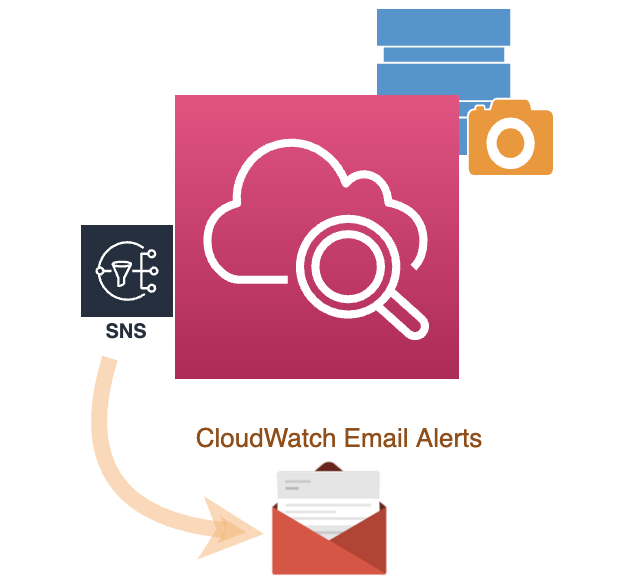
**Configure AWS CloudWatch Alerts to track EBS Snapshot State Changes**



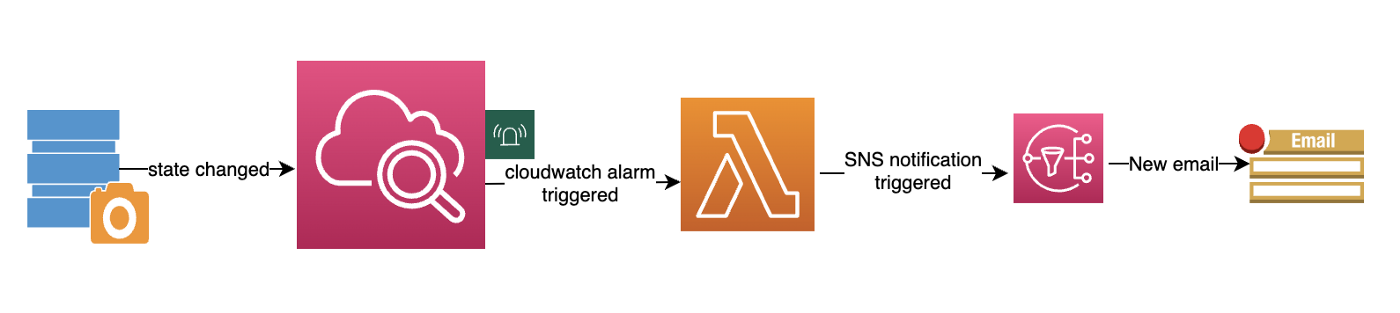
**Introduction**

The data on Amazon EBS volumes can be backed up in Amazon S3 as EBS snapshots. Snapshots generate various events indicating a state change. The list of snapshot events can be found in Amazon [documentation](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-cloud-watch-events.html#snapshot-events). In this article we will see how these events can be monitored in Amazon CloudWatch and how to setup alerts using Amazon SNS to track a snapshot status.

**Workflow**

For the sake of this discussion, let us assume we want to monitor EBS [snapshot archival events](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitor-snapshot-archiving.html). Archiving snapshots helps to save up to 75% on storage costs for snapshots that you don’t intend to use frequently but needs to be retained for a longer period. We will track when the snapshot is moved to the archive tier or restored to the standard tier so they are ready for use. EBS emits events when a snapshot state changes.

Here is a high level diagram of the workflow that is triggered when a snapshot state changes.



