

Para lograr que la función Lambda solo realice un snapshot completo la primera vez (o si el bucket está vacío), y luego haga snapshots parciales en los días siguientes, puedes modificar el código para verificar la existencia de datos en el bucket de S3. Aquí te muestro cómo hacerlo:

Código Lambda Modificado

```
import boto3
import datetime

def lambda_handler(event, context):
    rds_client = boto3.client('rds')
    s3_client = boto3.client('s3')
    s3_bucket_name = 'dbs3export'

    # Verificar si el bucket está vacío
    bucket_objects = s3_client.list_objects_v2(Bucket=s3_bucket_name,
Prefix='exported-snapshots/')
    is_bucket_empty = 'Contents' not in bucket_objects

    # Si el bucket está vacío, hacer un snapshot completo, de lo
    contrario, hacer un snapshot parcial
    snapshot_type = 'full' if is_bucket_empty else 'incremental'

    # Obtener el último snapshot de RDS si no es la primera ejecución
    latest_snapshot = None
    if not is_bucket_empty:
        snapshots = rds_client.describe_db_snapshots(
            DBInstanceIdentifier='prod-02',
            SnapshotType='manual'
        )
        latest_snapshot = max(snapshots['DBSnapshots'], key=lambda x:
x['SnapshotCreateTime'], default=None)

    # Crear un nuevo snapshot
    snapshot_idenfier = f"prod-02-snapshot-
{datetime.datetime.now().strftime('%Y-%m-%d-%H-%M-%S')}-{snapshot_type}"
    rds_client.create_db_snapshot(
        DBSnapshotIdentifier=snapshot_idenfier,
        DBInstanceIdentifier='prod-02'
    )

    # Exportar el snapshot a S3
    export_task_id = f"task-
{datetime.datetime.now().strftime('%Y%m%d%H%M%S')}"
    source_arn = f"arn:aws:rds:us-east-1:253021683072:snapshot:
{snapshot_idenfier}"
    iam_role_arn = 'arn:aws:iam::253021683072:role/iamroles3bck'
    kms_key_id = 'arn:aws:kms:us-east-1:253021683072:key/be6a11f7-16d9-
4ee2-bc0c-eab9e87bf7f1'

    rds_client.start_export_task(
```

```
        ExportTaskIdentifier=export_task_id,  
        SourceArn=source_arn,  
        S3BucketName=s3_bucket_name,  
        IamRoleArn=iam_role_arn,  
        KmsKeyId=kms_key_id,  
        S3Prefix='exported-snapshots/'  
    )  
  
    return {  
        'statusCode': 200,  
        'body': f'Successfully started {snapshot_type} export task  
{export_task_id}.'  
    }
```

Desplegar y Configurar la Función Lambda

Sigue los mismos pasos anteriores para desplegar y configurar la función Lambda:

1. Guardar el Código y Crear el Archivo ZIP

```
zip function.zip lambda_function.py
```

2. Crear y Configurar el Rol IAM

```
aws iam create-role --role-name lambda-execution-role --assume-role-  
policy-document file://trust-policy.json  
  
aws iam attach-role-policy --role-name lambda-execution-role --policy-  
arn arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole  
aws iam attach-role-policy --role-name lambda-execution-role --policy-  
arn arn:aws:iam::aws:policy/AmazonRDSFullAccess  
aws iam attach-role-policy --role-name lambda-execution-role --policy-  
arn arn:aws:iam::aws:policy/AmazonS3FullAccess  
aws iam attach-role-policy --role-name lambda-execution-role --policy-  
arn arn:aws:iam::aws:policy/AWSKeyManagementServicePowerUser
```

3. Desplegar la Función Lambda

```
ROLE_ARN=$(aws iam get-role --role-name lambda-execution-role --query  
'Role.Arn' --output text)  
aws lambda create-function --function-name export-rds-snapshot --  
runtime python3.8 --role $ROLE_ARN --handler  
lambda_function.lambda_handler --zip-file fileb://function.zip
```

4. Configurar el Disparador de CloudWatch Events

```
aws events put-rule --name daily-rds-snapshot-export --schedule-expression 'rate(1 day)'  
  
aws lambda add-permission --function-name export-rds-snapshot --statement-id daily-rds-snapshot-export --action 'lambda:InvokeFunction' --principal events.amazonaws.com --source-arn arn:aws:events:REGION:ACCOUNT_ID:rule/daily-rds-snapshot-export  
  
FUNCTION_ARN=$(aws lambda get-function --function-name export-rds-snapshot --query 'Configuration.FunctionArn' --output text)  
aws events put-targets --rule daily-rds-snapshot-export --targets "Id"="1", "Arn"="$FUNCTION_ARN"
```

Verificar y Monitorear

1. Probar la Función Manualmente

```
aws lambda invoke --function-name export-rds-snapshot output.txt
```

2. Verificar los Logs de Ejecución

```
aws logs describe-log-groups  
aws logs describe-log-streams --log-group-name /aws/lambda/export-rds-snapshot  
aws logs get-log-events --log-group-name /aws/lambda/export-rds-snapshot --log-stream-name <log-stream-name>
```

Mantenimiento

Para actualizar el código de la función Lambda o realizar ajustes en los permisos, sigue los mismos pasos descritos anteriormente:

1. Actualizar el Código:

```
zip function.zip lambda_function.py  
aws lambda update-function-code --function-name export-rds-snapshot --zip-file fileb://function.zip
```

2. Probar y Verificar:

```
aws lambda invoke --function-name export-rds-snapshot output.txt
```

Resumen de Funcionalidad

- La función Lambda verifica si el bucket S3 está vacío.
- Si el bucket está vacío, se realiza un snapshot completo y se exporta.
- Si el bucket contiene datos, se realiza un snapshot incremental y se exporta.