1. **AWS Backup Vault**

An **AWS Backup Vault** is a secure storage container where your backup data is stored. It helps organize, manage, and protect your backups in AWS.

**Why it’s used:**

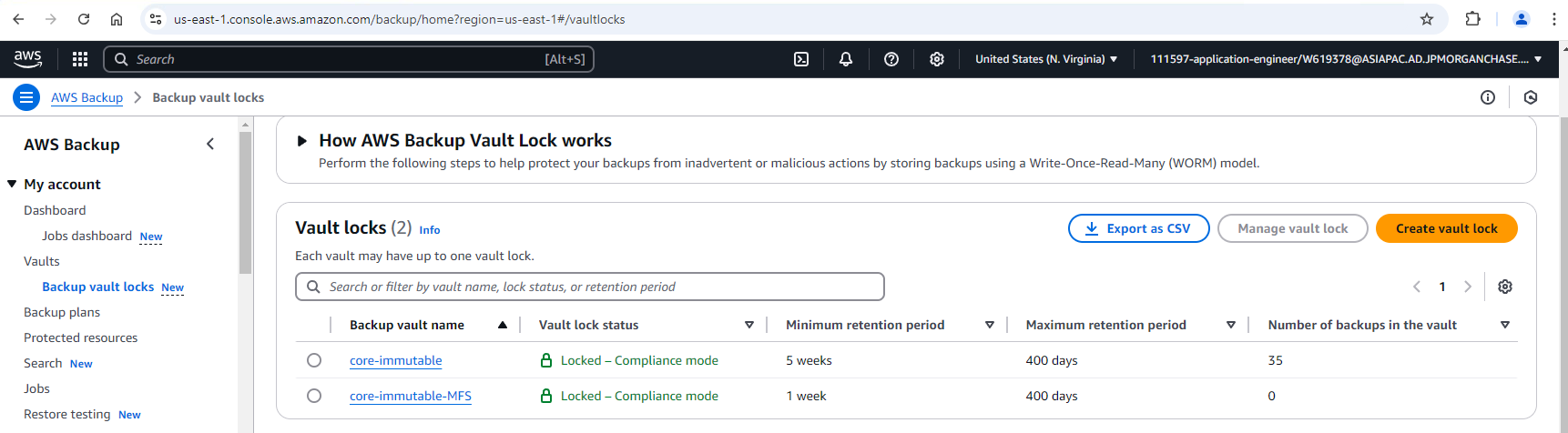
* It keeps all your backup data safe and organized.
* It allows you to manage backup retention policies and compliance.
* It enables encryption for secure data storage.

**Significance:**

* **Data protection**: Ensures backups are safe and can't be tampered with.
* **Compliance**: Helps meet regulatory requirements by storing backups securely.
* **Centralized management**: Lets you manage all your backups from one place in AWS.

**Existing aws backup vault locks in dev:**

* 1. core-immutable
  2. core-immutable-MFS



A screenshot of a computer

Description automatically generated

1. **AWS Backup Plan**:

An **AWS Backup Plan** is a set of rules that define how and when AWS resources (like databases, EC2 instances, or S3 buckets) should be backed up.

