Proceedings of the 11th North American International Conference on Industrial Engineering and Operations Management, Milwaukee, Wisconsin, USA, June 9-11, 2026

Proceedings of the International Conference on Industrial Engineering and Operations Management

Publisher: IEOM Society International, USA DOI: 10.46254/NA11.20260168
Published: June 9, 2026

Building a Scalable Data Lakehouse at Lamar University: Automated Infrastructure for Processing, Analytics, Machine Learning, and Archival

Naman Subedi, Md Masud Rana, Bo Sun and Frank Sun
Department of Computer Science, Lamar University
Beaumont, Texas 77705, USA
nsubedi1@lamar.edu, mrana15@lamar.edu

Helen H. Lou
Department of Chemical Engineering, Lamar University
Beaumont, Texas 77705, USA
hhlou@lamar.edu

Alek Hutson
Center for Midstream Management and Science (CMMS)
Lamar University, Beaumont, Texas 77705, USA
ahutson@lamar.edu

Abstract

Driven by the rising need for scalable, accessible, and interdisciplinary data infrastructure in university settings, this paper presents the development of a campus-scale Data Lakehouse at Lamar University (Texas), designed to bridge gaps in existing approaches to unified data platforms for both research and instruction. Built on an S3-compatible MinIO object store, the system addresses four key challenges: (1) limited access and governance for diverse users such as students, staff, and faculty, (2) fragmented technology stacks, (3) inconsistent dataset organization and retrieval, and (4) weak integration with academic workflows. These problems are tackled through the deployment of a web-based upload portal and a Python utility, enabling seamless data ingestion, processing, and retrieval. Early implementations—including a water system monitoring pipeline for forecasting and a CMMS workflow for operational analytics—demonstrate the Lakehouse's dual value in research and teaching. Findings highlight the effectiveness of accessible interfaces and scalable practices in fostering data-driven learning, with broader implications for replicability across institutions seeking to enhance curriculum integration, research capability, hands-on experience, and institutional data strategy.

Keywords
Data Lakehouse, Object Storage, MinIO, Educational Automation, Machine Learning

1. Introduction
Universities increasingly need storage that accepts raw, heterogeneous data at scale while remaining queryable for analytics and machine learning. Lamar University is building a campus Data Lakehouse that combines the scalability of a data lake with the governance and analytic performance of a data warehouse. In this architecture, an object store

provides the storage backbone and manages data as objects with rich metadata (SNIA 2025), which is well suited to modern IoT and ML workloads.

At Lamar University, the motivation for an object store is driven by several needs. IoT devices produce frequent, structured and unstructured outputs that benefit from metadata-rich organization. ML workflows require versioned storage for models, checkpoints, and large datasets. The university also needs durable archival and backup, with access control and S3 compatibility to support common tools.

We deployed a MinIO object store and exposed it through two paths to reduce friction for contributors. First, a simple web upload page on the Lakehouse site allows non-technical users to place data directly into the store. Second, a programmatic path provides a Python utility that supports progress-tracked uploads and downloads, automatic bucket creation, bucket visualization, file listing and deletion, and environment-based configuration. A lightweight testing harness exercises these operations during upgrades and records results for comparison across iterations.

Early use cases validate this approach. A water monitoring pipeline ingests periodic detector CSVs, consolidates them into combined datasets, and stores them in the object store for forecasting and analysis. Beyond research infrastructure, the system also functions as an educational automation tool. As shown by Armbrust et al. (2020), by offering both graphical and programmatic interfaces, it enables students, instructors, and research staff to engage with real-world datasets in a structured but accessible environment. This dual-access model supports classroom integration, student-led projects, and experiential learning, while giving faculty a platform for designing assignments and demonstrations that reflect modern data practices.

This paper contributes: (i) a campus-aligned rationale for adopting an S3-compatible object store within a Lakehouse, (ii) a pragmatic ingestion surface that serves both non-technical and programmatic users, (iii) an open testing harness for operational verification, and (iv) case studies that illustrate how these components support real research workflows at Lamar University.

1.1 Objectives

The objectives of this research are:

1. To deploy and configure an S3-compatible MinIO object store as the foundational storage layer for the Lamar University Data Lakehouse.
2. To design and implement dual ingestion paths—a web-based upload portal and a Python programmatic utility—that lower the barrier to data contribution for both technical and non-technical users.
3. To develop a testing harness that validates core operations (upload, download, list, delete) and supports regression detection during system upgrades.
4. To demonstrate the system's applicability through case studies in water monitoring and CMMS workflows, showcasing its dual value in research and education.

2. Literature Review

Many universities are shifting from traditional siloed databases to object storage systems (e.g., MinIO, Amazon S3, Azure Blob) as the foundation for modern data lakehouses (Mondal et al. 2024). These infrastructures allow research institutions to centralize raw and processed data from multiple disciplines, ranging from satellite imagery and drone surveys in environmental studies to electronic health records in medical research. The lakehouse model (Armbrust et al. 2020) combines the low-cost scalability of object storage with structured query layers, enabling both big-data analytics and machine learning workflows.

