

# Configuration Manual

MSc Research Project Cloud Computing

# Anuja Mahendrasingh Solanki Student ID: x22139524

School of Computing National College of Ireland

Supervisor: Prof. Sean Heeney

#### National College of Ireland Project Submission Sheet School of Computing



Student Name:	Anuja Mahendrasingh Solanki	
Student ID:	x22139524	
Programme: Cloud Computing		
Year:	2024	
Module:	dule: MSc Research Project	
Supervisor:	Sean Heeney	
Submission Due Date: 25th April 2024		
Project Title:	Configuration Manual	
Word Count:	979	
Page Count:	8	

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

<u>ALL</u> internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Anuja Mahendrasingh Solanki
Date:	25th April 2024

#### PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).Attach a Moodle submission receipt of the online project submission, to<br/>each project (including multiple copies).You must ensure that you retain a HARD COPY of the project, both for

your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

## Configuration Manual

# Anuja Mahendrasingh Solanki x22139524

#### 1 IAM Role Setup

Log in to your AWS Account and perform the below:

Search for the IAM management console. Set up a separate IAM role with only the rights needed for a certain service or task.

- Click on 'Roles' in the sidebar as shown below and click on the 'Create Role' button to create a new IAM role named "**Drift-role-27s3qi0o**". Select 'Custom role policy' for the role type and continue. Amazon Web Services (2024b)
- Policies are JSON documents in AWS that let you specify who has access to AWS resources and what actions they can perform on those resources.

Identity and Access	14	<u>M &gt; R</u>	oles >	Drift-role-27s3qi0o								
Management (IAM)	C	Drift-	rol	e-27s3qi0o տ								Delete
Q, Search IAM		Sumn	nary									Edit
Dashboard		Creatio	n date						ARN			
Access management		April 05	5, 2024	I, 00:00 (UTC+01:00)					am:aws:iam:471112828084:role/service	-role/Drift	t-role-27s3qi0o	
Liter century		Lastact	livity						Maximum session duration			
lkers		Ø 23 m	ninutes	s ago					1 hour			
Roles												
Policies												
Identity providers		Permis	sions	Trust relationships Tags	Access Advisor Revok	e ses	sions					
Arrount settings				-								
		Permi	issior	ns policies (5) into						C	Simulate 🛃 Rem	Add permissions v
<ul> <li>Access reports</li> </ul>		You can	attach u	p to 10 managed policies.								
Access Analyzer								Filter by 1	Type			
External access		0.9	unch					All type		1		(1) 0
Unused access		-										/ •
Analyzer settings			Polic	y name 🛃		a	Type		7	Attache	d entities	Ψ.
Credential report			-	·			AND second link former					
Organization activity		Π.	(±	Administratoraccess			www.imanaged - job func	bon		2		
Service control policies			÷	AmazonEC2FullAccess			AWS managed			2		
			ŧ	AWSLambdaBasicExecutionRole-c57	7dee-b91c-4209-97a6-69a07		Customer managed			1		
Related consoles			÷	CloudFormationDriftDetectionPolicy			Customer managed			1		
IAM Identity Center 2			Đ	DriftDetectionPolicy_SNS_CF			Customer managed			1		
····· · · · · · · · · · · · · · · · ·												

Figure 1: IAM Roles

- In the policy section, attach the already-available AWS-managed policies:
  - 1. AWS AdministorAccess
  - 2. AmazonEC2FullAccess
  - 3. AWS LambdaBasicExecutionrole
- As the account is logged in as the root user, any resource can be created in the cloudformation, and while detecting the drift, the Lambda function requires permission to use a variety of resources. Provisioning a resource without the permissions granted to Lambda will prevent the Lambda function from detecting the drift. In this case, AdministratorAccess is helpful. Grant the administrator restricted access to see if the cloudformation is unable to access permission for any resource; it can be identified. And then we can give access to that particular resource.

IAM > Policies > DriftDetectionPolicy_SN:           DriftDetectionPolicy_SN:	L_CF IS_CF Info		Edit Delete
Policy details			
Type Customer managed	Creation time April 03, 2024, 23:11 (UTC+01:00)	Edited time April 07, 2024, 00:38 (UTC+01:00)	ARN  am:aws:lam:471112828084:policy/DriftDetectionPolicy _SNS_CF
Permissions Entities attached 1	lags Policy versions (3) Access Advisor		
Permissions defined in this policy Permissions defined in this policy document specify	Info which actions are allowed or denied. To define permissions for an IAM identity (i	user, user group, or role), attach a policy to it	Edit Summary JSON
Q. Search Allow (2 of 409 services)			Show remaining 407 services
Service Acce	ss level v Resource	Request condition	
<u>CloudFormation</u> Limit	ed: Read All resources	None	
SNS Limit	ed: Write All resources	None	

