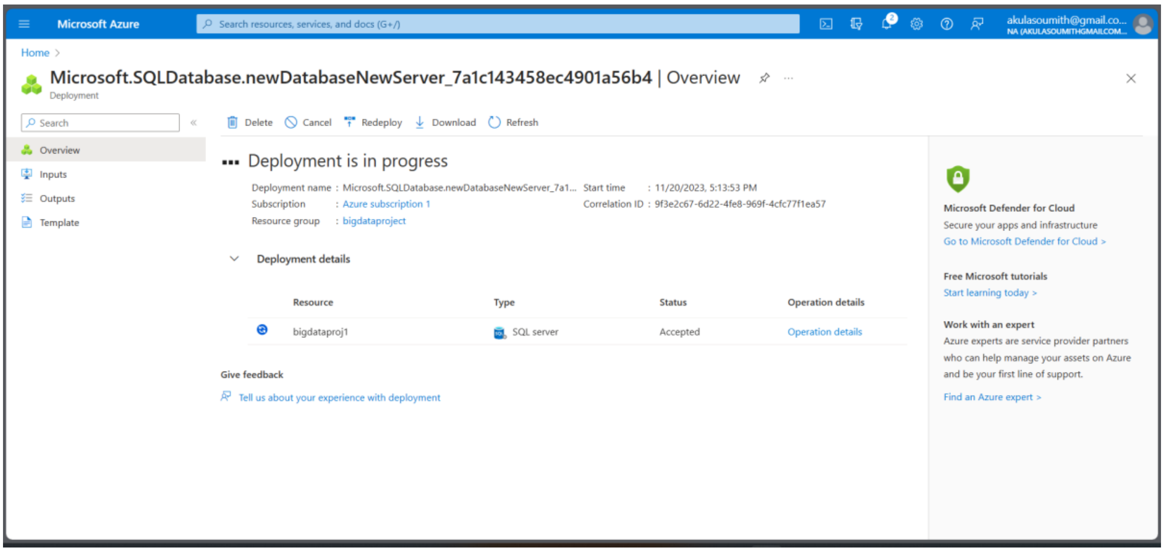
Authentication method
○ Use Microsoft Entra-only authentication
○ Use both SQL and Microsoft Entra authentication
● Use SQL authentication

Server admin login * bigdata

Password * ••••••••••••

Confirm password * ••••••••••••

Figure 7.19 : Azure SQL database server creation



Microsoft.SQLDatabase.newDatabaseNewServer_7a1c143458ec4901a56b4 | Overview

Deployment

Deployment is in progress

Deployment name : Microsoft.SQLDatabase.newDatabaseNewServer_7a1...
Subscription : Azure subscription 1
Resource group : bigdataproject
Start time : 11/20/2023, 5:13:53 PM
Correlation ID : 9f3e2c67-6d22-4fe8-969f-4cfc77f1ea57

Deployment details

Resource	Type	Status	Operation details
bigdataproj1	SQL server	Accepted	Operation details

Give feedback

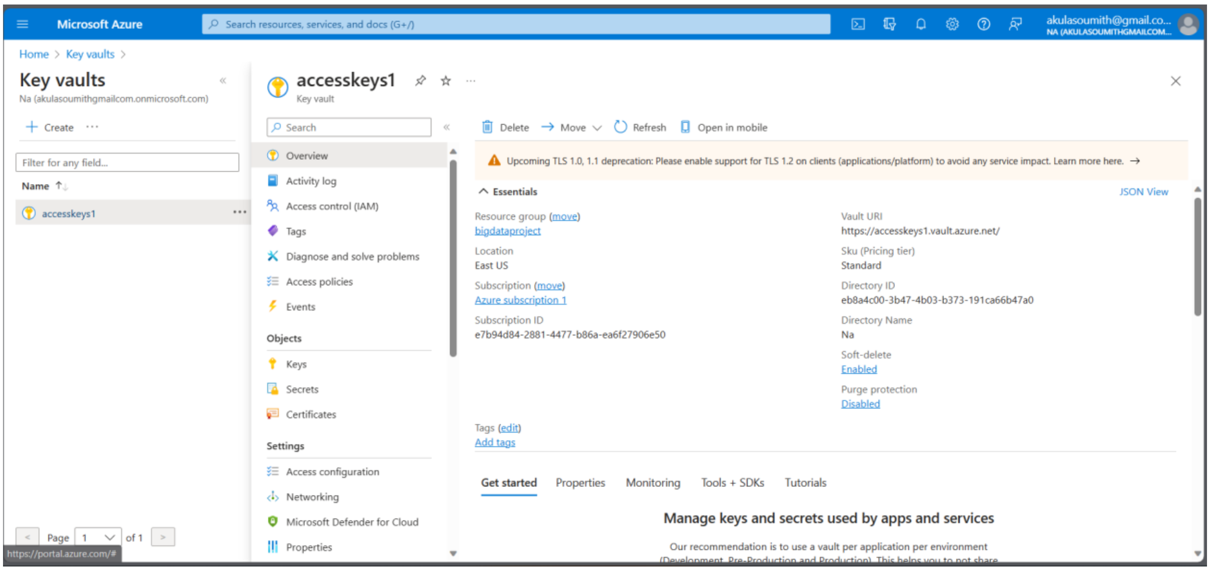
Tell us about your experience with deployment