Several universities have adopted object storage at scale. The University of Wisconsin–Madison offers a campus-hosted, S3-compatible Research Object Storage service with a 50 TB free tier, supporting archival storage, instrument integration, and static web hosting (University of Wisconsin–Madison 2025). Purdue University operates a Ceph-based Depot Object Store with approximately 4 PB capacity, providing centralized datasets, ML models, and long-term group retention (Purdue University 2025). Stanford University leverages AWS S3 for cloud-scale storage with intelligent tiering and integration with cloud workflows (Stanford University IT 2025). Libraries and museums also employ on-premises or cloud object stores for digital preservation of manuscripts, photographs, and rare collections.

The University of California system reports using cloud-based lakehouses to integrate genomic data, climate records, and imaging datasets, making them queryable with SQL-like engines and accessible to interdisciplinary teams. Similarly, the European Open Science Cloud emphasizes lakehouse-style storage to comply with FAIR (Findable, Accessible, Interoperable, Reusable) data principles in cross-institutional projects.

Beyond storage, universities are leveraging AI-driven forecasting systems to support academic and administrative planning. Student success offices use predictive models (Gama et al. 2014) to forecast enrollment patterns, dropout risks, and financial aid demand. Facilities departments apply similar models for energy consumption forecasting, optimizing HVAC scheduling and reducing operational costs. In the public sector, forecasting models are applied to budget allocation, workforce planning, and disease spread prediction (Marz and Warren 2015).

Prior work has demonstrated how ML applied to large, fast-changing data streams can directly support decision-making. For example, Joshi et al. (2024) leveraged Twitter data and deep learning models to rapidly detect and characterize natural disasters, illustrating the feasibility of near real-time forecasting from unstructured sources. Our work extends this direction into university data infrastructure, where object storage provides the backbone for both research datasets and predictive applications.

Universities are increasingly adopting AI across multiple functions. Georgia State University's "Pounce" texting assistant (Castleman and Page 2014) reduced "summer melt" in a randomized trial by 4 percentage points, helping more admitted students enroll. Arizona State University became the first university partner for ChatGPT Enterprise, supporting course help, research, and AI tutoring avatars (Arizona State University and OpenAI 2024).

In the public sector, OMB's M-24-10 memo (U.S. OMB 2024) requires every US agency to strengthen AI governance, name responsible officials, and adopt minimum risk management practices. New York City's AI Action Plan sets 37 actions for safe deployment across agencies (City of New York 2024), while San Jose publishes generative-AI guidelines and an algorithm inventory for transparency (City of San Jose 2024). The U.S. Treasury used AI to recover USD 375M in FY2023 payment integrity cases (U.S. Treasury 2023), and the UK NHS is funding AI imaging tools and expanding clinical trials (UK Government 2025).

Regulatory clearance is also accelerating: the FDA maintains a public list of AI-enabled devices, with a 2025 JAMA analysis counting 903 AI devices by August 2024, signaling mainstream clinical adoption (FDA 2024).

Although object storage systems offer scalability, rich metadata management, and API-driven access, in practice universities often deploy them in conventional ways, treating them as file systems rather than fully leveraging their strengths. For example, UW–Madison's Research Object Storage (University of Wisconsin–Madison 2025) largely preserves traditional backup and hosting workflows rather than enabling advanced metadata-driven research workflows. By contrast, Purdue University's Depot Object Store (Purdue University 2025) effectively harnesses object storage capabilities, providing scalable, S3-compatible, group-oriented storage. This discrepancy highlights a gap in academic adoption where the design advantages of object stores are underutilized despite their suitability for modern research demands.

3. Methods

The proposed LU Lakehouse architecture anchors on an S3-compatible MinIO deployment that exposes three user-facing paths: a website upload portal for non-technical contributors, a programmatic Python utility for power users, and an administrative console for operations. A lightweight test harness continuously verifies core paths during upgrades (Figure 1 and Figure 2).