Figure 2: DriftDetectionPolicy\_SNS\_CF

- Addition to these AWS-managed policies, attach the customer-based policies:
  - $1. \ CloudFormationDriftDetectionPolicy\\$
  - 2. DriftDetectionPolicy\_SNS\_CF
- Cloudformation and SNS are the two services attached to these policies, defining their permissions. When modifications are made to an IAM customer-managed policy, the new policy takes first place over the old one. IAM makes a new version for the controlled policy instead.
- DescribeStackDriftDetectionStatus, DetectStackDrift, and DetectStackResourceDrift and publish are defined in a version below:

```
"Statement": [
    {
        "Sid": "VisualEditor0",
        "Effect": "Allow",
        "Action": [
            "cloudformation:DetectStackResourceDrift",
            "cloudformation:DetectStackDriftDetectionStatus",
            "cloudformation:DetectStackDrift",
            "sns:Publish"
        ],
        "Resource": "*"
    }
]
```

• The IAM role, along with these policies, will be granted the necessary authorizations for accessing AWS services such as EC2, Lambda, and CloudFormation to help with remediation and drift detection workflows.

#### 2 Creation of Stacks using Cloudformation

Now Go to the AWS Console and search for "Cloudformation". For provisioning and setting up resources.

	Prerequisite - Prepare template
ity stack details	Prenare template
3 igure stack options	Every stack is based on a template. A template is a JSON or VAML file that contains configuration information about the AWS resources you want to include in the stack
4	Choose an existing template     Upload or choose an existing template.     O Use a sample template library.     Crosse template using a visual builder.
w and create	
	Amazon S3 URL     Novide an Amazon S3 URL to your template.     Sync a template from your template.     Sync a template from your OC repository.
	Upload a template file
	JSUN OF TAME FORMATION

Figure 3: Create Stacks

- Click on "Create Stack" and you will be drawn to the below page:
- Select the requirements as shown in the Figure 3. Click on Upload a template file to upload the ".YAML" file from your local system that will be used to define the stack.
- Provide a stack name as "EC2instance" and click next.
- Review the stack details and click on Submit. Now Ec2Instance Stack is created as shown in Figure 4
- Similarly, upload the '.YAML' files for S3Bucket, and DynamoDB stacks and create them as shown in Figure 5
- Note the Stack ID(ARNs) of all three stacks. In this case for EC2 instance it is arn:aws:cloudformation:ap-south-1:471112828084:stack/EC2instance/74c 0e490-f209-11ee-a0af-0aac48631e59

### 3 Configure SNS

Create an SNS Topic named "Detect\_Drift" to send notifications whenever the drift is detected in the stacks. Amazon Web Services (2024a)

- While creating the subscription, select the protocol as EMAIL and provide an email ID as the endpoint (where the notifications will appear); here it is given as "x22139524@student.ncirl.ie"
- Copy the ARN Id: arn:aws:sns:ap-south-1:471112828084:Detect\_Drift

Overview	C
Copied     D amaws:cloudformation:ap-south- 1:471112828084:stack/EC2instance/74c0e490-f209-11ee-a0af- 0aac48631e59	Description -
Status	Detailed status
O CREATE_COMPLETE	-
Status reason	Root stack
-	-
Parent stack -	Created time 2024-04-03 23:28:12 UTC+0100 Updated time -
Deleted time	Drift status
-	⊘ IN_SYNC
Last drift check time	Termination protection
2024-04-23 19:38:56 UTC+0100	Deactivated

Figure 4: EC2 Stack Details

Stac	<b>ks</b> (4)			C Dele	te Upo	late	Stack actions 🔻	Ci	eate sta	ick	•
QF	Filter by stack name			Complete	•	0	/iew nested		< 1	>	0
	Stack name	Stack ID	Status		Created time						
0	Dynamo2	am:aws:cloudformation:ap- south- 1:471112828084:stack/Dynamo 2/6fd12ba0-fd0f-11ee-89bc- 0606155f89a7	⊘ CREATE_COMPLETE		2024-04-18	00:08:43	UTC+0100				
0	DynamoD8	am:aws:cloudformation:ap- south- 1:471112828084:stack/Dynamo DB/eac28c70-f209-11ee-8b09- 0a02e01a9aa3	⊘ CREATE_COMPLETE		2024-04-03	23:31:30	UTC+0100				
0	EC2Instance	am:aws:cloudformation:ap- south- 1:471112828084:stack/EC2insta nce/74c0e490-f209-11ee-a0af- 0aac48631e59	⊘ CREATE_COMPLETE		2024-04-03	23:28:12	UTC+0100				
0	<u>S3Bucket</u>	am:aws:cloudformation:ap- south- 1:471112828084:stack/S3Bucke t/edefa5f0-f208-11ee-bd3f- 06859283fa06	⊘ CREATE_COMPLETE		2024-04-03	23:24:26	UTC+0100				

