Project - 3: Publishing Amazon SNS Messages Privately

Assignment Submitted By:-Hitesh Chauhan
Course Offered: -Advanced Cloud Computing and Devops
Assignment By: -Intellipaat
Trainer: -Puneet Gavri
Date Of Submission: -16/12/2024

Industry: Healthcare

Problem Statement:
How to secure patient records online and send it privately to the intended party

Topics:
In this project, you will be working on a hospital project to send reports online and develop a platform so the patients can access the reports via mobile and push notifications. You will publish the report to an Amazon SNS keeping it secure and private. Your message will be hosted on an EC2 instance within your Amazon VPC. By publishing the messages privately, you can improve the message delivery and receipt through Amazon SNS.

Highlights:
1. AWS CloudFormation to create a VPC
2. Connect VPC with AWS SNS
3. Publish message privately with SNS

Project- 3: Solution

Answer:

Create a Could formation stack be using given cloud formation template. Cloud formation will create all the required resources using given yaml script.

This is The Parameter Section.

This is the code below

```yaml
AWSTemplateFormatVersion: 2010-09-09
Description: CloudFormation Template for SNS VPC Endpoints Tutorial
Parameters:
  KeyName:
    Description: Name of an existing EC2 KeyPair to enable SSH access to the instance
    Type: 'AWS::EC2::KeyPair::KeyName'
    ConstraintDescription: must be the name of an existing EC2 KeyPair.
  SSHLocation:
    Description: The IP address range that can be used to SSH to the EC2 instance
    Type: String
    MinLength: '9'
    MaxLength: '18'
    Default: 0.0.0.0/0
   AllowedPattern: '(\d{1,3})\.(\d{1,3})\.(\d{1,3})\.(\d{1,3})/(\d{1,2})'
    ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.
Mappings:
  RegionMap:
   us-east-1:
```

```
    AMI: ami-428aa838
us-east-2:
    AMI: ami-710e2414
us-west-1:
    AMI: ami-4a787a2a
us-west-2:
    AMI: ami-7f43f307
ap-northeast-1:
    AMI: ami-c2680fa4
ap-northeast-2:
    AMI: ami-3e04a450
ap-southeast-1:
    AMI: ami-4f89f533
ap-southeast-2:
    AMI: ami-38708c5a
ap-south-1:
    AMI: ami-3b2f7954
ca-central-1:
    AMI: ami-7549cc11
eu-central-1:
    AMI: ami-1b2bb774
eu-west-1:
    AMI: ami-db1688a2
eu-west-2:
    AMI: ami-6d263d09
eu-west-3:
    AMI: ami-5ce55321
```

sa-east-1:

  AMI: ami-f1337e9d

```yaml
AWSTemplateFormatVersion: 2010-09-09
Description: CloudFormation Template for SNS VPC Endpoints Tutorial
Parameters:
  KeyName:
    Description: Name of an existing EC2 KeyPair to enable SSH access to the instance
    Type: 'AWS::EC2::KeyPair::KeyName'
    ConstraintDescription: must be the name of an existing EC2 KeyPair.
  SSHLocation:
    Description: The IP address range that can be used to SSH to the EC2 instance
    Type: String
    MinLength: '9'
    MaxLength: '18'
    Default: 0.0.0.0/0
    AllowedPattern: '(\d{1,3})\.(\d{1,3})\.(\d{1,3})\.(\d{1,3})/(\d{1,2})'
    ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.
Mappings:
  RegionMap:
    us-east-1:
      AMI: ami-428aa838
    us-east-2:
      AMI: ami-710e2414
    us-west-1:
      AMI: ami-4a787a2a
    us-west-2:
      AMI: ami-7f43f307
    ap-northeast-1:
      AMI: ami-c2680fa4
    ap-northeast-2:
      AMI: ami-3e04a450
    ap-southeast-1:
      AMI: ami-4f89f533
    ap-southeast-2:
      AMI: ami-38708c5a
    ap-south-1:
      AMI: ami-3b2f7954
    ca-central-1:
      AMI: ami-7549cc11
```

```yaml
    eu-central-1:
      AMI: ami-1b2bb774
    eu-west-1:
      AMI: ami-db1688a2
    eu-west-2:
      AMI: ami-6d263d09
    eu-west-3:
      AMI: ami-5ce55321
    sa-east-1:
      AMI: ami-f1337e9d
```

This is Resource Code
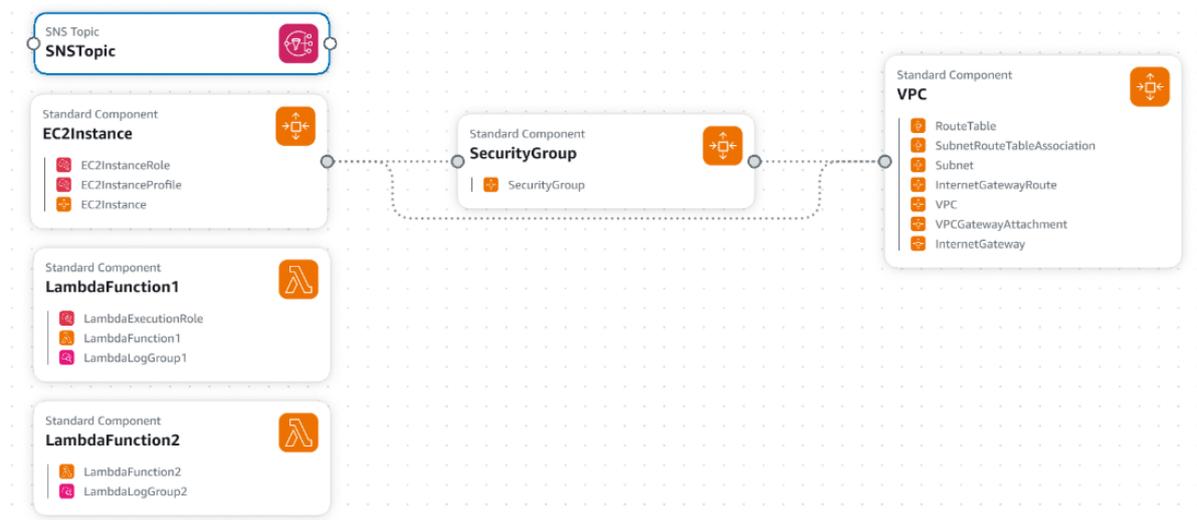
Resources:

  VPC:

   Type: 'AWS::EC2::VPC'

   Properties:

    CidrBlock: 10.0.0.0/16

    EnableDnsSupport: 'true'

    EnableDnsHostnames: 'true'

    Tags:

     - Key: Name

      Value: VPCE-Tutorial-VPC

  Subnet:

   Type: 'AWS::EC2::Subnet'

   Properties:

