AWS Architecture: Automated RDS Snapshots with EventBridge and Lambda

This documentation describes an AWS architecture that automates the creation of RDS PostgreSQL snapshots every hour using Amazon EventBridge and AWS Lambda. The architecture consists of the following components:  
  
1. Amazon EventBridge: A rule is set to trigger every hour to invoke the Lambda function.  
2. AWS Lambda: This function is responsible for creating snapshots of the RDS PostgreSQL instance.  
3. Amazon RDS (PostgreSQL): The database for which snapshots are created.

# IAM Role for Lambda Function

An IAM role is required for the Lambda function to create and delete RDS snapshots. Below is the IAM policy that grants the necessary permissions:

{  
 "Version": "2012-10-17",  
 "Statement": [  
 {  
 "Effect": "Allow",  
 "Action": [  
 "rds:CreateDBSnapshot",  
 "rds:DeleteDBSnapshot",  
 "rds:DescribeDBSnapshots",  
 "rds:DescribeDBInstances"  
 ],  
 "Resource": "\*"  
 },  
 {  
 "Effect": "Allow",  
 "Action": [  
 "logs:CreateLogGroup",  
 "logs:CreateLogStream",  
 "logs:PutLogEvents"  
 ],  
 "Resource": "\*"  
 }  
 ]  
 }

# Lambda Function Code

Below is the Python code for the Lambda function, which creates snapshots of the RDS PostgreSQL instance and deletes snapshots older than 7 days:

import boto3  
import datetime  
  
def lambda\_handler(event, context):  
 rds = boto3.client('rds')  
 db\_instance\_id = 'your-db-instance-id' # Replace with your RDS instance ID  
  
 # Create a snapshot identifier with a specific naming convention  
 snapshot\_identifier = f"lambda-rds-snapshot-{db\_instance\_id}-{datetime.datetime.now().strftime('%Y%m%d%H%M')}"  
  
 # Create the snapshot  
 try:  
 rds.create\_db\_snapshot(  
 DBInstanceIdentifier=db\_instance\_id,  
 DBSnapshotIdentifier=snapshot\_identifier  
 )  
 print(f"Snapshot {snapshot\_identifier} created successfully.")  
 except Exception as e:  
 print(f"Error creating snapshot: {str(e)}")  
   
 # Calculate the cutoff date for deletion (7 days ago)  
 cutoff\_date = datetime.datetime.now() - datetime.timedelta(days=7)  
  
 # List all manual snapshots for the specified DB instance  
 try:  
 snapshots = rds.describe\_db\_snapshots(  
 DBInstanceIdentifier=db\_instance\_id,  
 SnapshotType='manual' # Only consider manual snapshots  
 )['DBSnapshots']  
  
 # Iterate through snapshots and delete those older than 7 days  
 for snapshot in snapshots:  
 snapshot\_id = snapshot['DBSnapshotIdentifier']  
 snapshot\_create\_time = snapshot['SnapshotCreateTime'].replace(tzinfo=None)  
  
 # Check if the snapshot was created by this Lambda function  
 if snapshot\_create\_time < cutoff\_date and snapshot\_id.startswith(f"lambda-rds-snapshot-{db\_instance\_id}-"):  
 rds.delete\_db\_snapshot(DBSnapshotIdentifier=snapshot\_id)  
 print(f"Deleted snapshot: {snapshot\_id}")  
  
 except Exception as e:  
 print(f"Error listing or deleting snapshots: {str(e)}")  
  
 return {  
 "statusCode": 200,  
 "body": f"Snapshot {snapshot\_identifier} created, old snapshots deleted."  
 }

# EventBridge Rule for Hourly Trigger

The EventBridge rule triggers the Lambda function every hour to automate the RDS snapshot process. The following code snippet shows how to set up the rule with a schedule expression for hourly triggers:

aws events put-rule \  
 --name HourlyRDSBackup \  
 --schedule-expression "rate(1 hour)"  
  
aws events put-targets \  
 --rule HourlyRDSBackup \  
 --targets "Id"="1","Arn"="arn:aws:lambda:us-east-1:123456789012:function:RDSHourlySnapshot"  
  
aws lambda add-permission \  
 --function-name RDSHourlySnapshot \  
 --statement-id EventBridgeInvoke \  
 --action 'lambda:InvokeFunction' \  
 --principal events.amazonaws.com \  
 --source-arn arn:aws:events:us-east-1:123456789012:rule/HourlyRDSBackup

# AWS Architecture Diagram

The diagram below shows the architecture where EventBridge triggers the Lambda function every hour, which in turn creates a snapshot of the RDS PostgreSQL instance.

