**Project**

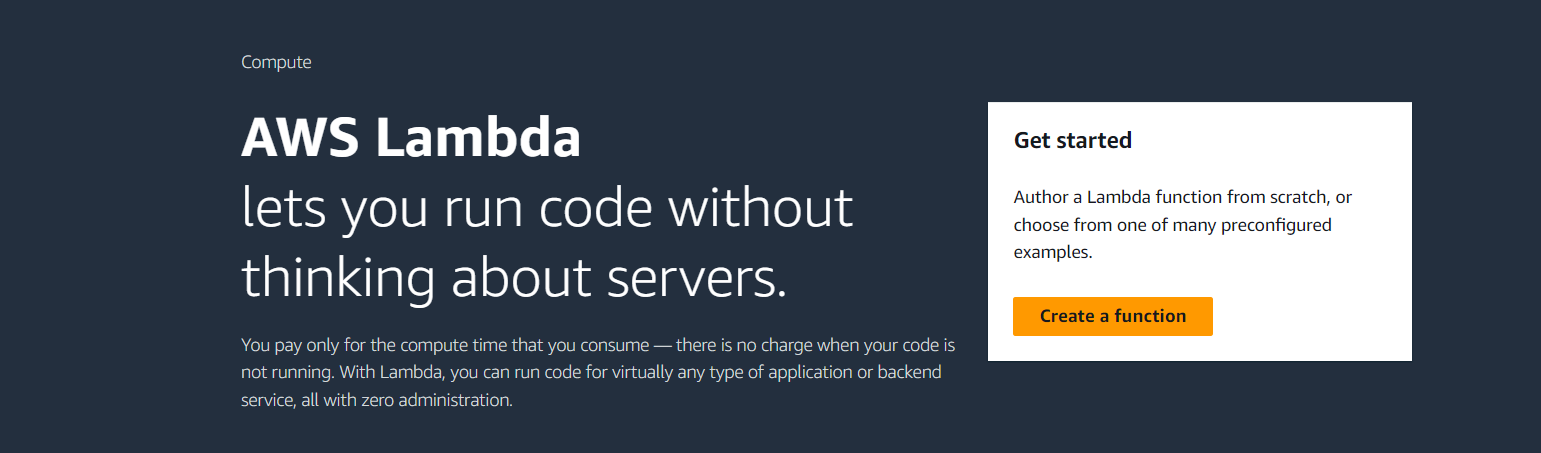
**Topic:** Maintaining Compliance for EBS Volume.

Project Description:

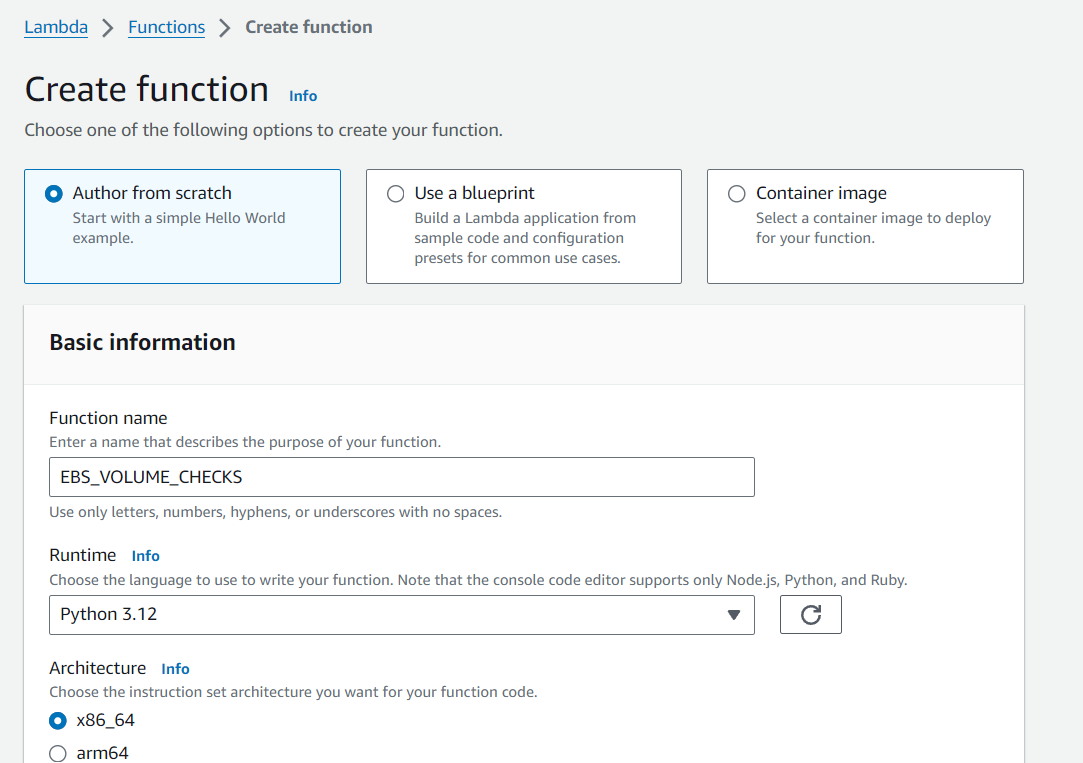
As a Cloud Engineering team, we take care of the AWS environment and make sure it complies with the organizational policies. We use AWS cloud watch in combination with AWS Lambda to govern the resources according to the policies. For example, we Trigger a Lambda function when an Amazon Elastic Block Store (EBS) volume is created. We use Amazon CloudWatch Events. CloudWatch Events allows us to monitor and respond to EBS volumes that are of type GP2 and convert them to type GP3.

