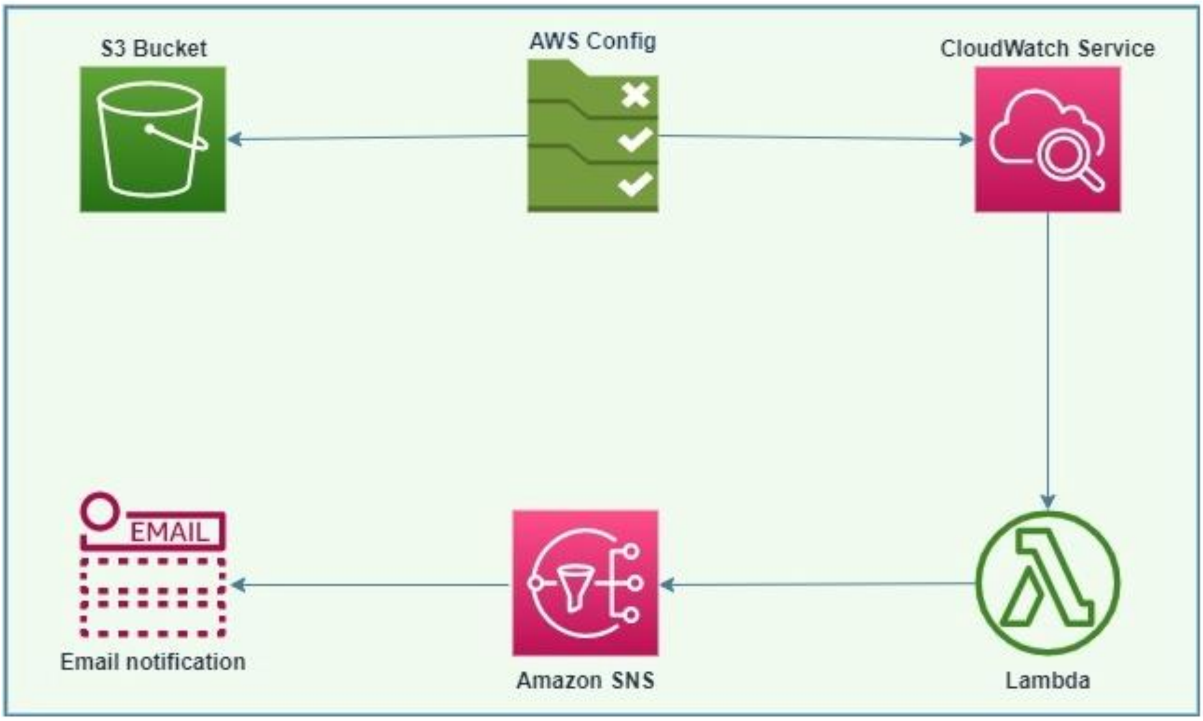
Automating a solution that will evaluate S3 server-side encryption status and remediate noncompliant buckets using Python code in Lambda

To achieve this, I utilized the combination of AWS services which include AWS Config, S3 Storage, CloudWatch, Lambda, IAM and SNS as shown in the blueprint below.



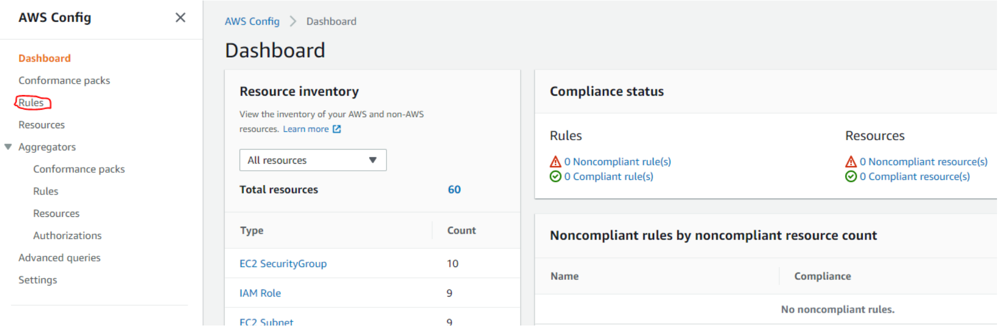
- AWS Config: used to run the baseline rule which is continuously compared with snapshot of the resources in the account to identify resources that are compliant and also the ones that are noncompliant with the defined configuration rule. This will create a list of compliant buckets and also another list of any noncompliant bucket.
- S3 Storage: This is the object storage resource which we are running a configuration check for in this project to identify if the buckets in the region are having default encryption enabled or not.
- CloudWatch: This is a monitoring service which can evoke lambda whenever there is any noncompliant bucket at any time from the AWS Config process. A JSON script is used to create an event that will trigger Lambda from AWS Config.
- Lambda: This is a very important service of AWS used to run code in a cost-efficient manner. It is a serverless service that will only run the code whenever it is triggered and we will only be billed for the period only. A python script is

used to enforce remediation action on noncompliant S3 buckets whenever AWS Config identify any. The S3 bucket parameters are passed to the Lambda through the CloudWatch event created.

- IAM roles: These are created to grant permission for services to perform required actions on other AWS resources. For instance, Lambda will be granted IAM role to allow it to perform S3 bucket encryption update.
- Amazon SNS: It is very important to communicate the details of the noncompliant S3 buckets found by the AWS config whenever it identifies any. An email will also be sent to designated person/group whenever the Lambda code remediate any noncompliant S3 bucket for audit purpose.

Setting up the AWS Config

- Click on the Rules option from the AWS Config dashboard page.



