

Automatic Snapshot of EBS Volume by using Lambda Function

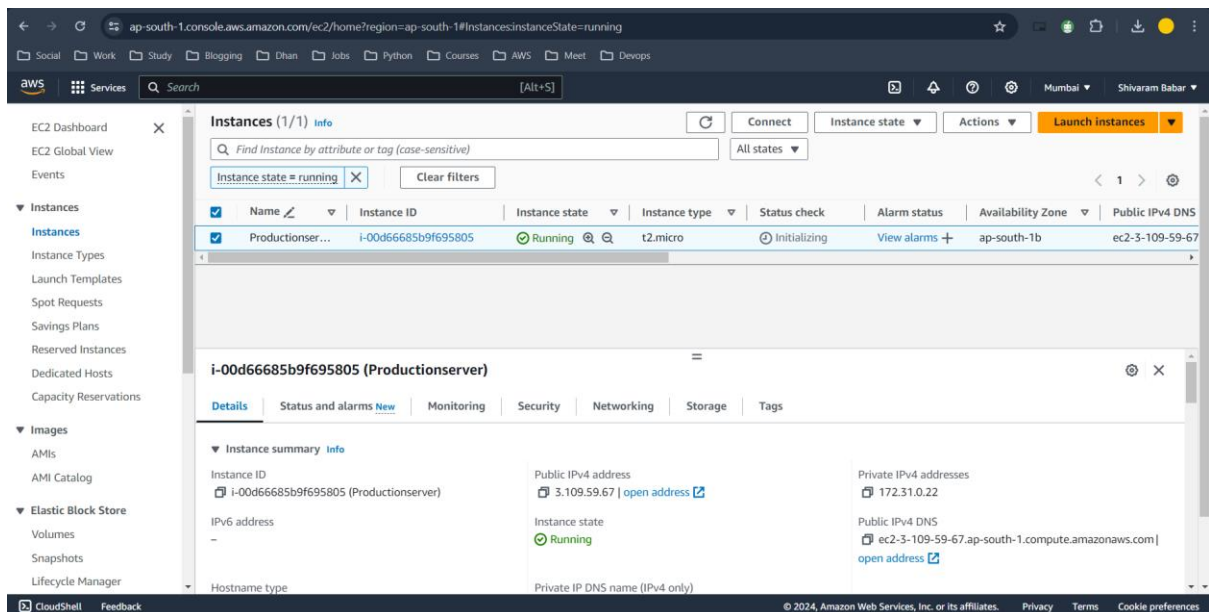
Introduction

In this project, I developed an automated solution for creating and managing EBS snapshots using AWS services. Leveraging AWS Lambda, EventBridge, SNS, and IAM policies, I ensured regular backups of EBS volumes with minimal manual intervention. This system includes notifications for snapshot creation status, providing a reliable and scalable backup strategy for critical data in AWS.

Step-by-Step Implementation

Step 1: Launch an EC2 Instance

- Navigated to the AWS Management Console.
- Went to the EC2 console and clicked "Launch Instance".
- Chose the Amazon Linux Amazon Machine Image (AMI).
- Selected an instance type (e.g., t2.micro) and configured instance details.
- Launched the instance (**Productionserver**) and connected to verify it was running properly.