**Step 1: Create a New Lambda Function**

1. **Navigate to Lambda Console**:
   * Go to the AWS Management Console.
   * Open the Lambda console.



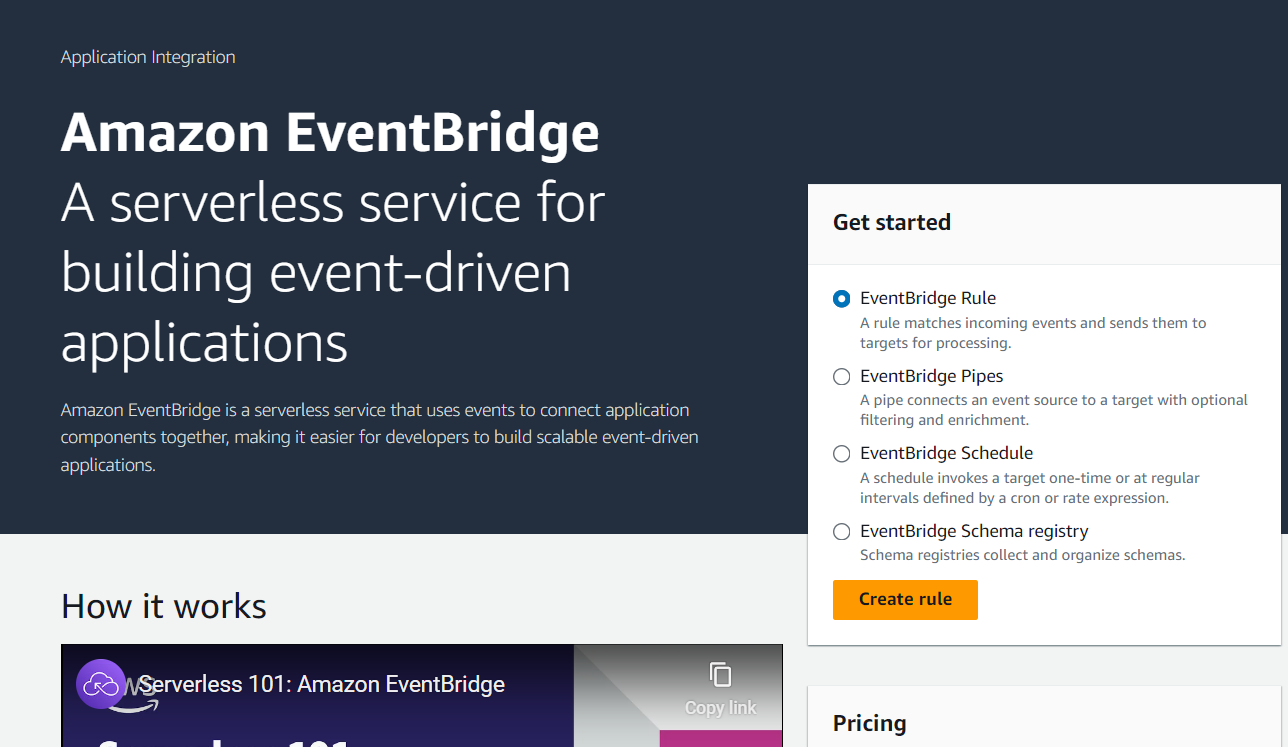
1. **Create Function**:
   * Click **Create function**.