**Steps**

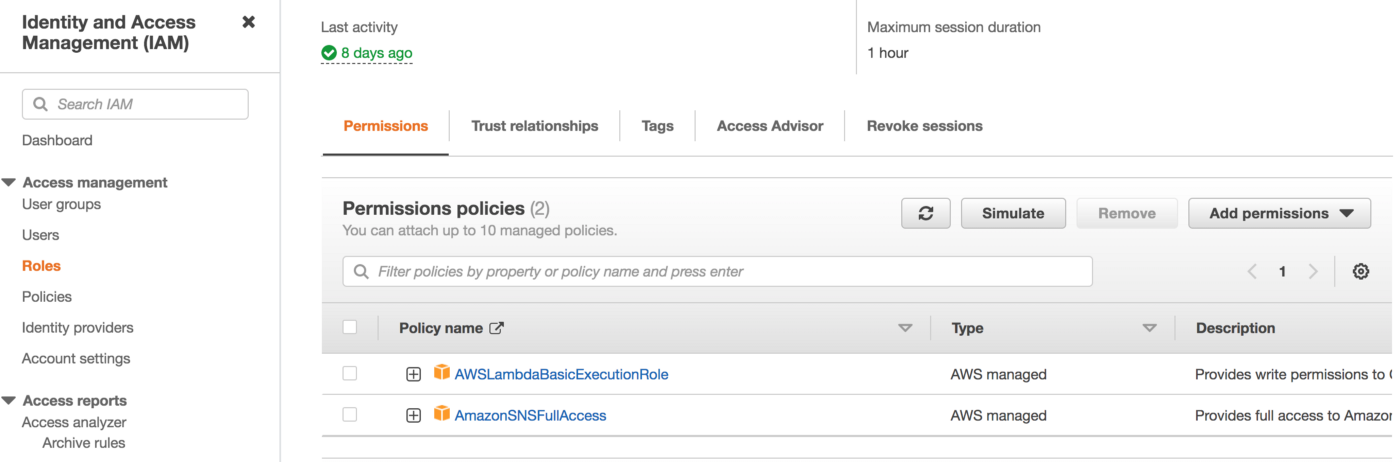
Below are the steps to configure the workflow.

1. Configure IAM role enabling Lambda and SNS access permissions.
2. Configure SNS notification service to send email notification with the event details to the subscriber.
3. Configure Lambda function to process the CloudWatch event and trigger an SNS notification.
4. Configure a rule in CloudWatch to track the snapshot notification event and invoke the Lambda function.

**IAM Setup**

The first step is to create an IAM Role to enable Lambda and SNS operations.

1. Navigate to **IAM** Service in the Management Console.
2. Create a new role by selecting **Access Management - >Roles -> Create role**
3. In the **Select Trusted Entity**page, select **AWS service**as the Trusted entity type. Under **Common Use cases** select **Lambda**.
4. Next, in the **Add Permissions** page, select two permission policies: **AWSLambdaBasicExecutionRole** and **AmazonSNSFullAccess**.



Select Next to go to the next page.

