Module 8: CloudFormation Assignment - 1

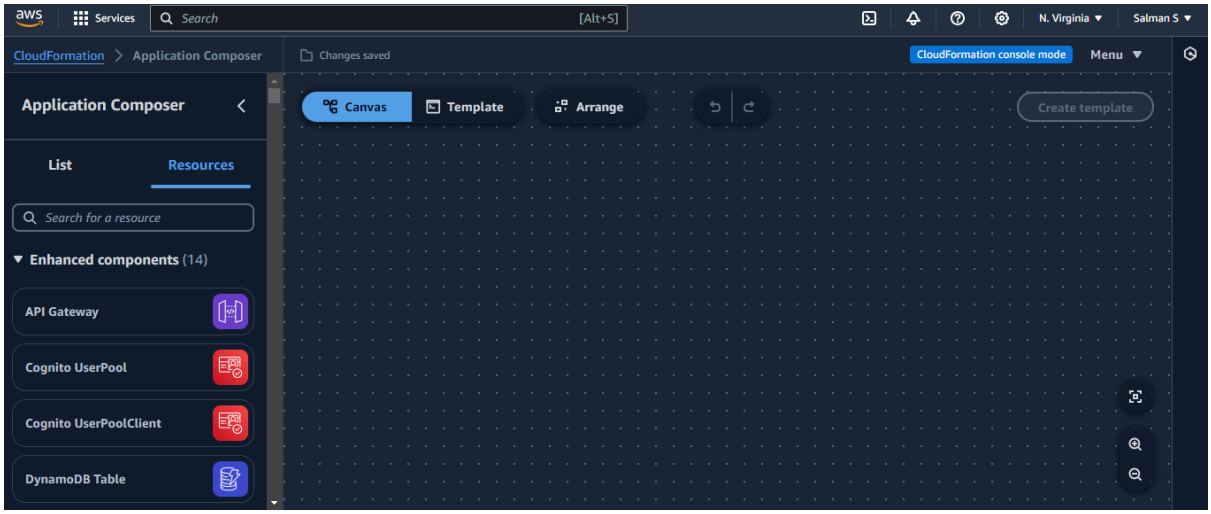
Problem Statement:

You work for XYZ Corporation. Your team is asked to deploy similar architecture multiple times for testing, development, and production purposes. Implement CloudFormation for the tasks assigned to you below.

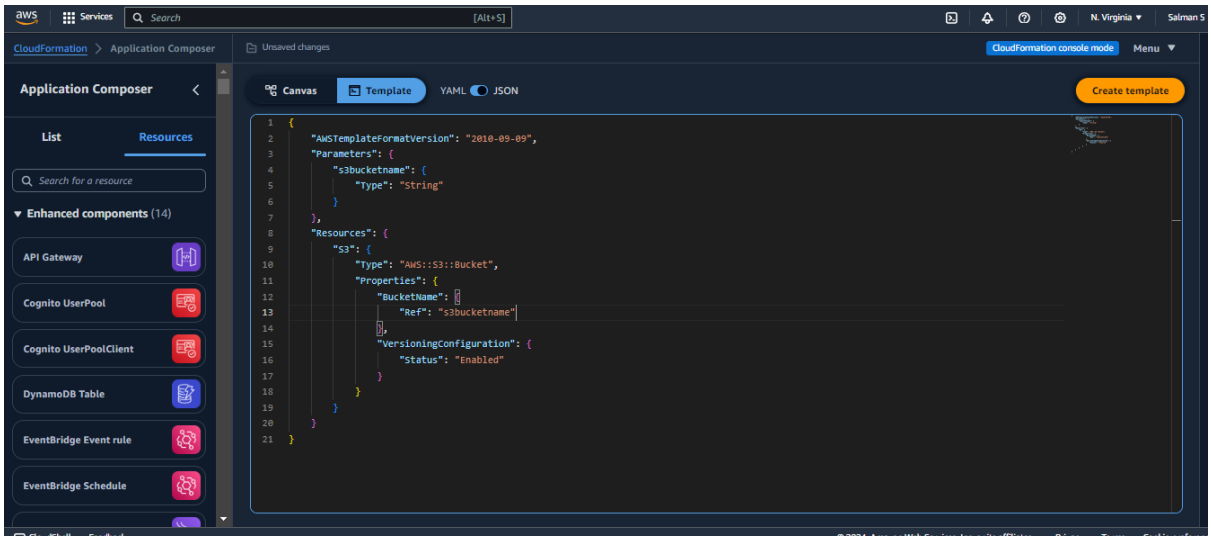
Tasks To Be Performed:

1. Create a template which can create an S3 bucket named "Intellipaat-<yourname >"
2. The template should be able to enable versioning for the bucket created.

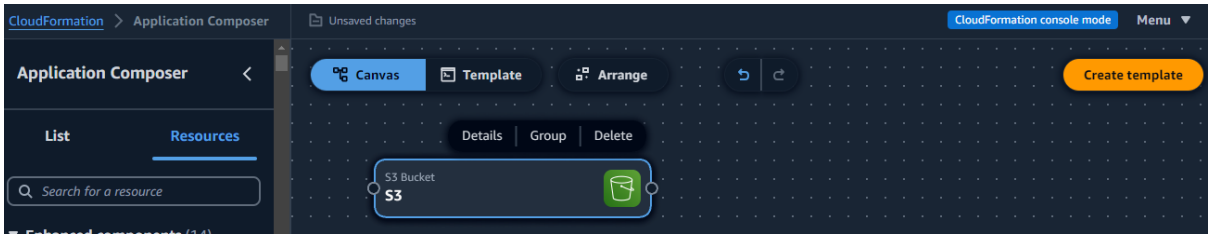
1. Go to Cloudformation Composer and write the code as per the assignment task and save



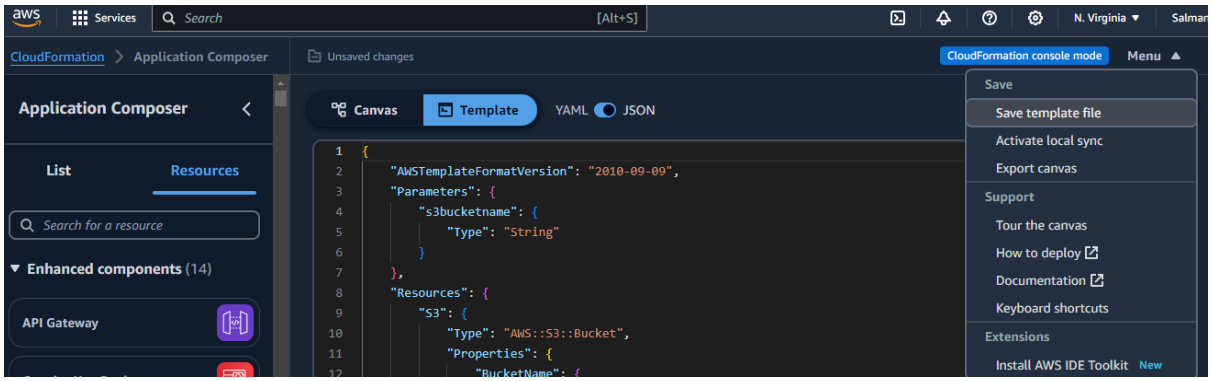
2. Creating a Parameter and Set type as string and Created the S3 Resources and Enabling Versioning in Cloudformation