    VpcId: !Ref VPC

    CidrBlock: 10.0.0.0/24

    Tags:

     - Key: Name

      Value: VPCE-Tutorial-Subnet

```yaml
  InternetGateway:
    Type: 'AWS::EC2::InternetGateway'
    Properties:
      Tags:
        - Key: Name
          Value: VPCE-Tutorial-InternetGateway
  VPCGatewayAttachment:
    Type: 'AWS::EC2::VPCGatewayAttachment'
    Properties:
      VpcId: !Ref VPC
      InternetGatewayId: !Ref InternetGateway
  RouteTable:
    Type: 'AWS::EC2::RouteTable'
    Properties:
      VpcId: !Ref VPC
      Tags:
        - Key: Name
          Value: VPCE-Tutorial-RouteTable
  SubnetRouteTableAssociation:
    Type: 'AWS::EC2::SubnetRouteTableAssociation'
    Properties:
      RouteTableId: !Ref RouteTable
      SubnetId: !Ref Subnet
  InternetGatewayRoute:
    Type: 'AWS::EC2::Route'
    Properties:
      RouteTableId: !Ref RouteTable
```

```yaml
      GatewayId: !Ref InternetGateway
      DestinationCidrBlock: 0.0.0.0/0
  SecurityGroup:
    Type: 'AWS::EC2::SecurityGroup'
    Properties:
      GroupName: Tutorial Security Group
      GroupDescription: Security group for SNS VPC endpoint tutorial
      VpcId: !Ref VPC
      SecurityGroupIngress:
        - IpProtocol: '-1'
          CidrIp: 10.0.0.0/16
        - IpProtocol: tcp
          FromPort: '22'
          ToPort: '22'
          CidrIp: !Ref SSHLocation
      SecurityGroupEgress:
        - IpProtocol: '-1'
          CidrIp: 10.0.0.0/16
      Tags:
        - Key: Name
          Value: VPCE-Tutorial-SecurityGroup
  EC2Instance:
    Type: 'AWS::EC2::Instance'
    Properties:
      KeyName: !Ref KeyName
      InstanceType: t2.micro
      ImageId: !FindInMap
```

```yaml
            - RegionMap
            - !Ref 'AWS::Region'
            - AMI
        NetworkInterfaces:
          - AssociatePublicIpAddress: 'true'
            DeviceIndex: '0'
            GroupSet:
              - !Ref SecurityGroup
            SubnetId: !Ref Subnet
        IamInstanceProfile: !Ref EC2InstanceProfile
        Tags:
          - Key: Name
            Value: VPCE-Tutorial-EC2Instance
  EC2InstanceProfile:
    Type: 'AWS::IAM::InstanceProfile'
    Properties:
      Roles:
        - !Ref EC2InstanceRole
      InstanceProfileName: EC2InstanceProfile
  EC2InstanceRole:
    Type: 'AWS::IAM::Role'
    Properties:
      RoleName: VPCE-Tutorial-EC2InstanceRole
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
```

```yaml
      Principal:
        Service: ec2.amazonaws.com
      Action: 'sts:AssumeRole'
    ManagedPolicyArns:
      - 'arn:aws:iam::aws:policy/AmazonSNSFullAccess'
  LambdaExecutionRole:
    Type: 'AWS::IAM::Role'
    Properties:
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
        - Effect: Allow
          Principal:
            Service: lambda.amazonaws.com
          Action: 'sts:AssumeRole'
      ManagedPolicyArns:
        - 'arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole'
  LambdaFunction1:
    Type: 'AWS::Lambda::Function'
    Properties:
      Code:
        ZipFile: |
          from __future__ import print_function
          print('Loading function')
          def lambda_handler(event, context):
            message = event['Records'][0]['Sns']['Message']
```

```yaml
        print("From SNS: " + message)

        return message
  Description: SNS VPC endpoint tutorial lambda function 1
  FunctionName: VPCE-Tutorial-Lambda-1
  Handler: index.lambda_handler
  Role: !GetAtt
    - LambdaExecutionRole
    - Arn
  Runtime: python3.9
  Timeout: '3'
LambdaPermission1:
 Type: 'AWS::Lambda::Permission'
 Properties:
  Action: 'lambda:InvokeFunction'
  FunctionName: !Ref LambdaFunction1
  Principal: sns.amazonaws.com
  SourceArn: !Ref SNSTopic
LambdaLogGroup1:
 Type: 'AWS::Logs::LogGroup'
 Properties:
  LogGroupName: !Sub "/aws/lambda/${LambdaFunction1}"
  RetentionInDays: '7'
LambdaFunction2:
 Type: 'AWS::Lambda::Function'
 Properties:
  Code:
   ZipFile: |
```

```
        from __future__ import print_function
        print('Loading function')
        def lambda_handler(event, context):
          message = event['Records'][0]['Sns']['Message']
          print("From SNS: " + message)
          return message
    Description: SNS VPC endpoint tutorial lambda function 2
    FunctionName: VPCE-Tutorial-Lambda-2
    Handler: index.lambda_handler
    Role: !GetAtt
      - LambdaExecutionRole
      - Arn
    Runtime: python3.9
    Timeout: '3'
LambdaPermission2:
  Type: 'AWS::Lambda::Permission'
  Properties:
    Action: 'lambda:InvokeFunction'
    FunctionName: !Ref LambdaFunction2
    Principal: sns.amazonaws.com
    SourceArn: !Ref SNSTopic
LambdaLogGroup2:
  Type: 'AWS::Logs::LogGroup'
  Properties:
    LogGroupName: !Sub "/aws/lambda/${LambdaFunction2}"
    RetentionInDays: '7'
SNSTopic:
```

```
  Type: 'AWS::SNS::Topic'
 Properties:
  DisplayName: VPCE-Tutorial-Topic
  TopicName: VPCE-Tutorial-Topic
  Subscription:
   - Endpoint: !GetAtt
     - LambdaFunction1
     - Arn
    Protocol: lambda
   - Endpoint: !GetAtt
     - LambdaFunction2
     - Arn
    Protocol: lambda
```

