Name: Nishant Gupta
Email: kr.nish99@gmail.com

Industry: Healthcare

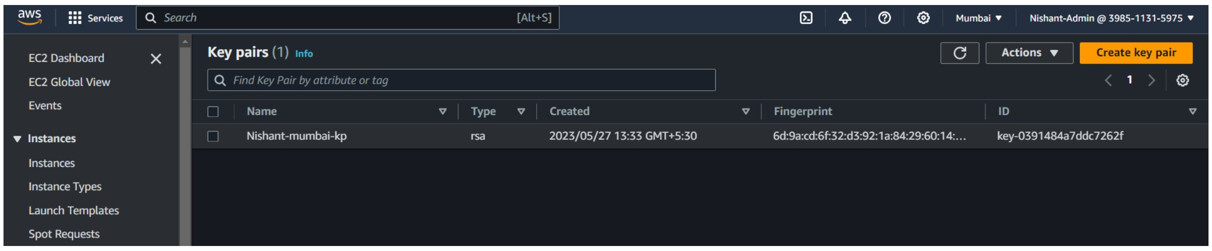
Problem Statement:
How to secure patient records online and send it privately to the intended party Topics: In this project, you will be working on a hospital project to send reports online and develop a platform so the patients can access the reports via mobile and push notifications. You will publish 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 messages privately with SNS.

Task 1:

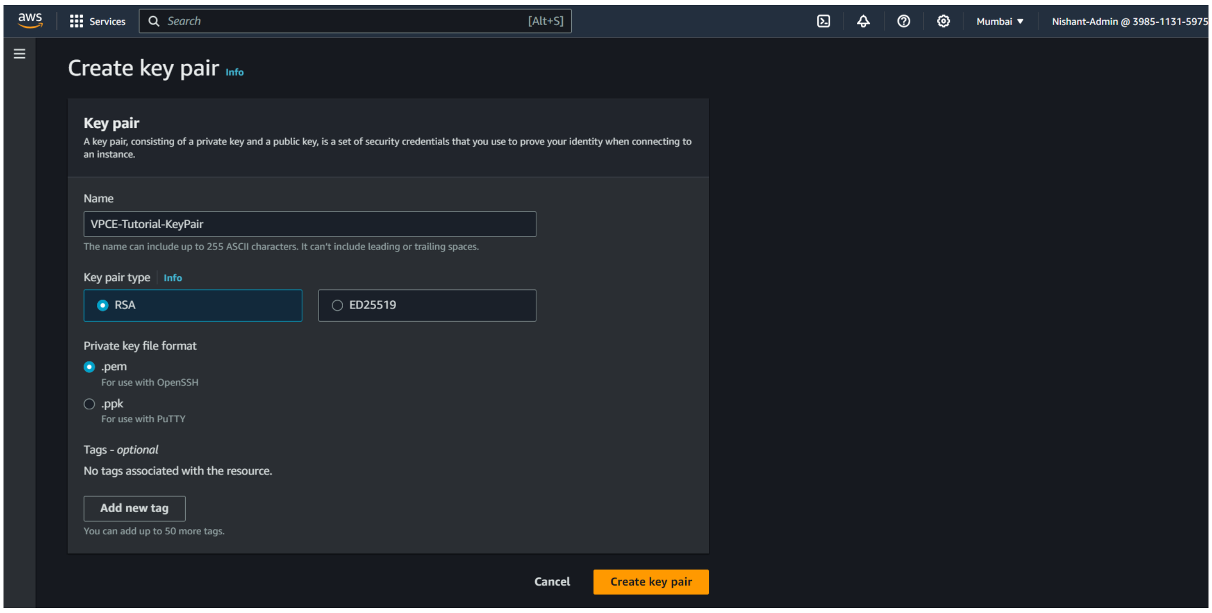
- We have to create keypair in EC2, Sign in to the AWS Management Console and open the Amazon EC2 console.
- find the Network & Security section. Then, choose Key Pairs.