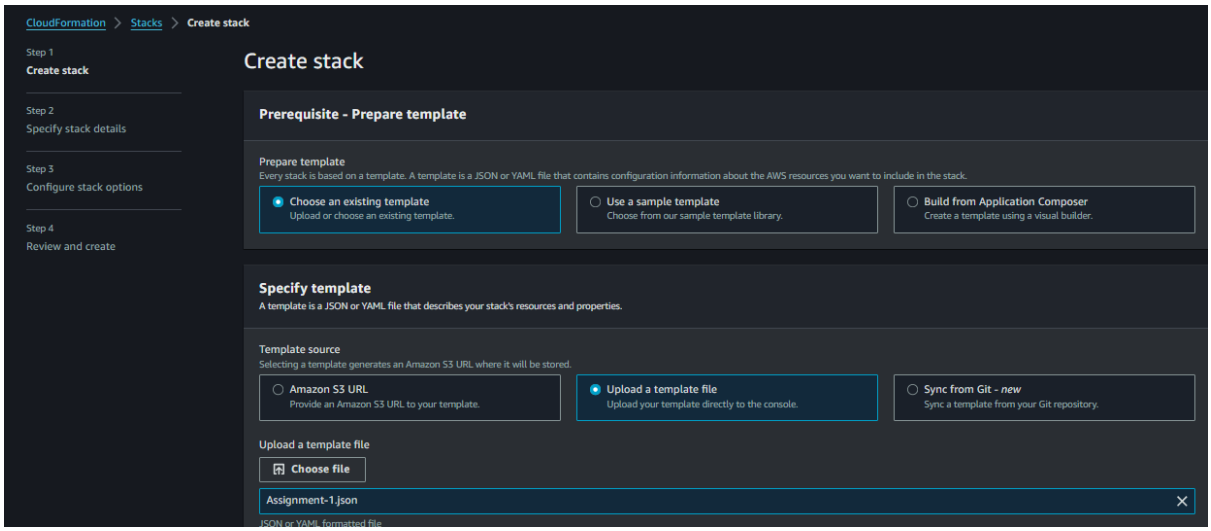
3. Canva vision



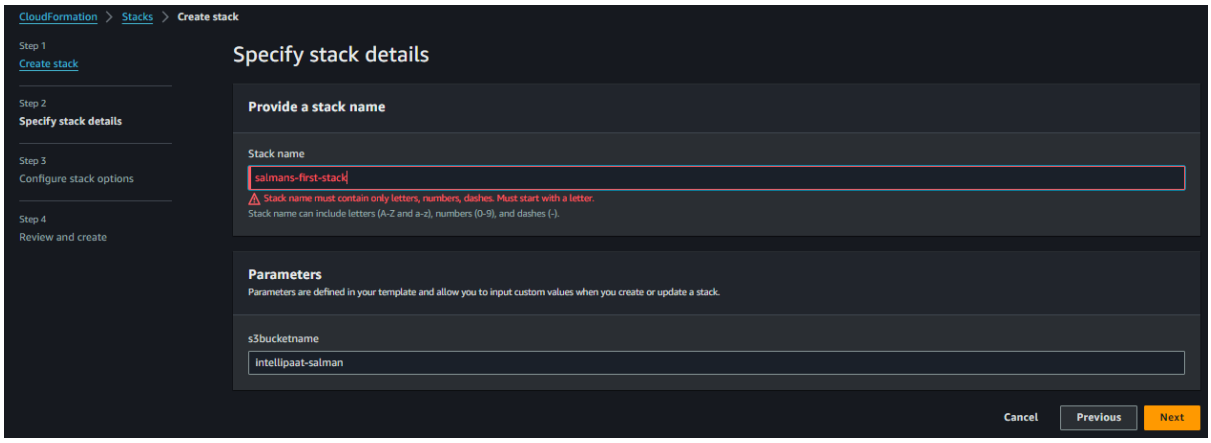
4. Go to Template and Save Template file



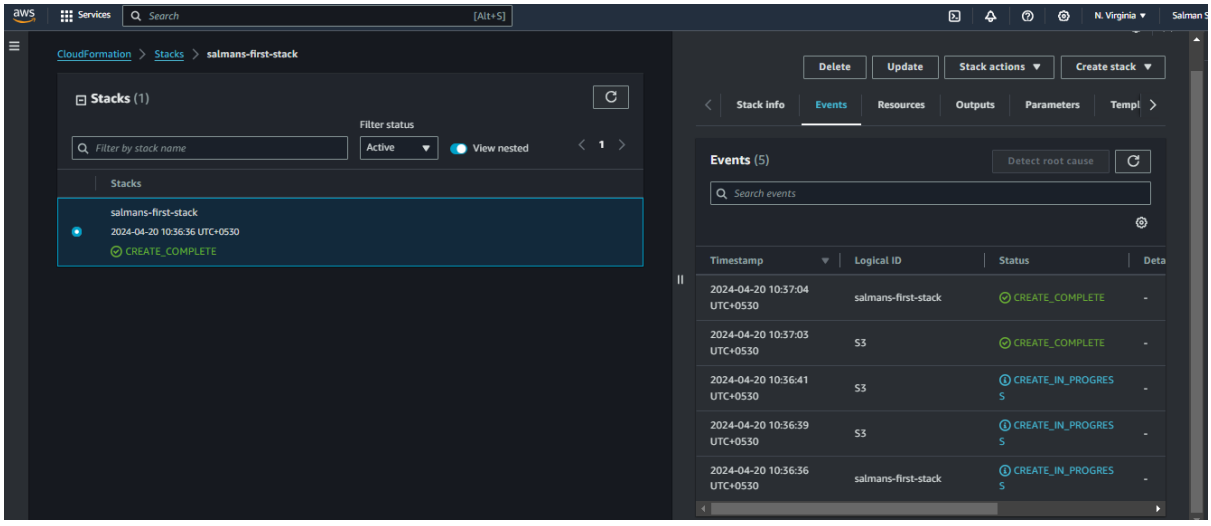
5. Go To Cloud Stack and Choose existing template and Upload a Template file which we Already Created



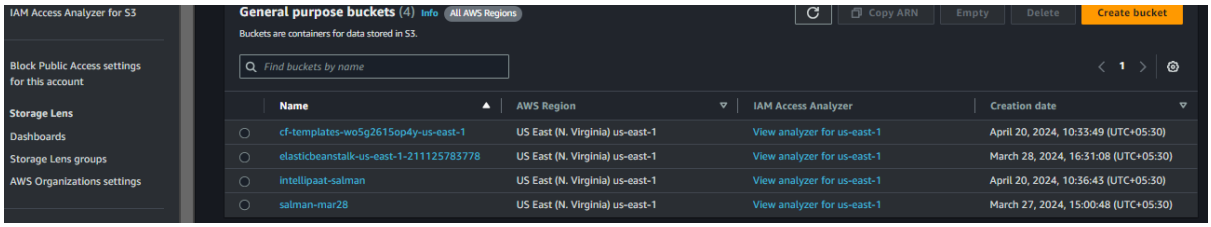
6. Don't give spaces and we set parameter as a string so we can give a bucketname here



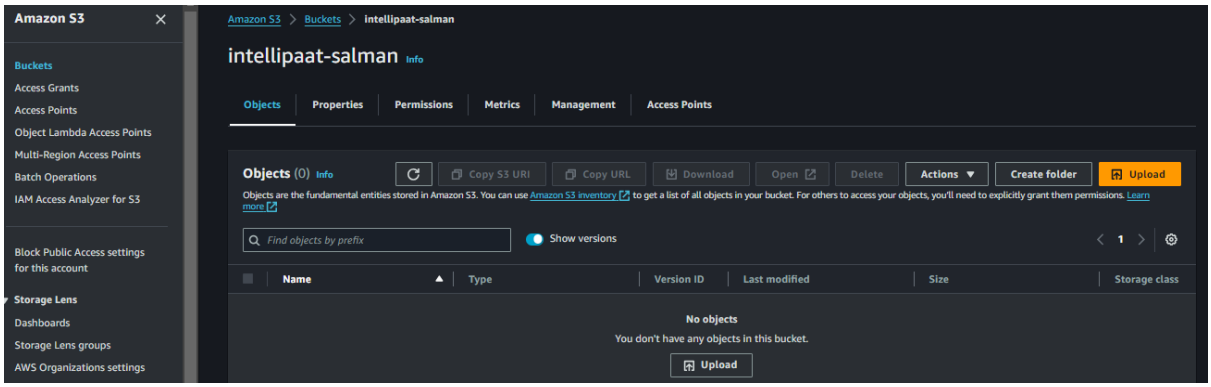
7. Now the Stacks Created Successfully



8. We can able to see the intellipaat-salman bucket



9. We also see the versioning and this is Assignment task which we completed



Module 8: CloudFormation Assignment - 2

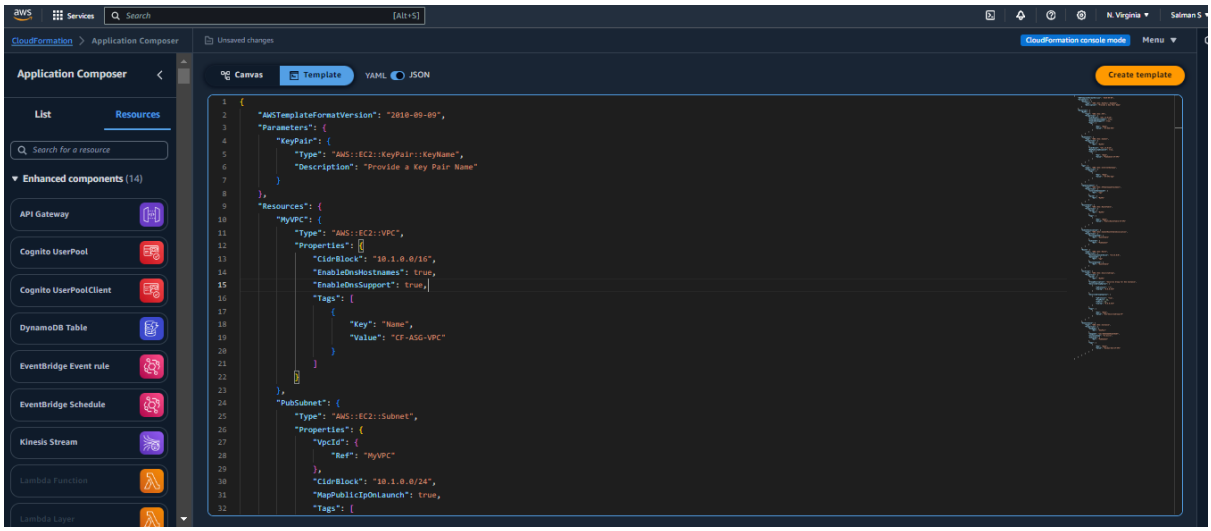
Problem Statement:

You work for XYZ Corporation. Your team is asked to deploy similar architecture multiple times for testing, development, and production purposes. Implement CloudFormation for the tasks assigned to you below.