Microsoft Defender for Cloud
Secure your apps and infrastructure
Go to Microsoft Defender for Cloud >

Free Microsoft tutorials
Start learning today >

Work with an expert
Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support.
Find an Azure expert >

Figure 7.20 : Azure SQL database server deployment in progress



Upcoming TLS 1.0, 1.1 deprecation: Please enable support for TLS 1.2 on clients (applications/platform) to avoid any service impact. Learn more here. →

Essentials

Resource group (move)
bigdataproject

Location
East US

Subscription (move)
Azure subscription 1

Subscription ID
e7b94d84-2881-4477-b86a-ea6f27906e50

Vault URI
https://accesskeys1.vault.azure.net/

Sku (Pricing tier)
Standard

Directory ID
eb8a4c00-3b47-4b03-b373-191ca66b47a0

Directory Name
Na

Soft-delete
Enabled

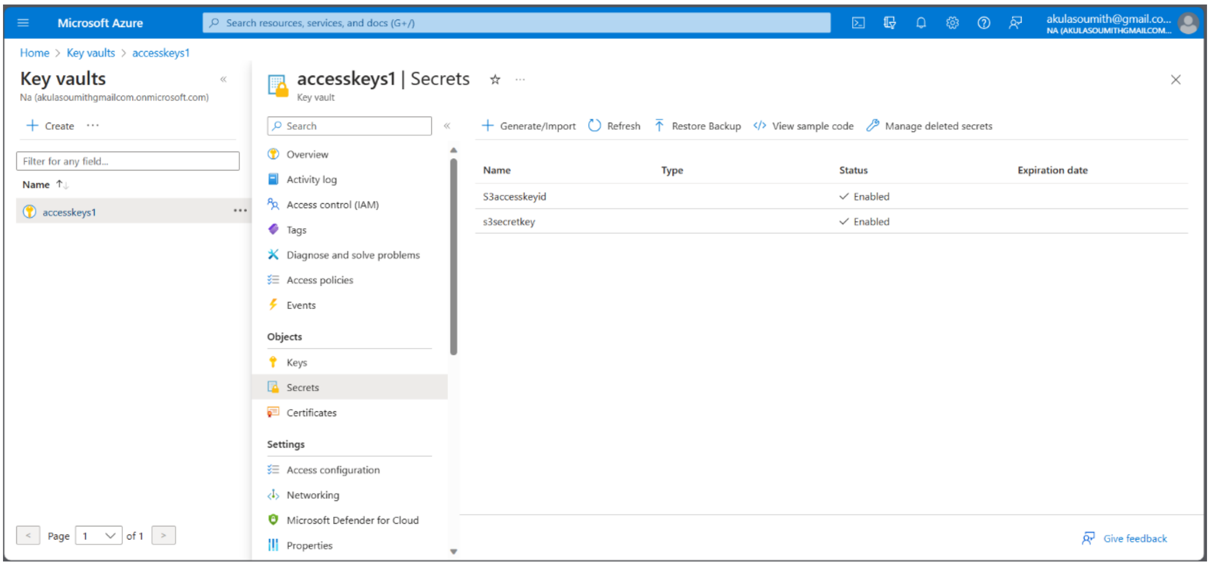
Purge protection
Disabled

Tags (edit)
Add tags

Manage keys and secrets used by apps and services

Our recommendation is to use a vault per application per environment
(Development, Pre-Production and Production). This helps you to not share