Key pairs (1) Info

Name	Type	Created	Fingerprint	ID
Nishant-mumbai-kp	rsa	2023/05/27 13:33 GMT+5:30	6d:9a:cd:6f:32:d3:92:1a:84:29:60:14:...	key-0391484a7ddc7262f

- Click on create key pair.



Create key pair Info

Key pair
A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name
VPCE-Tutorial-KeyPair
The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type Info
[✓ RSA] ED25519

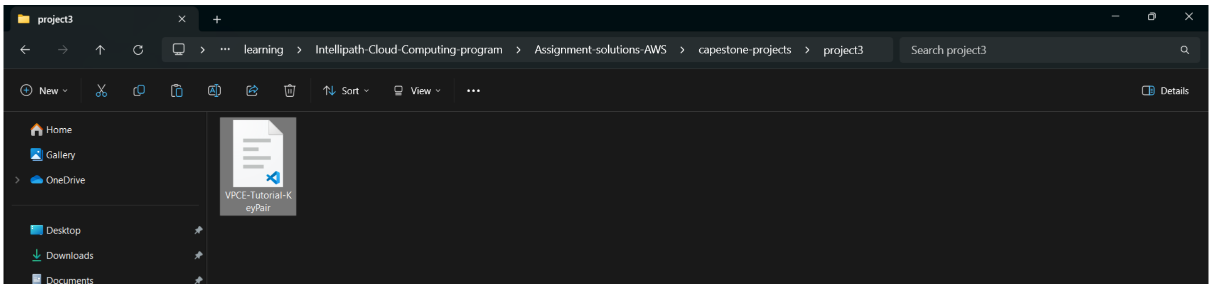
Private key file format
[✓ .pem] For use with OpenSSH
.ppk For use with PuTTY

Tags - *optional*
No tags associated with the resource.

Add new tag
You can add up to 50 more tags.

Cancel | Create key pair

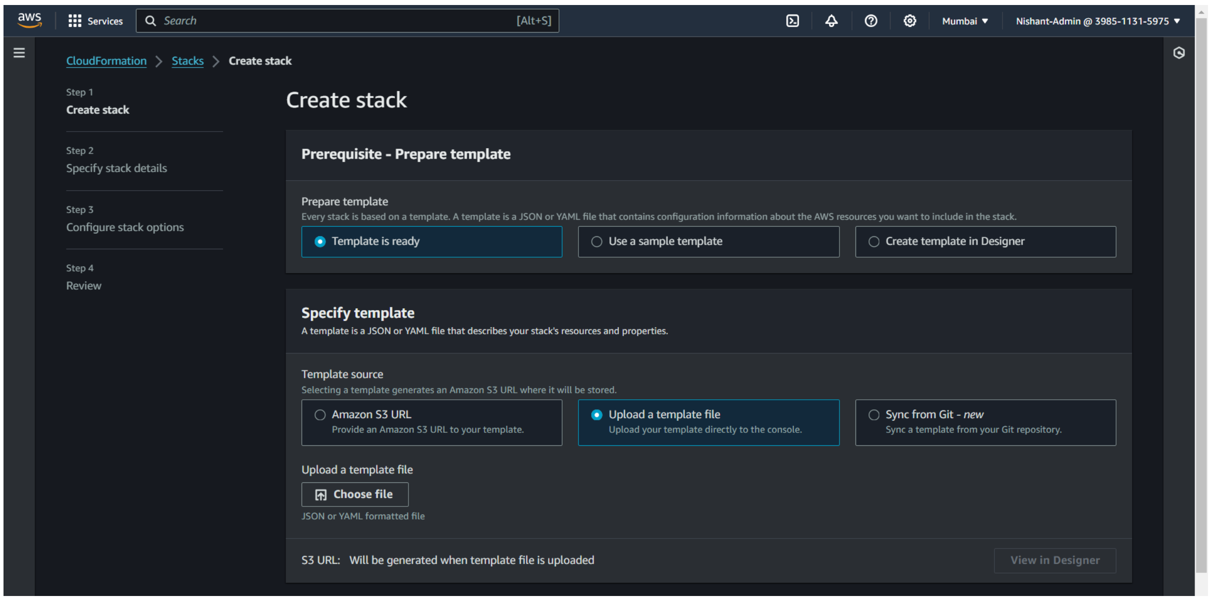
- Give the key pair with the name `VPCE-Tutorial-KeyPair` and then click on create key pair.
- Download and save the keypair.