Tasks To Be Performed:

1. Create a template with 1 VPC and 1 public subnet.
2. Launch an Amazon Linux EC2 instance in the public subnet and tag the instance as “CFinstance”

1. Develop the Code as per the Assignment in Cloudformation Created parameters,Resources



2. Code of 2nd Assignment and Creating a public sub and IGW, IGW Attachment to VPC Creating RT

3. Code for RT and Adding Tags and Value giving name and Creating Subnetasso through code in Cloudformation and Route

```
            "Type": "AWS::EC2::RouteTable",
            "Properties": {
                "VpcId": {
                    "Ref": "MyVPC"
                },
                "Tags": [
                    {
                        "Key": "Name",
                        "Value": "PublicRouteTable-CF-VPC"
                    }
                ]
            }
        },
        "SubnetAssociation": {
            "Type": "AWS::EC2::SubnetRouteTableAssociation",
            "Properties": {
                "RouteTableId": {
                    "Ref": "RouteTable"
                },
                "SubnetId": {
                    "Ref": "PubSubnet"
                }
            }
        },
        "Route": {
            "Type": "AWS::EC2::Route",
            "Properties": {
                "DestinationCidrBlock": "0.0.0.0/0",
                "GatewayId": {
                    "Ref": "Igw"
                },
                "RouteTableId": {
```

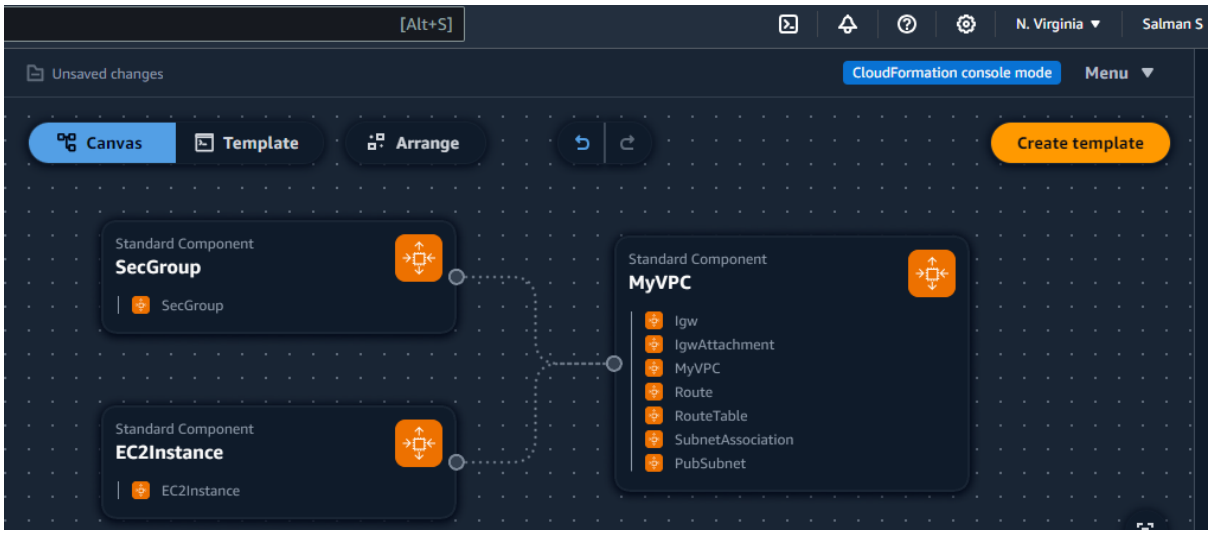
4. SecGroup, SGEgress and SGIngress

```
            "RouteTableId": {
                "Ref": "RouteTable"
            }
        }
    },
    "SecGroup": {
        "Type": "AWS::EC2::SecurityGroup",
        "Properties": {
            "VpcId": {
                "Ref": "MyVPC"
            },
            "GroupDescription": "Security Group for EC2 Instance",
            "SecurityGroupEgress": [
                {
                    "IpProtocol": "-1",
                    "CidrIp": "0.0.0.0/0"
                }
            ],
            "SecurityGroupIngress": [
                {
                    "IpProtocol": "tcp",
                    "FromPort": 22,
                    "ToPort": 22,
                    "CidrIp": "0.0.0.0/0"
                }
            ],
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "EC2-SecurityGroup-CF"
                }
            ]
```

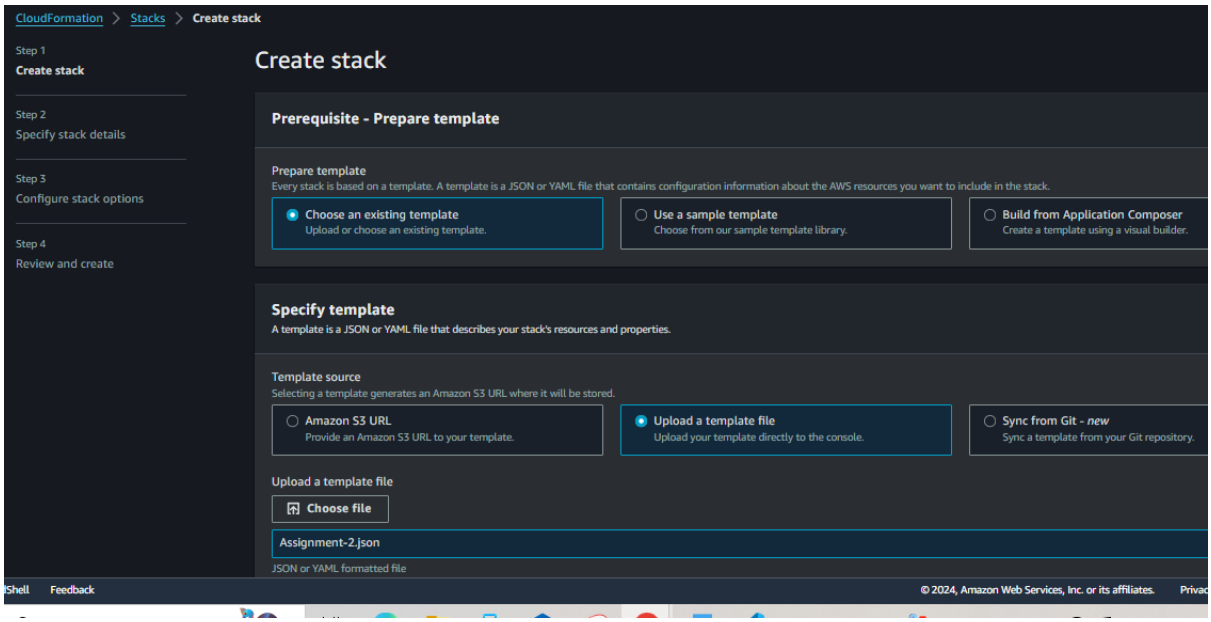
5. Finally Creating EC2 Instances and ref to pub subnet and Giving Tag I mean Name for EC2 Instances

```
            }
        ]
    }
},
"EC2Instance": {
    "Type": "AWS::EC2::Instance",
    "Properties": {
        "KeyName": {
            "Ref": "KeyPair"
        },
        "ImageId": "ami-080e1f13689e07408",
        "InstanceType": "t2.micro",
        "SubnetId": {
            "Ref": "PubSubnet"
        },
        "Tags": [
            {
                "Key": "Name",
                "Value": "Salman-EC2-CF-VPC"
            }
        ]
    }
}
}
```

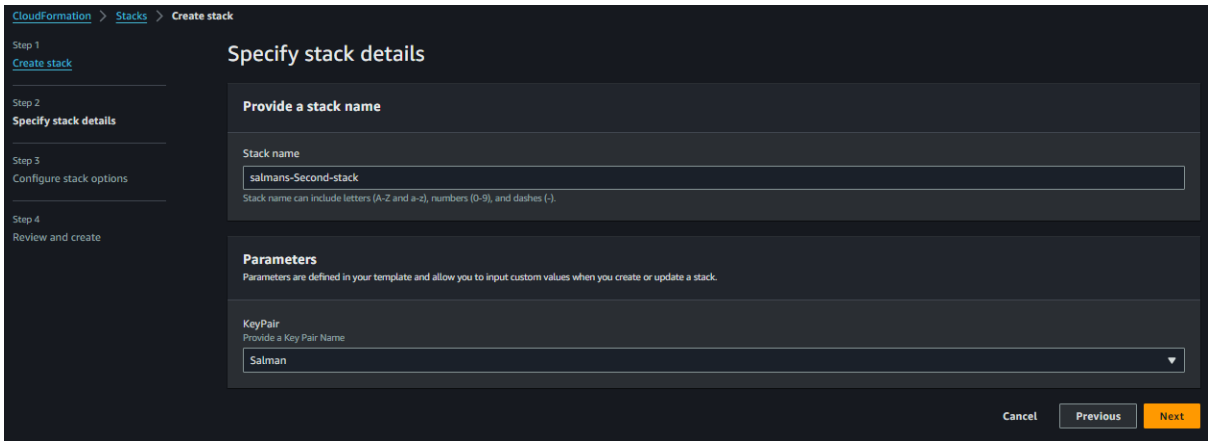
6. Canva Vision(HLD)



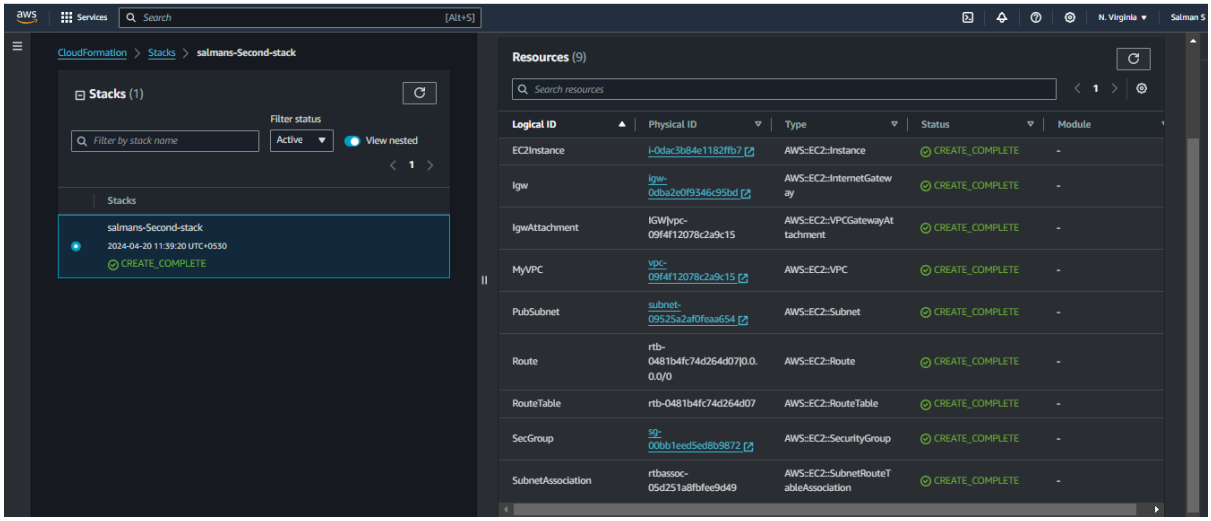
7. Creating Stack and existing template and Upload it