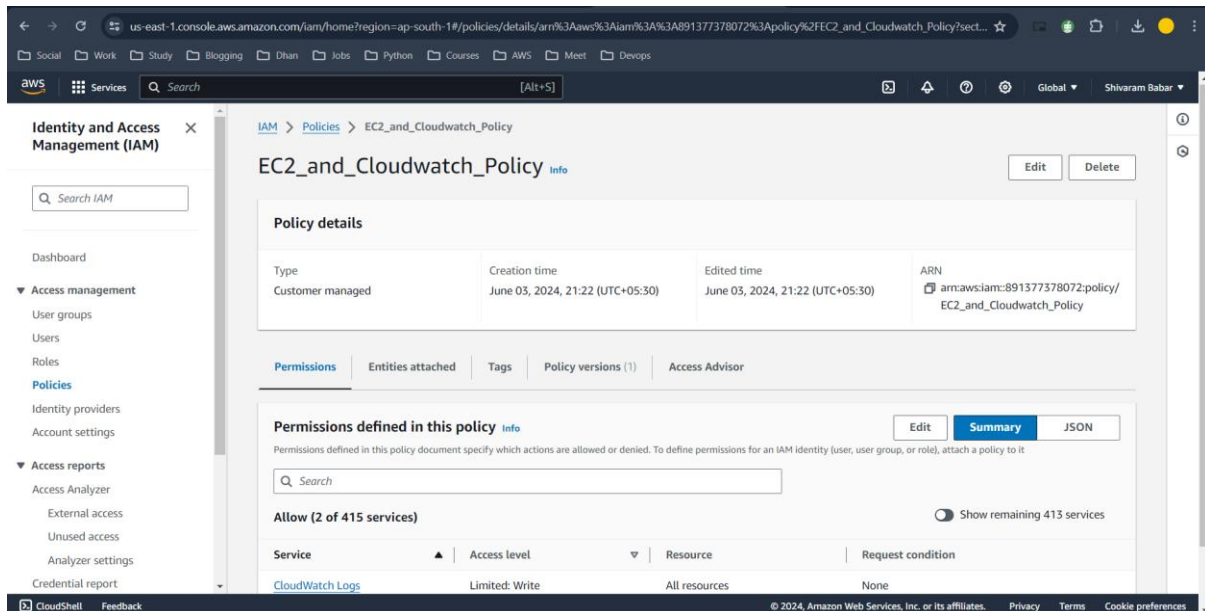
Step 2: Create IAM Policies

Created EC2 Service Policy:

- Accessed the IAM console, clicked on "Policies", and then "Create policy".
- Configured a policy to allow snapshot creation and volume description with the following actions: CreateSnapshot, DescribeVolumes, DescribeTags, DeleteSnapshot, DescribeSnapshots.

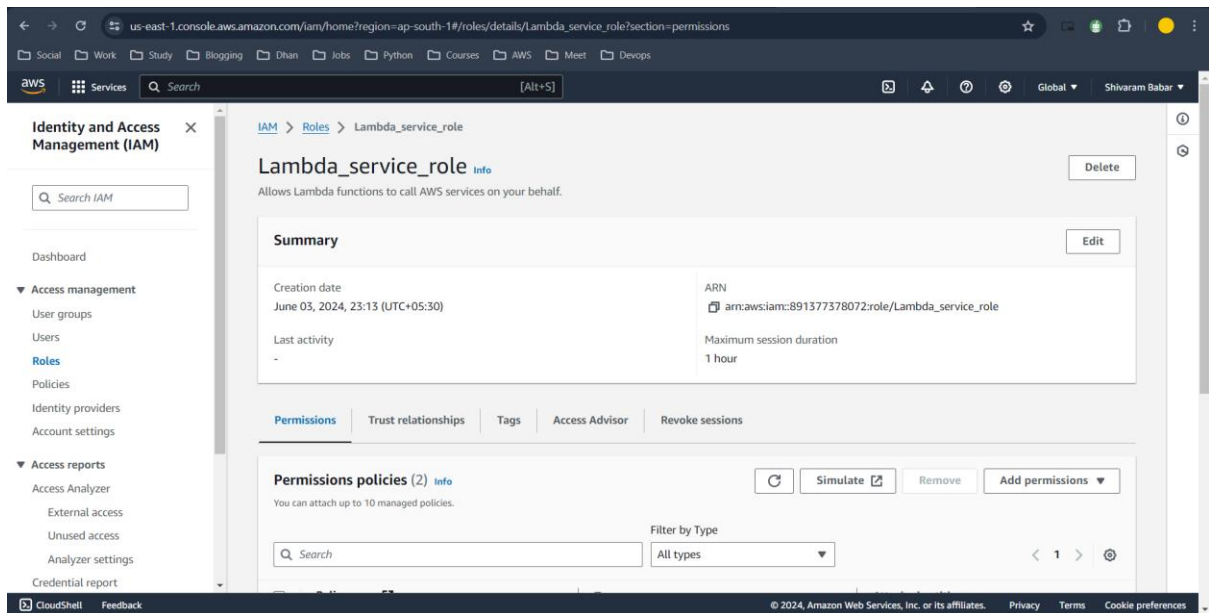
Created CloudWatch Logs Policy:

- In the add more permission Configured a policy to allow CloudWatch Logs actions: CreateLogGroup, CreateLogStream, PutLogEvents.
- Named the policy Named the policy EC2_and_Cloudwatch_Policy and created it.



Step 3: Create an IAM Role for Lambda Service

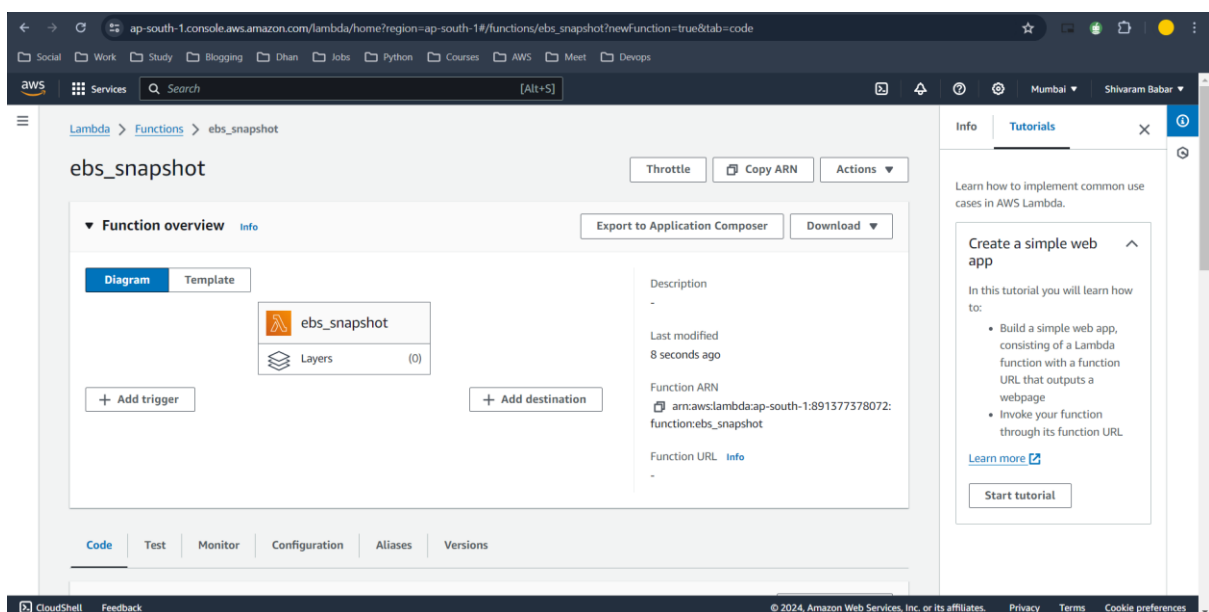
- Created IAM Role:
- Went to the IAM console, clicked on "Roles", and then "Create role".
- Selected "AWS service" and choose "Lambda".
- Attached the EC2_and_Cloudwatch_Policy, along with the AWSLambdaBasicExecutionRole policy.
- Named the role Lambda_service_role and created it.