Go to Aws Management Console you will see like this page
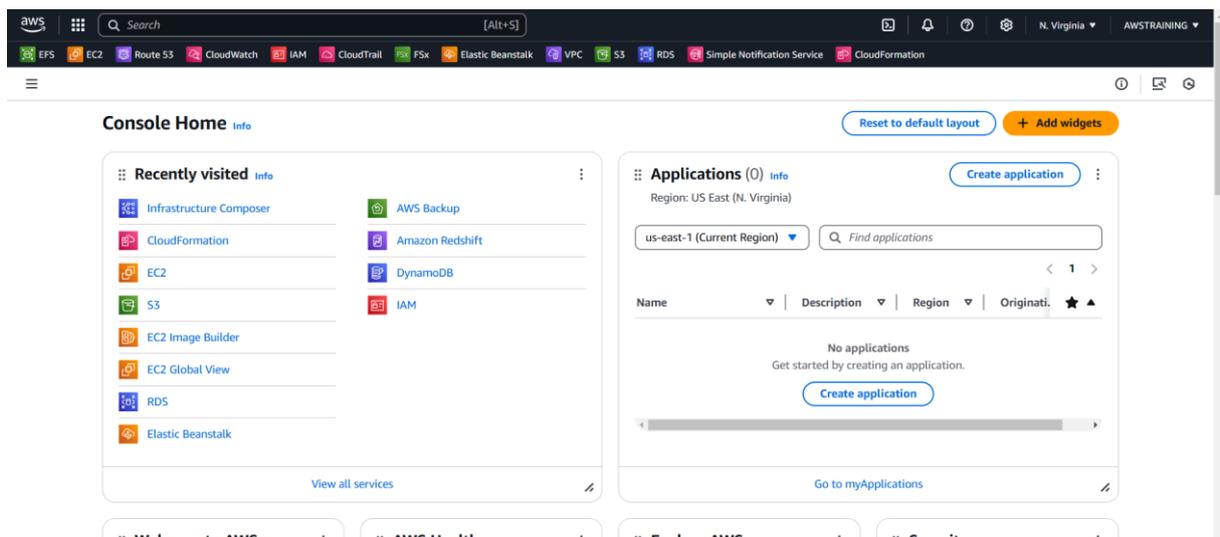
Then Search Cloud Formation
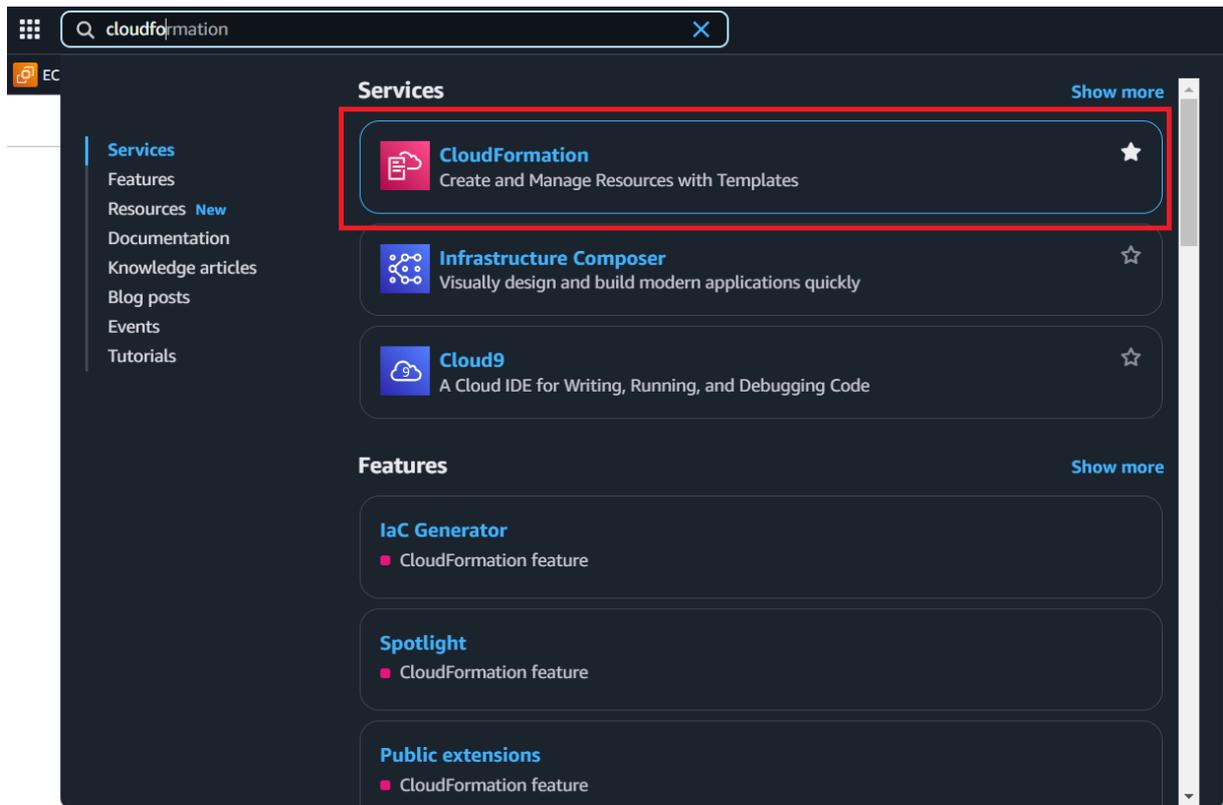
Click the cloud Formation service.
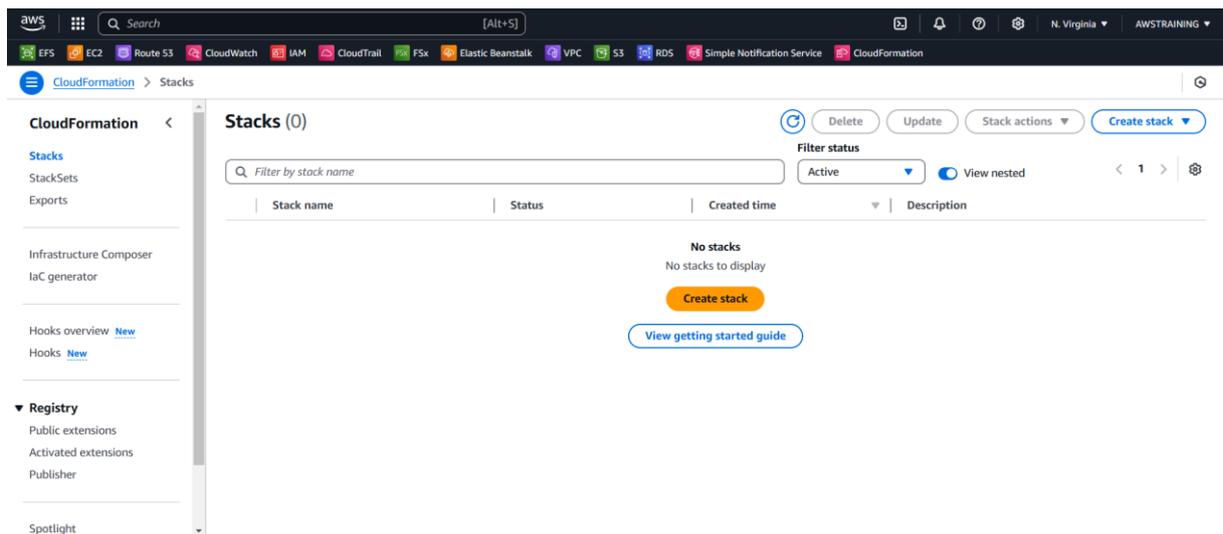
Then Click Infrastructure Composer
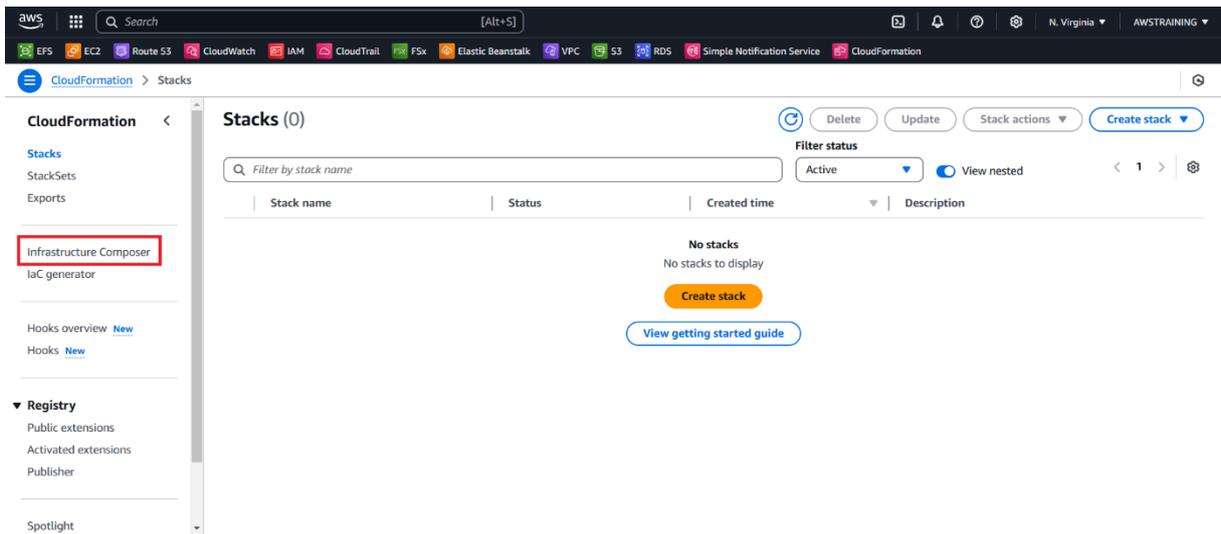
After click the Infrastructure composer you will see this page.