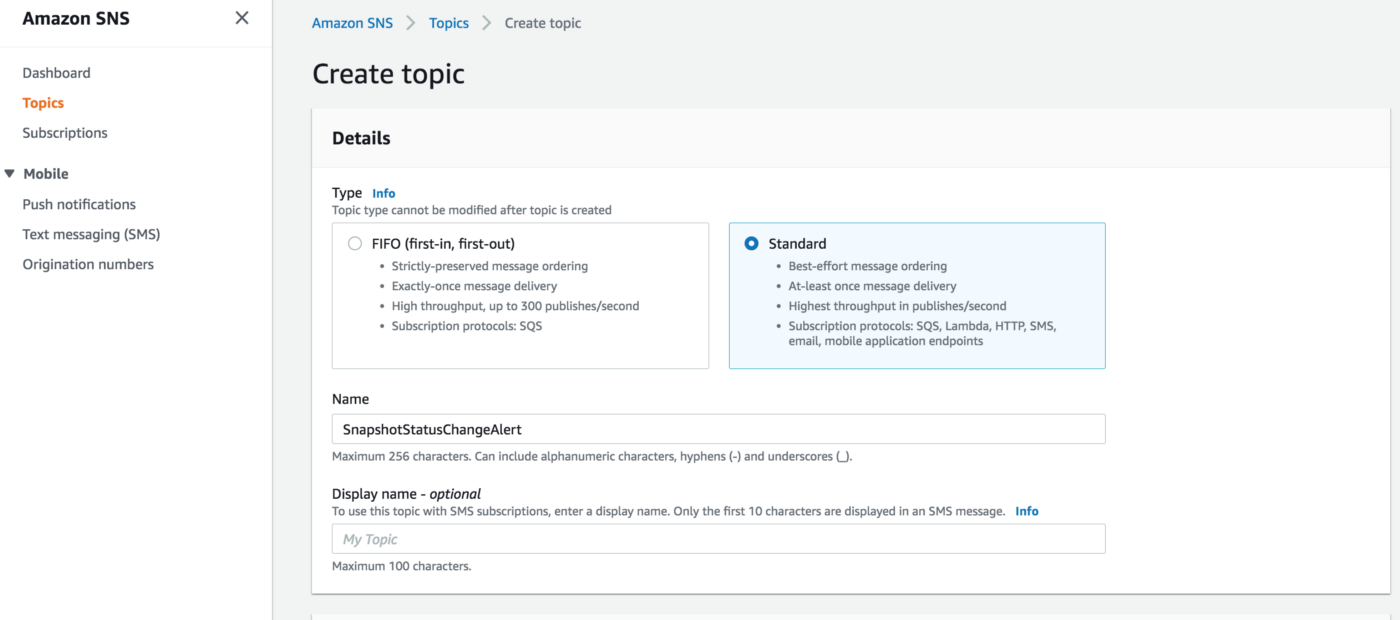
5. Fill out **Role name** and **Description**. For our example, let us assume role name as **SnapshotManager**and fill it accordingly.

6. Click **Create role**to create the role.

**SNS Notification Setup**

To get notified, we will setup and subscribe to an SNS topic.

1. Navigate to Amazon **SNS**service in the Management Console.
2. Select the option **Topics** -> **Create Topic**to create a topic.
3. In the Create Topic page select type **Standard**.
4. In the **Name**edit box enter a name for the topic. Let us assume the name as **SnapshotStatusChangeAlert** for this example.