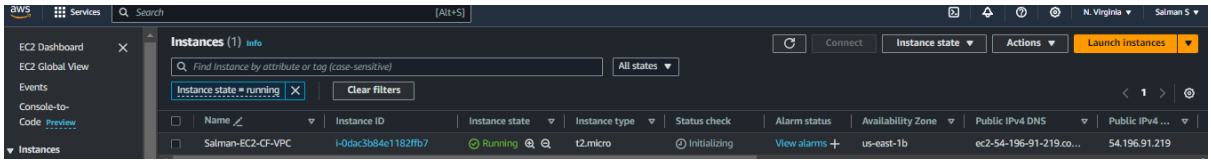
8. Stack Name and select Key Pair and Next



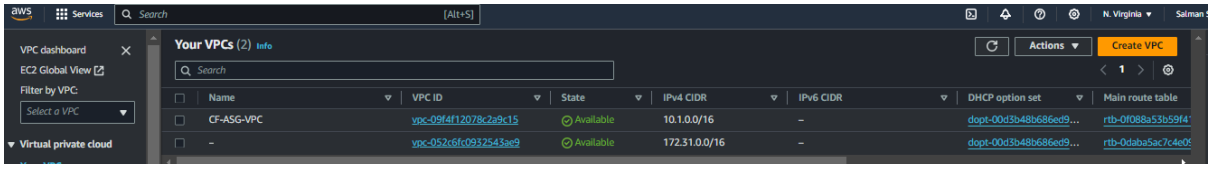
9. Stack Created Successfully EC2, IGW, IGWAttachment, MYVPC, PubSubnet, Route, RouteTable, SecGroup, SubnetAssociation



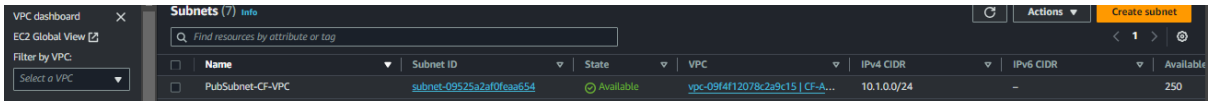
10. Successfully Created EC2 in Console



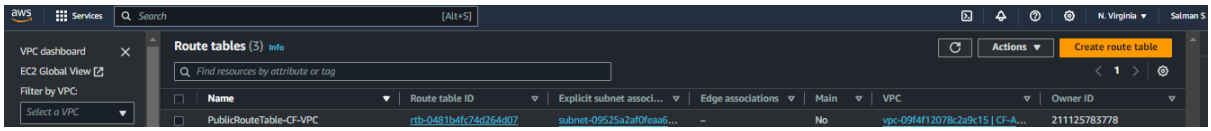
11. Successfully Created VPC in Console



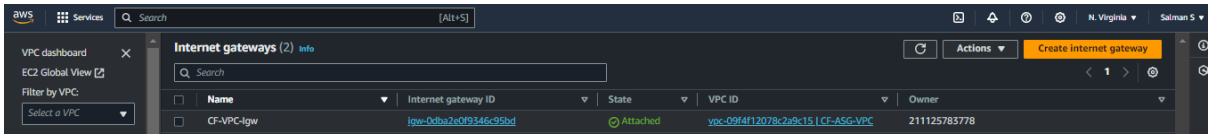
12. Successfully Created Public Subnet in Console



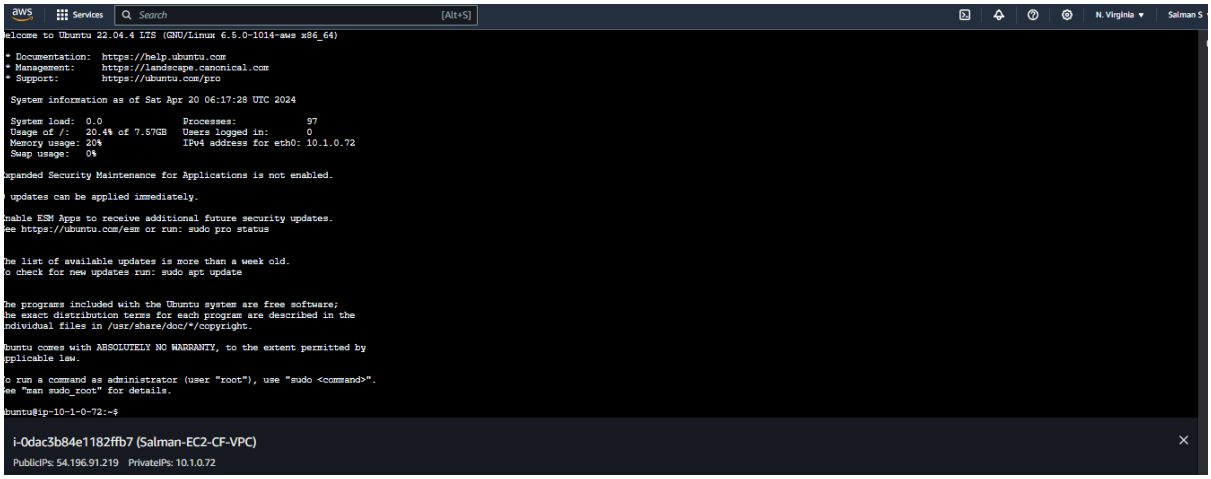
13. Successfully Created Public Route Table in Console



14. Successfully Created VPC-IGW in Console



15. Launch Instances Successfully and Once we delete the Stack All Resources will be deleted Automatically



Module 8: CloudFormation Assignment - 3

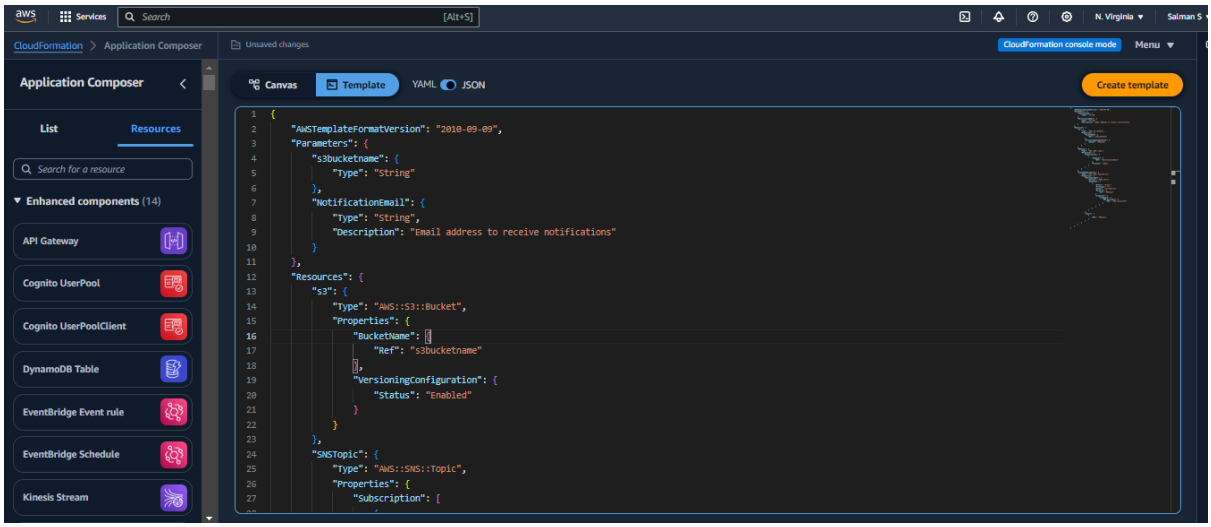
Problem Statement:

You work for XYZ Corporation. Your team is asked to deploy similar architecture multiple times for testing, development, and production purposes. Implement CloudFormation for the tasks assigned to you below.

Tasks To Be Performed:

1. Use the template from CloudFormation task 1.
2. Add Notification to the CloudFormation stack using SNS so that you get a notification via mail for every step of the stack creation process.