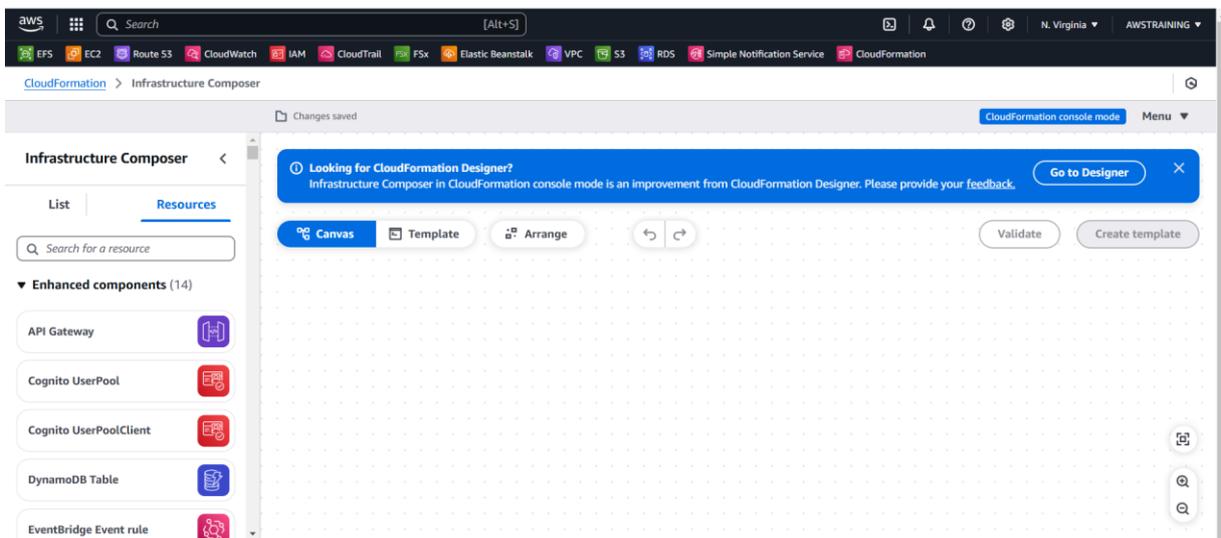
Then Click Templates and paste the code.
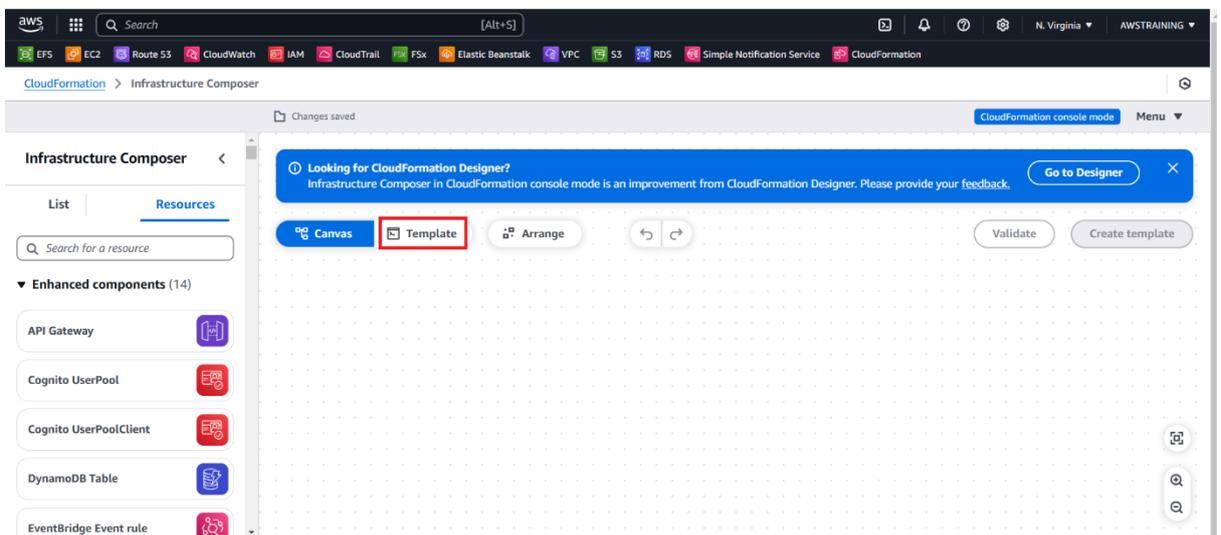
You will see like this page then click canvas
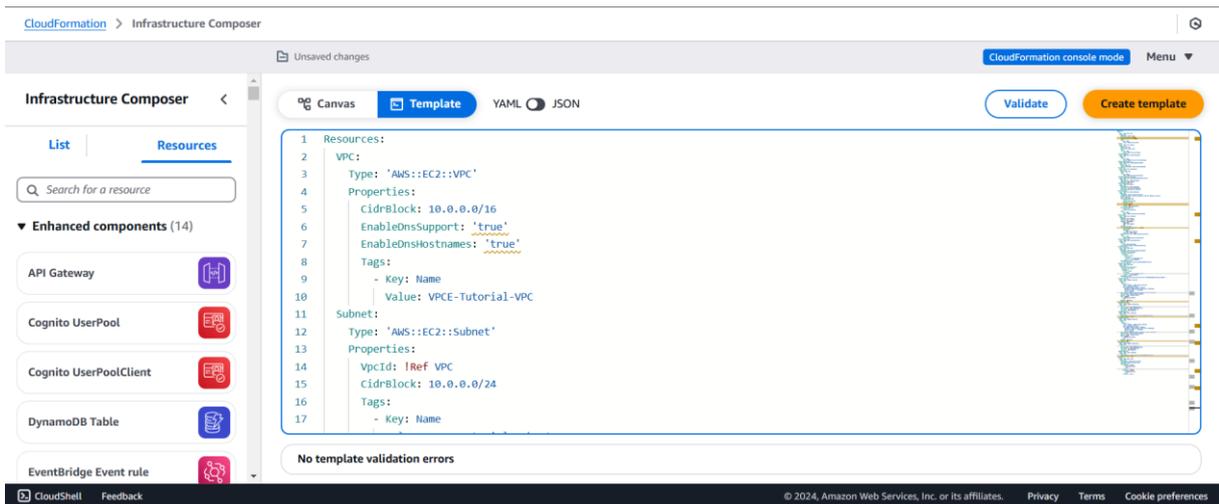
After Click the canvas you will see below architecture of cloud formation code.
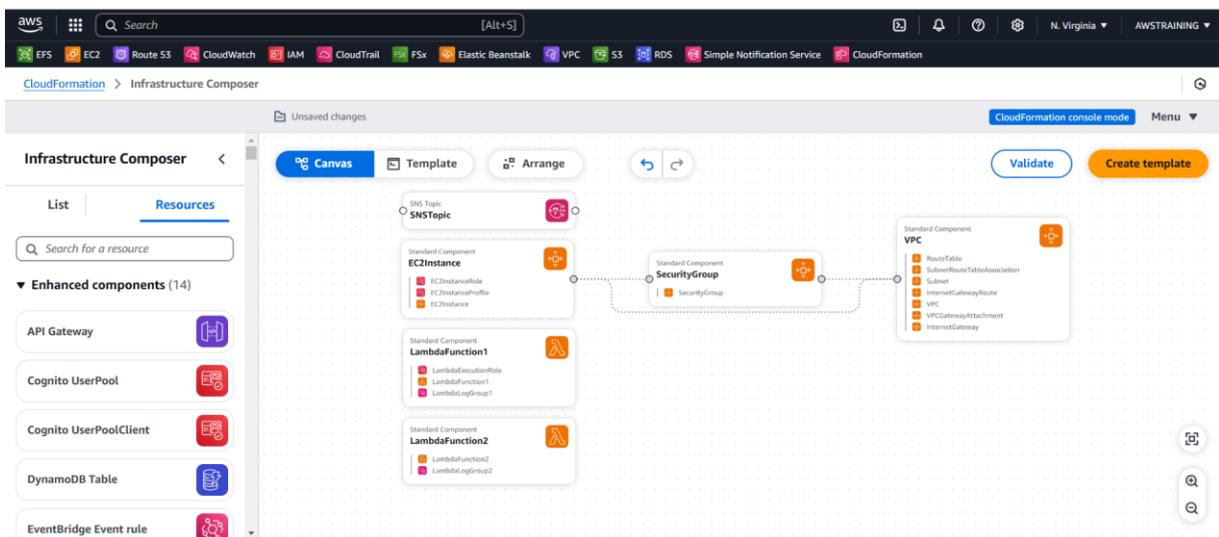
Now Click Create Template
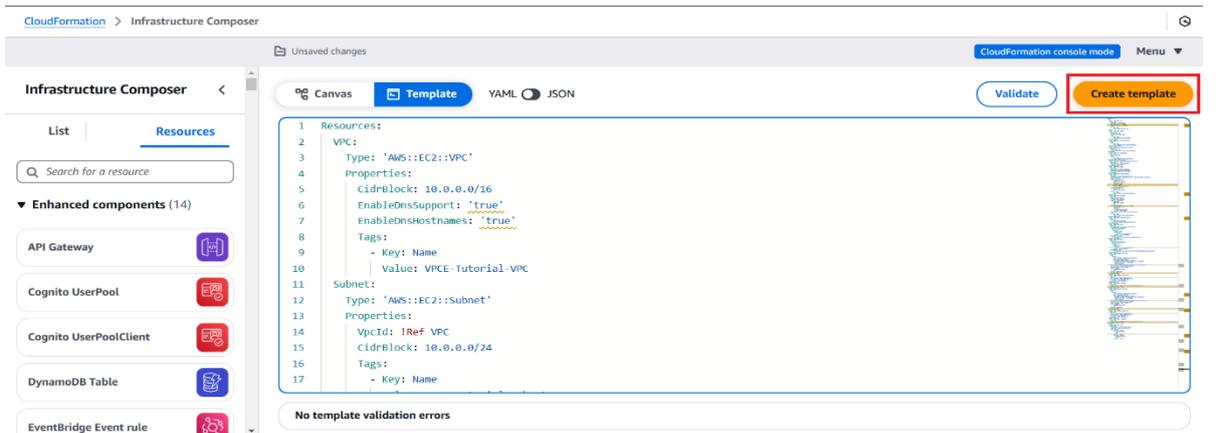