Figure 7.21 : Azure key vault



Name	Type	Status	Expiration date
S3accesskeyid		✓ Enabled	
s3secretkey		✓ Enabled	

Figure 7.22 : Azure key vault secret lists

CHAPTER-8
CONCLUSION

In conclusion, the implemented project represents a sophisticated and well-structured solution for the acquisition, processing, and analysis of telemetry data from connected vehicles. The journey begins with the systematic ingestion of data into an Amazon S3 bucket, utilizing a well-organized folder structure for effective data management. Leveraging Azure Data Factory, the project seamlessly orchestrates the movement of data from AWS S3 to Azure SQL Database, embodying best practices in the Extract, Transform, Load (ETL) process. The inclusion of Azure Key Vault enhances security by safeguarding sensitive credentials. Real-time validation through Azure Function App ensures the integrity of incoming data, directing valid information to the staging folder for subsequent processing. The project culminates in the structured storage of validated data within Azure SQL Database, creating a robust end-to-end data pipeline. Triggered automation, monitoring mechanisms, and optimization strategies contribute to the overall efficiency of the system. With a focus on facilitating data analysis, the project empowers stakeholders to derive meaningful insights from connected vehicle telemetry data, setting the stage for informed decision-making within the Internet of Things landscape.

CHAPTER-9

FUTURE ENHANCEMENTS

1. **Real-Time Processing and Analytics:**

   Integrate real-time processing capabilities to analyze and respond to telemetry data in near real-time. Utilize technologies like Apache Kafka or Azure Stream Analytics for stream processing.

2. **Integration with Additional Data Sources:**

   Extend the project's capabilities by integrating data from additional sources beyond telemetry data. This might include weather data, traffic information, or other external datasets to provide a more comprehensive view.

3. **Automated Data Quality Checks:**

   Implement automated data quality checks within the pipeline to identify and address issues proactively. This ensures that only high-quality data is processed and stored, improving the reliability of analytics.

4. **Cost Optimization:**

   Explore cost optimization strategies, such as leveraging serverless computing or optimizing cloud resource usage, to ensure efficient resource utilization and minimize operational costs.

5. **User Authentication and Authorization:**

   Implement robust user authentication and authorization mechanisms to control access to sensitive data and ensure that only authorized users can interact with the system.

6. **Continuous Monitoring and Alerting:**

   Enhance monitoring capabilities with continuous monitoring and alerting systems. Implement alerts for performance bottlenecks, data quality issues, or any anomalies in the system.

7. **Documentation and Knowledge Transfer:**

   Develop comprehensive documentation for the entire system architecture, deployment processes, and maintenance procedures. Facilitate knowledge transfer to new team members and stakeholders.

8. **Scalability and Performance Optimization:**

   Implement strategies for horizontal scalability to handle an increasing volume of data. Optimize the performance of data pipelines, ensuring they can efficiently process and transfer large datasets.

9. **Advanced Data Visualization:**

   Explore advanced data visualization techniques and tools to create more insightful and interactive dashboards. This could involve incorporating geospatial visualizations or utilizing advanced charting libraries.

10. **Security and Compliance Enhancements:**

    Strengthen security measures by implementing advanced encryption, access controls, and monitoring features. Ensure compliance with industry-specific regulations and standards governing data privacy and security.

By incorporating these future enhancements, the project can evolve into a more sophisticated and adaptive system, capable of meeting the evolving demands of connected vehicle telemetry data processing and analysis.

REFERENCES

- https://spark.apache.org/
- https://docs.aws.amazon.com/s3/index.html
- https://learn.microsoft.com/en-us/azure/data-factory/
- https://learn.microsoft.com/en-us/azure/synapse-analytics/overview-what-is
- https://docs.microsoft.com/en-us/azure/key-vault/
- https://docs.microsoft.com/en-us/azure/azure-functions/
- https://docs.microsoft.com/en-us/azure/azure-sql/
- https://docs.microsoft.com/en-us/azure/security/