Figure 5: Created Stacks

### 4 Creation of Python script in Lambda

To create a Python function to write a program for Automating Drift Detection in the stacks, follow the steps below:

- In the Lambda Console, create a Lambda function with the name **Drift**". Use the Runtime as Python. and click on the create function.
- Once the function is created, write your Python code for detecting drift in the three stacks mentioned in Section 2.
- Make sure you have the below installed before running your Python script for detecting the drift.
  - 1. Python
  - 2. Boto3 library

6	New Feature Amazon SNS now supports in-place message archiving and replay for FIFO	topics. Learn more 🖸			×
	Amazon SNS > Topics > Detect_Drift				
	Detect_Drift			Edit Delete	Publish message
	Details				
	Name Detect_Drift	Display name noti			
	ARN am:aws:sns:ap-south-1:471112828084:Detect_Drift	Topic owner 471112828084			
	Type Standard				
	Subscriptions Access policy Data protection policy Deli	very policy (HTTP/S)	Delivery status logging Encryption Ta	igs Integrations	
	Subscriptions (1)		Edit Delete Request	confirmation Confirm subscription	Create subscription
	Q, Search		]		< 1 > ©
	ID		▼ Status	♥ Protocol	▽
	• <u>c86983e1-6d92-4472-8040-d342e2356d1e</u> x221395	4@student.ncirl.ie	⊘ Confirmed	EMAIL	

Figure 6: Created Subscription

ambda > Functions > Create function       Create function		
hoose one of the following options to create your function.		
Author from scratch     Start with a simple Helio World example.	Use a blumprint Build a Lambda application from sample code and configuration presets for common use cases.	Container image Select a container image to deploy for your function.
Basic information		
Function name Enter a name that describes the purpose of your function.		
myFunctionName		
Use only letters, numbers, hyphens, or underscores with no spaces.		
Runtime Info Choose the language to use to write your function. Note that the console code editor supports only	Nodejs, Python, and Ruby.	
Python 3.12		• C
Architecture Info Choose the instruction set architecture you want for your function code.		
• x86_64		
🔾 arm64		
Permissions into Re default: Lambda will create an execution role with permissions to unload loos to Amazon Cloud	Natch Loos, You can customize this default role later when adding triggers	
Change default execution role		

Figure 7: Creation of Lambda Function

- If boto3 is not installed, use this command to install it. "**pip install boto3**". Amazon Web Services (2023a)
- Copy and paste the ARN IDs for the three stacks and the SNS subscription from sections 2 and 3 in the code to identify the resources. as shown in Figure 8



Figure 8: Stack ARN and SNS topic ARN

## 5 EventBridge Schedule for Triggering Lambda Function

- Create an Eventbridge schedular named "**Driftrule**" to trigger the Lambda function "**Drift**" and run the drift detection Python program with the below specifications.
- A cron expression generates a time-sensitive recurring schedule with precise configuration. Set the corn expression for the schedule as **0 0**,**8**,**16** \* ? \* Figure 9. Amazon Web Services (2023b) It means that the lambda function will trigger every 8 hours to check the drift of the resources.



Figure 9: Cron Expression

- Click on Next and search for AWS Lambda from the list of APIs provided. Select the target API as an AWS Lambda from the given options. Now scroll down and write the payload specifications with the below script. Figure 10
- The JSON Script:

```
{
    "comment": "Scheduled drift detection trigger",
    "stackName": "DynamoDB, EC2instance, S3Bucket"
}
```

Step 2: Target	Edit
Target detail	
Target AWS Lambda Drift [2] Payload { "comment": "Scheduled drift detection trigger	Target ARN  arm:aws:lambda:ap-south- 1:471112828084:function:Drift  ", "stackName": "DynamoDB, EC2instance, S3Bucket" }

Figure 10: Set Target

• Click on Next and select the permission: Select as an existing role and choose "Amazon\_EventBridge\_Scheduler\_LAMBDA\_84821d0ca4" Figure 11

tep 3: Settings	Edit
Schedule state and permissions	
Schedule state	Execution role
Enabled	Amazon_EventBridge_Scheduler_LAMBDA_84821d0ca4
NONE	

Figure 11: Setting up the permissions

### 6 Performing the Drift Detection

Once the above configurations are performed, drift detection will start as per the schedule defined in the AWS Eventbridge. It will then trigger the AWS Lambda template, where the drift detection script will start running.