Now Click the Confirm And Continue to cloud formation
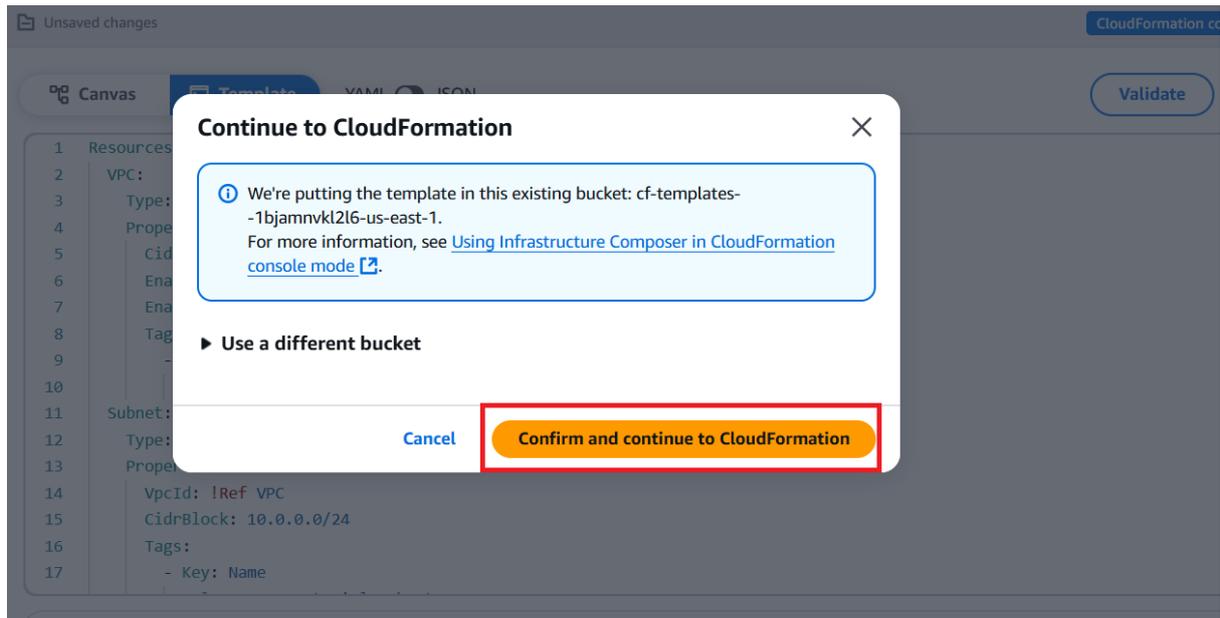
After Click the Confirm and Continue to cloud formation then you will this stack creation page.
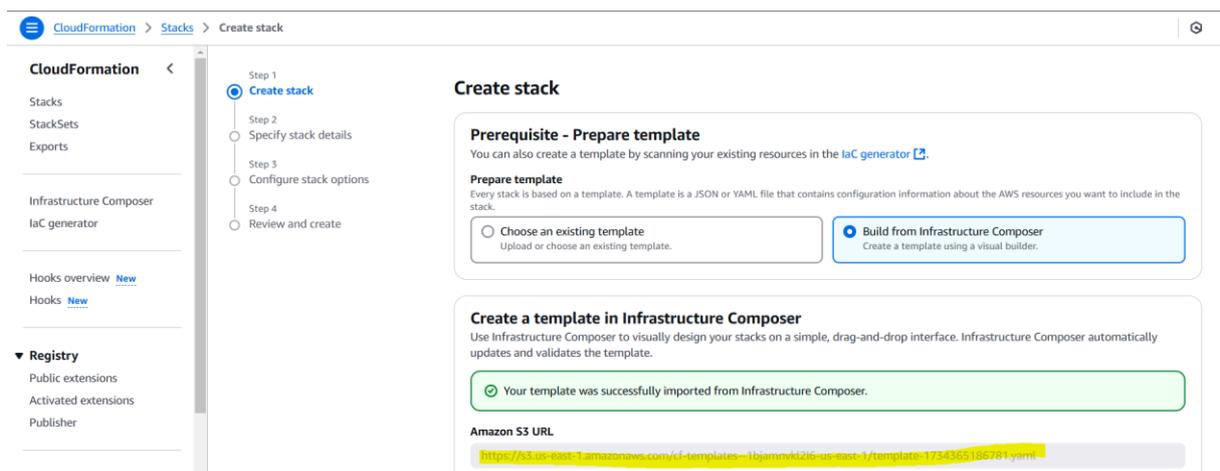
Click Next
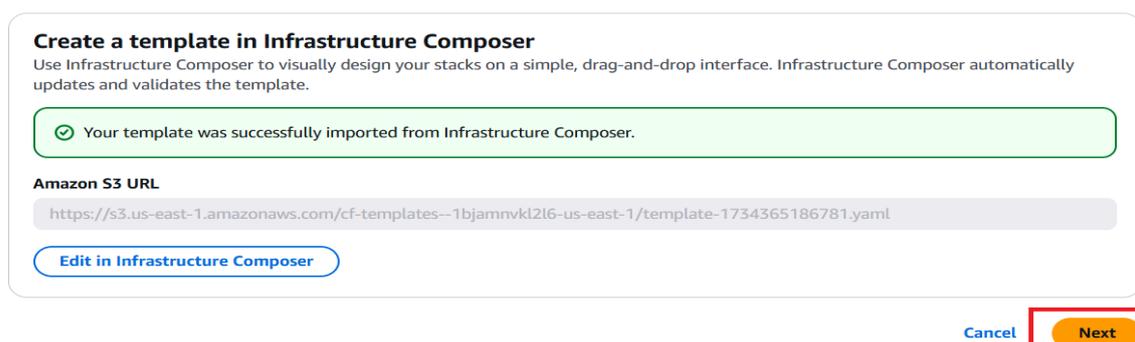
After Click the next you need to specify the stack details then click next.