- Now we have to create the resources in order to create the resources we are going to make use of CloudFormation.
- Go to cloudformation and click on create stack. We are going to use the following template.

```
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
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
  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
      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
        - 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
            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/AWSLambdaBasicExecutionR
ole'
  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']
            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
```


CloudFormation > Stacks > Create stack

Step 1
Create stack

Step 2
Specify stack details

Step 3
Configure stack options

Step 4
Review

Create stack

Prerequisite - Prepare template

Prepare template
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

[✓ Template is ready] Use a sample template Create template in Designer

Specify template
A template is a JSON or YAML file that describes your stack's resources and properties.

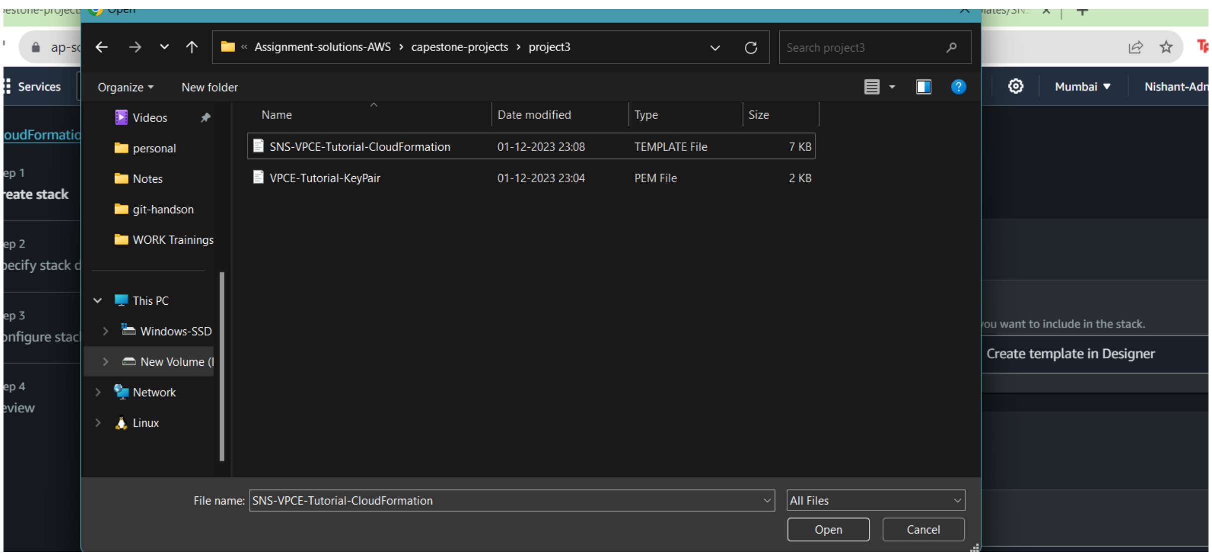
Template source
Selecting a template generates an Amazon S3 URL where it will be stored.

Amazon S3 URL — Provide an Amazon S3 URL to your template.
[✓ Upload a template file — Upload your template directly to the console.]
Sync from Git - *new* — Sync a template from your Git repository.

