**AWS Database Migration Service (DMS) is a service designed to migrate databases between different environments, such as on-premises to AWS, or between AWS accounts/regions. Here's how it can be used for backups or cross-account data sharing:**

**Steps for Using AWS DMS for Backup or Cross-Account Sharing**

1. **Setup Source and Target Endpoints:**
   * **Define the source endpoint as the database you want to back up or share.**
   * **Define the target endpoint as the destination database (in the same or a different AWS account/region).**
2. **Replication Instance:**
   * **Launch an AWS DMS replication instance, which is the compute resource used to run the migration/replication tasks.**
3. **Create a Migration Task:**
   * **Create a DMS task to copy the data from the source to the target.**
   * **You can configure DMS for:**
     + **Full Load: To copy all data as a backup.**
     + **Ongoing Replication: To sync ongoing changes in near real-time.**
4. **Use Amazon S3 as a Target (Optional):**
   * **If you want to store the backup in S3 for further processing or cost efficiency, you can configure S3 as the target endpoint.**
5. **Data Validation:**
   * **DMS allows you to validate that the data was copied correctly.**
6. **Restore in Target Account:**
   * **In the destination AWS account, restore the database or continue using the copied data.**

**Comparison: Approach 1 vs. Approach 2**

| **Criteria** | **Approach 1: Cross-Account Snapshot Sharing** | **Approach 2: AWS DMS** |
| --- | --- | --- |
| **Ease of Use** | Simple: Share snapshots via RDS console or API. | Moderate: Requires DMS setup, endpoints, and configuration. |
| **Data Freshness** | Only includes the state of the database at the snapshot time. | Can include real-time replication with ongoing changes. |
| **Storage Cost** | Snapshot storage billed as part of RDS backup storage in S3. | Additional S3 costs if S3 is used as the target. |
| **Compute Cost** | None. AWS handles snapshots natively. | DMS replication instance costs (e.g., $0.048/hour for t3.medium). |
| **Flexibility** | Limited: Snapshots are only RDS-compatible. | High: Can migrate data between different DB types or to S3. |
| **Cross-Region/Account** | Easy: Snapshots can be shared directly across accounts or regions. | Requires configuration of endpoints and inter-region data transfer. |
| **Complexity** | Low: Simple snapshot sharing. | Higher: Requires migration task creation and monitoring. |
| **Cost for Large Databases** | More cost-effective for static backups. | Higher for ongoing replication and larger databases. |
| **Use Cases** | Best for static backups or one-time sharing. | Best for continuous replication or heterogeneous migrations. |

**Cost Analysis Example**

**Assumptions:**

* Database size: 100 GB
* Snapshot storage in S3: $0.095/GB per month (in standard AWS regions).
* DMS instance: dms.t3.medium running for 24 hours.

**Approach 1: Snapshot Sharing**

1. **Snapshot Storage**:
   * 100 GB × $0.095/GB = **$9.50/month**.
2. **Cross-Account Sharing**:
   * No additional cost.
3. **Total Monthly Cost**:
   * **$9.50/month**.

**Approach 2: AWS DMS**

1. **Replication Instance**:
   * dms.t3.medium = $0.048/hour.
   * 24 hours × $0.048 = **$1.15**.
2. **S3 Storage (Optional)**:
   * 100 GB × $0.023/GB = **$2.30** (if S3 is used).
3. **Inter-Region Data Transfer (Optional)**:
   * 100 GB × $0.02/GB = **$2.00** (if cross-region).
4. **Total Cost for 1-Time Migration**:
   * **$3.45** (without S3) or **$5.75** (with S3).

**How AWS DMS Charges:**

* **DMS Cost**:
  + The DMS replication instance cost and associated costs (like temporary storage) are for the **duration of the migration task**. For example, if a migration task takes 24 hours, you only pay for those 24 hours.
  + Example: A dms.t3.medium instance running for 24 hours costs **$1.15** ($0.048/hour × 24 hours).

**Conclusion**

* **Choose Approach 1 (Snapshot Sharing)** if:
  + You need a static backup or one-time data transfer.
  + Cost efficiency is a priority.
  + Simplicity and ease of use are essential.
* **Choose Approach 2 (DMS)** if:
  + You need ongoing replication or real-time synchronization.
  + You’re migrating to a different database type or to S3.
  + Flexibility and data freshness are important.

https://stackoverflow.com/questions/66483418/can-you-use-aws-dms-to-move-aurora-db-from-one-account-to-another

https://docs.aws.amazon.com/prescriptive-guidance/latest/patterns/migrate-an-amazon-rds-db-instance-to-another-vpc-or-account.html

Detailed Steps perform the restoration using AWS DMS (Data Migration Service):-

 Prerequisites in Source Account:

* Create a VPC peering connection between source and target VPCs
* Configure route tables to allow traffic between VPCs
* Set up security groups to allow database access
* Ensure source database has binary logging enabled (if using MySQL/Aurora)

 Set up IAM Roles and Permissions:

* In source account: Create IAM role allowing DMS to access the source database
* In target account: Create IAM role for DMS service
* Configure cross-account IAM roles if needed

 Create Replication Instance (Target Account):

* Go to DMS console
* Create a replication instance
* Select appropriate instance size based on data volume
* Choose the VPC that has connectivity to source database
* Configure advanced settings like Multi-AZ if needed

 Configure Endpoints: Source Endpoint:

* Create endpoint for source database
* Use the database credentials
* Test connectivity using DMS console

Target Endpoint:

* Create endpoint for target database
* Configure appropriate settings
* Test connectivity

 Create and Start Migration Task:

* Create new migration task
* Choose migration type:
  + Full load (one-time copy)
  + Full load + CDC (continuous replication)
  + CDC only
* Select tables to migrate
* Configure table mappings
* Set up transformation rules if needed

 Migration Monitoring:

* Monitor task progress in DMS console
* Check CloudWatch metrics
* Verify data integrity
* Monitor for any errors or warnings

 Post-Migration Steps:

* Validate data completeness
* Update application connection strings
* Configure monitoring and backups
* Test application functionality

AWS RDS automatically creates backups for your Aurora PostgreSQL database. However, automated backups do not occur in real-time or close to real-time.

**How Automated Backups Work in AWS RDS**

1. **Backup Frequency:**
   * Automated backups are taken **once per day** during a specified backup window.
   * AWS also captures **transaction logs** every 5 minutes to allow point-in-time recovery.
2. **Storage Location:**
   * The backup data, including snapshots and transaction logs, is stored in **Amazon S3** by AWS.
3. **Real-Time Backup?**
   * **No**, automated snapshots are not real-time. They occur at a scheduled time once a day.
   * However, **point-in-time recovery (PITR)** allows restoring data from transaction logs to a specific moment.

**Checking the Last Backup Time**

* The screenshot shows:
  + **Snapshot Creation Time:** January 14, 2025, at 02:03 UTC
  + **Instance Creation Time:** January 10, 2025, at 04:31 UTC
  + This suggests that the latest backup was taken **on schedule, not in real-time**.

**Alternative for Near-Real-Time Backups**

If you need backups closer to real-time, consider:

1. **Manual Snapshots** – You can create snapshots anytime.
2. **Cross-Region Replication** – Continuously replicates data to another region.
3. **AWS DMS (Database Migration Service)** – For real-time replication.
4. **Aurora Backtrack** – Allows rolling back within a set time frame.

Would you like guidance on setting up one of these solutions?