Phase 4

Project Title: *Data Backup and Recovery System Using Cloud Object Storage*

1. Overview of Backup and Recovery Implementation:

In today's digital landscape, data is a critical asset for any organization. This project aims to establish a robust, scalable, and secure backup and recovery system by utilizing **IBM Cloud Object Storage**.

The system ensures:

- **Data Durability**: Reliable cloud storage to prevent data loss due to hardware failures.
- **Data Accessibility**: Seamless access to backups for quick recovery.
- **Automation**: Minimized manual effort through scripts and scheduled jobs.

The project provides a complete lifecycle, from backup creation to automated monitoring and future-ready disaster recovery strategies.

2. Configuring IBM Cloud Object Storage:

a. Create and Configure Buckets

Step 1: Log in to the IBM Cloud Dashboard

- Visit the IBM Cloud website and sign in using your credentials.
- If you don't have an account, create one by following the registration process.

Step 2: Navigate to the "Object Storage" Section

- Locate the Search bar at the top of the dashboard and type Object Storage.
- Click on the Object Storage service.
- If Object Storage is not provisioned in your account, click Create Resource and select Cloud Object Storage from the catalog.
- Choose the appropriate Region (e.g., *us-south*) and Resource Group to proceed.

Step 3: Create a Storage Bucket

- Inside the Object Storage service dashboard, navigate to the Buckets tab and click Create Bucket.
- Enter a unique name for your bucket. Remember that bucket names must be globally unique across all IBM Cloud users.
- Choose the desired Storage Class: Standard, Vault and Cold Vault
- Specify the Location for your data:
  - Regional: Stores data in a specific region (e.g., *us-south*).
  - Cross-Regional: Distributes data across multiple regions for higher availability.
  - Single-Site: Stores data in a single data center.

Step 4: Apply Access Control and Encryption Settings

- Access Control:
  - Use IBM Cloud IAM (Identity and Access Management) to assign roles (e.g., Viewer, Editor) to users or service IDs.
  - Ensure buckets are private unless explicitly required for public access.

        - Apply fine-grained policies, such as read-only or write-only permissions for specific users or services.
    - Encryption Settings:
        - Enable Server-Side Encryption (SSE) to encrypt objects automatically when they are stored in the bucket.
        - Use IBM-managed encryption keys or your own keys via IBM Key Protect or Hyper Protect Crypto Services for enhanced security.
        - Verify encryption status in the bucket settings.

b. **Integrating Backup Scripts**

   Step 1: Using PowerShell Backup Scripts

   Your project uses automated PowerShell scripts to create backups of MongoDB databases and upload them to the configured bucket.

    - **Backup Process**:
        - MongoDB data is dumped locally using mongodump with the connection string.
        - Data is compressed into a zip file.
        - The zip file is uploaded to the IBM Cloud Object Storage bucket.
    - **Script Validation**:
        - Ensure that the bucket name, API key, and endpoint are correctly configured in the backup script.
        - Test the script to confirm successful uploads to the bucket.

   Step 2: Using Recovery Scripts

    - The recovery script is used to download backup files from the bucket and restore MongoDB databases.
    - Ensure proper permissions are applied to the recovery script to allow downloading from the bucket.
    - Verify the bucket structure and file naming conventions to simplify restoration.

c. **Enabling Monitoring and Logging**

   To ensure the backup and recovery system functions as expected, enable monitoring and logging for your storage buckets.

    - **Monitoring Setup:**
        - Go to the **Metrics Routing** section in IBM Cloud.
        - Configure metrics to track storage usage, API request volume, and performance data.
        - Define routing rules to send metrics to an IBM Cloud Monitoring instance.

3. Automating Backup and Recovery with Schedulers:

3.1 Backup Scripts

The backup process is automated using a PowerShell script to capture MongoDB data, compress it into a .zip file, and upload it to IBM Cloud Object Storage.

Key Features of the Script:

- Backup MongoDB databases using the mongodump command.
- Use secure credentials to interact with MongoDB and IBM Cloud Object Storage.
- Compress backups for storage efficiency.
- Automatically upload the compressed backup to the cloud.

Script Details:

```
# Backup Script with MongoDB Connection String
# Define variables for backup folder, file, and bucket details
$BackupFolder = "C:\Mongo-Backups"
$BackupFile = "Mongo-Backups.zip"
$BucketName = "mongodb-data-backup-recovery"
$CosKey = "Mongo-Backups.zip"
$DesktopPath = "C:\Users\Asus\OneDrive\Desktop\Mongo-Backups.zip"

# MongoDB connection string
$MongoConnectionString =
"mongodb+srv://<username>:<password>@cluster0.mongodb.net/"

# Ensure backup folder exists
if (-not (Test-Path $BackupFolder)) {
   New-Item -ItemType Directory -Path $BackupFolder
}

# Create a MongoDB backup
mongodump --uri=$MongoConnectionString --out $BackupFolder

# Compress the backup directory
Compress-Archive -Path "$BackupFolder\*" -DestinationPath $DesktopPath -Force

# Upload to IBM Cloud Object Storage
ibmcloud cos upload --bucket $BucketName --key $CosKey --file $DesktopPath
Write-Host "Backup and upload completed successfully!"
```

3.2 Recovery Scripts

The recovery process is automated using PowerShell, enabling seamless restoration of data from IBM Cloud Object Storage.