1. Using Cloudformation Task 1 and Template and Add Notification to the Cloudformation Stack using SNS you get notification via mail for every step of the stack creation process and Creating SNSTopic and Email Subscription



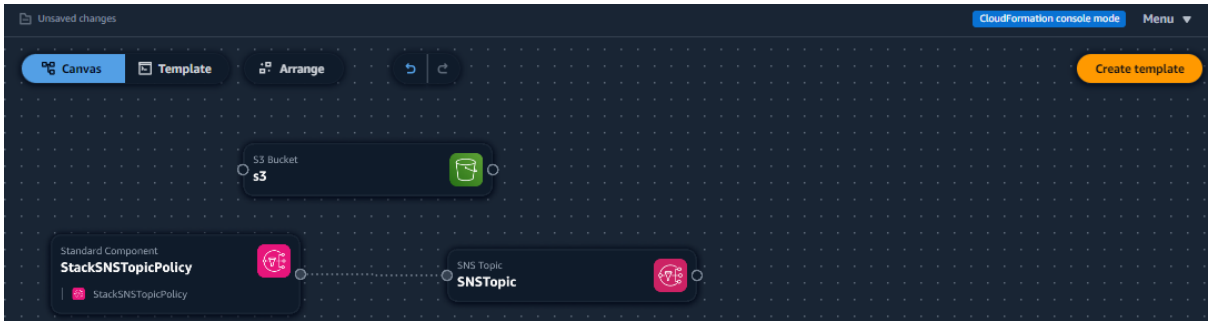
2. In the Subscription The Endpoint is to NotificationEmail and Protocol Email and Adding StackSNSTopicPolicy, Properties, PolicyDocument, Version, Statement, Effect and Principal, Action, Resource is Res SNSTopic and Condition and Ref AWS AccountID

```
            "Subscription": [
                {
                    "Endpoint": {
                        "Ref": "NotificationEmail"
                    },
                    "Protocol": "email"
                }
            ]
        }
    },
    "StackSNSTopicPolicy": {
        "Type": "AWS::SNS::TopicPolicy",
        "Properties": {
            "PolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": "*",
                        "Action": "sns:Publish",
                        "Resource": {
                            "Ref": "SNSTopic"
                        },
                        "Condition": {
                            "ArnEquals": {
                                "AWS:SourceArn": {
                                    "Ref": "AWS::AccountId"
```

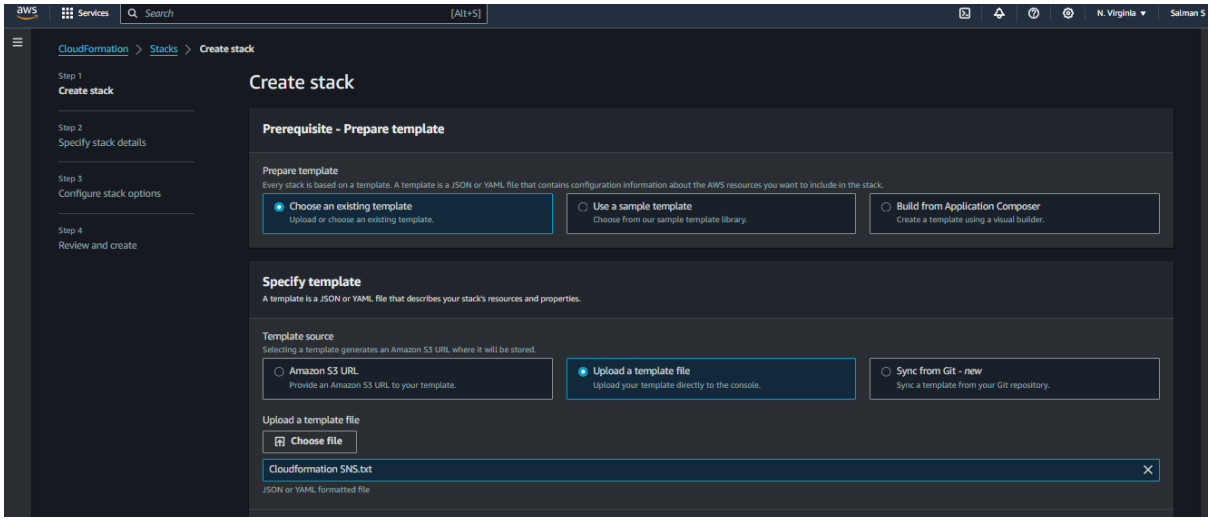
3. Topics and Ref to SNSTopic

```
                                "Ref": "
                            }
                        }
                    }
                }
            ]
        },
        "Topics": [
            {
                "Ref": "SNSTopic"
            }
        ]
    }
}
}
```

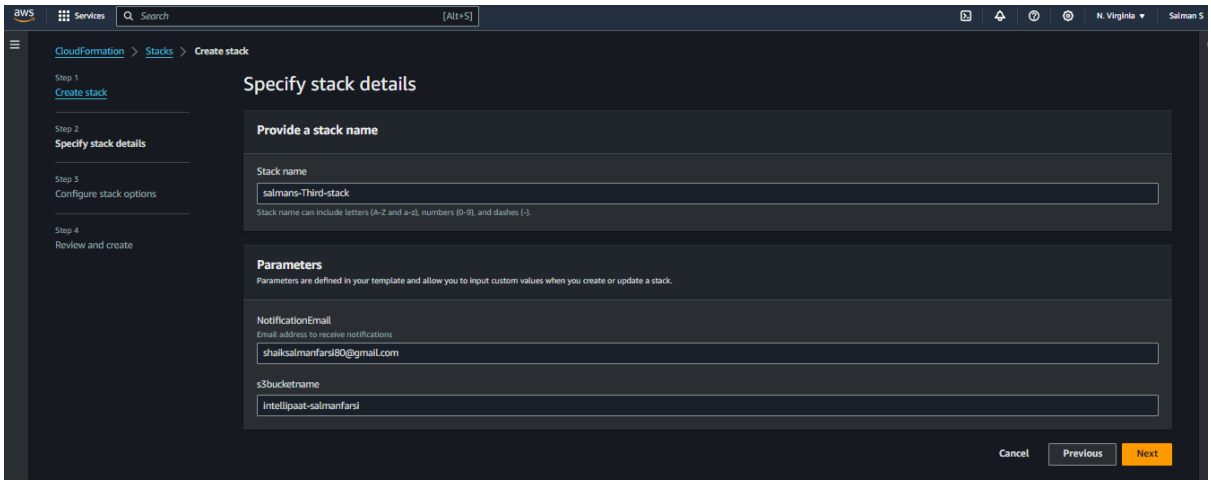
4. Canva Vision



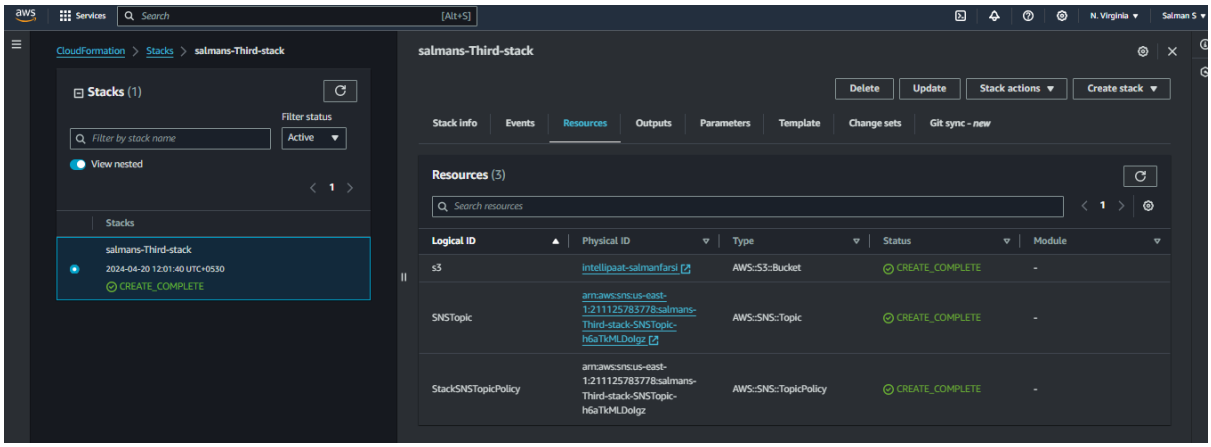
5. Create Stack and Existing one and Upload it



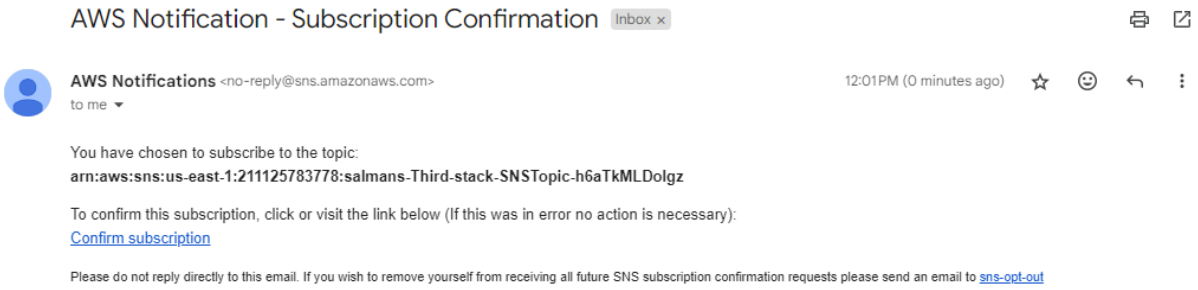
6. Giving Email and Bucket Name



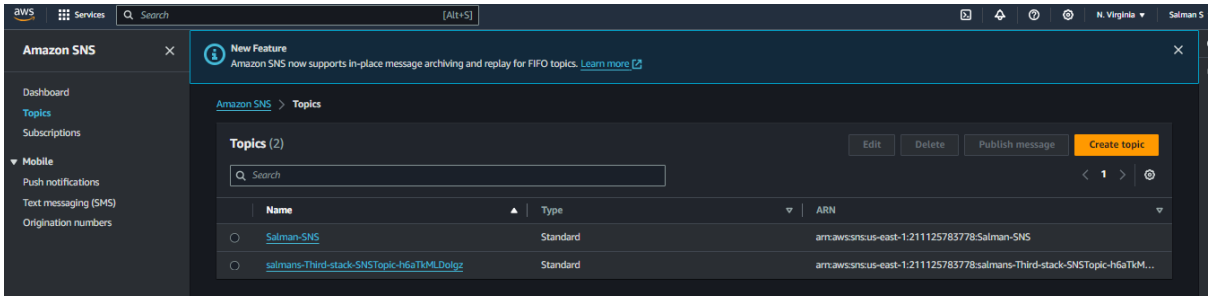
7. Stack Created Successfully and Resources too