Upload a template file
Choose file
JSON or YAML formatted file

S3 URL: Will be generated when template file is uploaded

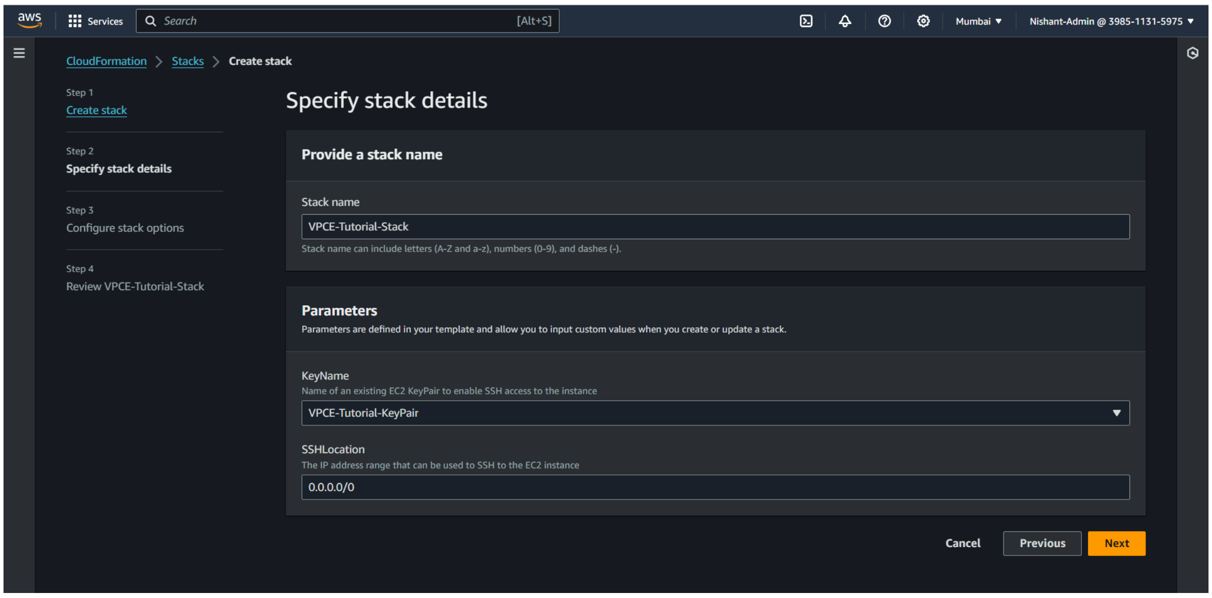
- Upload the template and click on next.


Assignment-solutions-AWS > capestone-projects > project3

Name	Date modified	Type	Size
SNS-VPCE-Tutorial-CloudFormation	01-12-2023 23:08	TEMPLATE File	7 KB
VPCE-Tutorial-KeyPair	01-12-2023 23:04	PEM File	2 KB

File name: SNS-VPCE-Tutorial-CloudFormation
All Files

- Provide the stack a name and Key Pair name, then click next.


CloudFormation > Stacks > Create stack

Step 1
Create stack

Step 2
Specify stack details

Step 3
Configure stack options

Step 4
Review VPCE-Tutorial-Stack

Specify stack details

Provide a stack name

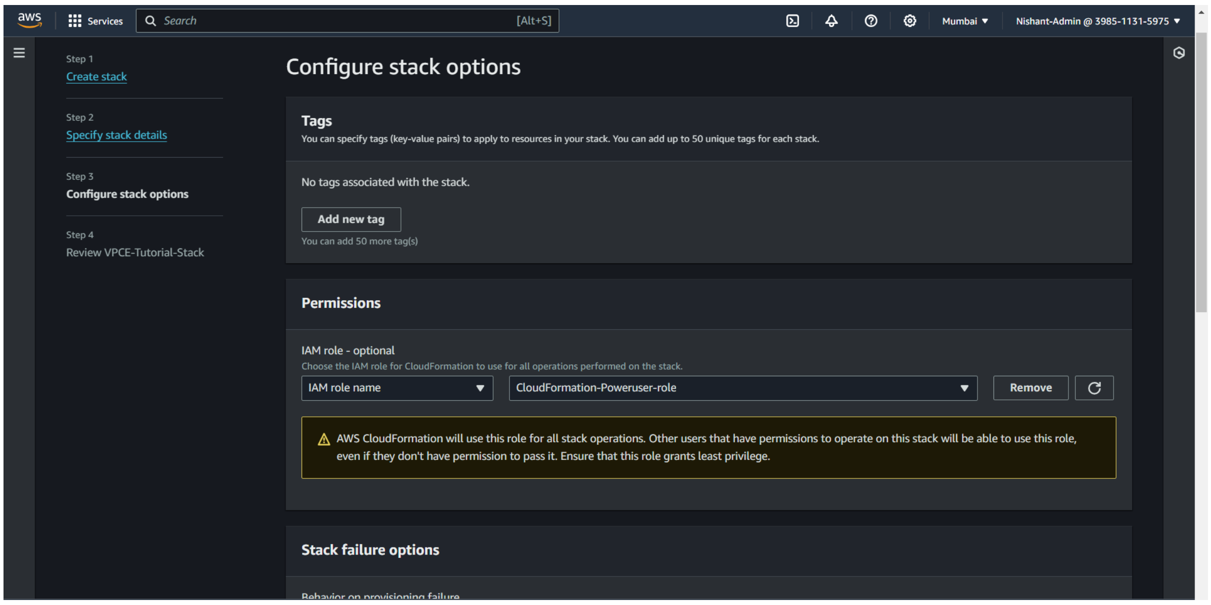
Stack name: VPCE-Tutorial-Stack
Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters
Parameters are defined in your template and allow you to input custom values when you create or update a stack.