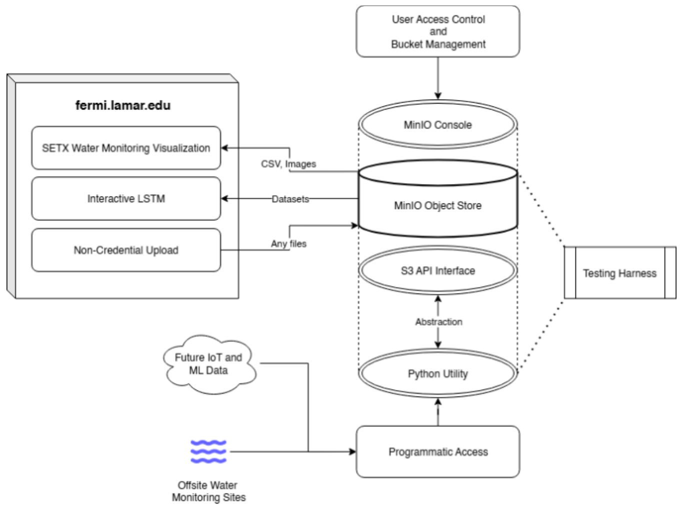


Figure 1. Proposed system diagram and interfacing

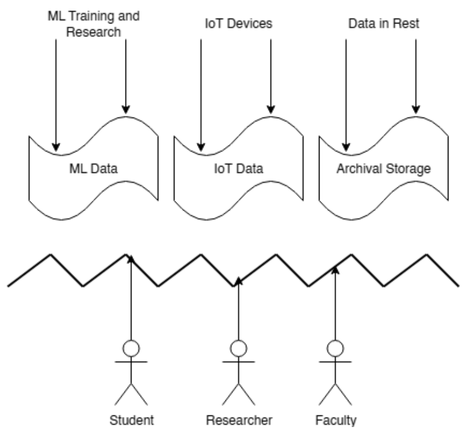


Figure 2. Proposed LU Lakehouse architecture

The stack is designed around S3 compatibility plus access control, which aligns with common analytics and ML tools and supports role-appropriate interfaces for different research groups. The programmatic layer's environment-based configuration helps separate development and production contexts while keeping credentials out of code.

We implement transfer progress as a lightweight byte counter sampled at fixed time intervals. Each upload or download maintains a cumulative counter b of bytes transferred and a known total size B. The instantaneous rate is r_inst = delta_b / delta_t, and a smoothed rate uses an exponential moving average. Percent complete is p = 100 * b/B, and the estimated time remaining is ETA = (B - b) / max(r_t, epsilon).

For parallel part transfers, each worker maintains a local counter. A thread-safe aggregator samples the sum at each interval to report unified progress. The reporter is injected into I/O paths via a minimal interface: on_start(B), on_update(delta_b), and on_finish().

We infer a tabular schema from CSV inputs using a small sample of rows, then validate and normalize the header. Given a file with M columns and N rows, we sample n = min(N, n_max) rows without replacement. Header validation includes trimming, collapsing whitespace, normalizing to snake_case, enforcing naming patterns, and de-duplicating collisions.

Type candidates include {bool, int64, float64, datetime, string} with a promotion lattice. For each sampled value, the algorithm tries types in order of specificity, promoting on failure. The output schema emits a triple (name, dtype, nullable) for each column with quality metrics including sample coverage, null rate, and confidence score.

The system comprises three key data flows:

Human upload path: A contributor opens the Lakehouse upload page, selects target data, and submits. The portal writes objects into the appropriate bucket in MinIO.

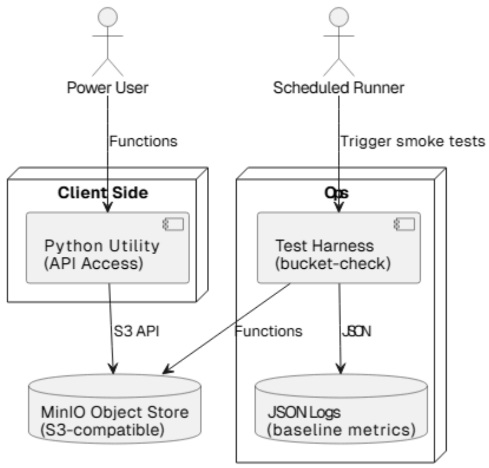


Figure 3. Human upload path flowchart

Programmatic path: Power users use the Python utility to push or pull datasets with progress bars, automated bucket creation, and structure visualization.

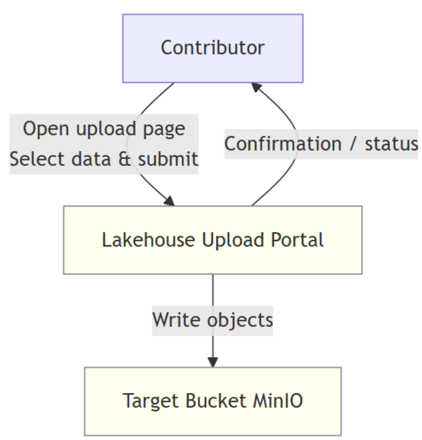


Figure 4. Testing harness flowchart

Testing harness flow: The process begins with initializing the test runner, which sequentially executes phase-specific scripts. Successful phases are logged, while failures trigger both logging and an alert with the option to continue (Figure 3 and Figure 4).

Multiple detectors continuously produce water data transmitted to a central server. A scheduled cron job triggers a Python script which stores and combines incoming streams into a unified master file. This master file is uploaded to the object store, where the frontend website fetches it for near-real-time visualization (Figure 5).

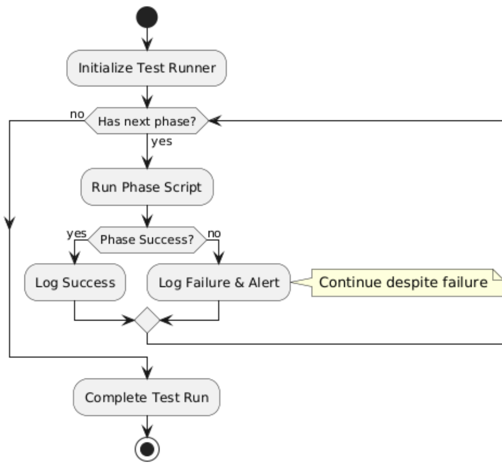


Figure 5. Water monitoring pipeline flowchart

4. Data Collection

We used a variety of datasets from different sources. This data is publicly available and may differ from internal or proprietary sources. It is provided solely for educational and informational purposes.