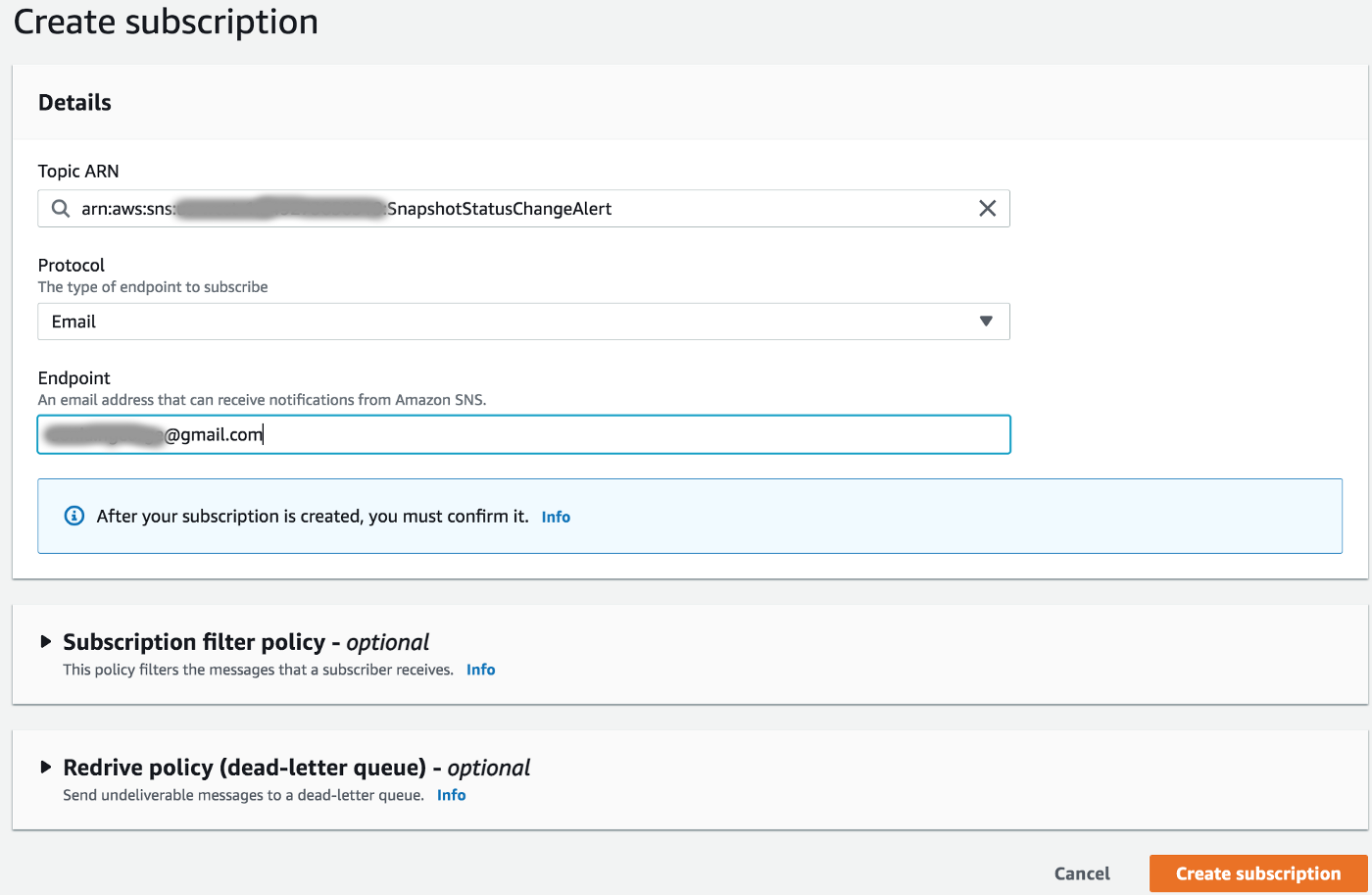
5. Click the **Create Topic**button to finalize the changes and create the topic.