KeyName
Name of an existing EC2 KeyPair to enable SSH access to the instance
VPCE-Tutorial-KeyPair

SSHLocation
The IP address range that can be used to SSH to the EC2 instance
0.0.0.0/0

- Provide an Iam role with poweruser permissions.



Step 1
Create stack

Step 2
Specify stack details

Step 3
Configure stack options

Step 4
Review VPCE-Tutorial-Stack

Configure stack options

Tags
You can specify tags (key-value pairs) to apply to resources in your stack. You can add up to 50 unique tags for each stack.

No tags associated with the stack.

Add new tag
You can add 50 more tag(s)

Permissions

IAM role - optional
Choose the IAM role for CloudFormation to use for all operations performed on the stack.

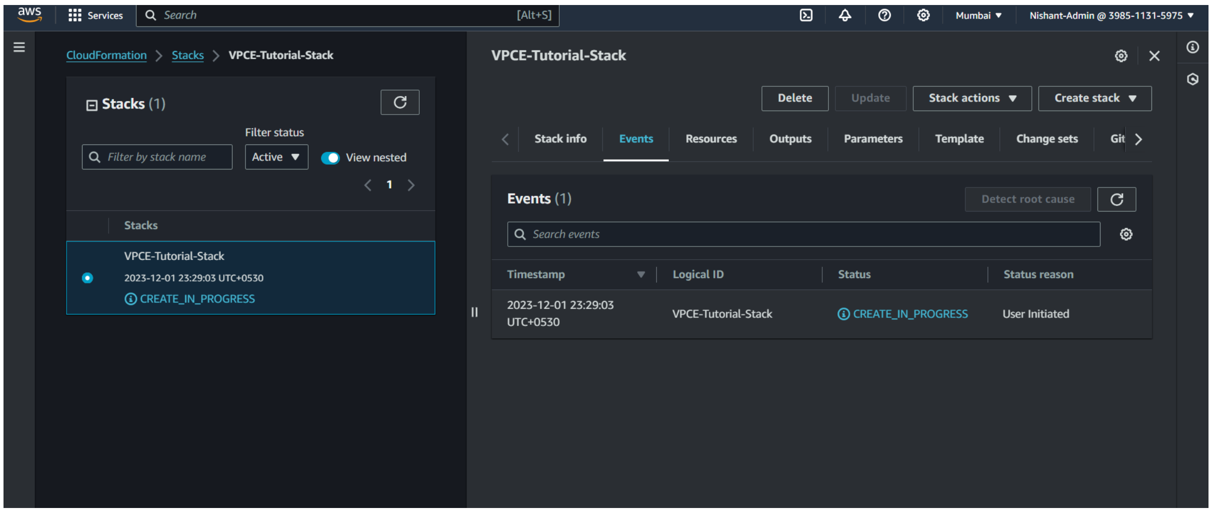
IAM role name | CloudFormation-Poweruser-role | Remove

AWS CloudFormation will use this role for all stack operations. Other users that have permissions to operate on this stack will be able to use this role, even if they don't have permission to pass it. Ensure that this role grants least privilege.