This is a cleaned version of data received from three monitoring sites in the Beaumont, Texas area: Saltwater Barrier, Neches River, and Pine Island Bayou. Data is collected every 20 minutes via a cron job and uploaded to the object store. Key variables include datetime, temperature (Celsius and Fahrenheit), and conductivity.

This is a case study on research performed by colleagues currently using the object store as a database with the Python utility. Building on prior NSF-funded research, our group has accumulated extensive water quality datasets spanning the past decade. These data are particularly valuable for studying water quality trends in Southeast Texas. Accordingly, this dataset serves as the primary data source for the experiments and evaluations presented in this paper (Table 1).

Table 1. Water-quality Sample

Datetime (America/CST)	**Temp (C)**	**Temp (F)**	**COND (Ms/cm)**
2024-11-21 09:22:30	19.60	67.27	0.12
2024-11-21 10:22:30	19.60	67.27	0.12
2024-11-21 11:22:30	19.60	67.27	0.12
2024-11-21 12:22:30	19.60	67.26	0.12
2024-11-21 13:22:30	19.59	67.27	0.12
…	…	…	…

For CMMS (Computerized Maintenance Management System) studies, detector outputs are stored as CSV, ingested, and preprocessed, with emphasis on maintaining a consistently updated, analysis-ready file in the object store.

5. Results and Discussion

5.1 Numerical Results

Operational Validation: We verified core paths by running a lightweight harness that performs uploads, downloads, listings, and deletions, then writes outcomes to logs for comparison across iterations. The test harness output confirmed successful operation across all phases, with SHA256 integrity checks passing for files of varying sizes (empty, 1 MB, 10 MB).

Test results showed:
- Phase 4 (Object Retrieval and Integrity Check) completed successfully with SHA256 hash verification for all test objects.
- All seven test phases passed without errors during configuration changes and upgrades.
- The harness confirmed that behavior remained stable across system iterations.

User-Facing Reliability: Non-technical contributors were able to place files in the store through the Lakehouse upload page without SDKs or command-line tools, reducing handoffs and keeping datasets centralized at the time of collection.

Programmatic Instrumentation: For power users, the Python utility exposed progress-tracked transfers and automatic bucket creation. It reported transfer speed and duration, helping diagnose network or endpoint bottlenecks during large moves.

5.2 Graphical Results

Frontend Interactivity: The website, available at fermi.lamar.edu from within the Lamar University network, allows users to interact with all ongoing and planned projects (Figure 6 and Figure 7).

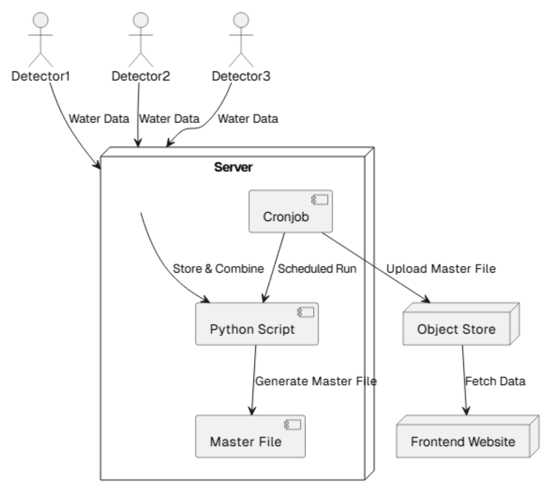


Figure 6. fermi.lamar.edu website interface

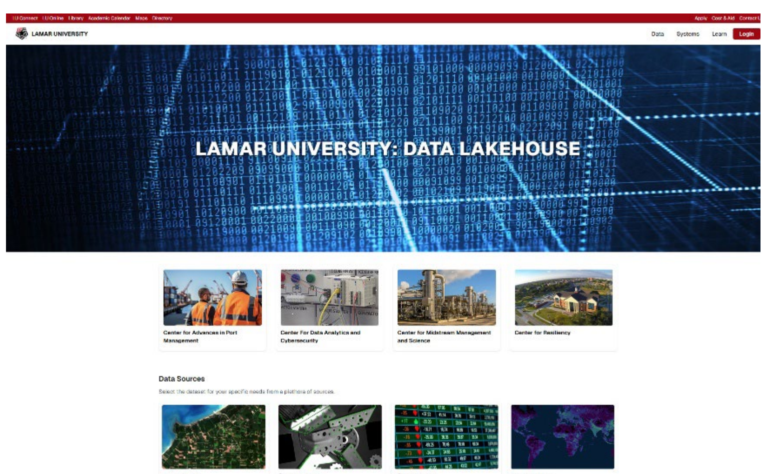


Figure 7. Frontend data upload interface

SETX Water Monitoring: Users and stakeholders can view detector locations in an interactive map of the Beaumont area, showcasing three detector sites (Figure 8 - Figure 13).

