Hackathon Problem Statement

Data in Motion — Building the Future of Intelligent Cloud Storage

As organizations increasingly adopt hybrid and multi-cloud environments, they face challenges in efficiently managing, accessing, migrating, and analyzing large volumes of distributed data. Ensuring scalability, performance, cost efficiency, and real-time responsiveness is key to building resilient cloud data architectures.

Your challenge is to design and implement an intelligent data management solution that can dynamically analyze, tier, and move data across different storage environments (on-premise, private cloud, and public cloud), while also handling real-time data streams to enable continuous insight generation.

Your Solution Should Aim to:

1. Optimize Data Placement:

Build a mechanism that automatically determines where a given dataset should reside depending on:

- Access frequency (hot, warm, cold data)
- Latency requirements
- Cost per GB
- Predicted future access trends

2. Enable Multi-Cloud Data Migration:

Demonstrate a prototype or provide a structured plan that outlines how data can be migrated or synchronized across multiple cloud environments while ensuring security, performance efficiency, and minimal disruption.

3. Integrate Real-Time Data Streaming:

Use a streaming technology such as Apache Kafka or MQTT to simulate continuous data flow. Your system should be able to process or react to data while it is actively moving through the environment.

4. Enable Predictive Insights:

Integrate a machine learning component that learns data usage patterns and recommends pre-emptive data movements or storage class changes.

5. Ensure Data Consistency and Availability:

Your solution must support synchronization across distributed environments and handle cases of network failures or data conflicts gracefully.

6. Provide a Unified Dashboard:

Create an intuitive web or command-line interface that visualizes data distribution, performance metrics, and streaming or migration activity.

Bonus Tracks (Optional Enhancements):

- Implement data encryption and access control policies that adapt to storage location.
- Include containerized deployment (e.g., Docker, Kubernetes) to simulate multi-cloud scalability.
- Integrate with cloud APIs (AWS S3, Azure Blob, GCP Storage) for realistic testing.
- Add automated alerting or policy triggers based on cost, latency thresholds, or streaming data patterns.

Tech Stack Guidelines (Indicative Only):

- Backend: Python, Java, or Go
- Data Layer: SQLite, MongoDB, or JSON-based simulation
- Streaming: Kafka or MQTT
- ML Framework (Optional): TensorFlow, PyTorch, or scikit-learn

- Cloud Simulation: AWS Free Tier, Azure, GCP, or local mock APIs

Expected Deliverables:

1. A working prototype (web app, CLI, or microservice) demonstrating the solution.
2. A short technical presentation covering:
- Problem understanding and architecture
- Data management logic and streaming or migration approach
- Performance insights or simulation results
- Future scalability roadmap

Judging Criteria:

Category	Weightage
Innovation and Relevance to NetApp’s Domain	25%
Technical Depth and Architecture	25%
Scalability, Efficiency, and Data Handling	20%
User Experience (UI/UX or Usability)	15%
Presentation and Clarity of Thought	15%

Why This Matters:

In the era of data-driven intelligence, organizations are challenged not by the lack of storage capacity, but by the need for intelligent, automated, and adaptive data management. Your solution will showcase your ability to think like a next-generation cloud and data engineer — optimizing performance, automating decisions, and driving innovation — exactly what NetApp’s technology ecosystem thrives on.