* + Select **Author from scratch**.
  + Enter the function name as EBS\_VOLUME\_CHECKS.
  + Choose **Python 3.x** as the runtime.
  + Select **Create a new role with basic Lambda permissions**.
  + Click **Create function**.

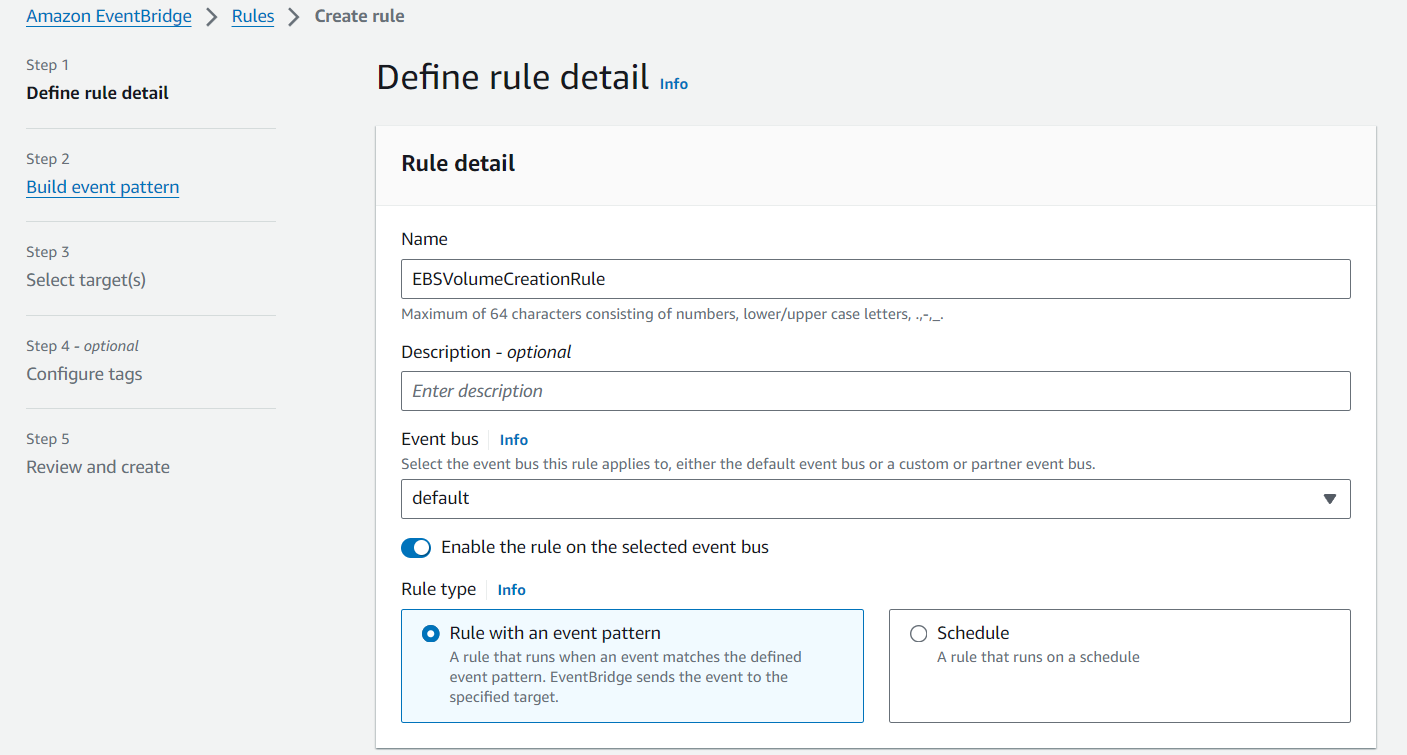
**Step 2: Create an EventBridge Rule**

1. **Navigate to EventBridge Console**:
   * Go to the AWS Management Console.