Step 4: Create a Lambda Function

1. Created Lambda Function:

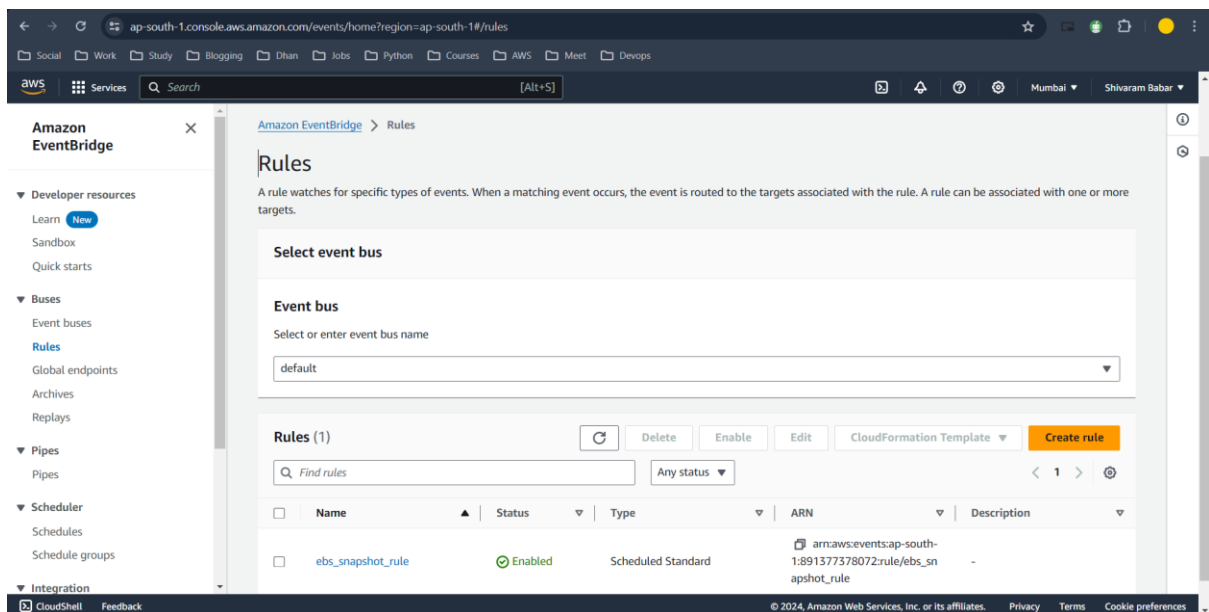
- Navigated to the Lambda console and clicked "Create function".
- Chose "Author from scratch" and entered the following details:
 - **Function name:** ebs_snapshot
 - **Runtime:** Python 3.12
 - **Role:** Selected the role created in Step 3 (Lambda_service_role).
- Clicked "Create function".



