Take EBS Snapshot of a particular instance at 10:30 AM IST daily and drop a mail with Instance Id and snapshot id and Timestamp

Step 1: Create an IAM Role for the Lambda Function

- Go to the IAM Service in the AWS Management Console.
- Click on Roles and then Create role.
- Select AWS service and choose Lambda.
- Click Next: Permissions.
- Click Create policy and go to the JSON tab.
- Enter the following policy definition:

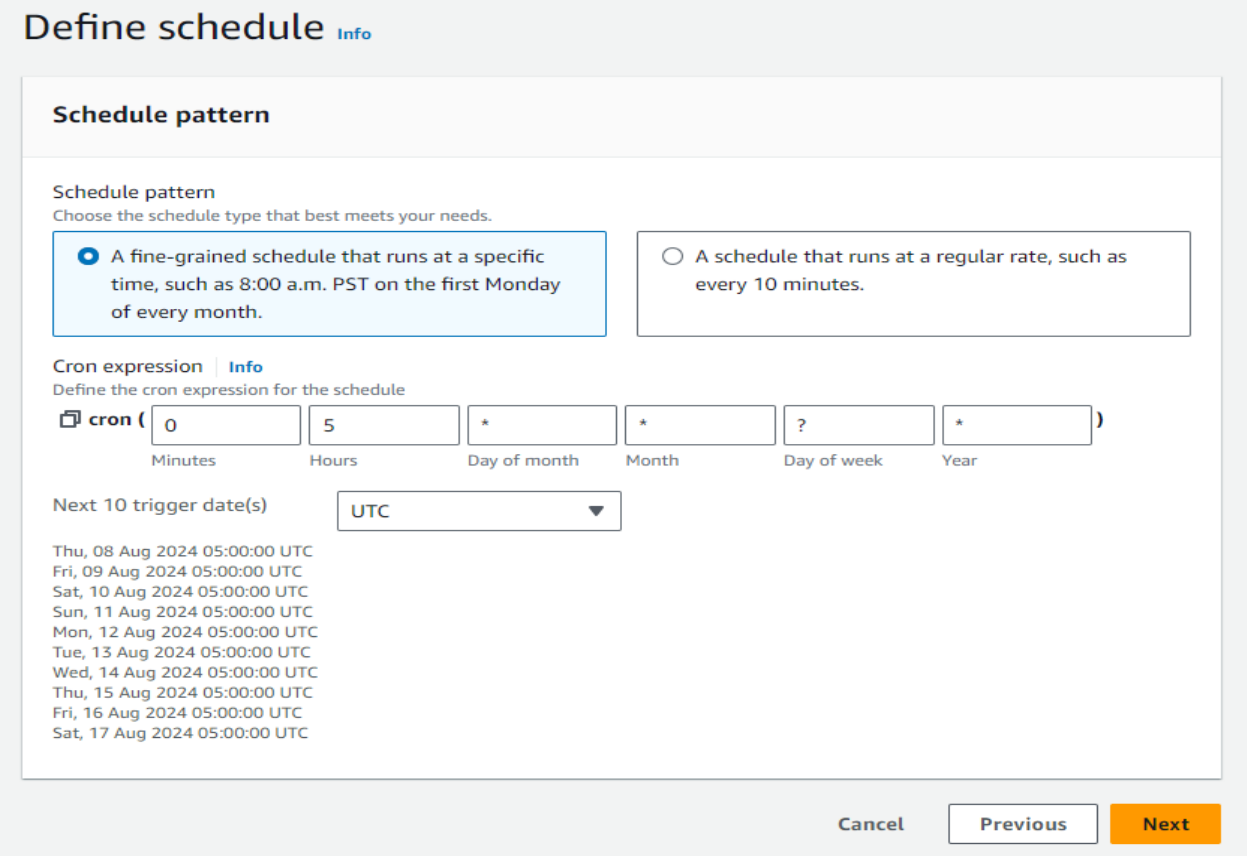
```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "ec2:CreateSnapshot",
    "ec2:DescribeInstances",
    "ec2:DescribeVolumes",
    "ec2:DescribeVolumeStatus",
    "sns:Publish"
   ],
   "Resource": "*"
  }
 ]
}
```

- Click Review policy, name the policy (e.g., LambdaEBSPolicy), and create it.
- Go back to the IAM role creation wizard, select Refresh to see the policy, and attach it to the role.
- Name the role (e.g., LambdaEBSRole) and create it.