Resource inventory

View the inventory of your AWS and non-AWS resources. Learn more

All resources

Total resources 60

Type	Count
EC2 SecurityGroup	10
IAM Role	9
EC2 Subnet	9

Compliance status

Rules

0 Noncompliant rule(s)
0 Compliant rule(s)

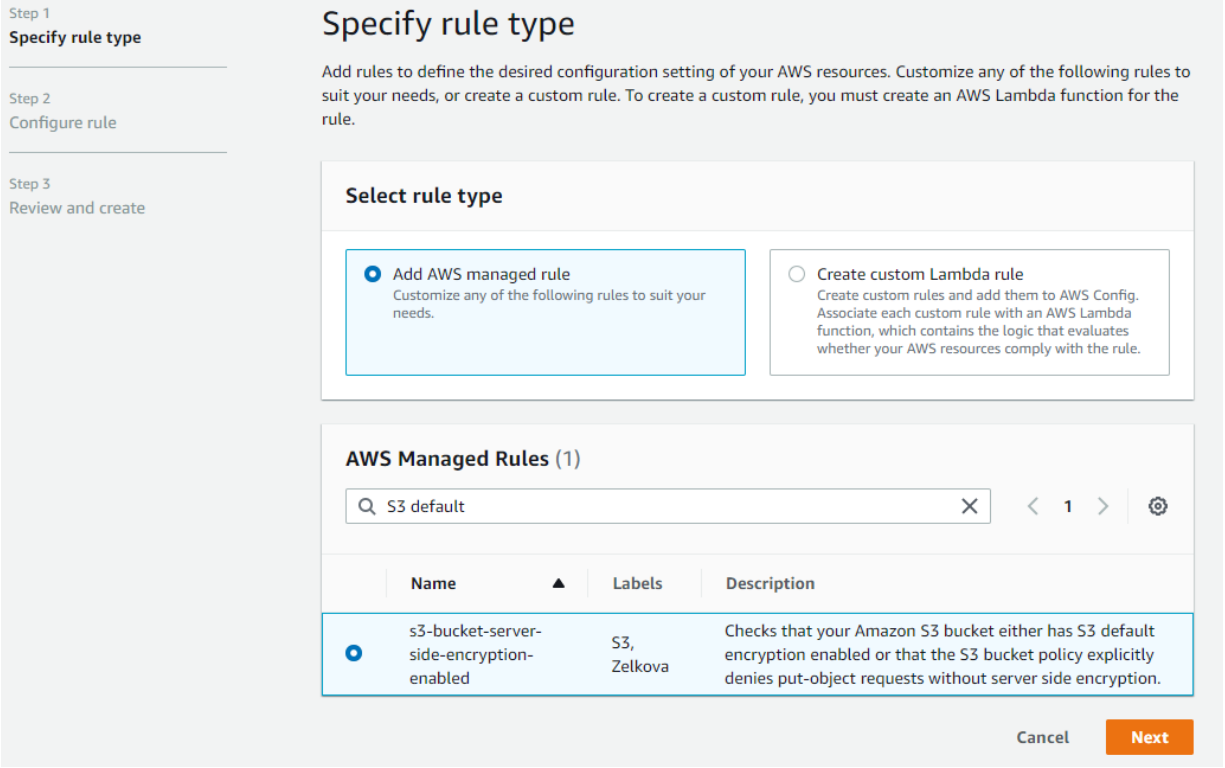
Resources

0 Noncompliant resource(s)
0 Compliant resource(s)

Noncompliant rules by noncompliant resource count

Name	Compliance
No noncompliant rules.	

- Click on "Add rule" to specify the rule type. There are AWS managed rules one can pick from and there is an alternative to create a customs Lambda rule. In this project, we will be using an AWS managed rule for S3 default encryption which is the S3 bucket server-side encryption enabled rule. Once selected from the search rule box, we click next.



Step 1
Specify rule type

Step 2
Configure rule

Step 3
Review and create

Specify rule type

Add rules to define the desired configuration setting of your AWS resources. Customize any of the following rules to suit your needs, or create a custom rule. To create a custom rule, you must create an AWS Lambda function for the rule.

Select rule type

● Add AWS managed rule
Customize any of the following rules to suit your needs.

○ Create custom Lambda rule
Create custom rules and add them to AWS Config. Associate each custom rule with an AWS Lambda function, which contains the logic that evaluates whether your AWS resources comply with the rule.

AWS Managed Rules (1)