• The first step is initializing Cloudformation and SNS in the Python script.



Figure 12: Initilization of Cloudformation and SNS

- Provide the stack ARN and SNS ARN for locating the resources. Check Figure 8
- After the stacks are defined, the program will trigger the drift detection for all the stacks, one after the other for drift.



Figure 13: Triggering Drift Detection

- Once the drift detection is completed, it extracts the details from the stack and sends the status to the "endpoint".
- The below response will be received after a successful run of the Python script. Figure 15



Figure 14: Drift Detection SNS Notification

■ Iambda_function × Environment Vari × Execution result: × ⊕	
▼ Execution results	Status: Succeeded Max memory used: 80 MB Time: 95576.46 ms
Test Event Name	
(unsaved) test event	
Response	
<pre>{     "statusCode": 200,     "body": "\"Drift detection and notification sent for all stacks.\"" }</pre>	
Function Logs	
START RequestId: 4368378-bfc-485-a37c-382-1672366c Version: SLIFST [INF0] 2024-04-247811-049:10.322 4196575-bfc-485-a37c-3827672966 Trup printing error for response ('Stackld': armassicloudformationse-south-14711 [INF0] 2024-04-247811-049:10.332 4196575-bfc-485-a37c-3827672966 Trip [INF0] 2024-04-247811-049:10.4332 4196575-bfc-485-a37c-3827672984c Drif printing error for response ('Stackld': armassicloudformationse-south-14711 [INF0] 2024-04-247811-049:10.4332 4196575-bfc-485-a37c-3827672984c Drif printing error for response ('Stackld': armassicloudformationse-south-14711 [INF0] 2024-04-247811-99:10.4052 4196575-bfc-485-a37c-3827672984c Drif printing error for response ('Stackld': armassicloudformationse-south-14711 [INF0] 2024-04-247811-99:10.4052 4196575-bfc-485-a37c-3827672984c Drif printing error for response ('Stackld': armassicloudformationse-south-14711 printing error for response ('Stackld': armassicloudformationse-south-14711 [INF0] 2024-04-247819:59:10-5522 41965375-bfc-458-a37c-3827637986c Drif Brift 4365378-bfc-458-a37c-3827617298cc Drift [INF0] 2024-04-247819:59:10-5524 51962 41965375-bfc-458-a37c-3827637986c Drift [INF0] 2024-04-247819:59:10-5524 51962 41965375-bfc-458-a37c-382763798cc Drift [INF0] 2024-04-247819:59:10-5552 41965375-bfc-458-a37c-382763798cc Drift [INF0] 2024-04-247819:59:10-5552 41965375-bfc-458-a37c-382763798cc Drift [INF0] 2024-04-247819:59:10-5552 41965375-bfc-458-a37c-38276376376456767676 [INF0] 2024-04-247819:59:10-5552 41965375-bfc-458-a37c-3827637966 Drift [INF0] 2024-04-247819:59:10-5552 41965375-bfc-458-a37c-3827637637645637676767676767676767676767676767676767	d credentiais in environment veriables. gring drift detection for stack: annias:cloudformation:ap-south-1:471112828084:stack/DynamoDB/ee228/7 1/282864:stack/DynamoDfee228/278-7289-118e-8089-808260139a3', 'StackDriftDetection1d': 'd9b03328-01ac' t detection completed for stack: annias:cloudformations-p-south-1:471112828084:stack/CDynamoDB/ee228/7 gring drift detection for stack: annias:cloudformations-p-south-1:471112828084:stack/CEClinstance/7408 12828084:stack/EC2Instance/7408490-7809-11ee-808f-0asc48631e59', 'StackDriftDetection1d': 'deteff09-01 t detection completed for stack: annias:cloudformation:ap-south-1:471112828084:stack/EC2Instance/7408 12828084:stack/S2084:stack/
Portuget ID	
41968578-bfec-485a-a37c-83e716729a6c	

Figure 15: Response after successful run

#### References

Amazon Web Services (2023a). Boto3 Documentation: AWS CloudFormation.

Amazon Web Services (2023b). Schedule types in aws scheduler.

Amazon Web Services (2024a). Amazon sns developer guide.

Amazon Web Services (2024b). Iam roles - aws identity and access management.