Step 5: Create an EventBridge Rule

1. Created EventBridge Rule:

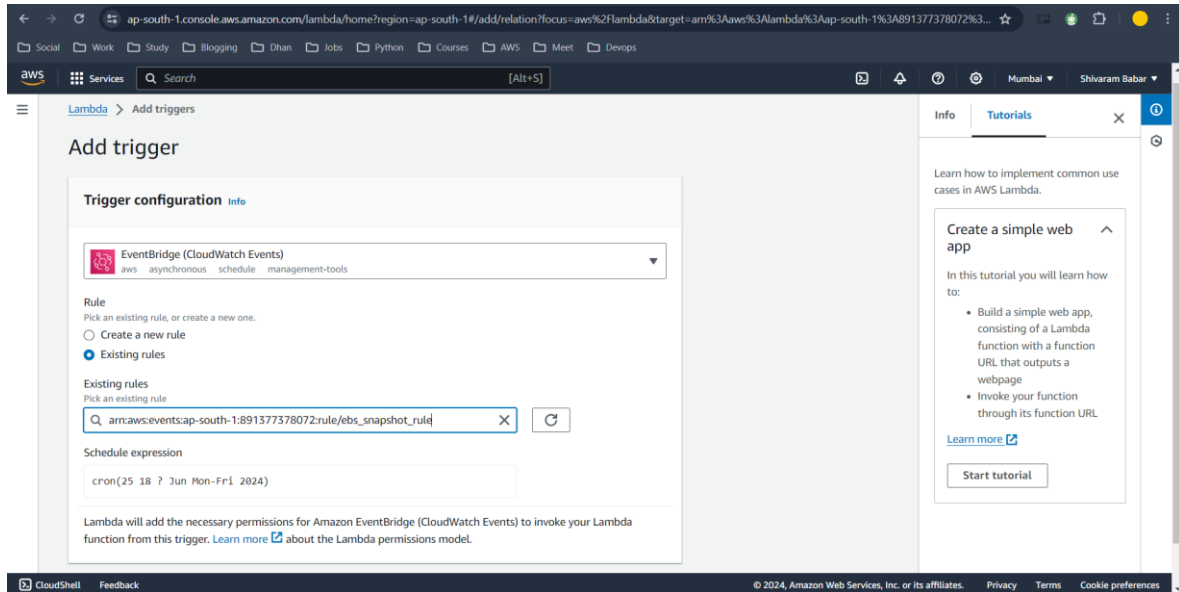
- Navigated to the EventBridge console and clicked "Create rule".
- Entered the following details:
 - **Name:** ebs_snapshot_rule
 - **Description:** Schedule to create EBS snapshots daily.
- Under "Define pattern", chose "Event Source" and selected "Schedule".
- Used a rate expression, `rate(1 day)`, to schedule daily snapshot creation.
- Clicked "Next".



Step 6: Add Trigger for Lambda Function

1. Added EventBridge Rule as Trigger:

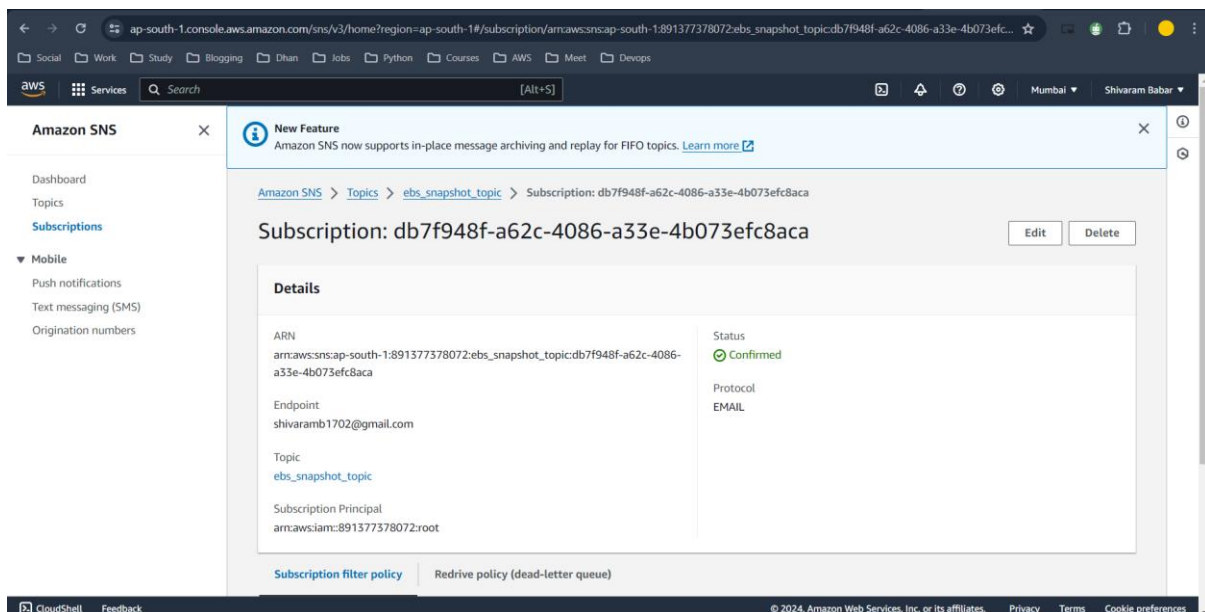
- In the target section of the EventBridge rule, chose "Lambda function".
- Selected the Lambda function created earlier (ebs_snapshot_rule).
- Clicked "Next" and then "Create rule".