8. I got a Subscription Mail From AWS Notification



9. Topics Created for Every Step of Stack Creation and Now the Assignment is Completed



Module 8: SQS and SES Assignment

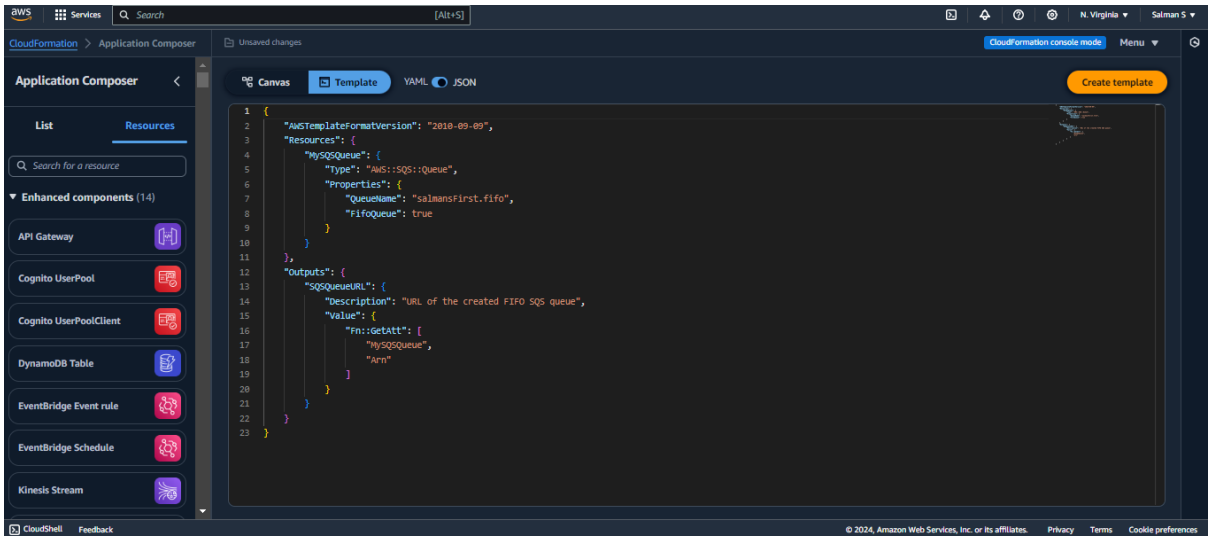
Problem Statement:

You work for XYZ Corporation. Your team is asked to deploy similar architecture multiple times for testing, development, and production purposes. Implement CloudFormation for the tasks assigned to you below.

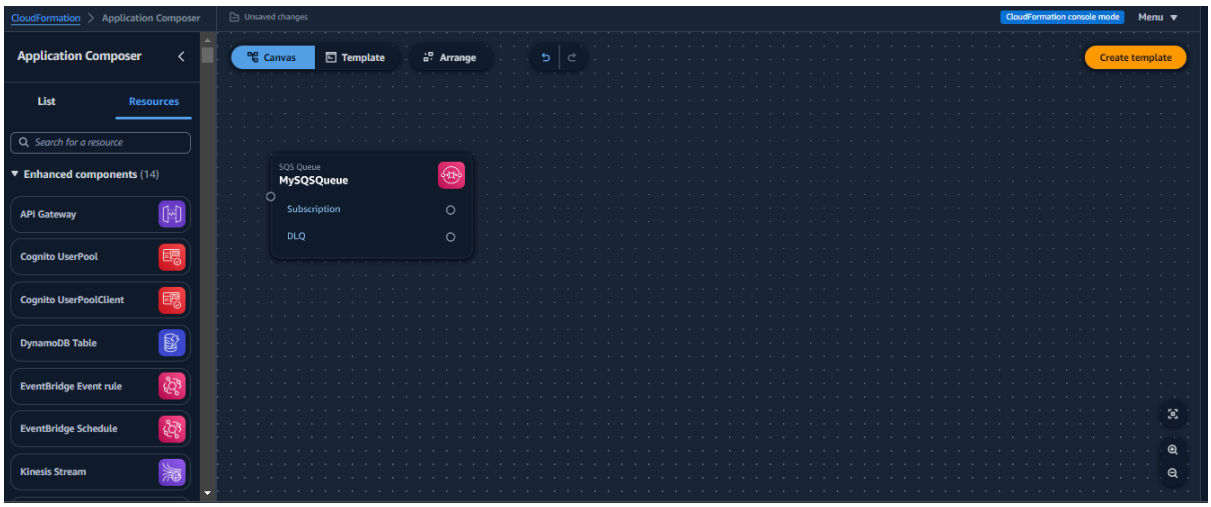
Tasks To Be Performed:

1. Create a FIFO SQS queue and test by sending messages.
2. Register your mail in SES and send a test mail to yourself.

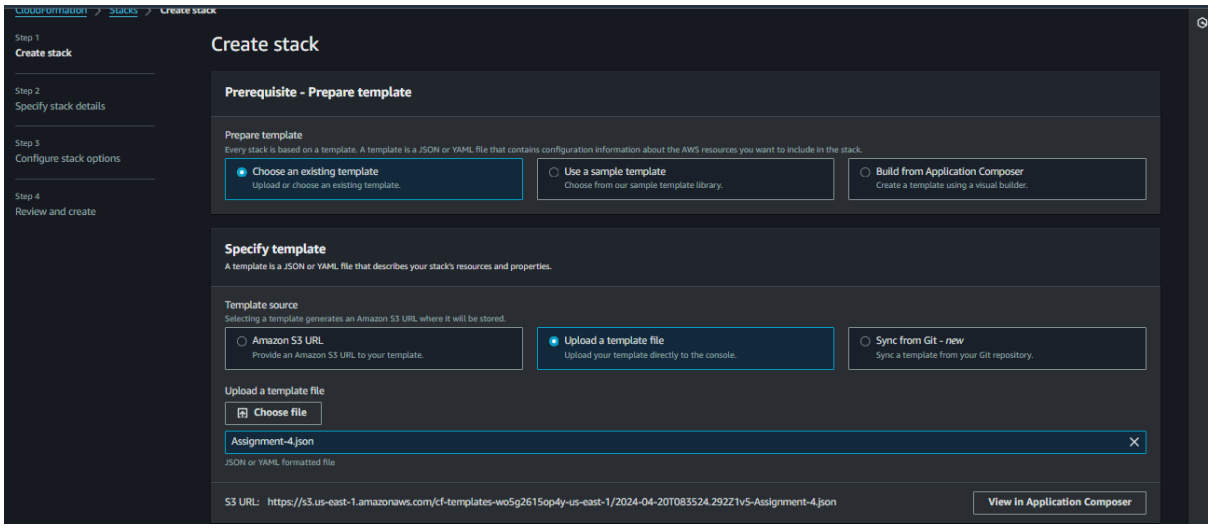
1. Creating FIFO SQS Queue only through json code and AWS::SES::Identity Is showing Unrecognized



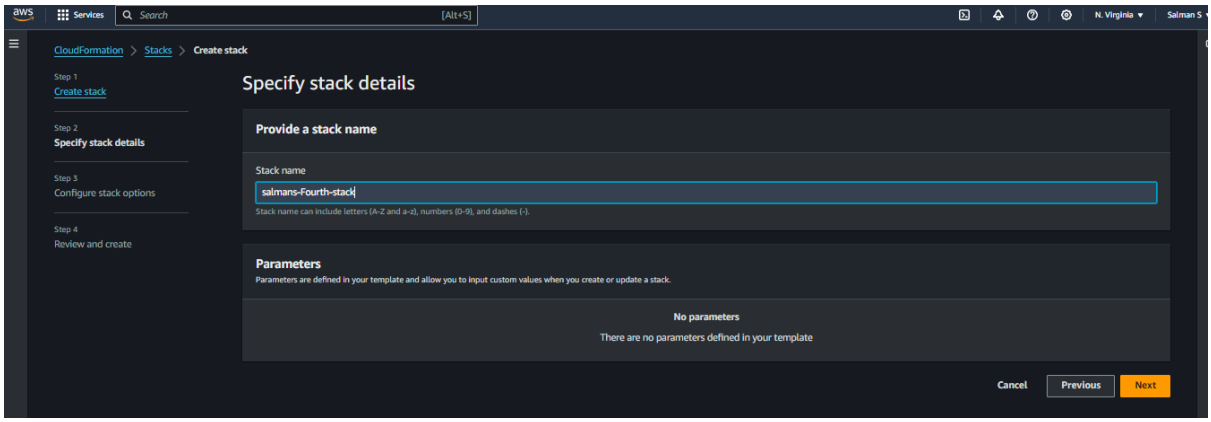
2. Canva Vison(HLD)



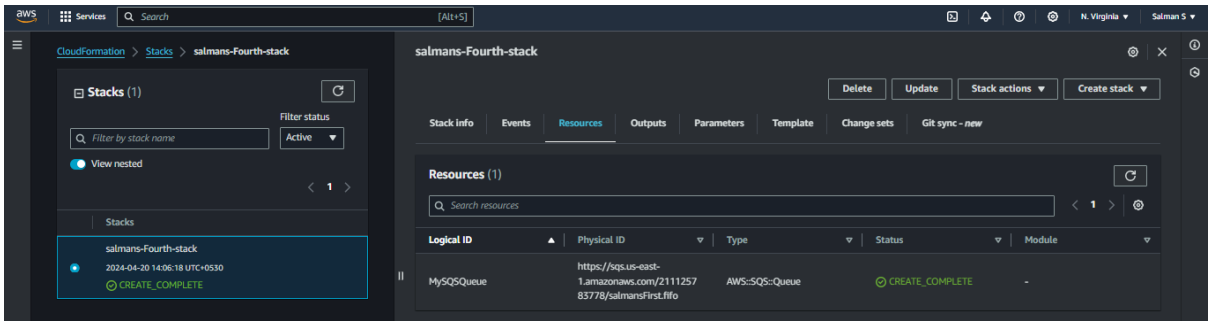
3. Create Fourth Stack Means 4th Assignment and Choose existing one and upload it and next