Stack failure options

Behavior on provisioning failure

- Click on next and then review and click on create stack.



Stacks (1)

Filter status: Active | View nested

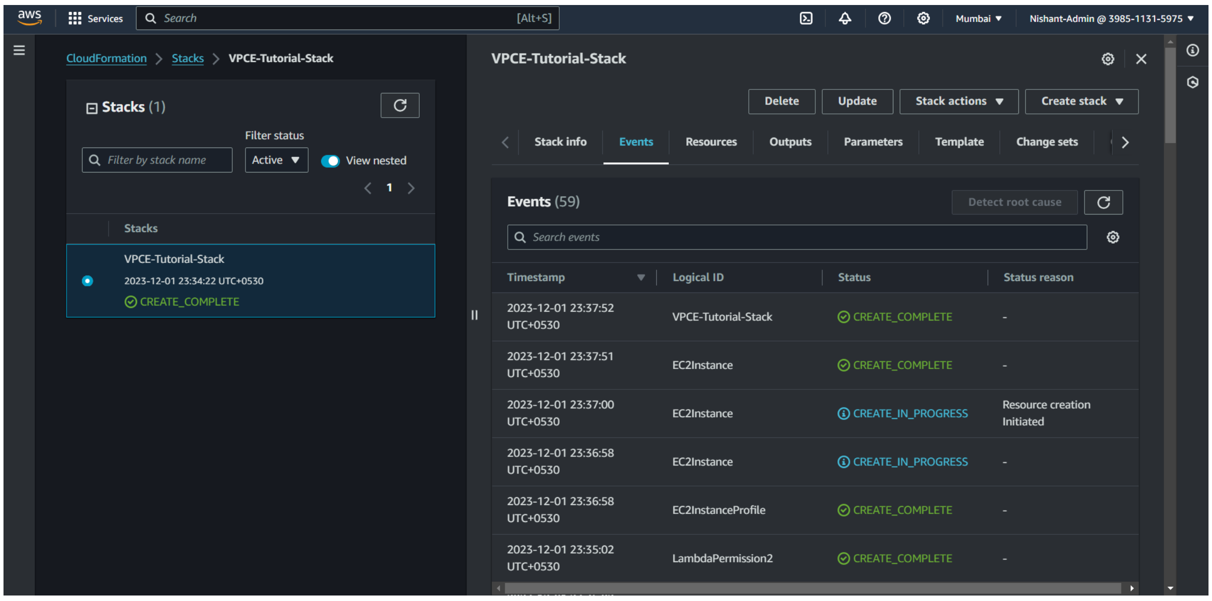
Stacks
VPCE-Tutorial-Stack 2023-12-01 23:29:03 UTC+0530 CREATE_IN_PROGRESS

VPCE-Tutorial-Stack

Events (1)

Timestamp	Logical ID	Status	Status reason
2023-12-01 23:29:03 UTC+0530	VPCE-Tutorial-Stack	CREATE_IN_PROGRESS	User Initiated

- Wait for the stack to be created.



Stacks (1)