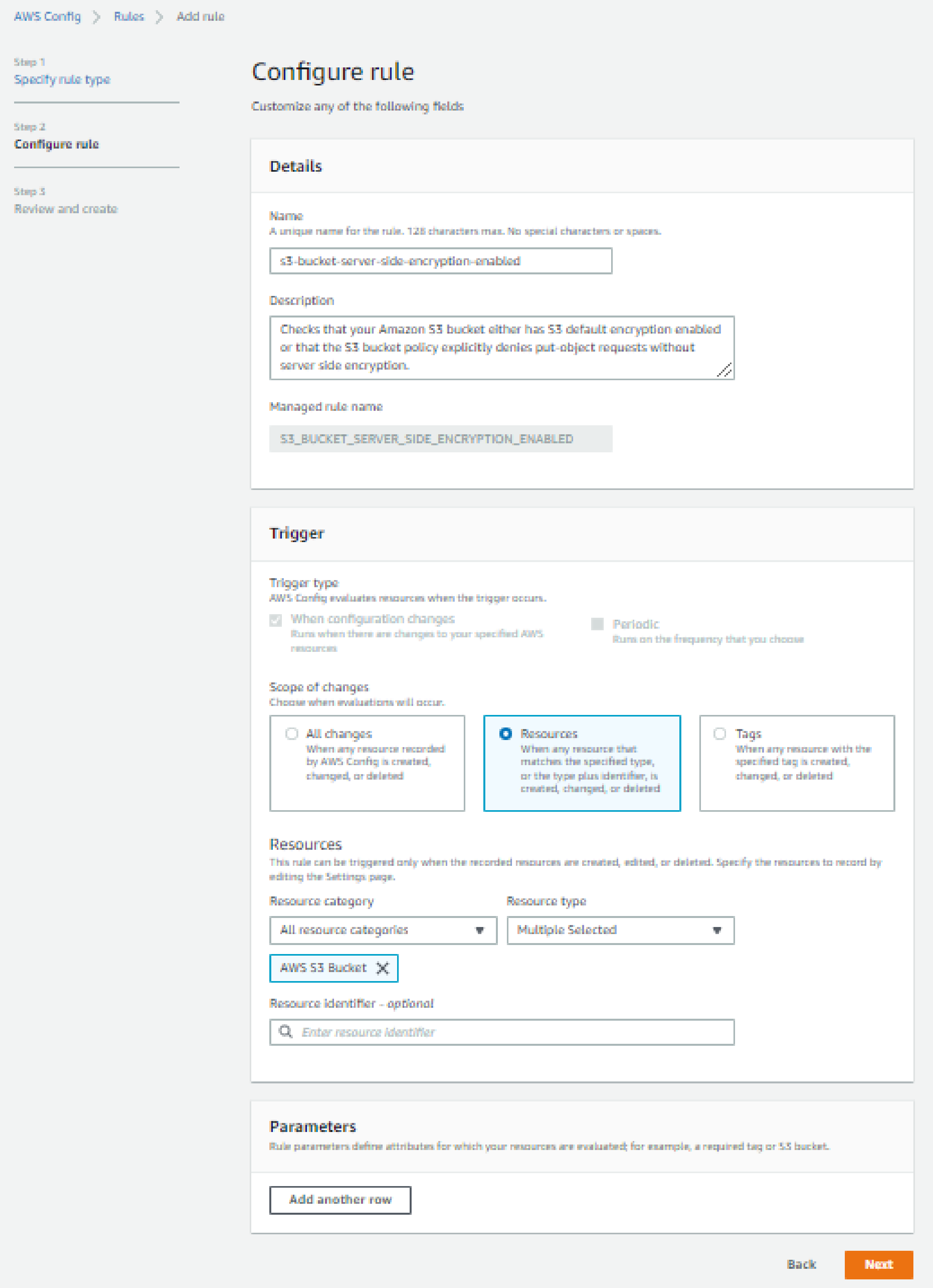
S3 default

	Name ▲	Labels	Description
●	s3-bucket-server-side-encryption-enabled	S3, Zelkova	Checks that your Amazon S3 bucket either has S3 default encryption enabled or that the S3 bucket policy explicitly denies put-object requests without server side encryption.

- Under the Configure rule step, the rule selected will automatically filed and the second section of this page is for Trigger. This refer to the changes that will evoke config rule and the resource/s that the config rule will be evaluated against. Kindly select “Resources” on the scope of changes, select “All resource categories” under Resource category and select specific resource under resource type option (AWS S3 Bucket) was selected in this case.
- Resource identifier is optional and left blank as we are planning to have this configuration rule evaluated against any of the S3 buckets in the region. Parameters can be used to specify tags or specific bucket is that is the requirement, we will leave that and click next to review the config rule.
- The review and create page give us an overview of the selected rule and resources we have specified from above. Kindly click on back to change anything if necessary or click the add rule button to create our config rule.
- AWS Config is integrated with Amazon CloudWatch and therefore send information about configuration changes and notifications to CloudWatch Events.

Amazon CloudWatch Events rule

AWS Config sends detailed information about the configuration changes and notifications to Amazon CloudWatch Events. To create rules, visit the Step 1: Create Rule page in the Amazon CloudWatch Events console.



AWS Config > Rules > Add rule

Step 1
Specify rule type

Step 2
Configure rule

Step 3
Review and create

Configure rule

Customize any of the following fields

Details

Name
A unique name for the rule. 128 characters max. No special characters or spaces.

s3-bucket-server-side-encryption-enabled

Description

Checks that your Amazon S3 bucket either has S3 default encryption enabled or that the S3 bucket policy explicitly denies put-object requests without server side encryption.

Managed rule name

S3_BUCKET_SERVER_SIDE_ENCRYPTION_ENABLED

Trigger

Trigger type
AWS Config evaluates resources when the trigger occurs.

☑ When configuration changes
Runs when there are changes to your specified AWS resources

□ Periodic
Runs on the frequency that you choose

Scope of changes
Choose when evaluations will occur.

○ All changes
When any resource recorded by AWS Config is created, changed, or deleted

● Resources
When any resource that matches the specified type, or the type plus identifier, is created, changed, or deleted

○ Tags
When any resource with the specified tag is created, changed, or deleted

Resources
This rule can be triggered only when the recorded resources are created, edited, or deleted. Specify the resources to record by editing the Settings page.

Resource category: All resource categories

Resource type: Multiple Selected

AWS S3 Bucket ✕

Resource identifier - *optional*

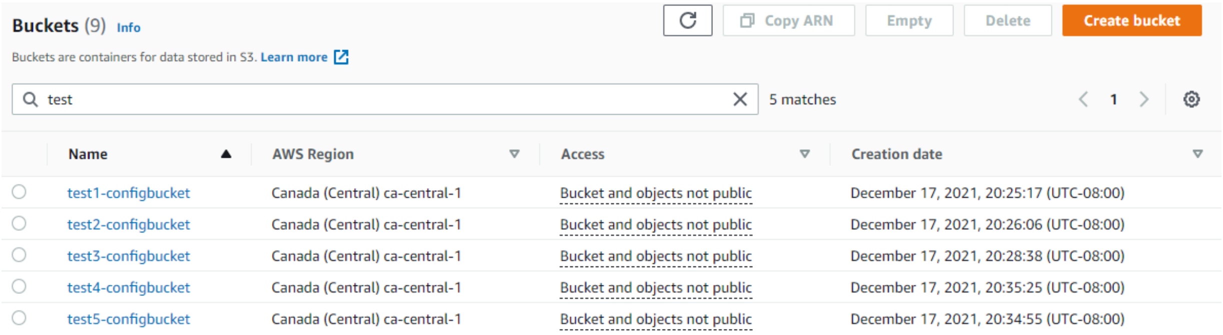
Enter resource identifier

Parameters

Rule parameters define attributes for which your resources are evaluated; for example, a required tag or S3 bucket.

Simple Storage Service -S3 Buckets creation

- We have created a couple of buckets for this project, some with S3 buckets with the name test 1, test2, test3, test4 and test5. Test1, test4 and test5 buckets all created with default encryption disabled so they can be flagged as noncompliant and will be remediated by the Lambda code. Both buckets named test2 and test3 are created with default encryption enabled and expected to meet the AWS Config evaluation as compliant.