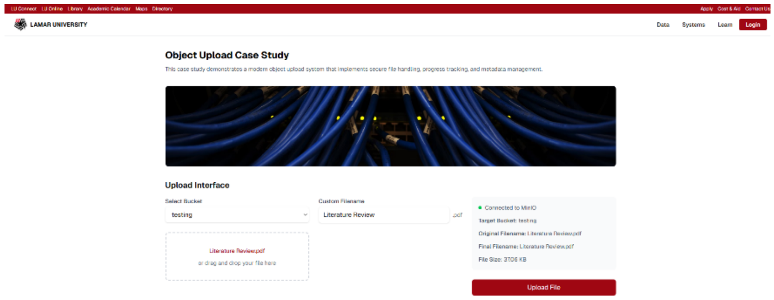


Figure 8: Map view of detector locations

The raw consolidated data can be viewed using the CSV viewer, updated every 20 minutes. The four main properties selected for graph visualization are pH value, Turbidity/NTU, Dissolved Oxygen, and Algal Pigment Fluorescence, presented in a normalized time series chart.

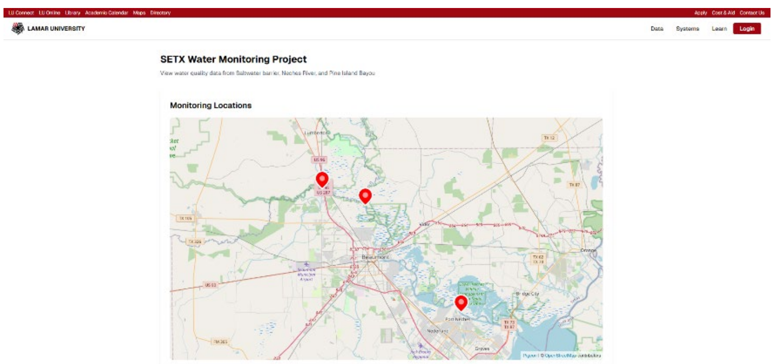
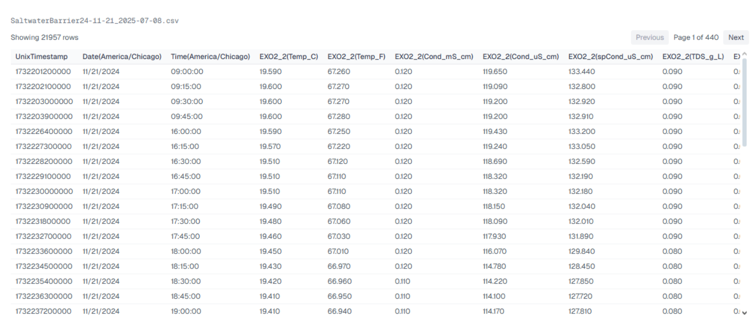


Figure 9. Data viewer interface



SaltwaterBarrier24-11-21_2025-07-08.csv

Showing 21957 rows

Previous Page 1 of 440 Next

UnixTimestamp	Date(America/Chicago)	Time(America/Chicago)	EXO2_2(Temp_C)	EXO2_2(Temp_F)	EXO2_2(Cond_mS_cm)	EXO2_2(Cond_uS_cm)	EXO2_2(spCond_uS_cm)	EXO2_2(TDS_g_L)	E
1732201200000	11/21/2024	09:00:00	19.590	67.260	0.120	119.650	133.440	0.090	0.
1732202100000	11/21/2024	09:15:00	19.600	67.270	0.120	119.090	132.800	0.090	0.
1732203000000	11/21/2024	09:30:00	19.600	67.270	0.120	119.200	132.920	0.090	0.
1732203900000	11/21/2024	09:45:00	19.600	67.280	0.120	119.200	132.910	0.090	0.
1732226400000	11/21/2024	16:00:00	19.590	67.250	0.120	119.430	133.200	0.090	0.
1732227300000	11/21/2024	16:15:00	19.570	67.220	0.120	119.240	133.050	0.090	0.
1732228200000	11/21/2024	16:30:00	19.510	67.120	0.120	118.690	132.590	0.090	0.
1732229100000	11/21/2024	16:45:00	19.510	67.110	0.120	118.320	132.190	0.090	0.
1732230000000	11/21/2024	17:00:00	19.510	67.110	0.120	118.320	132.180	0.090	0.
1732230900000	11/21/2024	17:15:00	19.490	67.080	0.120	118.150	132.040	0.090	0.
1732231800000	11/21/2024	17:30:00	19.480	67.060	0.120	118.090	132.010	0.090	0.
1732232700000	11/21/2024	17:45:00	19.460	67.030	0.120	117.930	131.890	0.090	0.
1732233600000	11/21/2024	18:00:00	19.450	67.010	0.120	116.070	129.840	0.080	0.
1732234500000	11/21/2024	18:15:00	19.430	66.970	0.120	114.780	128.450	0.080	0.
1732235400000	11/21/2024	18:30:00	19.420	66.960	0.110	114.220	127.850	0.080	0.
1732236300000	11/21/2024	18:45:00	19.410	66.950	0.110	114.100	127.720	0.080	0.
1732237200000	11/21/2024	19:00:00	19.410	66.940	0.110	114.170	127.810	0.080	0.