**In simple terms:**

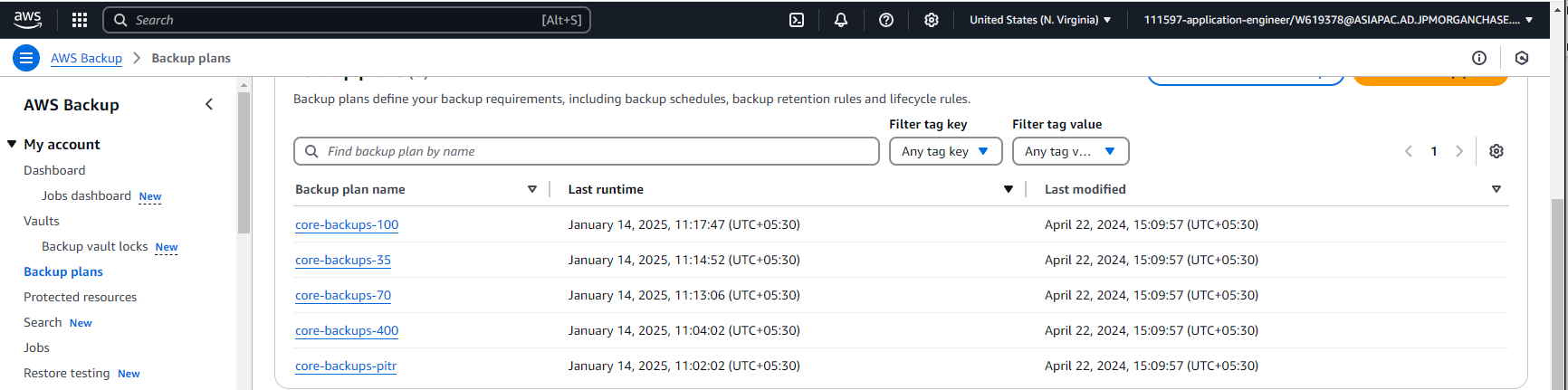
* It’s like a schedule that tells AWS when and how to take backups of your data.
* You can set how often backups happen (e.g., daily, weekly) and how long they should be kept.
* You can apply these plans to different AWS resources based on your needs.

**Why it's important:**

* **Automates backups**: You don't have to manually back up data every time.
* **Consistency**: Ensures all required resources are backed up in a consistent manner.
* **Compliance**: Helps meet backup retention policies for data protection.

**Existing Backup Plans:**

* **core-backups-100**
* **core-backups-35**
* **core-backups-70**
* **core-backups-400**
* **core-backups-pitr**



1. If your **RDS instance** is backed up using AWS Backup with the tag CORE\_BACKUPS\_RETENTION=35DAYS, and you've set up the **core-immutable** backup vault and the **core-backups-35** backup plan, you can retrieve backed-up data (e.g., in case of accidental deletion) using the following steps:
   * **Steps to Retrieve Backed-up Data:**
     + **Go to AWS Backup Console**:
       1. In the AWS Management Console, open the **AWS Backup** service.
     + **Select the Backup Vault**:
       1. Since your backups are stored in the core-immutable vault, go to the **Backup vaults** section and select **core-immutable**.
     + **Find Your Backup**:
       1. Under the **Recovery points** tab, search for the recovery point related to your **RDS instance**. You can use the tag CORE\_BACKUPS\_RETENTION=35DAYS to filter results, or search by resource name.
     + **Select the Recovery Point**:
       1. Once you locate the correct backup for your RDS instance (based on the date and time you need), click on the recovery point.
     + **Restore the Backup**:
       1. AWS Backup will give you options to **restore** the backup. You can choose to restore it to:
       2. A **new RDS instance** (if you want to keep the old one as-is).
       3. The **original RDS instance** (if you want to overwrite the deleted or corrupted data).
     + **Choose Restoration Settings**:
       1. Select the required **restore configuration**:
       2. For RDS, you can choose to restore it to the same instance or a new instance.
       3. Specify the necessary **restore time** if needed (e.g., point-in-time restoration for RDS).
     + **Complete the Restoration**:
       1. After selecting the restore options, click **Restore**. AWS will create the new RDS instance or restore the data as per your configuration.
     + **In Simple Terms:**
       1. If something happens to the main database (e.g., deletion), go to the **AWS Backup Console**.
       2. Find the **backup** of your RDS instance (from the core-immutable vault).
       3. Use the **restore** option to bring back the deleted or lost data.
     + **Key Points to Remember:**
       1. **Backup retention period**: Your backups are retained for 35 days as per your tag, so you can restore backups from that period.
       2. **Continuous backup**: This is disabled, so you will only restore backups from the specified period (i.e., the last 35 days in this case).
   * **Backup Vault**: Ensure you select the correct vault (core-immutable) where the backups are stored.
2. **How to add backup?**