Buckets (9) Info

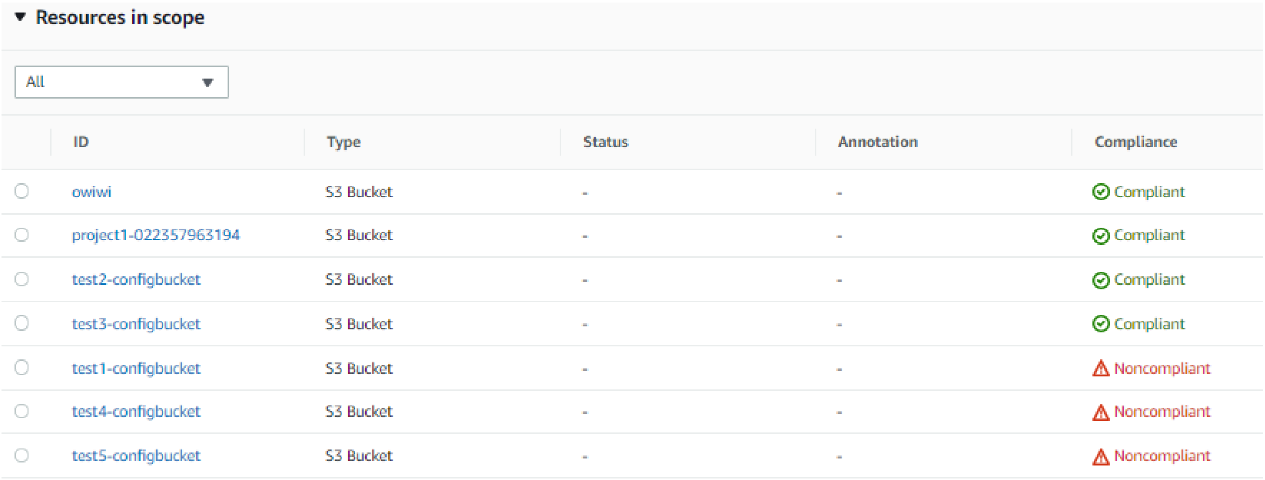
Buckets are containers for data stored in S3. Learn more

test

5 matches

	Name ▲	AWS Region	Access	Creation date
○	test1-configbucket	Canada (Central) ca-central-1	Bucket and objects not public	December 17, 2021, 20:25:17 (UTC-08:00)
○	test2-configbucket	Canada (Central) ca-central-1	Bucket and objects not public	December 17, 2021, 20:26:06 (UTC-08:00)
○	test3-configbucket	Canada (Central) ca-central-1	Bucket and objects not public	December 17, 2021, 20:28:38 (UTC-08:00)
○	test4-configbucket	Canada (Central) ca-central-1	Bucket and objects not public	December 17, 2021, 20:35:25 (UTC-08:00)
○	test5-configbucket	Canada (Central) ca-central-1	Bucket and objects not public	December 17, 2021, 20:34:55 (UTC-08:00)

- Once AWS Config evaluate the predefined configuration rule against all S3 buckets within the same region which the config rule was created (Canada central 1 used in this project), the complaint and noncompliant buckets are identified with appropriate tags created by AWS Config.



▼ Resources in scope

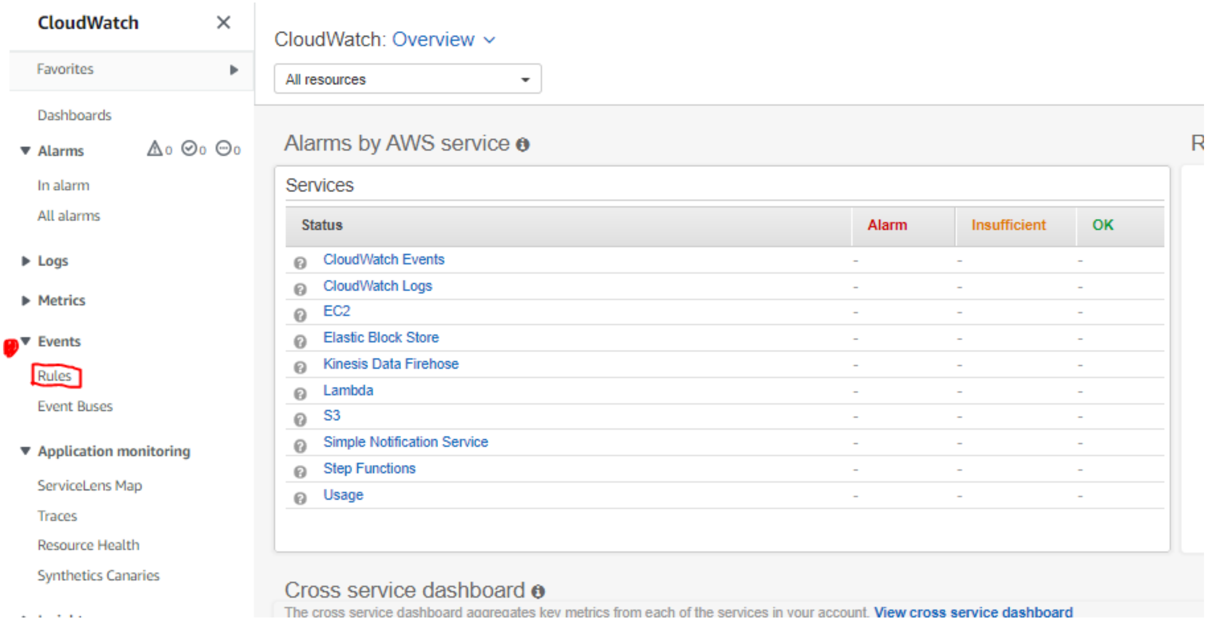
All ▼

	ID	Type	Status	Annotation	Compliance
○	owiwi	S3 Bucket	-	-	✓ Compliant
○	project1-022357963194	S3 Bucket	-	-	✓ Compliant
○	test2-configbucket	S3 Bucket	-	-	✓ Compliant
○	test3-configbucket	S3 Bucket	-	-	✓ Compliant
○	test1-configbucket	S3 Bucket	-	-	⚠ Noncompliant
○	test4-configbucket	S3 Bucket	-	-	⚠ Noncompliant
○	test5-configbucket	S3 Bucket	-	-	⚠ Noncompliant

Amazon CloudWatch Event

- This is very important to integrate Lambda into the architecture for the action we will like to take on noncompliant bucket/s whenever the created AWS

Config identifies any. To create this Amazon CloudWatch event, kindly click on the event under the CloudWatch dashboard and select rules.