Filter status: Active | View nested

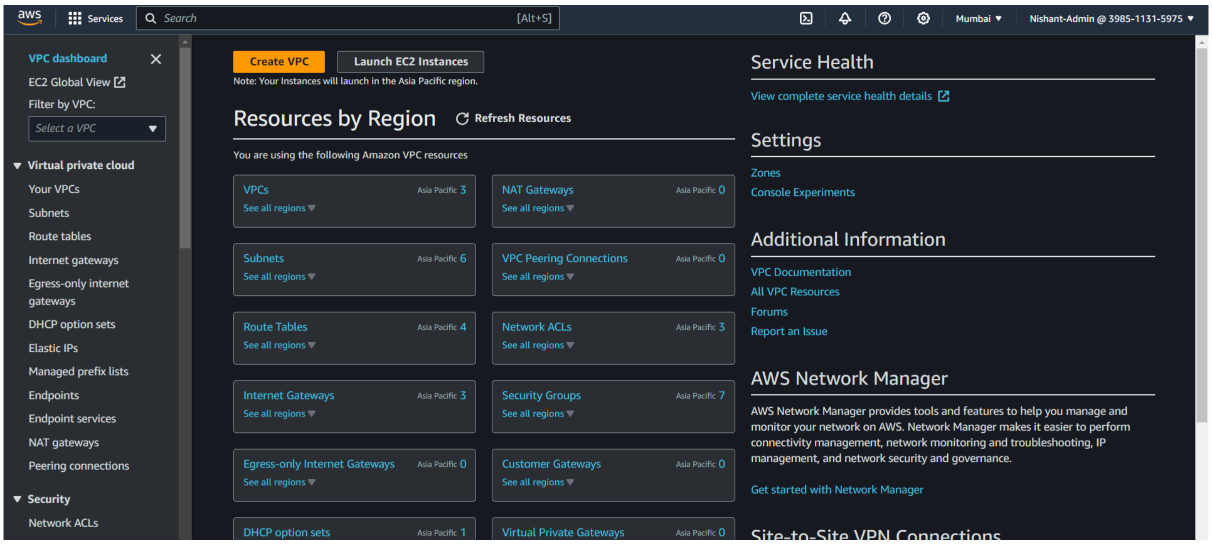
Stacks
VPCE-Tutorial-Stack 2023-12-01 23:34:22 UTC+0530 CREATE_COMPLETE

VPCE-Tutorial-Stack

Events (59)

Timestamp	Logical ID	Status	Status reason
2023-12-01 23:37:52 UTC+0530	VPCE-Tutorial-Stack	CREATE_COMPLETE	-
2023-12-01 23:37:51 UTC+0530	EC2Instance	CREATE_COMPLETE	-
2023-12-01 23:37:00 UTC+0530	EC2Instance	CREATE_IN_PROGRESS	Resource creation Initiated
2023-12-01 23:36:58 UTC+0530	EC2Instance	CREATE_IN_PROGRESS	-
2023-12-01 23:36:58 UTC+0530	EC2InstanceProfile	CREATE_COMPLETE	-
2023-12-01 23:35:02 UTC+0530	LambdaPermission2	CREATE_COMPLETE	-

- Now that we have all the necessary resources created we will create a VPC endpoint so that we can send the message.
- Open VPC.



Create VPC | Launch EC2 Instances

Note: Your Instances will launch in the Asia Pacific region.

Resources by Region — Refresh Resources

You are using the following Amazon VPC resources

VPCs — Asia Pacific 3	NAT Gateways — Asia Pacific 0
Subnets — Asia Pacific 6	VPC Peering Connections — Asia Pacific 0
Route Tables — Asia Pacific 4	Network ACLs — Asia Pacific 3
Internet Gateways — Asia Pacific 3	Security Groups — Asia Pacific 7
Egress-only Internet Gateways — Asia Pacific 0	Customer Gateways — Asia Pacific 0
DHCP option sets — Asia Pacific 1	Virtual Private Gateways — Asia Pacific 0

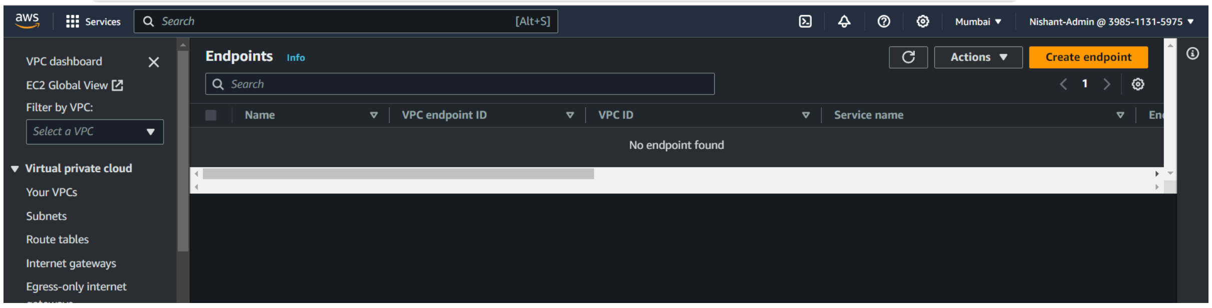
Service Health
View complete service health details