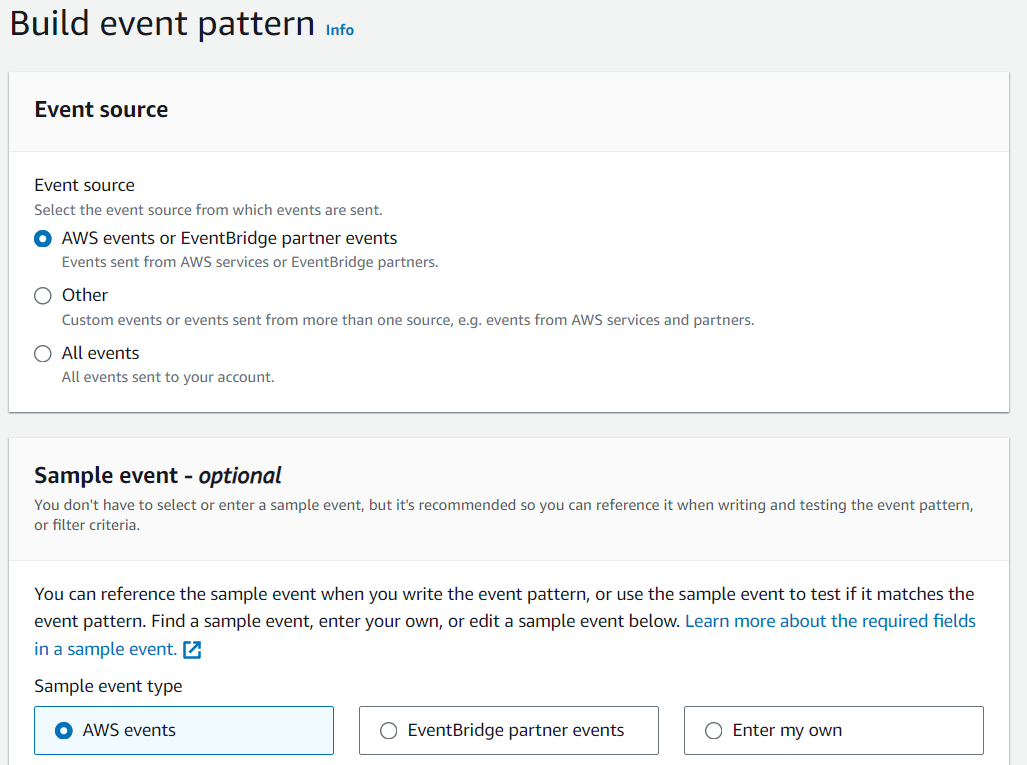


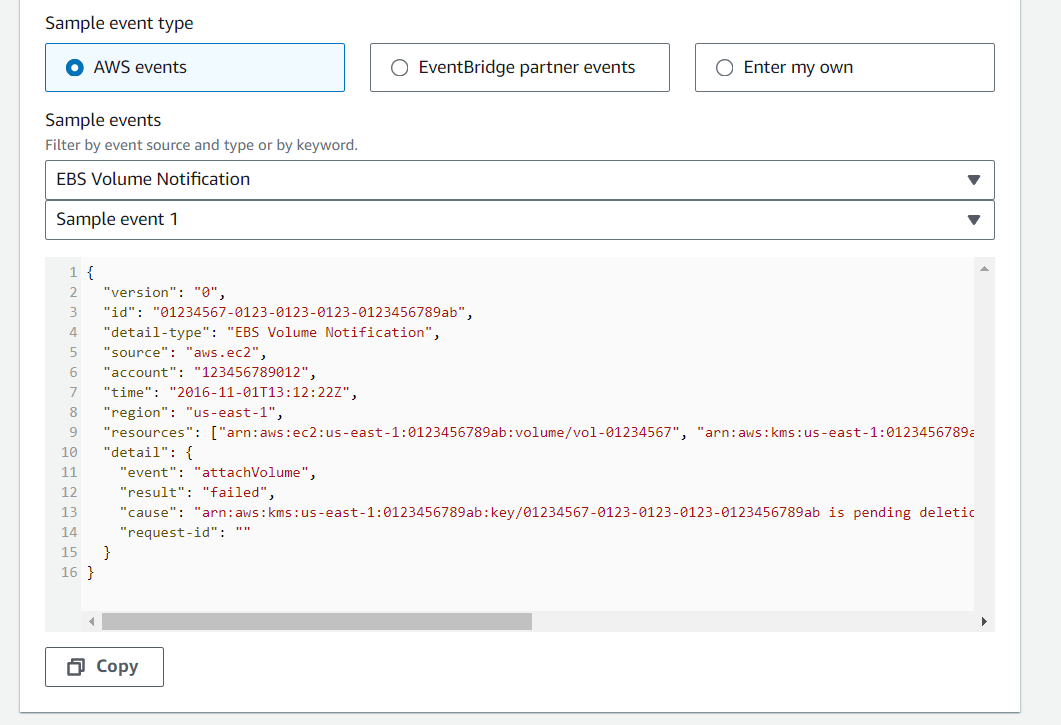
* + Open the Amazon EventBridge console (formerly CloudWatch Events).