Alarms by AWS service

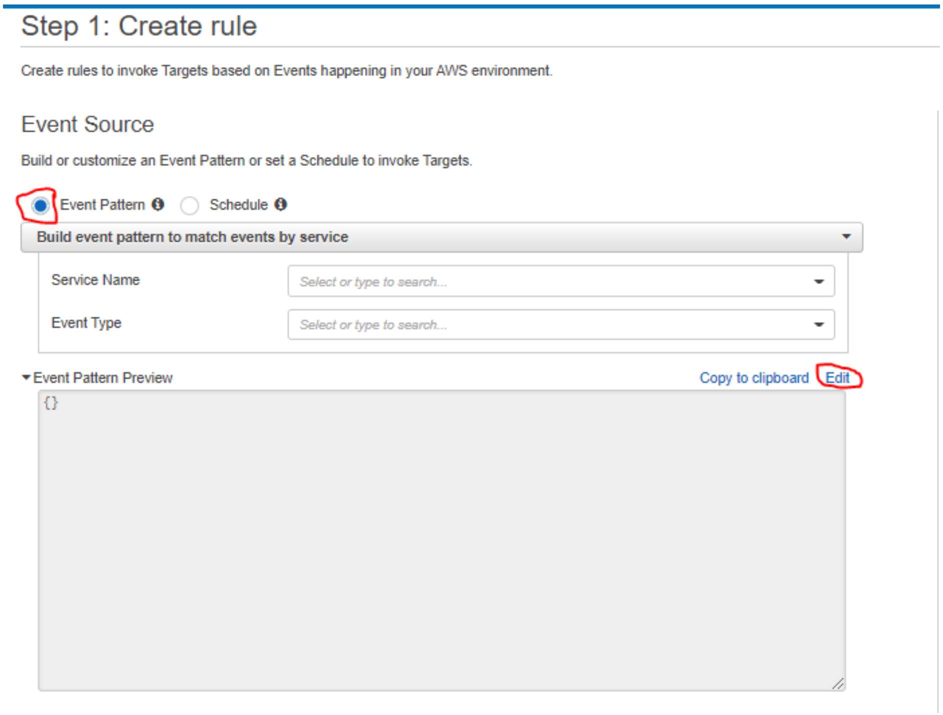
Services

Status	Alarm	Insufficient	OK
CloudWatch Events	-	-	-
CloudWatch Logs	-	-	-
EC2	-	-	-
Elastic Block Store	-	-	-
Kinesis Data Firehose	-	-	-
Lambda	-	-	-
S3	-	-	-
Simple Notification Service	-	-	-
Step Functions	-	-	-
Usage	-	-	-

Cross service dashboard

The cross service dashboard aggregates key metrics from each of the services in your account. View cross service dashboard

- Click on create rule button and under step 1, choose the event pattern that will be used to evoke Lambda. Click edit to paste the JSON script that will constantly lookup the AWS config generated output and pick necessary details of any noncompliant S3 bucket which will be passed to Lambda for remediation. Click to save the JSON after pasting.



Step 1: Create rule

Create rules to invoke Targets based on Events happening in your AWS environment.

Event Source

Build or customize an Event Pattern or set a Schedule to invoke Targets.

● Event Pattern ○ Schedule

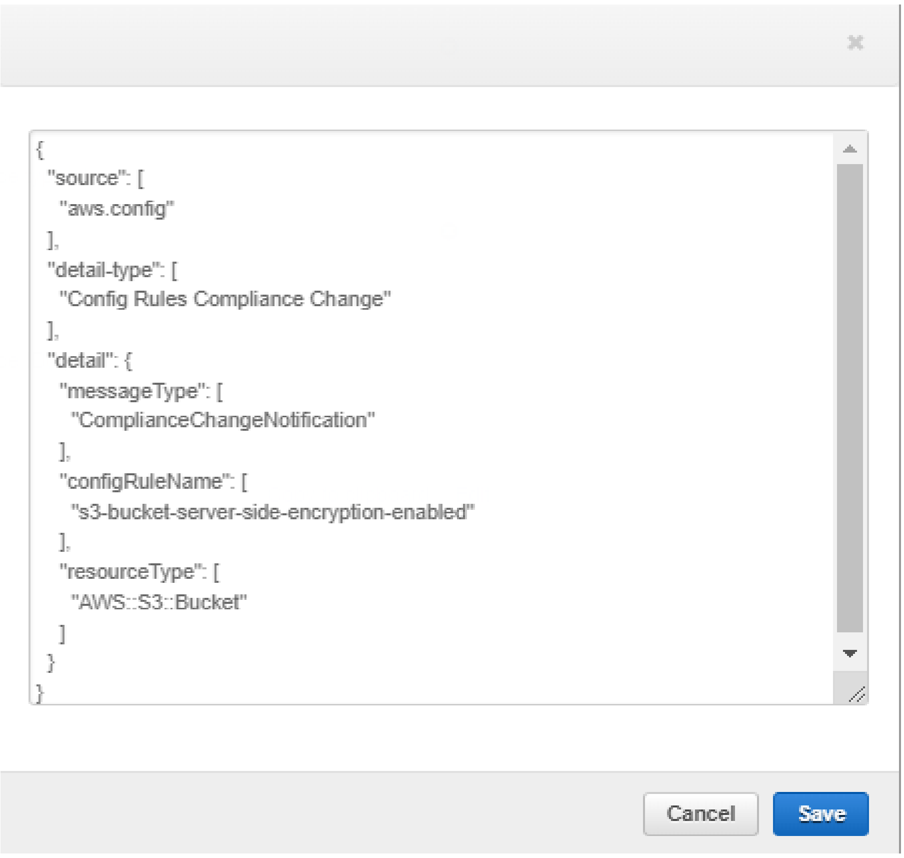
Build event pattern to match events by service

Service Name: Select or type to search...

Event Type: Select or type to search...

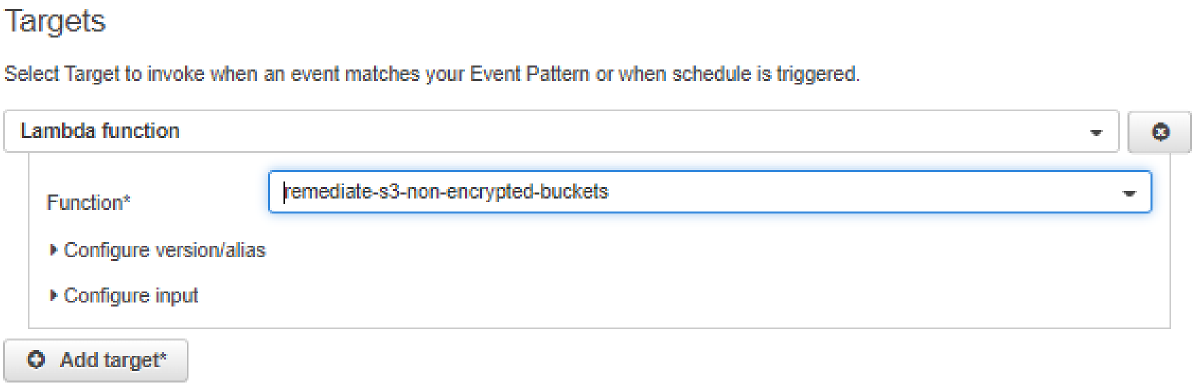
Event Pattern Preview

```
{}
```



```
{
  "source": [
    "aws.config"
  ],
  "detail-type": [
    "Config Rules Compliance Change"
  ],
  "detail": {
    "messageType": [
      "ComplianceChangeNotification"
    ],
    "configRuleName": [
      "s3-bucket-server-side-encryption-enabled"
    ],
    "resourceType": [
      "AWS::S3::Bucket"
    ]
  }
}
```

- Next is the “Targets” where we can select the service that should be invoke whenever there is an event that matches the defined event pattern specified through the JSON script. Kindly click add target and select Lambda function from the dropdown list of services available.
- The lambda function created will also be selected if one is already created with the python code to remediate any AWS Config identified noncompliant S3 bucket. The created lambda function was named “remediate-S3-non-encrypted-buckets and was available to be selected as shown below.