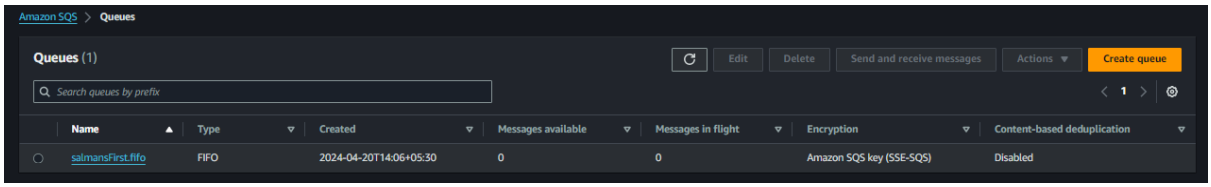
4. Stack Name and Next



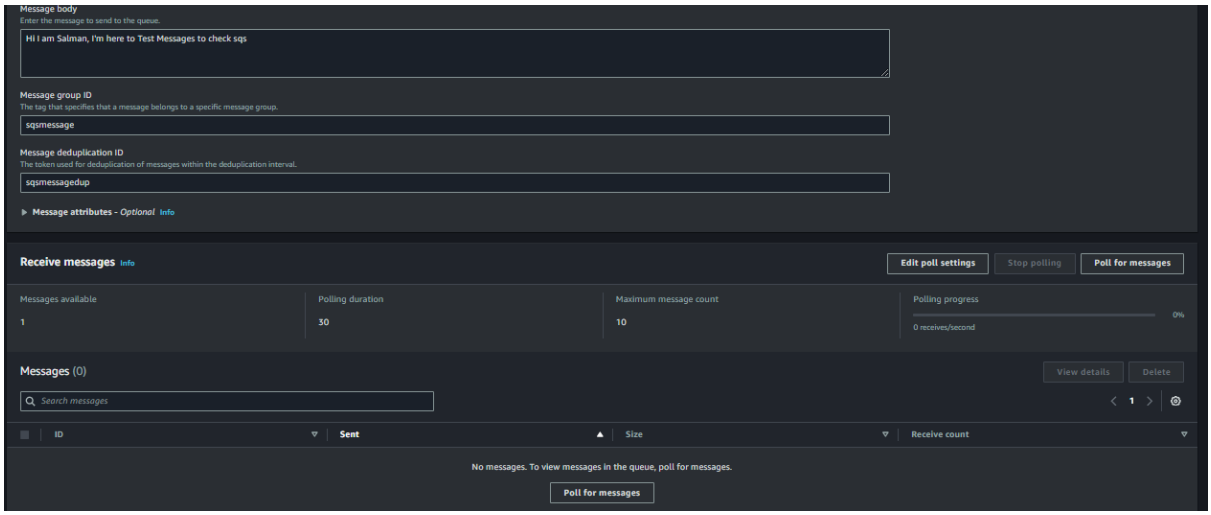
5. Created Successfully



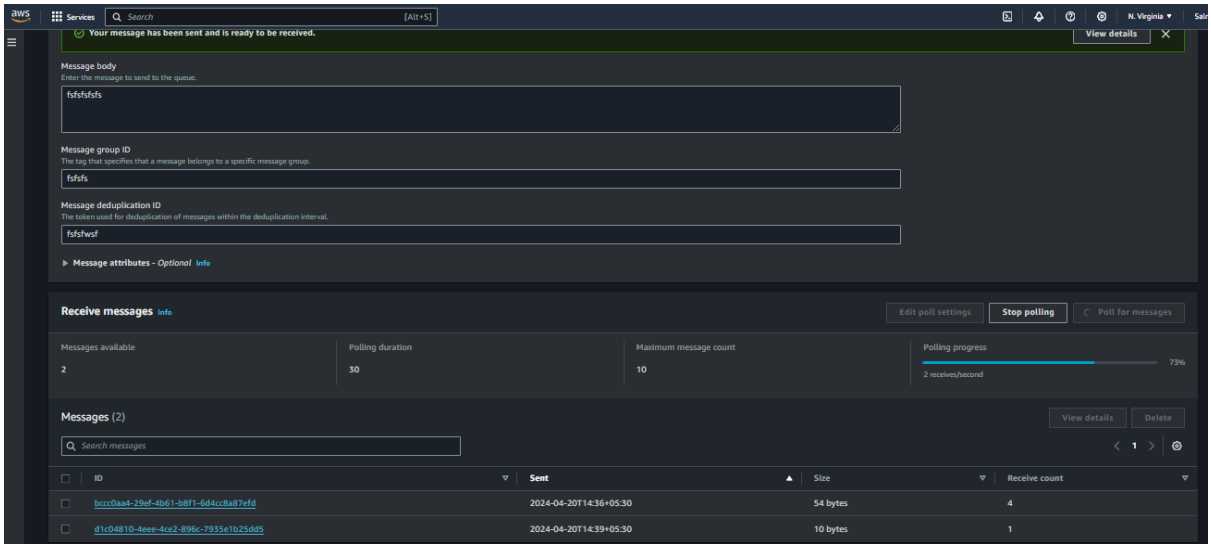
6. See the SalmanFirst.fifo is Created According to our given code



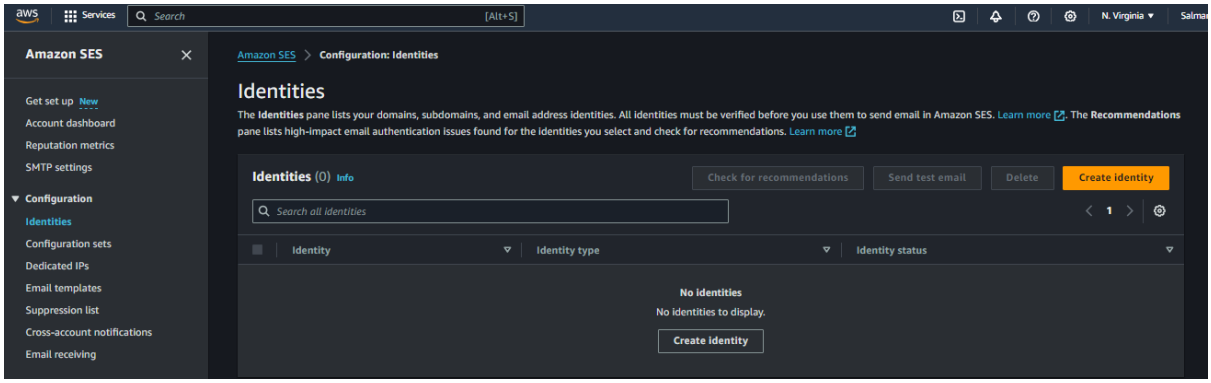
7. Create a Message in SQS



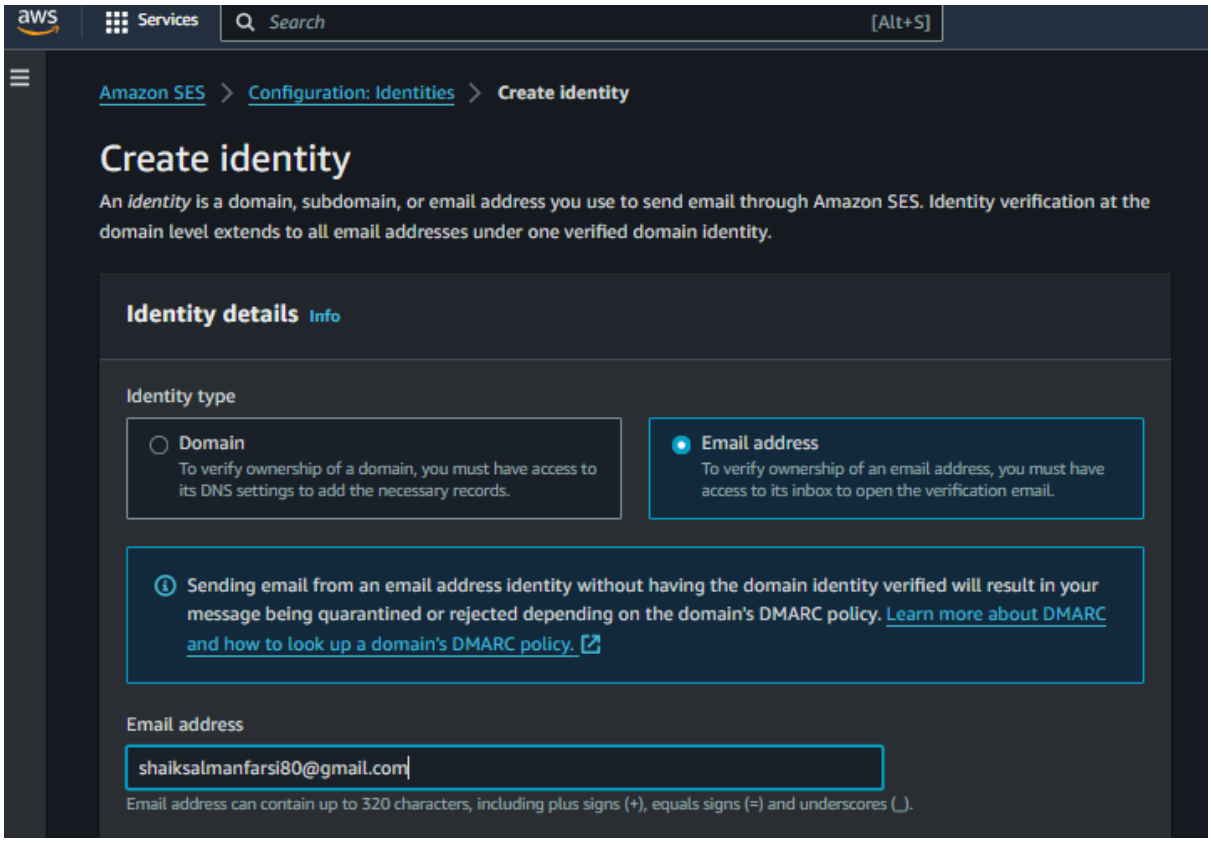
8. And Creating Multiple Messages and Poll for Messaging