1. **Create Rule**:
   * Click **Create rule**.

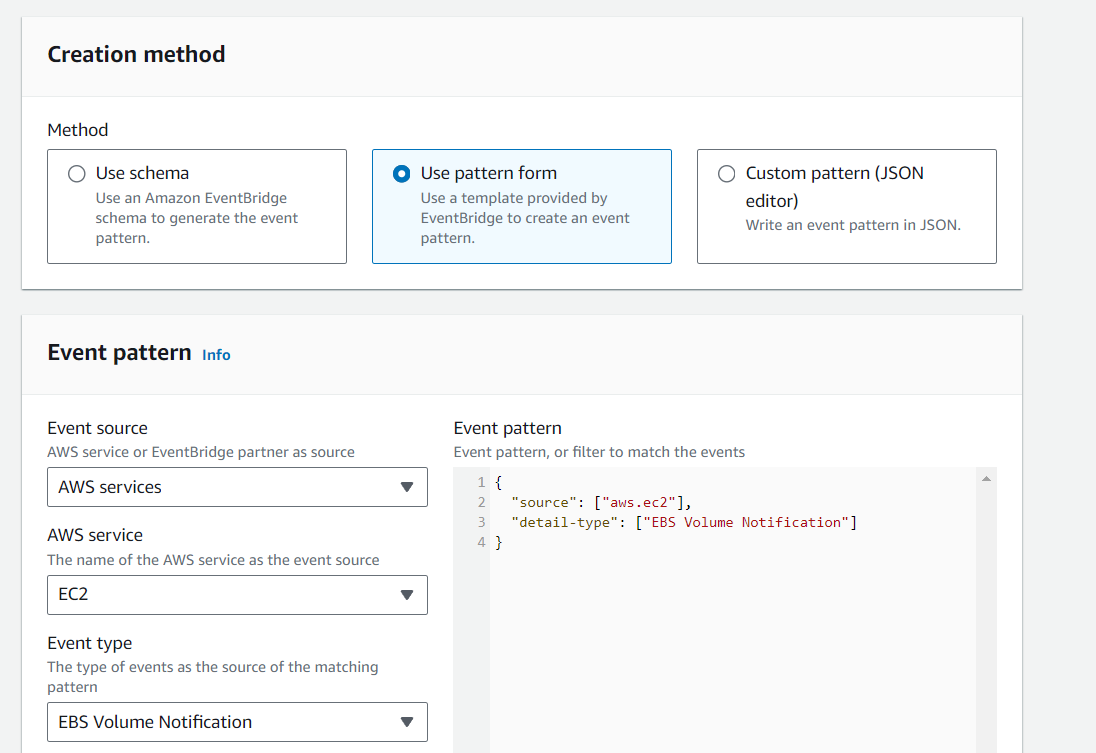


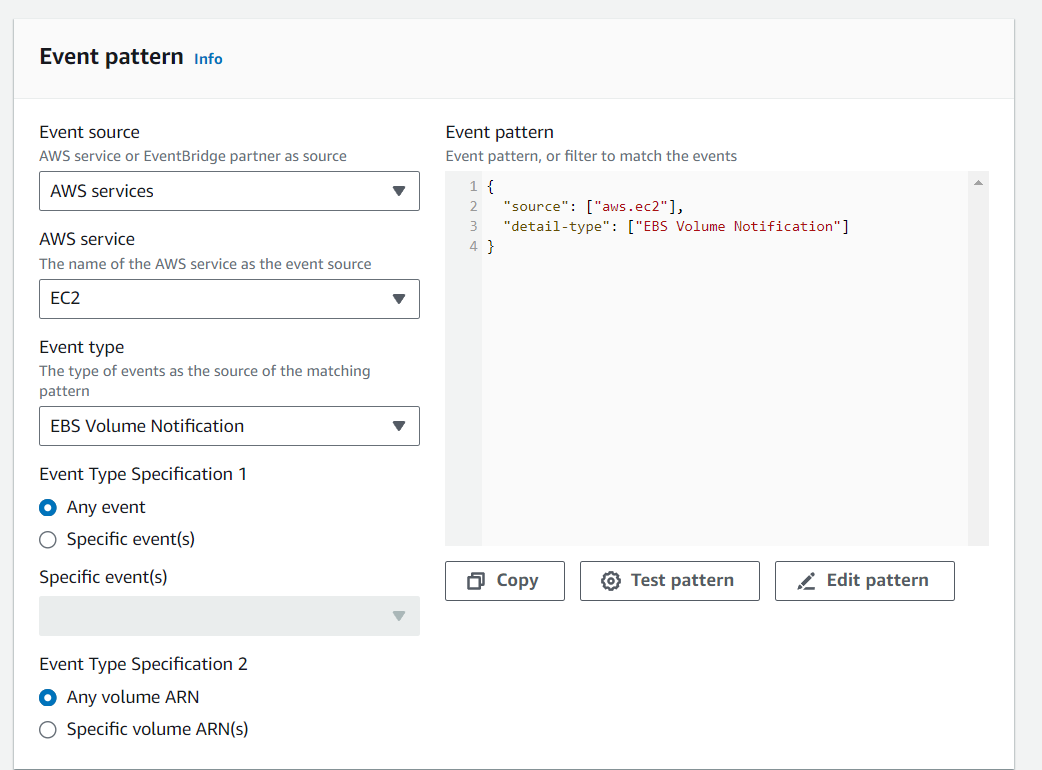
* + Enter a name for the rule, e.g., EBSVolumeCreationRule.