6. Next to subscribe to the topic, select **Subscriptions** in the menu and click on the **Create Subscription** button.

7. Under **Topic ARN**select the ARN of the **SnapshotStatusChangeAlert**topic.

8. Under **Protocol**select the **Email** protocol.

9. In the **Endpoint**edit box enter the email address where you want to be notified when the snapshot status changes.



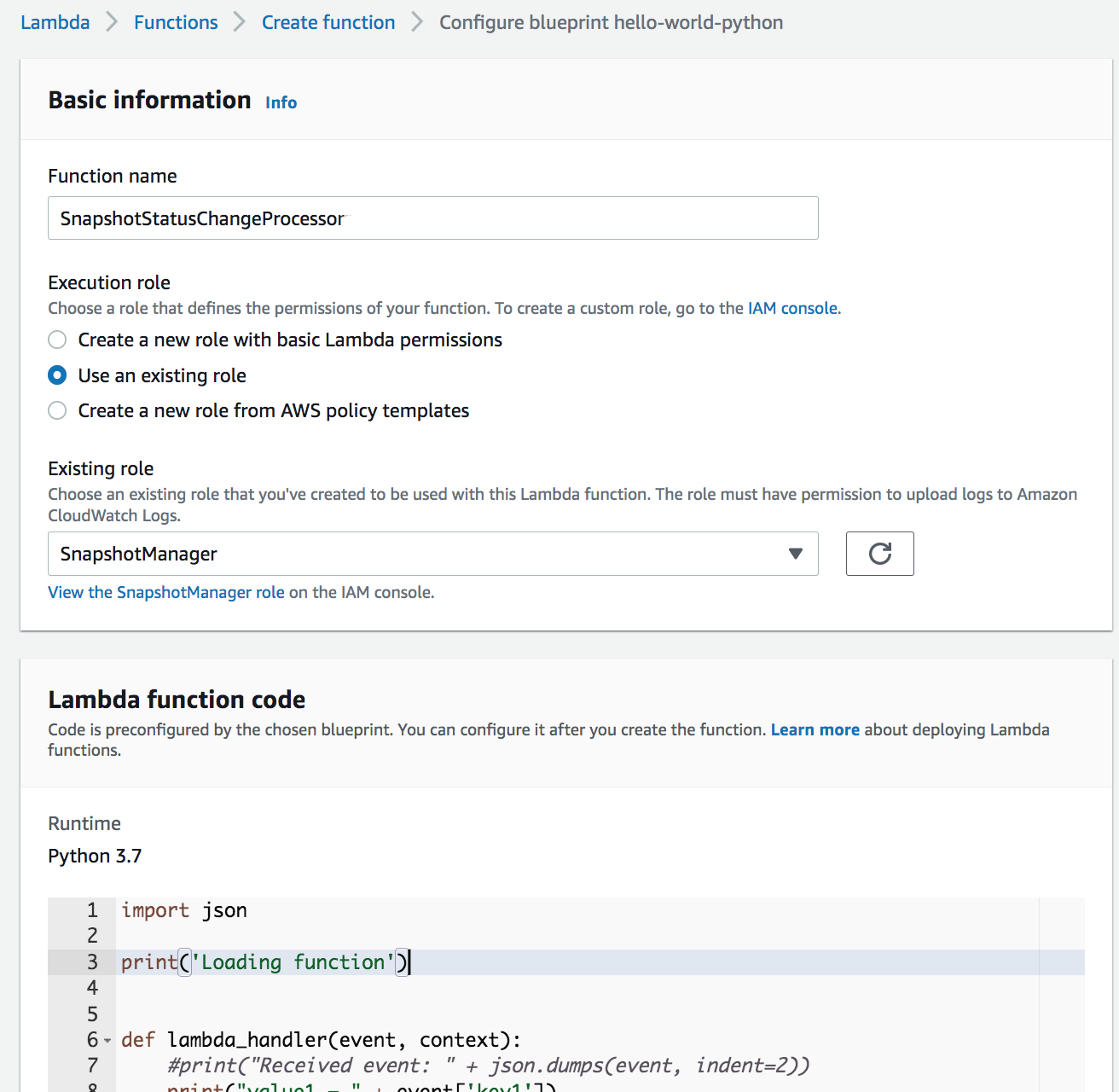
10. Finalize the changes by clicking on the **Create subscription** button.

11. Check for AWS confirmation email and confirm the same.

**AWS Lambda Configuration**

Next step is to configure a Lambda function to process the CloudWatch event generated when the EBS snapshot status changes.

1. Navigate to **Lambda** service in the AWS Management Console.
2. Select the option **Functions** -> **Create function**.
3. To get a basic python based template setup, in the Create function page select the option **Use a blueprint**.
4. Under **Blueprints** type *hello-world-python,*select the function and click the **Configure** button.
5. In the Basic Information page, enter the **Function name** and select the **Execution role**.
6. For this example assume **SnapshotStatusChangeProcessor**as the name and enter it in the **Function name** edit box.
7. For Execution role, select **Use an existing role**and in the Existing role dropbox select the IAM role **SnapshotManager** that was previously created.