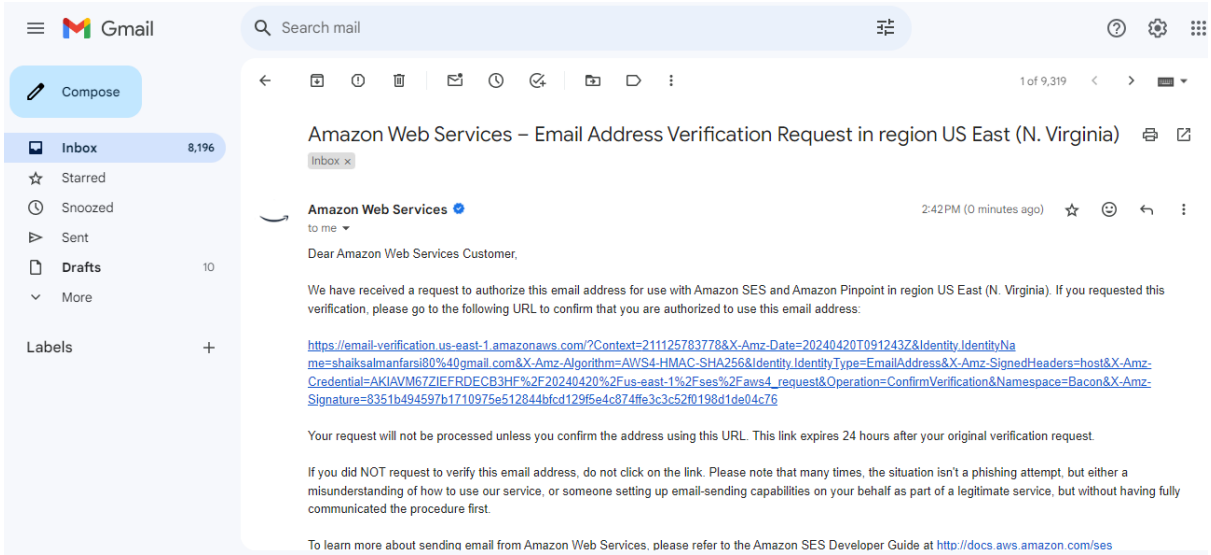
9. Go To Amazon SES and Click Identities and Create Identity



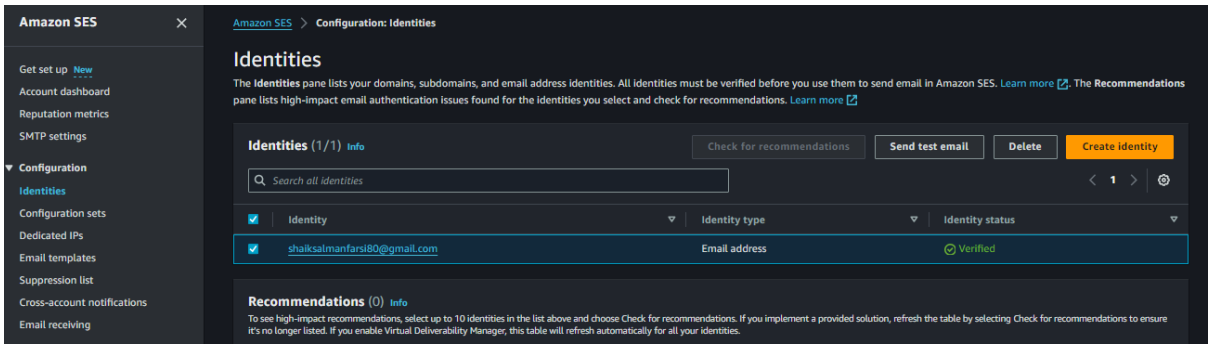
10. Click Email Address and Give Your Mail Id



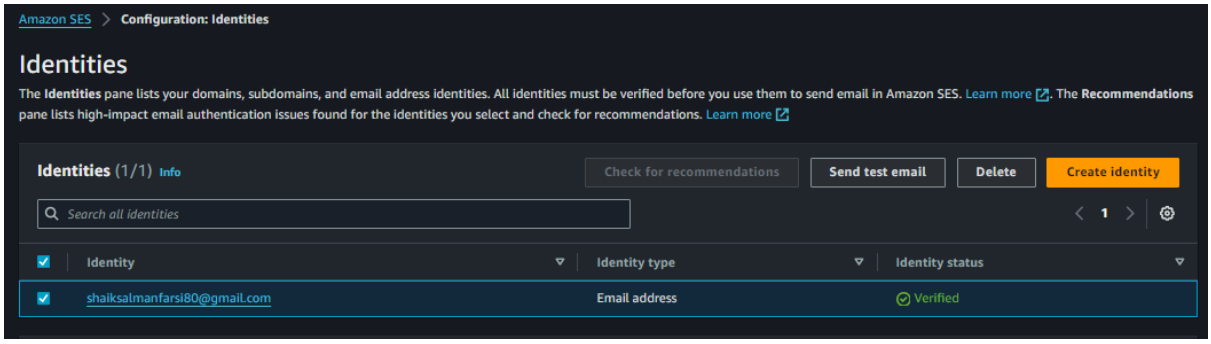
11. After u have to verify it



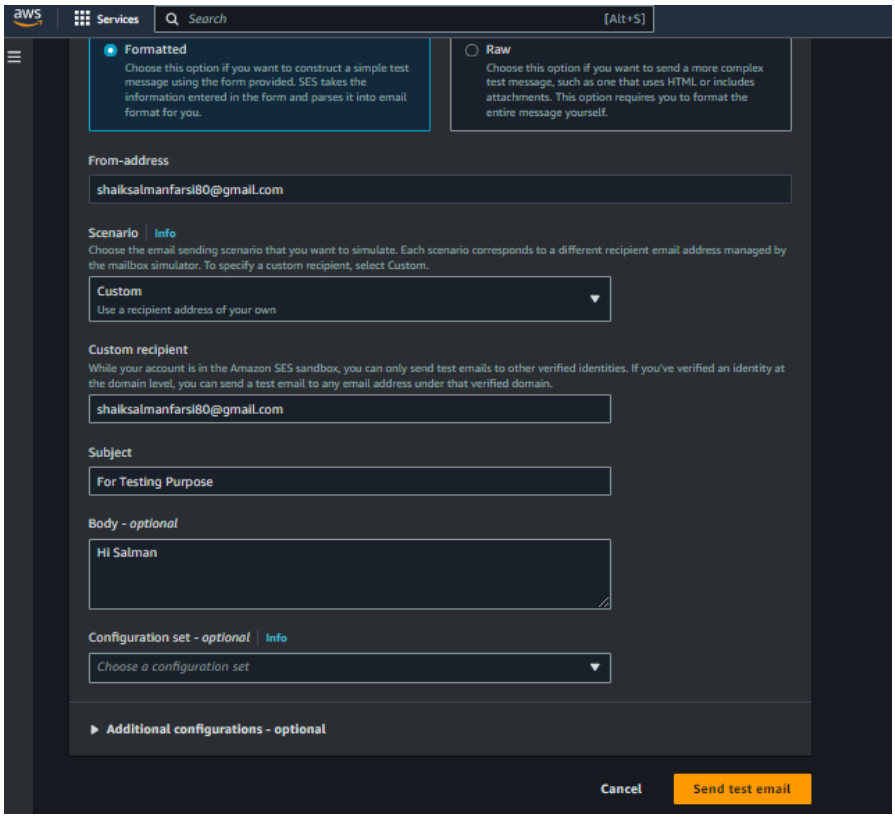
12. Select Email and Click Send Test Email



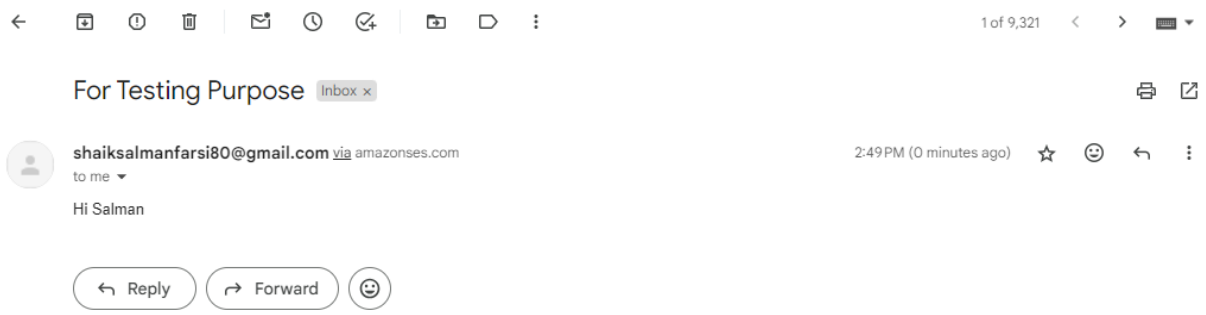
13. Same As it is Above the step



14. Simple Test Message then Select Formatted and From and To I selected Same And Subject is Testing Purpose and Body is Hi Salman



15. We Got Mail Via Amazonses.com and the Task Completed



Thank You