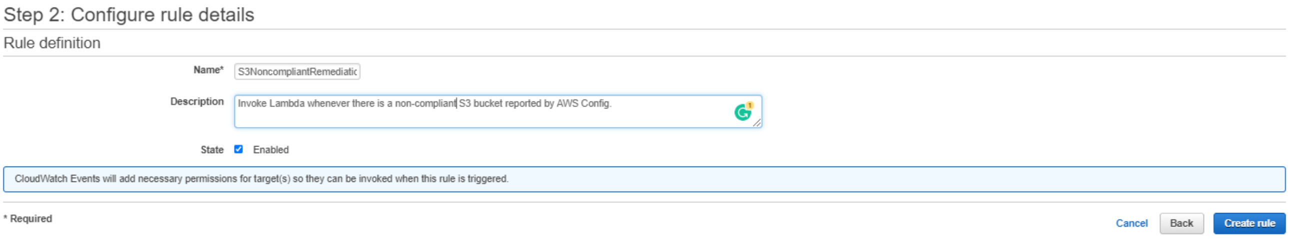
Targets

Select Target to invoke when an event matches your Event Pattern or when schedule is triggered.

Lambda function

Function* remediate-s3-non-encrypted-buckets

- In step 2, add a name for the CloudWatch event rule and click create rule.



Step 2: Configure rule details

Rule definition

Name* S3NoncompliantRemediatic

Description Invoke Lambda whenever there is a non-compliant S3 bucket reported by AWS Config.

State ☑ Enabled

CloudWatch Events will add necessary permissions for target(s) so they can be invoked when this rule is triggered.

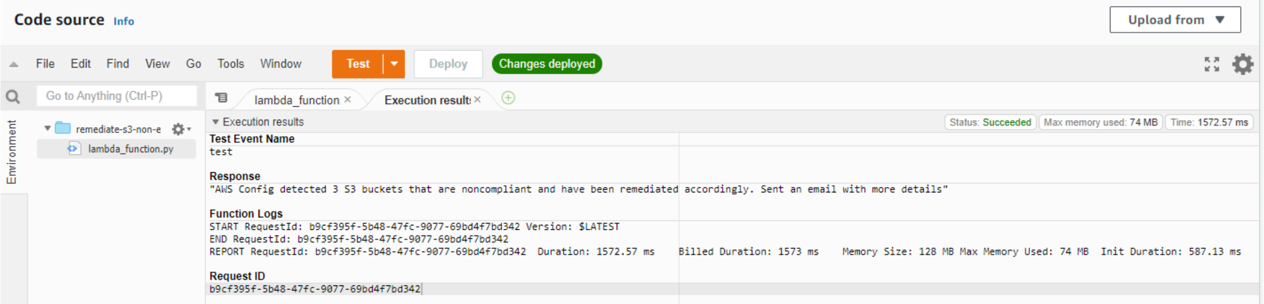
* Required

AWS Lambda Function

- Select the “functions” section from the Lambda dashboard and click on create function.
- On the page, please click on the “create function” button.
- Choose python as your runtime to use a python script for the API.
- Paste the code below to have Lambda pull the list of noncompliant S3 buckets from AWS Config and remediate each of them using defined SSE AES256.

```
import json
import boto3


config_client = boto3.client('config')
s3_client = boto3.client('s3')
sns_client = boto3.client('sns')
def lambda_handler(event, context):
    #Get all noncompliant resources from awsconfig
    response = config_client.get_compliance_details_by_config_rule(
    ConfigRuleName='s3-bucket-server-side-encryption-enabled',
    ComplianceTypes=['NON_COMPLIANT'],
    )
    non_compliant=json.loads(json.dumps(response, default=str))


    eval_result=non_compliant.get('EvaluationResults') #this will be a list of noncompliant s3 buckets from config


    for e in eval_result:
        non_comp_rsc = e.get('EvaluationResultIdentifier').get('EvaluationResultQualifier').get('ResourceId')


        response = s3_client.put_bucket_encryption(
    Bucket = non_comp_rsc,
    #set this bucket to default encryption enabled state
    ServerSideEncryptionConfiguration={
        'Rules': [
            {
                'ApplyServerSideEncryptionByDefault': {
                    'SSEAlgorithm': 'AES256'},
            },
        ]
    }
)



    eval_result=non_compliant.get('EvaluationResults') #this will be a list of noncompliant s3 buckets from config


    for e in eval_result:
        non_comp_rsc = e.get('EvaluationResultIdentifier').get('EvaluationResultQualifier').get('ResourceId')


        response = sns_client.publish(
    TopicArn="arn:aws:sns:ca-central-1:022357963194:S3-Compliance-topic",
    Subject= "Noncompliant S3 bucket Remediated",
    Message=f"Hello, {non_comp_rsc} bucket have been identified as non-compliant through the aws-config and have been remediated using Lambda to enforce SSE encryption on {non_comp_rsc}"
    )
```

- Below is the message printed after the Lambda code was successfully executed.