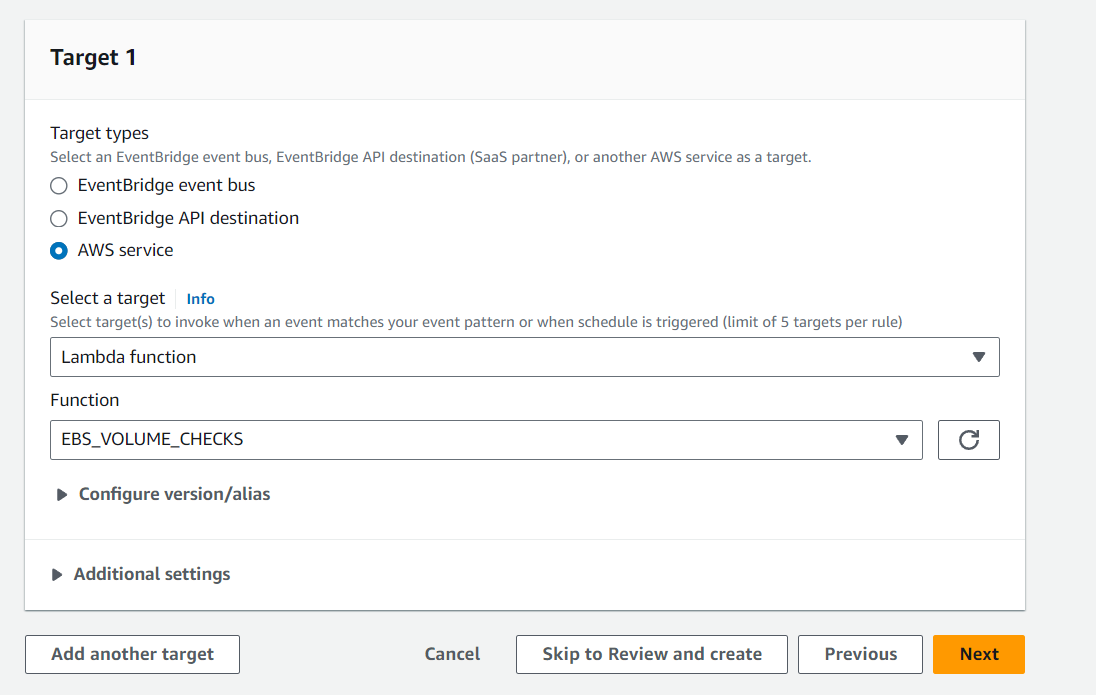


* + For **Event Source**, choose **Event Source** as **EC2** and specify **Volume State Change**.