8. Click the **Create function** button to create the Lambda function.

9. Modify the lambda function code to trigger the **SNS notification** with the event details.

import json  
import boto3print('Loading function')def lambda\_handler(event, context): event\_data = json.dumps(event, indent=2)  
   
 print("Received event: " + json.dumps(event, indent=2)) # Trigger SNS notification  
 client = boto3.client('sns')  
 response = client.publish(  
 TopicArn='arn:aws:sns:<region>:<accountid>:**SnapshotStatusChangeAlert**', # update region,accountid  
 Message=event\_data,  
 Subject='Snapshot Archive Tier Changed'  
 )

10. Click **Deploy** button to deploy the new code changes.

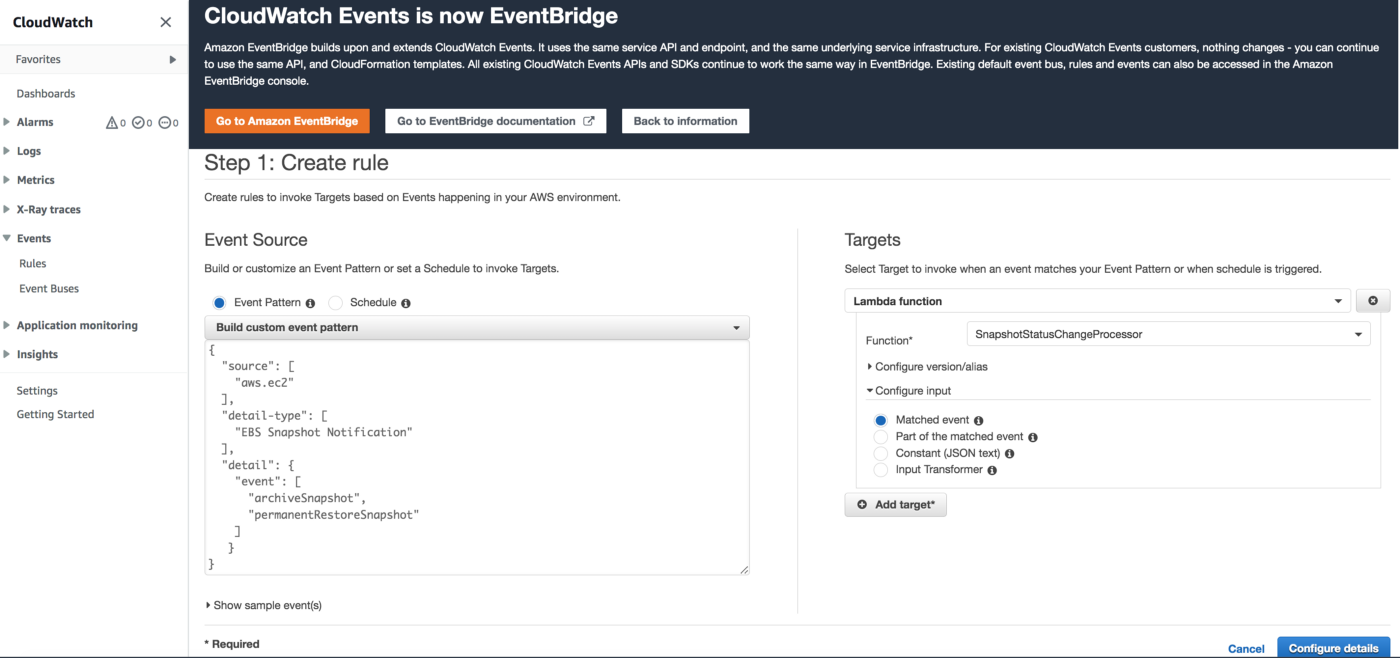
**AWS CloudWatch Rule Configuration**

The final step is to configure a CloudWatch Event rule that captures the snapshot status change event and kicks off the workflow.

1. Navigate to **CloudWatch** service in the AWS Management Console.
2. Under **Events** menu select **Rules**and click the **Create Rule** button to create a rule.
3. In the **Create Rule** page under **Event Source** select **Event Pattern**.
4. Edit **Event Pattern Preview**and enter the pattern to track archive and restore events. Check the examples in the AWS documentation on [snapshot archival events](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitor-snapshot-archiving.html) to construct the pattern string. The below example tracks the events generated when any snapshot is moved to the Archive Tier or restored to the Standard Tier.