Step 2: Create a CloudWatch Rule to Trigger the Lambda Function

- Go to the CloudWatch Service in the AWS Management Console.
- Click on Rules in the left-hand menu, then Create rule.
- In the Event Source section, select Event Source as Schedule.

- Enter the following cron expression to trigger the Lambda function daily at 10:30 AM IST:



- In the Targets section, click Add target and select Lambda function.
- Choose the Lambda function that you will create in the next step.
- Click Configure details and create the rule.

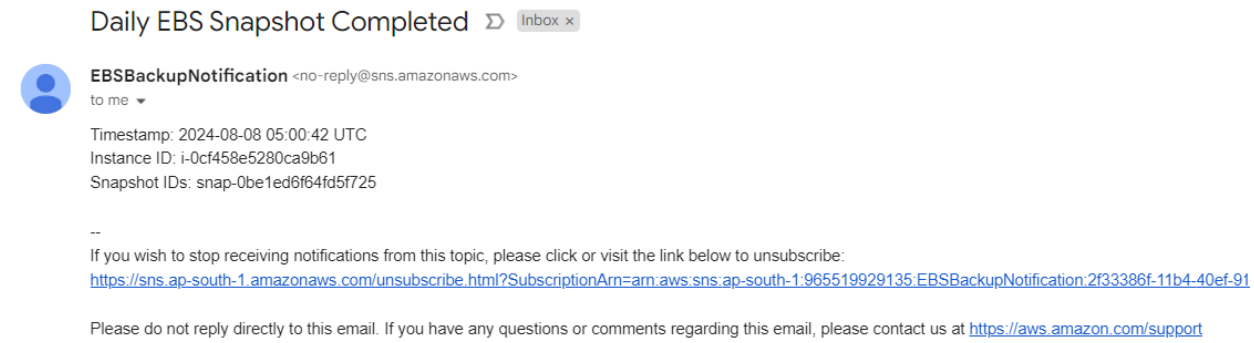
Step 3: Create an SNS Topic and Subscribe Your Email

- Go to the SNS Service in the AWS Management Console.
- Click on Create topic and choose Standard.
- Enter the topic name and create the topic.
- Click Create subscription, select Email as the protocol, and enter your email address.
- Confirm the subscription from the email you receive.

Step 4: Create the Lambda Function and attach the policy which we created in Step1.

```
import boto3
from datetime import datetime

ec2 = boto3.client('ec2')
sns = boto3.client('sns')

def lambda_handler(event, context):
    # Update with your instance ID
    instance_id = 'i-0cf458e5280ca9b61'

    # Describe the instance to get its volumes
    response = ec2.describe_instances(InstanceIds=[instance_id])
    volumes = [volume['Ebs']['VolumeId'] for reservation in response['Reservations'] for
instance in reservation['Instances'] for volume in instance['BlockDeviceMappings']]

    snapshot_ids = []
    timestamp = datetime.utcnow().strftime('%Y-%m-%d %H:%M:%S UTC')

    # Create a snapshot for each volume
    for volume_id in volumes:
        snapshot = ec2.create_snapshot(VolumeId=volume_id, Description='Daily snapshot')
        snapshot_ids.append(snapshot['SnapshotId'])

    # Prepare the message
    message = f"Timestamp: {timestamp}\n"
    message += f"Instance ID: {instance_id}\n"
    message += "Snapshot IDs: " + ", ".join(snapshot_ids)

    # Send the notification
    sns.publish(
        TopicArn='arn:aws:sns:ap-south-1:965519929135:EBSBackupNotification',
        Subject='Daily EBS Snapshot Completed',
        Message=message
    )
```

Output:



Daily EBS Snapshot Completed Inbox ×

EBSBackupNotification <no-reply@sns.amazonaws.com>
to me

Timestamp: 2024-08-08 05:00:42 UTC
Instance ID: i-0cf458e5280ca9b61
Snapshot IDs: snap-0be1ed6f64fd5f725

--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.ap-south-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:ap-south-1:965519929135:EBSBackupNotification:2f33386f-11b4-40ef-91

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at https://aws.amazon.com/support