Figure 10. Graph visualization for different time scales

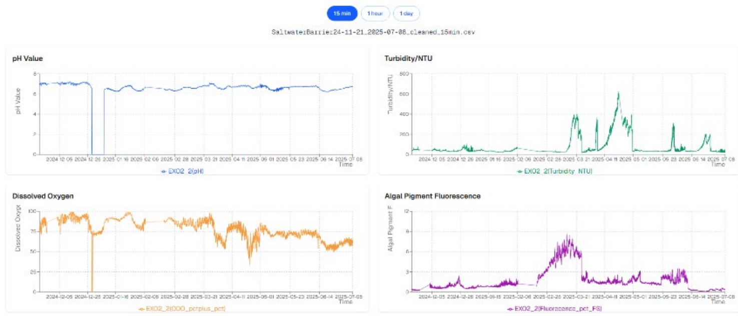


Figure 11. Graph visualization with camera dialog

As cameras are available at detector sites, photos are included as part of the visualization, time-synced to the time series chart.

LSTM Application: Using the object store and a lightweight LSTM model, we built an interactive LSTM forecast allowing users to change parameters on the fly—dataset, column to train on, number of years, window size, LSTM units, epochs, learning rate, and activation function.

Figure 12: Interactive LSTM forecast

Administrative Operations: The MinIO Console is used for access control, bucket management, and health checks. Administrators can change policies including Access Control, Encryption, Replication, Object Locking, Tagging, Quota, and Versioning.

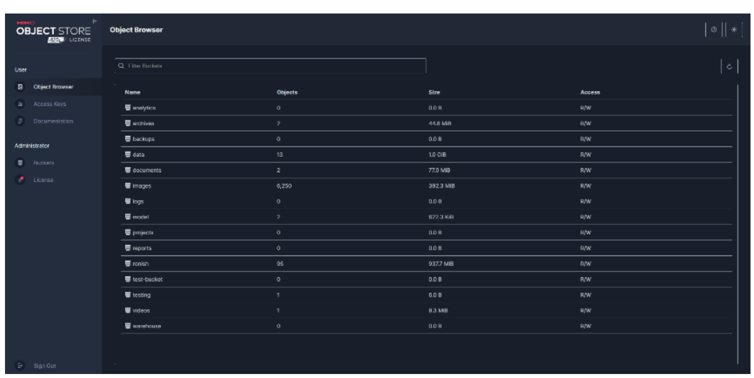


Figure 13. MinIO Store Console - Bucket Settings

5.3 Proposed Improvements

To strengthen reliability and performance, we propose expanding the object store deployment from a single-node VM setup to a 4-VM cluster, with 2 VMs dedicated for redundancy. The upgraded configuration will:
1. Increase fault tolerance through data replication across redundant nodes, ensuring continuity even if one or more VMs fail.

2. Boost throughput by distributing storage and compute requests across multiple VMs, improving parallelism for faster ingestion and retrieval.

3. Enable erasure coding, leveraging MinIO's erasure coding to balance storage efficiency with resiliency.
4. Prepare for scaling by laying the foundation for horizontal scaling where additional VMs can be added without redesigning the storage layer (Figure 14 and Figure 15).