This Section nothing to do.

In Stack failure options.there is two option.

1.Roll back all stack resources

2.Preserve successfully provisioned resources.

So choosed to Preserve successfully provisioned resources.

And There is another option in stack failure is

Delete newly created resources during a rollback

In this there is two options

1.use deletion policy

2.Preserve successfully provisioned resources.

Choosed to Preserve successfully provisioned resources.

In advanced options.we nothing to do changes.

In Capabilties Check the box and click next

Then we Need To Review all the things

Step 1
Create stack

Step 2
Specify stack details

Step 3
Configure stack options

Step 4
Review and create

Review and create

Step 1: Specify template

Edit

Prerequisite - Prepare template

Template
-

Template

Template URL
https://s3.us-east-1.amazonaws.com/cf-templates--1bjamnvkl2l6-us-east-1/template-1734365467767.yaml

Stack description
CloudFormation Template for SNS VPC Endpoints Tutorial

Step 2: Specify stack details

Edit

Provide a stack name

Stack name
CFFORVPC

Parameters (2)

Key ▲	Value ▽
KeyName	cloudwatch logs
SSHLocation	0.0.0.0/0

Step 3: Configure stack options

Edit

Tags

Key ▲	Value ▽
No tags	
There are no tags defined for this stack	

Permissions

No permissions

There is no IAM role associated with this stack

Stack failure options

Rollback on failure
Deactivated

Delete newly created resources during a rollback
Activated

Stack policy

No stack policy

There is no stack policy defined

Rollback configuration

Monitoring time
-

CloudWatch alarm ARN
-

Notification options

SNS topic ARN

No notification options

There are no notification options defined

Stack creation options

Timeout
-

Termination protection
Deactivated

Quick-create link

Use quick-create links to get stacks up and running quickly from the AWS CloudFormation console with the same basic configuration as this stack. Copy the URL on the link to share. Learn more [↗]

Open quick-create link [↗]

Create change set   Cancel   Previous   Submit

Then click submit.