{  
 "source": [  
 "aws.ec2"  
 ],  
 "detail-type": [   
 "EBS Snapshot Notification"  
 ],   
 "detail": {   
 "event": [   
 "archiveSnapshot",  
 "permanentRestoreSnapshot"  
 ]  
 }  
}

5. In the **Targets** section, select **Add Target**and in the dropdown select **Lambda function**. In the Function name edit box enter the function name **SnapshotStatusChangeProcessor**that was earlier created. Select **Configure input -> Matched event**.



6. Next click the **Configure details** button and proceed to the next step.

7. In the Rule definition page set the rule name in the **Name** field. For this example assume rule name as **SnapshotStatusChange**and set accordingly. Enter **Description** for the rule.

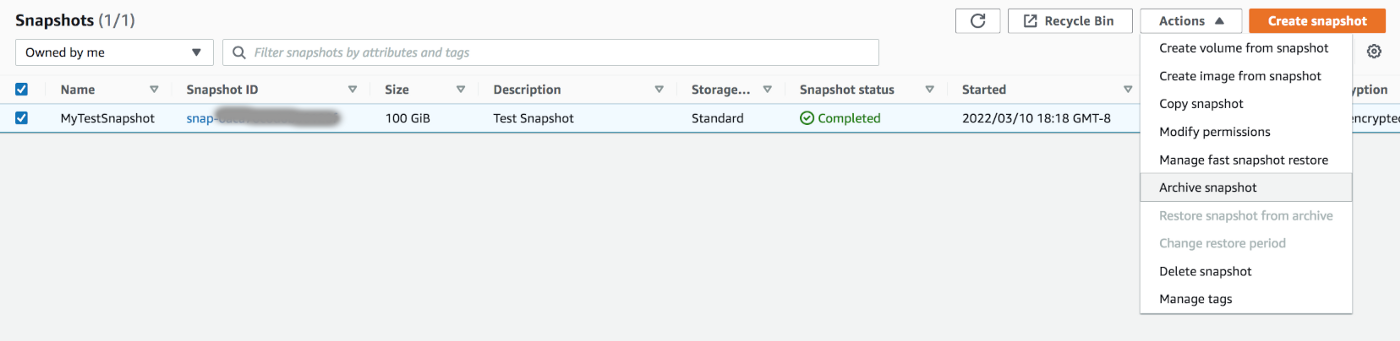
8. Click the **Create rule** button to confirm the changes and create a new rule.

The workflow is now setup and ready for testing.

**Test Setup**

To test the workflow, change a EC2 snapshot state to archive or restore.

1. Navigate to **EC2** service from the Management Console.
2. Select Snapshots from the menu. Select a snapshot to archive. Next select **Actions -> Archive snapshot**and confirm the operation.



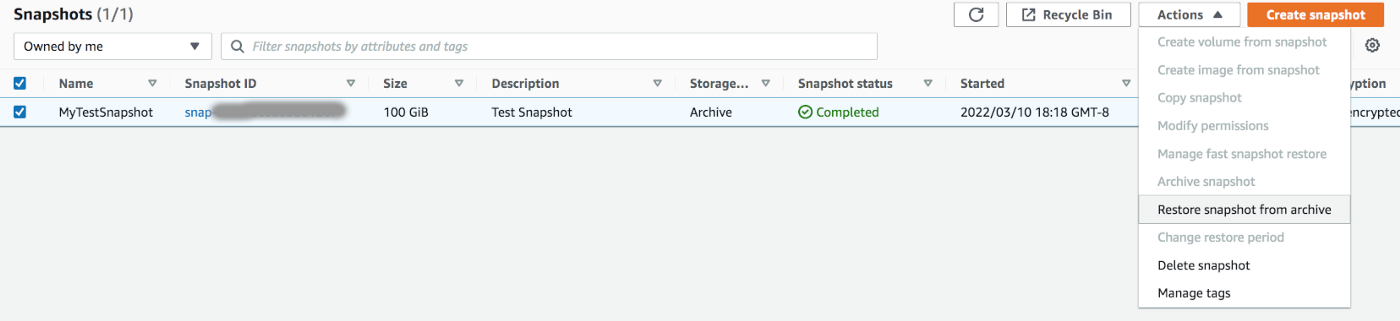
3. Click on the snapshot ID and in the snapshot page, under **Storage tier** you can view the status. When the status changes to indicate completion, an event is generated which triggers the workflow.