**AWS Backup Steps:**

**Step 1: Create AWS Backup Vault**

* **Why**: A secure storage container for organizing and protecting backup data.
* **Existing Vaults in Dev**:
  + core-immutable
  + core-immutable-MFS

**Step 2: Create Backup Plan**

* **Why**: Defines how and when AWS resources should be backed up.
* **Existing Backup Plans**:
  + core-backups-100
  + core-backups-35
  + core-backups-70
  + core-backups-400
  + core-backups-pitr

**Step 3: Add Tag**

* Add CORE\_BACKUPS\_RETENTION = 35 days to these DB instances in Dev:
  + aurora-cluster-ywoeyfwg
    - aurora-cluster-node-0-ywoeyfwg
    - aurora-reader-node-1-ywoeyfwg
  + rdsapg86a8-auroratdb-v1-cluster
    - rdsapg86a8-auroratdb-v1-node-0
    - rdsapg86a8-auroratdb-v1-node-1

**Step 4: Resource Assignment**

* Assign RDS instances to the core-backups-35 plan.

**Restoration**:

* In case of deletion, go to the AWS Backup Console, find the backup in the **core-immutable** vault, and use the restore option to recover the data.

# AWS Backup Implementation Guide

## Prerequisites

- AWS Console access with appropriate permissions

- Access to RDS instances and AWS Backup service

## Detailed Implementation Steps

### Step 1: Create AWS Backup Vault

1. Navigate to AWS Backup Console

2. Select "Backup vaults" from the left navigation

3. Use existing vaults:

- `core-immutable`

- `core-immutable-MFS`

### Step 2: Select Backup Plan

1. Navigate to "Backup plans" in AWS Backup Console

2. Use existing plan `core-backups-35` which has a 35-day retention period

3. Verify other available plans if needed:

- core-backups-100

- core-backups-70

- core-backups-400

- core-backups-pitr

### Step 3: Tag Resources

1. Navigate to RDS Console

2. Add the following tag to these DB instances:

```

Key: CORE\_BACKUPS\_RETENTION

Value: 35

```

Apply to these instances:

- aurora-cluster-ywoeyfwg

- aurora-cluster-node-0-ywoeyfwg

- aurora-reader-node-1-ywoeyfwg

- rdsapg86a8-auroratdb-v1-cluster

- rdsapg86a8-auroratdb-v1-node-0

- rdsapg86a8-auroratdb-v1-node-1

### Step 4: Resource Assignment

1. Go to the `core-backups-35` backup plan

2. Click "Assign resources"

3. Select RDS as the resource type

4. Choose the tagged instances from Step 3

5. Confirm the assignment

## Restore Process

1. Navigate to AWS Backup Console

2. Select the `core-immutable` vault

3. Locate the desired backup point

4. Click "Restore"

5. Select restore options:

- Instance specifications

- VPC settings

- Security groups

6. Initiate restoration

## Verification

After implementation, verify:

- Tags are correctly applied to all instances

- Resources appear in the backup plan

- Initial backup is created according to schedule

## Monitoring

- Regular backup status checks in AWS Backup Console

- Verify backup completion notifications

- Monitor backup storage usage in the vault

## Best Practices

- Document any custom settings applied

- Maintain a backup testing schedule

- Keep track of retention periods

- Monitor backup costs

Key changes made to the YAML file:

1. Added a new component of type AWSBackup that:
   * References your existing vault (core-immutable)
   * References your existing backup plan (core-backups-35)
   * Assigns the RDS cluster to the backup plan
   * Applies the required tags
2. Modified the RDS Aurora component:
   * Added CORE\_BACKUPS\_RETENTION: "35" to the tags section
   * Set enable\_immutable\_backups: true to ensure backups are immutable
   * Added dependency on the backup component via dependsOn
3. The backup configuration will:
   * Use the existing vault and backup plan
   * Apply the required tags to the RDS instances
   * Assign the resources to the backup plan

To implement this:

1. Replace your existing YAML with this updated version
2. Apply the configuration through your deployment system
3. Verify in AWS Console that:
   * Tags are applied to the RDS instances
   * Resources are assigned to the backup plan
   * Backups are being created according to the schedule