Test Event Name

```
test
```

Response

```
"AWS Config detected 3 S3 buckets that are noncompliant and have been remediated accordingly. Sent an email with more details"
```

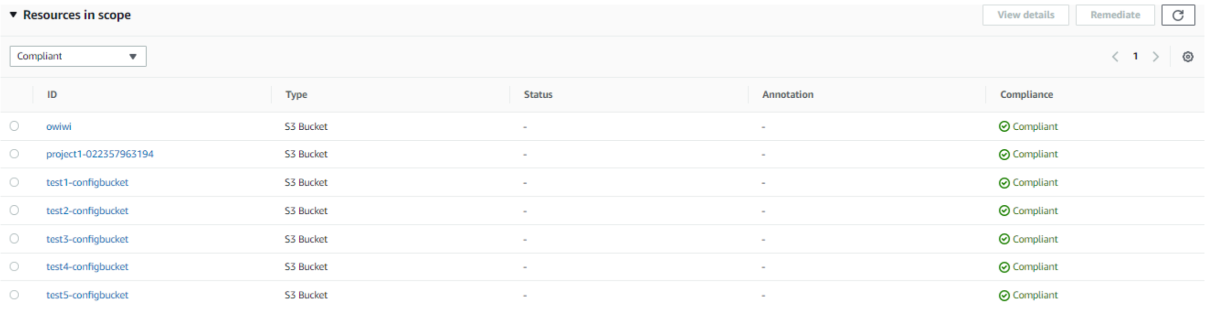
Function Logs

```
START RequestId: b9cf395f-5b48-47fc-9077-69bd4f7bd342 Version: $LATEST
END RequestId: b9cf395f-5b48-47fc-9077-69bd4f7bd342
REPORT RequestId: b9cf395f-5b48-47fc-9077-69bd4f7bd342  Duration: 1572.57 ms    Billed Duration: 1573 ms    Memory Size: 128 MB Max Memory Used: 74 MB  Init Duration: 587.13 ms
```

Request ID

```
b9cf395f-5b48-47fc-9077-69bd4f7bd342
```

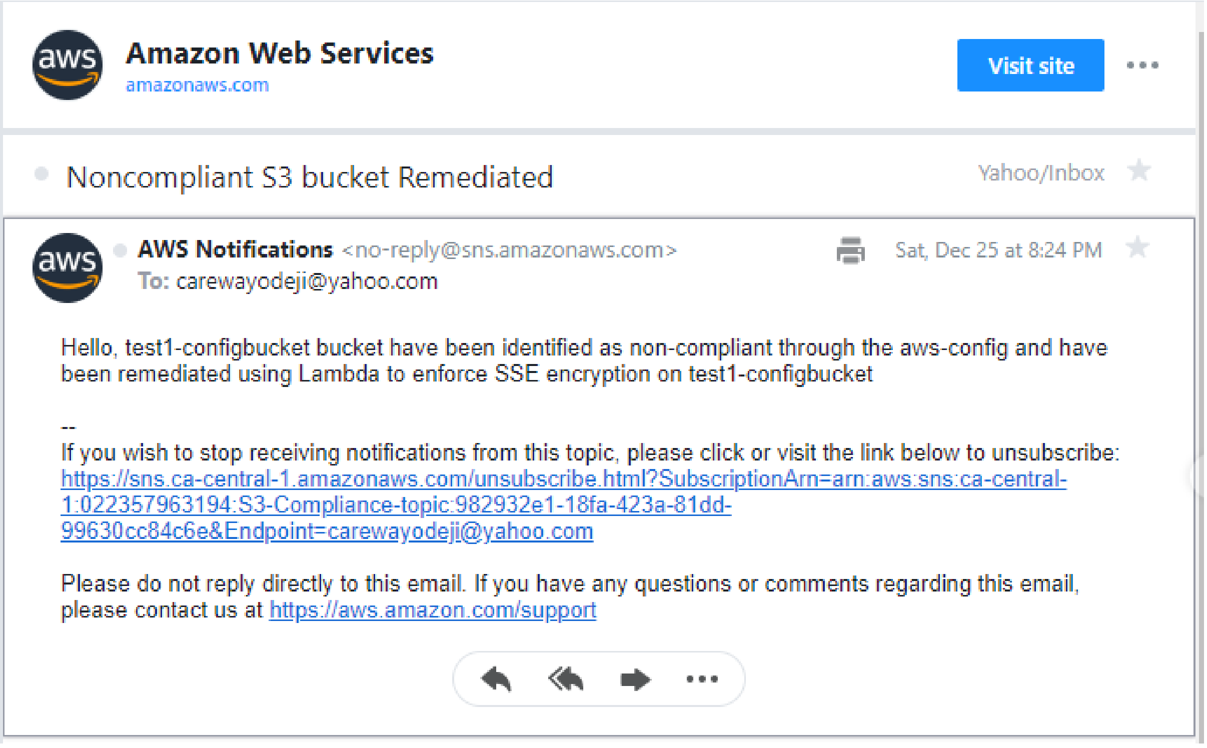
- The python code printed the number of noncompliant S3 buckets as detected by AWS Config and actioned them all to be compliant.



	ID	Type	Status	Annotation	Compliance
○	owiwi	S3 Bucket	-	-	✓ Compliant
○	project1-022357963194	S3 Bucket	-	-	✓ Compliant
○	test1-configbucket	S3 Bucket	-	-	✓ Compliant
○	test2-configbucket	S3 Bucket	-	-	✓ Compliant
○	test3-configbucket	S3 Bucket	-	-	✓ Compliant
○	test4-configbucket	S3 Bucket	-	-	✓ Compliant
○	test5-configbucket	S3 Bucket	-	-	✓ Compliant

Email Notifications

- The designated person/group will receive an email of each s3 bucket detected as noncompliant by AWS Config and remediated by Lambda.
- This can be customized as needed.



