Database Backup, Recovery Strategy and Advice

This document outlines a comprehensive backup and recovery strategy for the project Media Content Management and Browsing API to ensure data integrity, minimize downtime, and maintain business continuity.

1. Overview

The PostgreSQL database stores critical data for the system, including user accounts, profiles, subscriptions, media content, watch histories, admin information, etc. This strategy aims to prevent data loss due to hardware failures, human error, software bugs, or cyberattacks.

2. Backup Strategy

- **Frequency:**
  - **Full Backups:** Weekly (e.g., every Sunday at 2:00 AM)
  - **Incremental Backups:** Daily (e.g., every day at 3:00 AM)
  - **On-Demand Backups:** Before major updates, deployments, or planned maintenance.
- **Retention:**
  - **Full Backups:** Retain the last 4 full backups (one month).
  - **Incremental Backups:** Retain the last 30 incremental backups.
- **Storage:**
  - **Primary:** Cloud storage (AWS S3, Google Cloud Storage, Azure Blob Storage) with encryption for security and data protection.
  - **Secondary:** Off-site storage (external hard drives, tapes) for disaster recovery.
- **Types:**
  - **Logical Backups:** Using `pg_dump` for flexibility and ease of restoration.
  - **Physical Backups:** Using `pg_basebackup` for faster recovery in some cases.
  - **Transaction Log Archiving (WAL):** Enabled for point-in-time recovery.

3. Backup Protocol

1. **Preparation:**
   - Set up a dedicated server or a cloud-based solution for backup operations.
   - Configure WAL archiving in `postgresql.conf`:

```
archive_mode = on
archive_command = 'cp %p /path_to_archive/%f'
```

2. **Full Backups:**
   - Schedule weekly backups using `cron` or a similar scheduler:

```
0 2 * * 0 /path/to/backup_script.sh
```

- Backup script:

```
#!/bin/bash
pg_dump -U <username> -d <database_name> -F c -b -v -f
"/path/to/backups/full_backup_$(date +%F).backup"
```

3. **Incremental Backups:**
   - Use tools like Barman or pgBackRest for efficient incremental backups.
4. **Verification:**
   - Regularly test backup integrity by restoring to a test environment:

```
pg_restore -l /path/to_backup/full_backup_$(date +%F).backup
```

5. **Retention Policy:**
   - Automate backup removal using scripts or cloud lifecycle policies.

4. Recovery Strategy

1. **Full Database Recovery:**
   - Use `pg_restore` for logical backups:

```
pg_restore -U <username> -d <database_name> -v /path/to_backup/full_backup_$(date
+%F).backup
```

2. **Incremental Recovery:**
   - Restore the latest full backup and then apply subsequent incremental backups.
3. **Point-in-Time Recovery (PITR):**
   - Utilize WAL archives to recover to a specific point in time.
4. **Testing:**
   - Regularly test recovery procedures in a controlled environment.

5. Preventing Downtime

Replication:

- **Implementation:** Implement PostgreSQL replication (e.g., streaming replication) for high availability.
- **Benefits:** Provides a standby server ready to take over in case of primary server failure.

Read-Only Mode:

- **During Backups:** Schedule backups during off-peak hours or transition the primary server to read-only mode during backups to minimize impact on user experience.

Monitoring:

- **Implementation:** Use monitoring tools (e.g., pgAdmin, Zabbix, Prometheus) to track database performance, identify potential issues, and receive alerts.

Maintenance Windows:

- **Scheduling:** Schedule maintenance activities (including backups and updates) during off-peak hours or dedicated maintenance windows to minimize downtime.

6. Security

- **Encryption:**
  - **Data at Rest:** Encrypt backups both in transit and at rest using strong encryption algorithms.
  - **Data in Motion:** Use secure protocols (e.g., HTTPS/SSL) for data transfer.
- **Access Control:**
  - **Restrict Access:** Restrict access to backup files and scripts to authorized personnel.
  - **Strong Authentication:** Implement strong authentication mechanisms for accessing backup systems.
- **Regular Security Audits:**
  - **Conduct Audits:** Regularly audit security logs and vulnerability scans to identify and address potential security threats.

7. Documentation

- **Comprehensive Documentation:** Maintain detailed documentation of the entire backup and recovery process, including:
  - Backup schedule and retention policy.
  - Backup locations and storage methods.
  - Automated scripts and their configurations.
  - Recovery procedures and testing results.
  - Contact information for support.
- **Regular Review:** Regularly review and update the backup and recovery strategy based on evolving needs, new technologies, and security best practices.

8. Conclusion

This comprehensive backup, recovery strategy and advice provide a robust framework for protecting the Media Content Management and Browsing System's database. Implementing these measures and conducting regular reviews can significantly minimize the risk of data loss and ensure business continuity.