PHASE 2

Data Backup and Recovery with Using IBM Cloud Object Storage

PHASE 3 – DESIGN A SOLUTION BLUEPRINT

College Name : AMRUTA INSTITUTE OF ENGINEERING & MANAGEMENT SCIENCES

Group Members :

- **Name :** RAKESH G V
  CAN ID NUMBER : CAN_33908760
- **Name :** SANJAY K A
  CAN ID NUMBER : CAN_33846529
- **Name :** SHEKAR N R
  CAN ID NUMBER : CAN_33857769
  Work : Service Integration
- **NAME :** NANDAN M G
  CAN ID NUMBER:CAN_33849313

Tools and Platforms Selection

For the **Data Backup and Recovery System**, we selected tools and platforms that ensure scalability, reliability, and security:

- **IBM Cloud Object Storage**:
  - Serves as the primary storage solution for unstructured data backups.
  - Offers **99.999999999% durability** and **scalability**.
  - Supports storage classes like **Standard, Vault, and Cold Vault** to optimize cost-efficiency.
  - Provides **AES 256-bit encryption** to secure stored data.
  - Supports **cross-region replication** for disaster recovery.
- **IBM Cloud SDKs**:
  - Used **IBM COS SDK for Node.js** to integrate IBM Cloud Object Storage into the backend.
  - Simplified file operations including **upload, download, and deletion**.
  - Ensured **high availability** through regional redundancy.
- **Express.js**:
  - Lightweight **backend framework** used for server-side logic.

- Created **RESTful APIs** to facilitate interactions between the frontend and storage backend.
- Supports **JWT-based authentication** for secure API calls.

- **React.js**:
  - Built a **responsive frontend interface** for managing files.
  - Users can **upload, download, and delete files** seamlessly.
  - Implemented **role-based access control (RBAC)** for user management.
- **Multer**:
  - Middleware for handling file uploads in the backend.
  - Ensures smooth integration with IBM Cloud Object Storage.
  - Supports **file size restrictions** to prevent excessive data storage.

Solution Development Process

Step 1: Provisioning Resources

- **Created an IBM Cloud Object Storage instance** and configured a **bucket** (databackupandstoragesystem) for file storage.
- **Set up IAM roles and API keys** to restrict access and enhance security.
- **Enabled Multi-Factor Authentication (MFA)** for additional security.

Step 2: Backend Development

- **Developed the backend using Express.js** with the following functionalities:
  - **Upload:** Files uploaded via putObject API.
  - **Download:** Files retrieved via getObject API.
  - **Delete:** Files permanently removed using deleteObject API.
- **RESTful API Endpoints**:
  - POST /upload – Uploads a file.
  - GET /download/:filename – Downloads a file.
  - DELETE /delete/:filename – Deletes a file.
  - GET /files – Lists all files.
  - POST /restore/:filename – Restores soft-deleted files.

Step 3: Frontend Development

- Built a **React.js frontend** with:
  - **File Upload Interface** – Users can upload files.
  - **File List Dashboard** – Displays stored files with download and delete options.
  - **Download and Delete Features** – Users can manage stored data easily.
  - **File Preview Feature** – Users can preview supported file types before downloading.

Step 4: Security Measures

- **Enabled HTTPS** for encrypted data transmission.
- **Implemented IAM-based access control** to secure storage buckets.
- **Applied input validation** to prevent malicious file uploads.
- **Configured automatic virus scanning** for uploaded files.

API and SDK Integration

IBM COS SDK for Node.js

- Integrated SDK for smooth file operations:
  - putObject – Uploading files.
  - getObject – Retrieving files.
  - deleteObject – Deleting files.
  - restoreObject – Restoring soft-deleted files.

Error Handling

- **Implemented error handling mechanisms**:
  - **Retry logic** for transient errors.
  - **Proper responses** for invalid file operations.
  - **Auto-recovery mechanism** for interrupted file transfers.

Testing and Bug Fixes

Functional Testing

- **File Upload:** Verified for different file types and sizes.
- **File Download:** Ensured downloaded files maintain integrity.
- **File Deletion:** Checked for proper error handling on non-existent files.
- **File Restoration:** Tested restoring soft-deleted files.

Stress Testing

- Simulated **high traffic loads** to monitor performance.
- Used **IBM Cloud Monitoring** to detect potential bottlenecks.
- **Tested concurrent file operations** with up to **1000 requests per second**.

Error Simulation

- **Tested scenarios such as**:
  - Network interruptions.
  - Invalid API requests.
  - Storage capacity limits.
  - **Unsuccessful backup restoration attempts**.

Bug Fixes

- Addressed **incorrect file paths, API timeouts, and UI inconsistencies**.
- **Fixed authentication issues** related to JWT tokens.

Performance Measurement and Optimization

Monitoring Tools

- **IBM Cloud Monitoring**:
  - Tracked metrics including **upload/download times** and **API response rates**.
  - Set up **real-time dashboards** for performance insights.
- **Analytics Dashboards**:
  - Displayed daily file usage statistics.
  - Monitored **error rates and storage trends**.
  - **Generated weekly reports** for system administrators.

Alerting and Notifications

- Configured alerts for:
  - **Storage nearing capacity**.
  - **API failures and unusual traffic spikes**.
  - **Unauthorized access attempts**.

Optimization Strategies

- **Enhanced file transfer efficiency** by:
  - Implementing **chunked uploads for large files**.
  - Enabling **compression** for downloads.
  - **Utilizing CDN caching** for faster content delivery.

- **Cost optimization** by:
  - Moving rarely accessed files to **Cold Vault storage**.
  - **Automating file cleanup** for redundant files.

Conclusion and Future Enhancements

The **Data Backup and Recovery System** was successfully designed and implemented using **IBM Cloud Object Storage, Express.js, and React.js**. The system provides **scalable, secure, and efficient file management** with robust performance monitoring.

Future Enhancements:

1. **File Versioning** – Maintain multiple versions of files for backup history.
2. **Soft Deletion** – Introduce a recovery window before permanent deletion.
3. **Automated Backups** – Schedule periodic file backups for data redundancy.
4. **AI-Based Anomaly Detection** – Detect unusual patterns in file access and alert administrators.
5. **Blockchain Integration** – Ensure data integrity with a tamper-proof transaction ledger.

This project demonstrates the **reliability and efficiency** of cloud-based backup solutions, setting a foundation for further scalability and innovation.