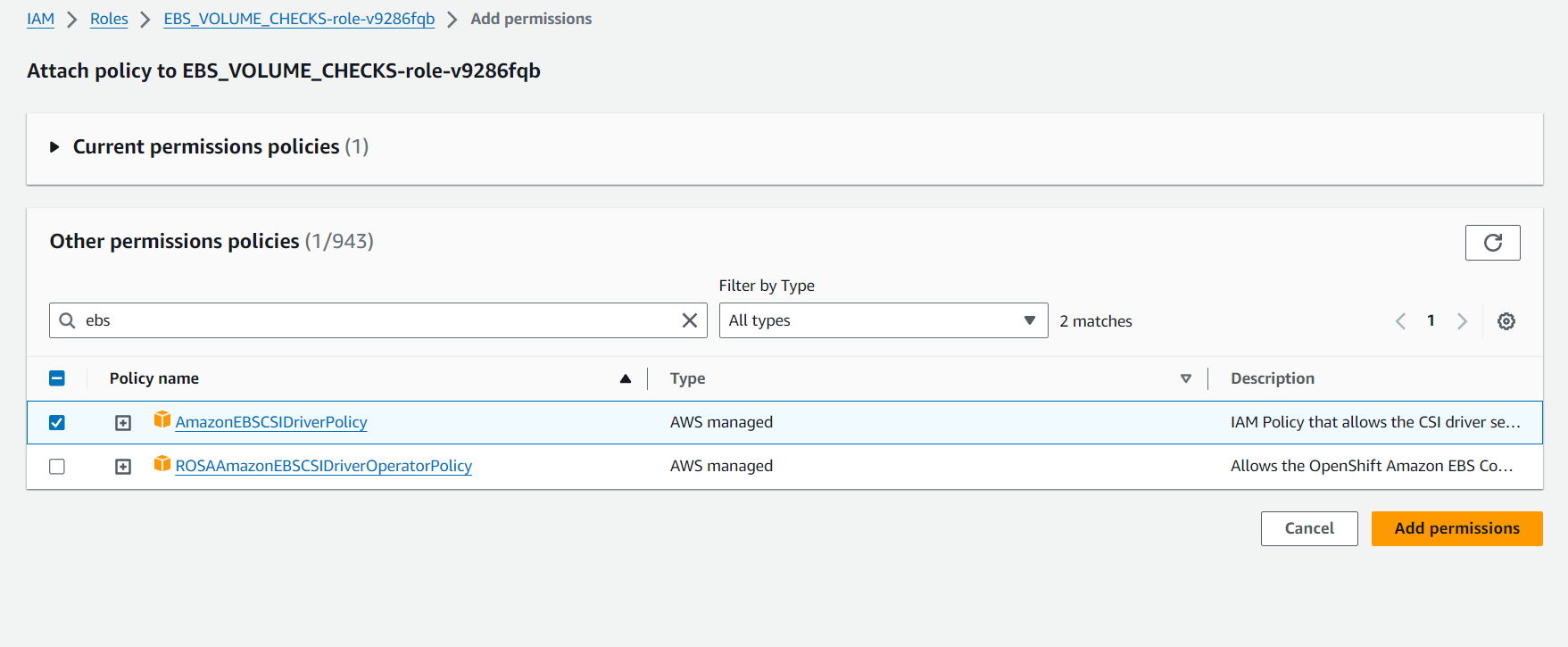
* + For **Event Pattern**, Use Event Source as AWS services.
  + Choose AWS service as EC2.
  + Event type -> EBS Volume Notification.



* + For **Targets**, select **Lambda function** and choose the EBS\_VOLUME\_CHECKS function created in Step 1.
  + Click **Create**.