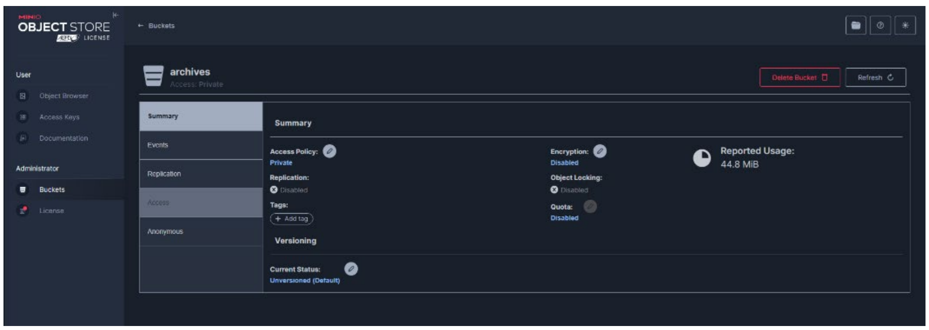


Figure 14. 4 VM setup for redundant object store

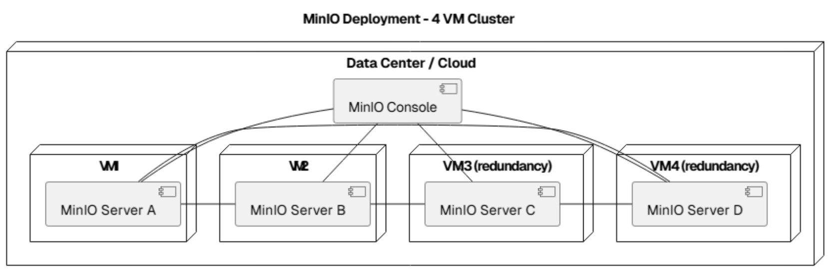


Figure 15. Upgrade performance change

Additionally, we propose a dual-path learning loop that uses the object store as the hub for both fast feedback and slower, high-accuracy modeling. Incoming data in transit will trigger lightweight fine-tuning of a smaller model for rapid feedback to detectors, while complete data accumulated in the store will train a more accurate model for planning and analysis.

Currently, Lamar University does not have a subscription to AWS S3, the industry-leading cloud storage service. As a result, a direct performance comparison between the proposed data lakehouse and AWS S3 is not feasible within the scope of this project. Instead, we plan to systematically benchmark the proposed system against widely adopted open-source alternatives, such as Ceph.

5.4 Validation

The system was validated through multiple approaches:
Integrity Verification: The test harness performed SHA256 hash comparisons between uploaded and downloaded files across three size categories (empty, 1 MB, 10 MB), with all integrity checks passing successfully.

Functional Validation: All seven phases of the test harness completed without errors, covering bucket creation, object upload, listing, retrieval, integrity check, deletion, and cleanup operations. This systematic validation confirms that the object store maintains data integrity and operational reliability.

User Acceptance: Non-technical contributors successfully uploaded data through the web portal without training, validating the accessibility of the interface. Power users confirmed that the Python utility's progress tracking and automatic bucket creation met their workflow requirements.

Pipeline Validation: The water monitoring pipeline demonstrated consistent 20-minute update cycles with reliable data consolidation from three detector sites, confirming the system's suitability for near-real-time data workflows.

6. Conclusion

We deployed an S3-compatible MinIO object store as the storage layer of the Lamar University Lakehouse and exposed it through two primary ingestion paths: a simple website upload page for non-technical contributors and a Python utility for programmatic users. A lightweight harness verifies core operations during upgrades. Early use cases in water monitoring and ML research demonstrate that centralizing raw and prepared artifacts in the object store reduces friction for research teams and shortens time-to-analysis.

In addition to supporting research workflows, deployment creates an educational resource that integrates seamlessly into teaching and training. By providing both web-based and programmatic access, the system allows students to practice data management, analytics, and machine learning tasks in an authentic environment, while giving instructors a platform to design assignments that reflect real-world practices.

All four research objectives have been met: (1) the MinIO object store was successfully deployed and configured as the foundational storage layer; (2) dual ingestion paths were implemented through the web portal and Python utility; (3) a comprehensive testing harness was developed for operational validation and regression detection; and (4) case studies in water monitoring and CMMS workflows demonstrated the system's dual value in research and education.

The unique contribution of this work lies in providing a complete, replicable framework that bridges research infrastructure with instructional goals. This dual function positions the Lakehouse as a transferable model that other universities can adopt to enhance curriculum integration, hands-on experience, and institutional data strategy.

The proposed Data Lakehouse project offers a compelling environment for both research and pedagogy. We plan to invite Lamar University faculty to pilot the system and provide structured feedback. Additionally, the lakehouse architecture will be incorporated as a case study in senior- and graduate-level courses in computer networks and distributed systems. Quantitative evaluation metrics will be developed, and relevant data will be collected to systematically assess system performance.

Acknowledgements

The authors gratefully acknowledge the support of LU CMMS for this work. Additionally, this research has been supported by the U.S. Department of Energy (DoE) under Grant No. DE-CR0000035. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the DoE.

References

Arizona State University and OpenAI, ChatGPT Enterprise partnership in higher education, Available: https://www.asu.edu/news/2024/openai-partnership, 2024.

Armbrust, M., Das, T., Sun, L., Yavuz, B., Zhu, S., Murthy, M., Torres, J., van Hovell, H., Ionescu, A., Luszczak, A., Switakowski, M., Szafranski, M., Li, X., Ueshin, T., Mokhtar, M., Boncz, P., Ghodsi, A., Paranjpye, S., Senster, P., Xin, R. and Zaharia, M., Delta Lake: High-performance ACID table storage over cloud object stores, Proceedings of the VLDB Endowment, vol. 13, no. 12, pp. 3411-3424, 2020.

Castleman, B. and Page, L., Summer nudging: Can personalized text messages and peer mentor outreach increase college going among low-income high school graduates?, Georgia State University, Tech. Rep., 2014.

City of New York, NYC Artificial Intelligence Action Plan, Available: https://www.nyc.gov/assets/oti/downloads/pdf/ai-action-plan.pdf, 2024.

City of San Jose, Generative AI guidelines and algorithmic transparency inventory, Available: https://www.sanjoseca.gov/your-government/departments-offices/information-technology/artificial-intelligence, 2024.