Step 7: Create an SNS Topic

1. Created SNS Topic:

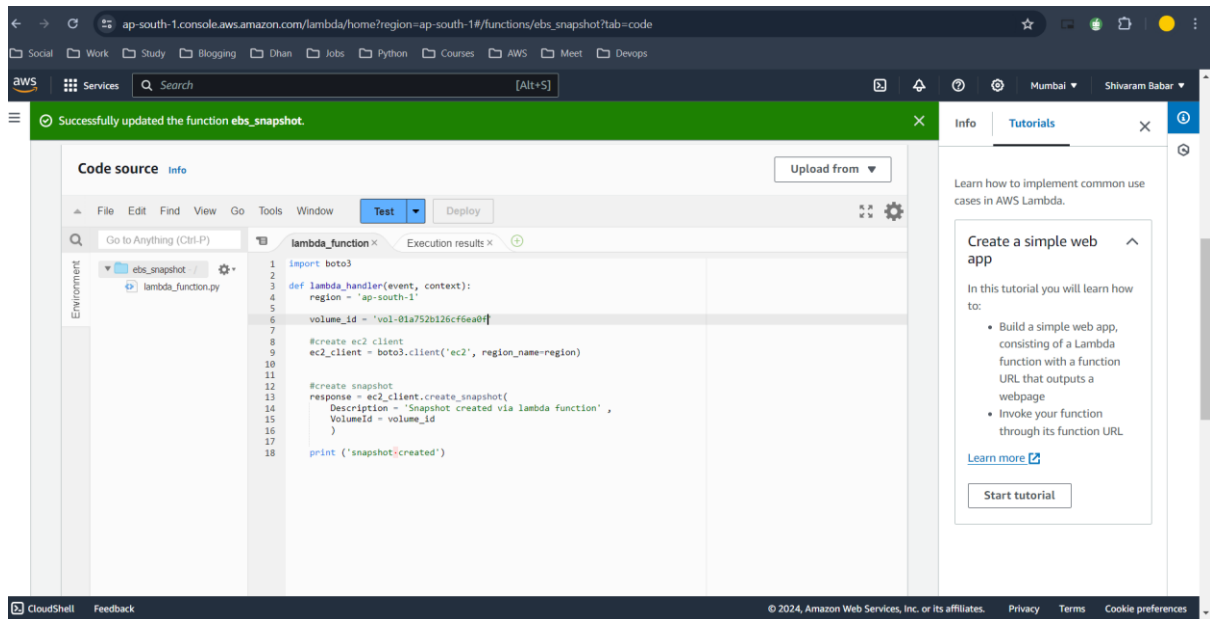
- Navigated to the SNS console and clicked "Create topic".
- Chose "Standard" for the type and entered a name for the topic, `ebs_snapshot_topic`.
- Clicked "Create topic".



Step 8: Add Destination to SNS Topic in Lambda Function

Configured SNS Topic in Lambda:

- In the function's configuration page, scrolled down to the "Function code" section.
- Ensured the SNS topic ARN was correctly added in the `sns.publish` call within the Lambda function code.
- Replaced `YOUR_REGION`, `YOUR_ACCOUNT_ID`, and `ebs_snapshot_topic` with appropriate values.



Conclusion

I successfully set up an automated system for creating and managing EBS snapshots using AWS Lambda, EventBridge, and SNS. By following these steps, I ensured that EBS volumes are backed up regularly, and I receive notifications about the status of these snapshots. This automation enhances data protection and reduces manual intervention, providing a scalable solution for EBS snapshot management.