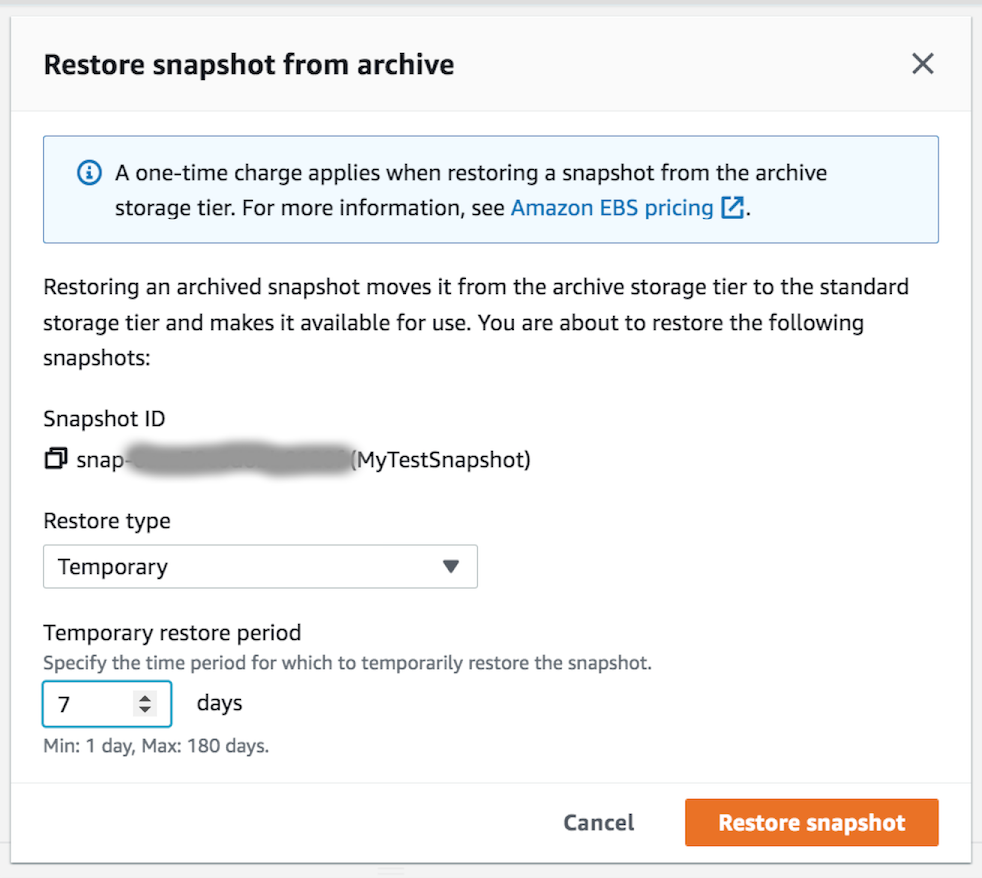
4. On successful completion, there should be a notification message at the subscribed email address with details of the snapshot state change.



The snapshot is archived. When the snapshot is required again, it has to be first restored before it can be used.

5. To restore the selected snapshot, click **Actions** -> **Restore snapshot from archive**. In the popup dialog, choose **Restore type**as **permanent** and click on the **Restore snapshot** button.





6. Once the restore is complete, the snapshot is restored to Standard tier which creates a new CloudWatch event. This triggers the workflow which generates a new email including the snapshot restore event details.



The snapshot is now restored and ready for use.