**Step 3: Update Lambda Execution Role**

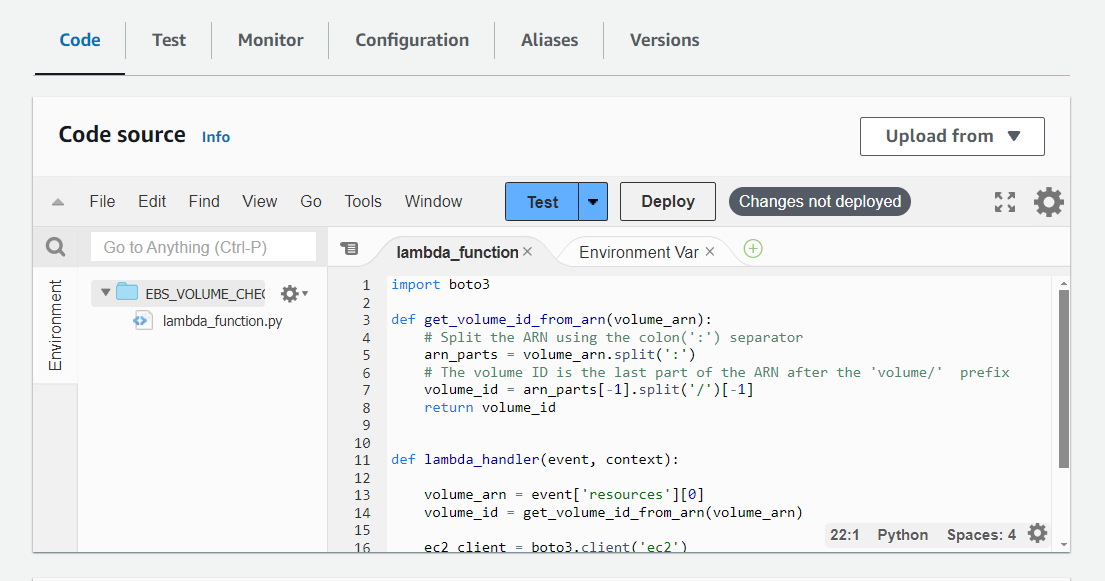
1. **Navigate to IAM Console**:
   * Go to the AWS Management Console.
   * Open the IAM console.
2. **Edit Role**:
   * In the navigation pane, choose **Roles**.
   * Select the role created for the Lambda function (e.g., EBS\_VOLUME\_CHECKS-role).
3. **Add Inline Policy**:
   * Scroll down to the **Permissions** tab.



* + Click **Add attach policy**.
  + Choose AmazonEBSCSIDriverPolicy.
  + Click **Add permissions**.

**Step 4: Update Lambda Function Code**

1. **Navigate to Lambda Console**:
   * Open the Lambda console.
   * Select the EBS\_VOLUME\_CHECKS function.
2. **Update Function Code**:



* + Add the following code to your Lambda function:

python

Copy code

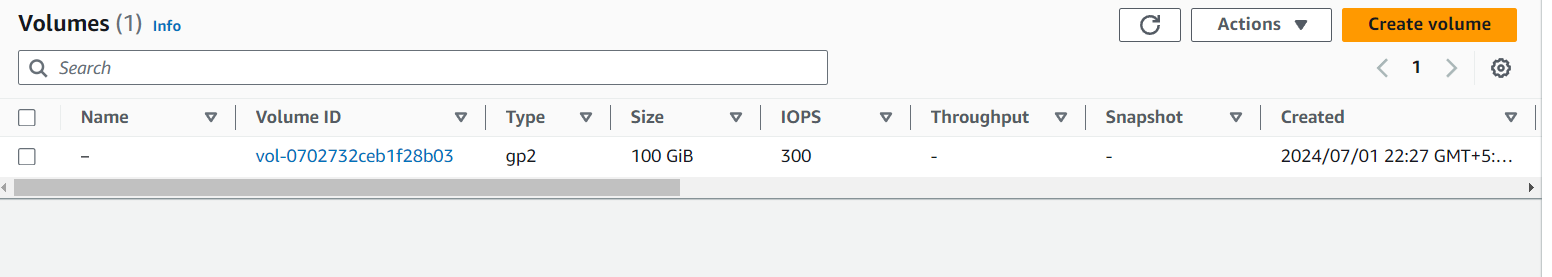
* import boto3
* def get\_volume\_id\_from\_arn(volume\_arn):
* # Split the ARN using the colon(':') separator
* arn\_parts = volume\_arn.split(':')
* # The volume ID is the last part of the ARN after the 'volume/' prefix
* volume\_id = arn\_parts[-1].split('/')[-1]
* return volume\_id

* def lambda\_handler(event, context):
* volume\_arn = event['resources'][0]
* volume\_id = get\_volume\_id\_from\_arn(volume\_arn)
* ec2\_client = boto3.client('ec2')
* response = ec2\_client.modify\_volume(
* VolumeId=volume\_id,
* VolumeType='gp3',
* )

Click **Deploy**.

**Step 5: Test the Lambda Function**

1. **Create a New GP2 Volume**:
   * Go to the EC2 console.



* + Create a new EBS volume with type GP2.

1. **Check Result:**