After Click the submit you see the event updates

CFFORVPC

Stacks (1)

Filter status

`Filter by stack name` | Active ▼

View nested

‹ **1** ›

Stacks
CFFORVPC
2024-12-16 21:53:26 UTC+0530
ⓘ CREATE_IN_PROGRESS

Events (60)

View root cause

`Search events`

Timestamp ▼	Logical ID	Status	Detailed status
2024-12-16 21:53:44 UTC+0530	SecurityGroup	ⓘ CREATE_IN_PROGRESS	-
2024-12-16 21:53:43 UTC+0530	RouteTable	ⓘ CREATE_IN_PROGRESS	⊘ CONFIGURATION_COMPLETE
2024-12-16 21:53:43 UTC+0530	Subnet	ⓘ CREATE_IN_PROGRESS	-
2024-12-16 21:53:42 UTC+0530	RouteTable	ⓘ CREATE_IN_PROGRESS	-
2024-12-16 21:53:42 UTC+0530	VPCGatewayAttachment	ⓘ CREATE_IN_PROGRESS	-
2024-12-16 21:53:41 UTC+0530	Subnet	ⓘ CREATE_IN_PROGRESS	-
2024-12-16 21:53:41 UTC+0530	RouteTable	ⓘ CREATE_IN_PROGRESS	-

---

CFFORVPC

Stacks (1)

Filter status

`Filter by stack name` | Active ▼

View nested

‹ **1** ›

Stacks
CFFORVPC
2024-12-16 21:53:26 UTC+0530
⊘ CREATE_COMPLETE

Events (63)

View root cause

`Search events`

Timestamp ▼	Logical ID	Status	Detailed status
2024-12-16 21:53:46 UTC+0530	LambdaFunction2	ⓘ CREATE_IN_PROGRESS	-
2024-12-16 21:53:46 UTC+0530	SubnetRouteTableAssociation	ⓘ CREATE_IN_PROGRESS	-
2024-12-16 21:53:46 UTC+0530	LambdaFunction1	ⓘ CREATE_IN_PROGRESS	-
2024-12-16 21:53:45 UTC+0530	Subnet	⊘ CREATE_COMPLETE	-
2024-12-16 21:53:45 UTC+0530	LambdaExecutionRole	⊘ CREATE_COMPLETE	-
2024-12-16 21:53:45 UTC+0530	InternetGateway	⊘ CREATE_COMPLETE	-
2024-12-16 21:53:44 UTC+0530	VPCGatewayAttachment	⊘ CREATE_COMPLETE	-

---

CFFORVPC

Stacks (1)

Filter status

`Filter by stack name` | Active ▼

View nested

‹ **1** ›

Stacks
CFFORVPC
2024-12-16 21:53:26 UTC+0530
⊘ CREATE_COMPLETE

Events (63)

View root cause

`Search events`

Timestamp ▼	Logical ID	Status	Detailed status
2024-12-16 21:53:48 UTC+0530	EC2InstanceProfile	ⓘ CREATE_IN_PROGRESS	-
2024-12-16 21:53:47 UTC+0530	SubnetRouteTableAssociation	⊘ CREATE_COMPLETE	-
2024-12-16 21:53:47 UTC+0530	LambdaFunction1	ⓘ CREATE_IN_PROGRESS	-
2024-12-16 21:53:47 UTC+0530	LambdaFunction2	ⓘ CREATE_IN_PROGRESS	-
2024-12-16 21:53:47 UTC+0530	SubnetRouteTableAssociation	ⓘ CREATE_IN_PROGRESS	-
2024-12-16 21:53:47 UTC+0530	EC2InstanceProfile	ⓘ CREATE_IN_PROGRESS	-
2024-12-16 21:53:46 UTC+0530	EC2InstanceRole	⊘ CREATE_COMPLETE	-

Stacks (1)

Filter status

Filter by stack name

Active ▼

View nested

‹ 1 ›

Stacks

CFFORVPC
2024-12-16 21:53:26 UTC+0530
⊘ CREATE_COMPLETE

Events (63)

View root cause

Search events

Timestamp	Logical ID	Status	Detailed status
2024-12-16 21:53:55 UTC+0530	InternetGatewayRoute	⊘ CREATE_COMPLETE	-
2024-12-16 21:53:55 UTC+0530	LambdaFunction1	⊘ CREATE_COMPLETE	-
2024-12-16 21:53:55 UTC+0530	LambdaFunction2	⊘ CREATE_COMPLETE	-
2024-12-16 21:53:55 UTC+0530	InternetGatewayRoute	ⓘ CREATE_IN_PROGRESS	-
2024-12-16 21:53:54 UTC+0530	InternetGatewayRoute	ⓘ CREATE_IN_PROGRESS	-
2024-12-16 21:53:53 UTC+0530	RouteTable	⊘ CREATE_COMPLETE	-
2024-12-16 21:53:51 UTC+0530	SecurityGroup	⊘ CREATE_COMPLETE	-

Stacks (1)

Filter status

Filter by stack name

Active ▼

View nested

‹ 1 ›

Stacks

CFFORVPC
2024-12-16 21:53:26 UTC+0530
⊘ CREATE_COMPLETE

Events (63)

View root cause

Search events

Timestamp	Logical ID	Status	Detailed status
2024-12-16 21:53:57 UTC+0530	EC2Instance	ⓘ CREATE_IN_PROGRESS	-
2024-12-16 21:53:57 UTC+0530	EC2InstanceProfile	ⓘ CREATE_IN_PROGRESS	⊘ CONFIGURATION_COMPLETE
2024-12-16 21:53:56 UTC+0530	LambdaLogGroup1	ⓘ CREATE_IN_PROGRESS	-
2024-12-16 21:53:56 UTC+0530	LambdaLogGroup2	ⓘ CREATE_IN_PROGRESS	-
2024-12-16 21:53:55 UTC+0530	SNSTopic	ⓘ CREATE_IN_PROGRESS	-
2024-12-16 21:53:55 UTC+0530	LambdaLogGroup1	ⓘ CREATE_IN_PROGRESS	-
2024-12-16 21:53:55 UTC+0530	LambdaLogGroup2	ⓘ CREATE_IN_PROGRESS	-

Stacks (1)

Filter status

Filter by stack name

Active ▼

View nested

‹ 1 ›

Stacks

CFFORVPC
2024-12-16 21:53:26 UTC+0530
⊘ CREATE_COMPLETE

Events (63)

View root cause

Search events

Timestamp	Logical ID	Status	Detailed status
2024-12-16 21:54:04 UTC+0530	LambdaPermission1	ⓘ CREATE_IN_PROGRESS	-
2024-12-16 21:54:04 UTC+0530	LambdaPermission2	ⓘ CREATE_IN_PROGRESS	-
2024-12-16 21:54:03 UTC+0530	SNSTopic	ⓘ CREATE_IN_PROGRESS	⊘ CONFIGURATION_COMPLETE
2024-12-16 21:54:03 UTC+0530	LambdaLogGroup1	⊘ CREATE_COMPLETE	-
2024-12-16 21:54:03 UTC+0530	LambdaLogGroup2	⊘ CREATE_COMPLETE	-
2024-12-16 21:54:03 UTC+0530	SNSTopic	ⓘ CREATE_IN_PROGRESS	-
2024-12-16 21:53:59 UTC+0530	EC2Instance	ⓘ CREATE_IN_PROGRESS	-

The Stack has been created successfully now we need to verify all the things.