Gama, J., Zliobaite, I., Bifet, A., Pechenizkiy, M. and Bouchachia, A., A survey on concept drift adaptation, ACM Computing Surveys, vol. 46, no. 4, pp. 44:1-44:37, 2014.

Internal Revenue Service, AI and analytics in compliance enforcement, Available: https://www.irs.gov/newsroom/irs-expands-ai-use-for-compliance, 2023.

Joshi, A., Subedi, N., Bista, H. and Ishwar, K. C., Harnessing the tweet stream: Deep learning for natural disaster detection, Machine Learning for Social Transformation, Springer Nature Singapore, pp. 85-97, 2024.

Marz, N. and Warren, J., Big Data: Principles and Best Practices of Scalable Realtime Data Systems, Manning, Greenwich, CT, USA, 2015.

Mondal, A. S., Sanyal, M., Barua, H. B., Chattopadhyay, S. and Mondal, K. C., Comparative analysis of object-based big data storage systems on architectures and services: A recent survey, Journal of the Institution of Engineers (India): Series B, 2024.

National Institute of Standards and Technology, AI Risk Management Framework (AI RMF 1.0) and Generative AI Profile, Available: https://www.nist.gov/itl/ai-risk-management-framework, 2023.

Purdue University, Research Computing, Depot Object Store, Available: https://www.rcac.purdue.edu/storage/object, 2025.

Stanford University IT, AWS S3 Object Storage, Available: https://uit.stanford.edu/storage/aws-s3-object-storage, 2025.

Storage Networking Industry Association (SNIA), What is object storage?, Available: https://www.snia.org/education/what-is-object-storage, 2025.

UK Government, Department of Health and Social Care, NHS AI diagnostic fund deployment and trial expansions, Available: https://www.england.nhs.uk/ai-lab/ai-diagnostic-fund, 2025.

U.S. Department of the Treasury, Payment integrity and fraud detection report, Available: https://home.treasury.gov/news/press-releases/jy1805, 2023.

U.S. Food and Drug Administration, Artificial intelligence and machine learning (AI/ML)-enabled medical devices, Available: https://www.fda.gov/medical-devices/software-medical-device-samd/artificial-intelligence-and-machine-learning-software-medical-device, 2024.

U.S. Office of Management and Budget, M-24-10: Advancing governance, innovation, and risk management for agency use of artificial intelligence, Available: https://www.whitehouse.gov/omb/information-for-agencies/m-24-10/, 2024.

University of Wisconsin-Madison, Division of Information Technology, Research Object Storage (S3), Available: https://it.wisc.edu/services/research-object-storage-s3/, 2025.

Biographies

Naman Subedi is a graduate research assistant at Lamar University, where he is pursuing an MS in Computer Science. His research focuses on applying machine learning to diverse domains, including data storage, image segmentation, and user experience. He has contributed to projects on invasive species detection using drone imagery, data lakehouse architectures for IoT-driven water monitoring, and object storage optimization. In addition to his technical research, Naman has led product design initiatives for startups and academic events, with a focus on creating user-centered systems that integrate advanced computation with accessible interfaces.

Md Masud Rana is an Assistant Professor of Computer Science at Lamar University. He earned his Ph.D. from the University of Technology Sydney. Dr. Rana is an expert in secure software engineering and a researcher specializing in cybersecurity and penetration testing. His research interests include offensive and defensive cybersecurity, artificial intelligence, IoT, nano particles, quantum dots, graph-based learning, and quantum machine learning.

Bo Sun received his Ph.D. degree in Computer Science from Texas A&M University, College Station TX, in 2004. He is currently a Professor in the Department of Computer Science at Lamar University. His research interests include various topics in Wireless Networks and Network Security (with a focus on Intrusion Detection). He has published about 80 journal and conference papers. His research has been supported by Lamar University, Texas ARP award, and the U.S. National Science Foundation.

Frank Sun is currently an Instructor and System Administrator in the Department of Computer Science at Lamar University. He holds an M.S. in Computer Science from the University of Houston. His professional and research interests include Linux and Unix system administration, as well as cybersecurity.

Helen H. Lou, a Fellow of American Institute of Chemical Engineering (AIChE), is the Director for the Center for Data Analytics and Cybersecurity (CDAC) and a Professor in the Dan F. Smith Department of Chemical Biomolecular Engineering at Lamar University. She also serves as the Faculty Advisor for the Center for Midstream Management and Science. Dr. Lou earned her Ph.D. in Chemical Engineering and M.A. in Computer Science from Wayne State University in 2001. She is a registered Professional Engineer in Texas.

Alek Hutson is the Associate Director of the Center for Midstream Management and Science (CMMS) at Lamar University. He leads workforce development initiatives and applied research addressing practical challenges in the midstream energy sector. Dr. Hutson holds degrees in Physics and Mathematics from Lamar University and earned his Ph.D. in Physics from the University of Houston through the ALICE collaboration at CERN. In addition to his academic role, Dr. Hutson is the CEO and co-founder of Labor Force Solutions (LFS), where he develops innovative safety and training programs for the heavy transport industry.