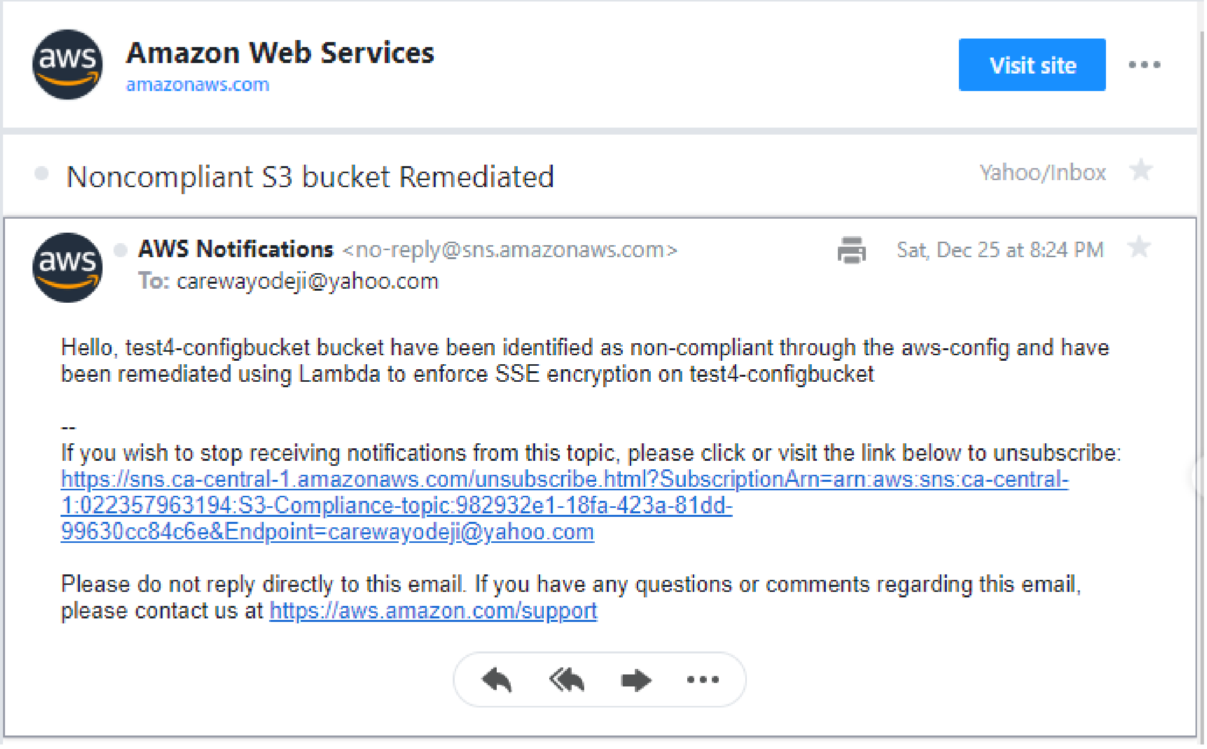
Noncompliant S3 bucket Remediated

AWS Notifications <no-reply@sns.amazonaws.com> Sat, Dec 25 at 8:24 PM
To: carewayodeji@yahoo.com

Hello, test1-configbucket bucket have been identified as non-compliant through the aws-config and have been remediated using Lambda to enforce SSE encryption on test1-configbucket

--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.ca-central-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:ca-central-1:022357963194:S3-Compliance-topic:982932e1-18fa-423a-81dd-99630cc84c6e&Endpoint=carewayodeji@yahoo.com

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



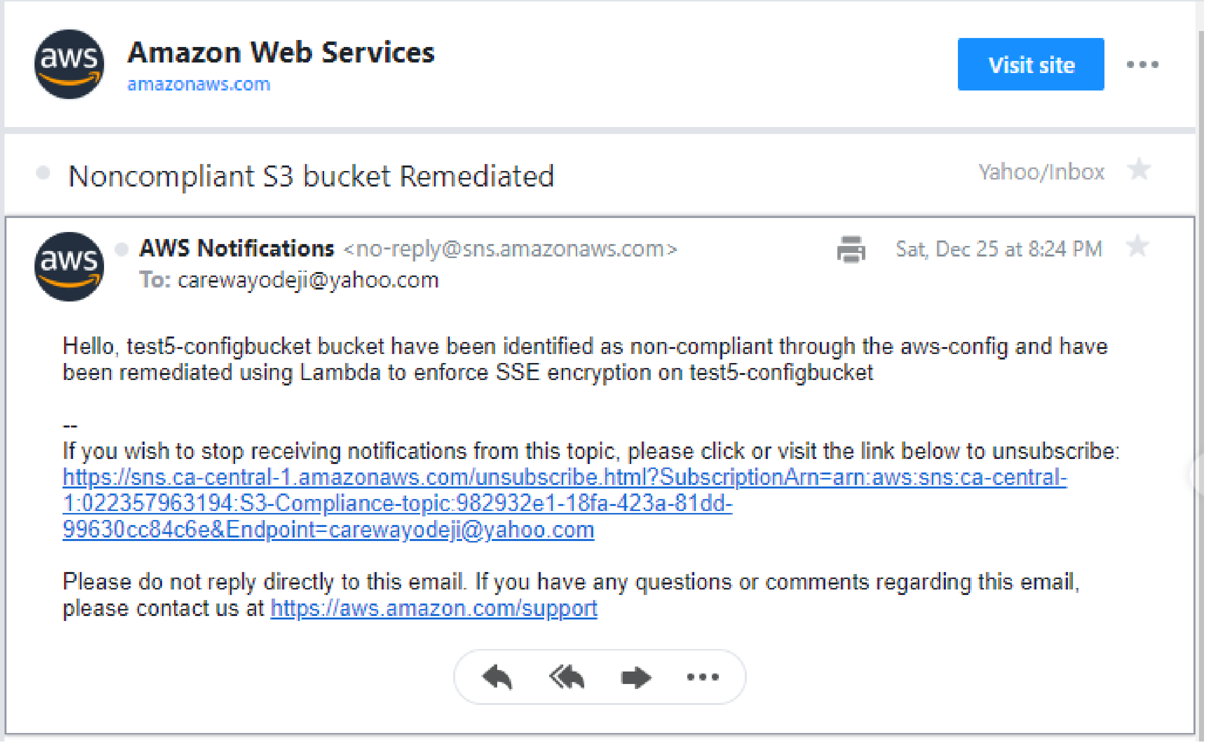
Noncompliant S3 bucket Remediated

AWS Notifications <no-reply@sns.amazonaws.com> Sat, Dec 25 at 8:24 PM
To: carewayodeji@yahoo.com

Hello, test4-configbucket bucket have been identified as non-compliant through the aws-config and have been remediated using Lambda to enforce SSE encryption on test4-configbucket

--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.ca-central-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:ca-central-1:022357963194:S3-Compliance-topic:982932e1-18fa-423a-81dd-99630cc84c6e&Endpoint=carewayodeji@yahoo.com

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



Noncompliant S3 bucket Remediated

AWS Notifications <no-reply@sns.amazonaws.com>

To: carewayodeji@yahoo.com

Sat, Dec 25 at 8:24 PM

Hello, test5-configbucket bucket have been identified as non-compliant through the aws-config and have been remediated using Lambda to enforce SSE encryption on test5-configbucket

--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.ca-central-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:ca-central-1:022357963194:S3-Compliance-topic:982932e1-18fa-423a-81dd-99630cc84c6e&Endpoint=carewayodeji@yahoo.com

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