Route tables (1/3) Info

Last updated 1 minute ago

Actions ▼   **Create route table**

Find resources by attribute or tag

< 1 >

	Name	▽	Route table ID	▽	Explicit subnet associ...	▽	Edge associations	▽	Main	▽	VPC
☐	–		rtb-095aabeb9b1593bb7		–		–		Yes		vpc-0ca4bd13727b758cb
☑	VPCE-Tutorial-RouteTable		rtb-01cb081ccf3becffc		subnet-07f4c7128b2b57...		–		No		vpc-0c42f0b520e4188f1 | V
☐	–		rtb-06d3471056bc003e8		–		–		Yes		vpc-0c42f0b520e4188f1 | V

VPC > Route tables > rtb-01cb081ccf3becffc

rtb-01cb081ccf3becffc / VPCE-Tutorial-RouteTable

Actions ▼

Details Info

Route table ID	Main	Explicit subnet associations	Edge associations
🗏 rtb-01cb081ccf3becffc	🗏 No	subnet-07f4c7128b2b576c1 / VPCE-Tutorial-Subnet	–
VPC	Owner ID		
vpc-0c42f0b520e4188f1 | VPCE-Tutorial-VPC	🗏 207567757353		

Routes | Subnet associations | Edge associations | Route propagation | Tags

Routes (2)

Both ▼   **Edit routes**

Filter routes

< 1 >

Destination	▽	Target	▽	Status	▽	Propagated	▽
0.0.0.0/0		igw-0eb71c7f57ed45c98		⊘ Active		No	
10.0.0.0/16		local		⊘ Active		No	

Internet gateways (2) Info

Actions ▼   **Create internet gateway**

Search

< 1 >

	Name	▽	Internet gateway ID	▽	State	▽	VPC ID	▽	Owner
☐	–		igw-01655718ff8c7f3a5		⊘ Attached		vpc-0ca4bd13727b758cb		207567757353
☐	VPCE-Tutorial-InternetGateway		igw-0eb71c7f57ed45c98		⊘ Attached		vpc-0c42f0b520e4188f1 | VPCE-Tutoria...		207567757353

VPC > Internet gateways > igw-0eb71c7f57ed45c98

igw-0eb71c7f57ed45c98 / VPCE-Tutorial-InternetGateway

Actions ▼

Details Info

Internet gateway ID	State	VPC ID	Owner
🗏 igw-0eb71c7f57ed45c98	⊘ Attached	vpc-0c42f0b520e4188f1 | VPCE-Tutorial-VPC	🗏 207567757353

Tags

Manage tags

Search tags

< 1 >

Key	Value
aws:cloudf...	CFFORVPC
Name	VPCE-Tutorial-InternetGateway
aws:cloudf...	InternetGateway
aws:cloudf...	arn:aws:cloudformation:us-east-1:207567757353:stack/CFFORVPC/13a3e230-bbca-11ef-900b-0ec39c392a3f

---

☰   🖵 ⊘ 🖵 🖵

Dashboard		**Instances (1)** Info	Last updated less than a minute ago	Connect	Instance state ▼	Actions ▼	**Launch instances** ▼

EC2 Global View
Events
▼ **Instances**
  Instances
  Instance Types

Find Instance by attribute or tag (case-sensitive)

All states ▼

< 1 >

	Name ✐	▽	Instance ID	Instance state	▽	Instance type	▽	Status check	Alarm status	Availability Zone	▽	Public IPv4
☐	VPCE-Tutorial-...		i-057a8c3ab9287d8f0	⊘ Running ⊕ ⊖		t2.micro		⊘ 2/2 checks passec	View alarms +	us-east-1f		ec2-3-238-...

We need to subscribe the sns topics.After Confirm the subscription you will notified.

Now We need to connect the ec2 instance from mobaxterm tool.

We need to do aws configure need to configure

We need provide Access Key And Secret Access key
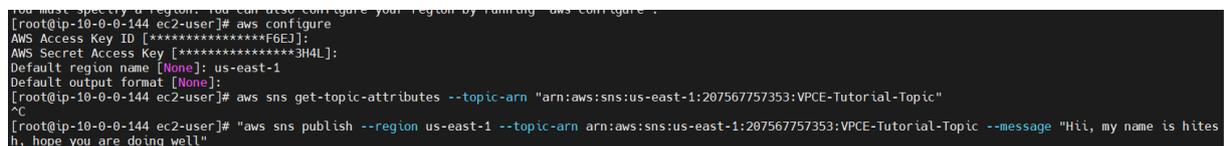
Then We need to run command **aws sns get-topic-attributes --topic-arn "arn:aws:sns:us-east-1:207567757353:VPCE-Tutorial-Topic"**

Then Run **"aws sns publish --region us-east-1 --topic-arn arn:aws:sns:us-east-1:207567757353:VPCE-Tutorial-Topic --message "Hii, my name is hitesh, hope you are doing well"**

This is the email confirmation.
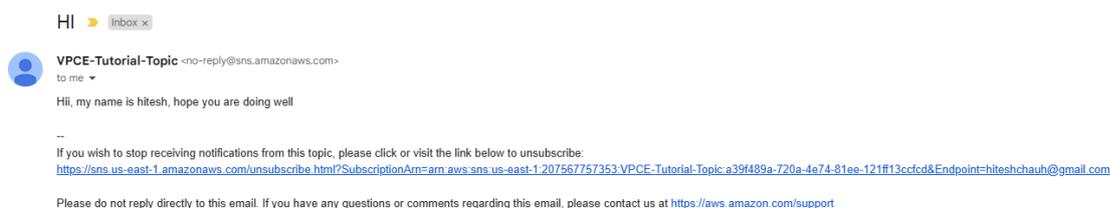
Note: There is another options if command is not working so go to simple notification service.then subscribe the those topic you will choose manually send the push message.