Key Features of the Script:

- Download backups from the configured cloud bucket.
- Extract the .zip file and restore the database using mongorestore.
- Overwrite the existing database if required.

Script Details:

```
# Recovery Script
# Variables
$bucketName = "mongodb-data-backup-recovery"
$objectKey = "Mongo-Backups.zip"
$downloadPath = "C:\Users\Asus\OneDrive\Desktop\Mongo-Backups.zip"
$extractPath = "C:\Mongo-Backups\Mongo-Backups"
$apiKey = "<Your IBM Cloud API Key>"

# Authenticate using IBM Cloud CLI
ibmcloud login --apikey $apiKey

# Download the backup
ibmcloud cos download --bucket $bucketName --key $objectKey $downloadPath

# Extract the backup
Expand-Archive -Path $downloadPath -DestinationPath $extractPath -Force

# Restore MongoDB backup
mongorestore --uri="mongodb+srv://<username>:<password>@cluster0.mongodb.net/" --
drop --dir="$extractPath"
Write-Host "Recovery process completed successfully!"
```

3.3 Scheduling Scripts

To ensure regular execution of backup and recovery tasks, the scripts can be scheduled using built-in tools like Task Scheduler in Windows.

Windows Task Scheduler:

1. Open **Task Scheduler** from the Start menu.
2. Click **Create Basic Task** and provide a name for the task
3. Choose the frequency for the task.
4. Select **Start a Program** as the action.
5. In the Program/Script field, enter the path to the PowerShell executable.
6. Add Arguments field.
7. Click Finish. The backup script will now run on the chosen schedule.

4. Monitoring and Alerts:

Configuring Monitoring Metrics

To track the health and usage of backups, **IBM Cloud Monitoring** is configured for the storage bucket.

Steps:

- **Enable Metrics on the Bucket**:
  - Go to the bucket settings in the Object Storage dashboard.
  - Enable **Usage Metrics** and **Request Metrics** to monitor operations.
- **Set Up Metrics Routing**:
  - Navigate to **Metrics Routing** under **Observability** in the IBM Cloud dashboard.
  - Configure routing to a **Monitoring Instance** for analyzing data trends.
- **Configure Alerts**:
  - Set thresholds for storage utilization and operation failures.
  - Enable notifications via email or Slack for real-time updates.

5. IBM Cloud Platform Features and Considerations:

Feature	**Benefits**	**Best Practices**
Scalability	Handles large datasets with automatic scaling.	Monitor usage and set alerts for proactive scaling during peak loads.
Security	Protects sensitive data with IAM policies, encryption, and access control.	Use least privilege policies to minimize access.
Monitoring	Provides insights into storage operations and trends over time.	Set up thresholds for storage utilization, failed operations, and abnormal activity.
	Helps identify anomalies or unusual patterns.	Use IBM Cloud Monitoring dashboards to analyze and visualize metrics.
Cost Efficiency	Reduces expenses through optimal storage class selection and lifecycle policies.	Choose storage classes based on data access patterns (e.g., Standard, Vault, Cold Vault)
Resilience	Ensures data availability through cross-regional or multi-zone redundancy.	Use cross-regional replication for critical data to protect against regional outages.
	Guarantees uptime and availability even during hardware failures or natural disasters.	Regularly test recovery scenarios to validate disaster recovery and business continuity plans.

6. Future Enhancements:

- **Disaster Recovery Planning**: Enable multi-region backups and redundant storage for high availability during disasters. Regular simulations will ensure system reliability.
- **Advanced Monitoring**: Develop dashboards for backup metrics and dynamic alerts. Use predictive analytics to address issues proactively.
- **Enhanced Security**: Incorporate AES-256 encryption, role-based access control, and immutable backups for maximum data protection.
- **Data Versioning**: Enable storage bucket versioning for point-in-time recovery. Use lifecycle policies to manage older backups efficiently.
- **Dynamic Automation**: Automate backup schedules based on system events. Include validation checks post-recovery for seamless operation.
- **User-Friendly Interface**: Build a React-based dashboard or mobile app for monitoring backups and initiating recoveries with ease.
- **CI/CD Integration**: Integrate the backup system with CI/CD pipelines for automated pre-deployment backups and post-recovery validation.
- **Analytics and Reporting**: Generate reports on storage usage, backup success rates, and costs. Include audit logs for compliance.
- **Support for More Databases**: Extend support to MySQL, PostgreSQL, and Oracle for a wider range of use cases.
- **System Reliability Testing**: Introduce chaos engineering to test resilience and automate failover mechanisms for uninterrupted service.

7. Conclusion:

The Data Backup and Recovery System project leverages the scalability, security, and efficiency of IBM Cloud Object Storage to safeguard critical data. With automated backup and recovery scripts, advanced monitoring, and secure storage practices, this system ensures data durability and quick restoration in case of failures. By implementing features like scheduling, encryption, and integration with cloud services, the project demonstrates a robust solution tailored for real-world needs. Future enhancements, including disaster recovery planning, advanced analytics, and multi-database support, pave the way for a more resilient and scalable system. This project serves as a cornerstone for organizations to ensure business continuity and data protection in an increasingly data-driven world.