Settings
Zones
Console Experiments

Additional Information
VPC Documentation
All VPC Resources
Forums
Report an Issue

AWS Network Manager
AWS Network Manager provides tools and features to help you manage and monitor your network on AWS. Network Manager makes it easier to perform connectivity management, network monitoring and troubleshooting, IP management, and network security and governance.
Get started with Network Manager

- On the left hand side we will select Endpoints.

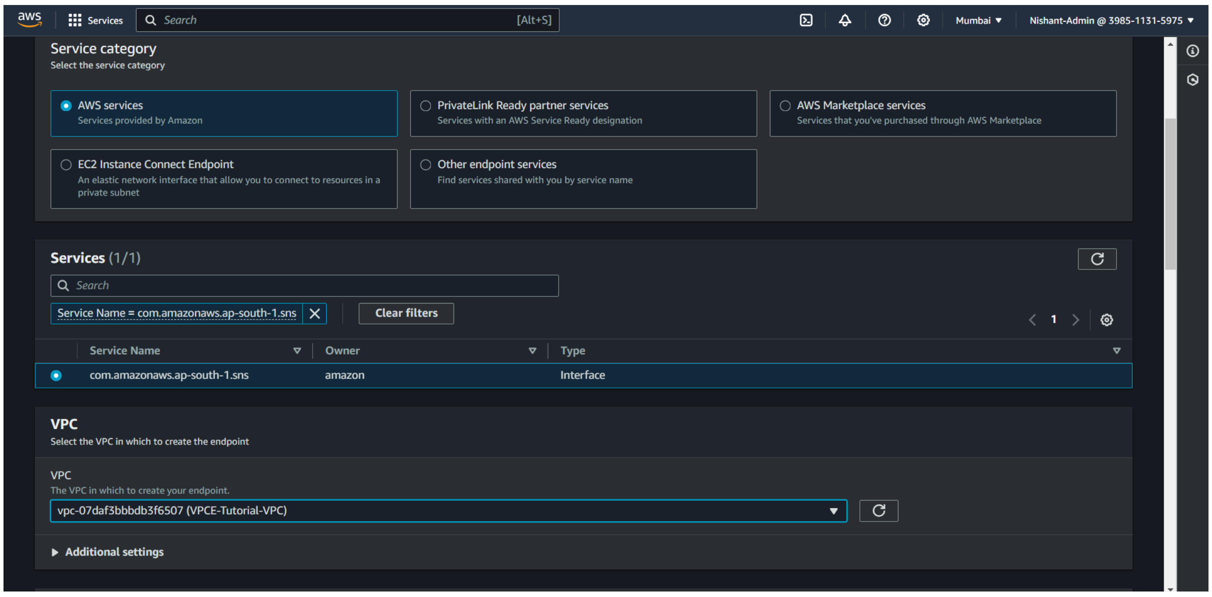


Endpoints Info

Name	VPC endpoint ID	VPC ID	Service name

No endpoint found

- Click on create endpoint.
- Select the SNS service and select our VPC.



Service category
Select the service category

- [✓ AWS services] Services provided by Amazon
- PrivateLink Ready partner services — Services with an AWS Service Ready designation
- AWS Marketplace services — Services that you've purchased through AWS Marketplace
- EC2 Instance Connect Endpoint — An elastic network interface that allow you to connect to resources in a private subnet
- Other endpoint services — Find services shared with you by service name

Services (1/1)

Service Name = com.amazonaws.ap-south-1.sns | Clear filters

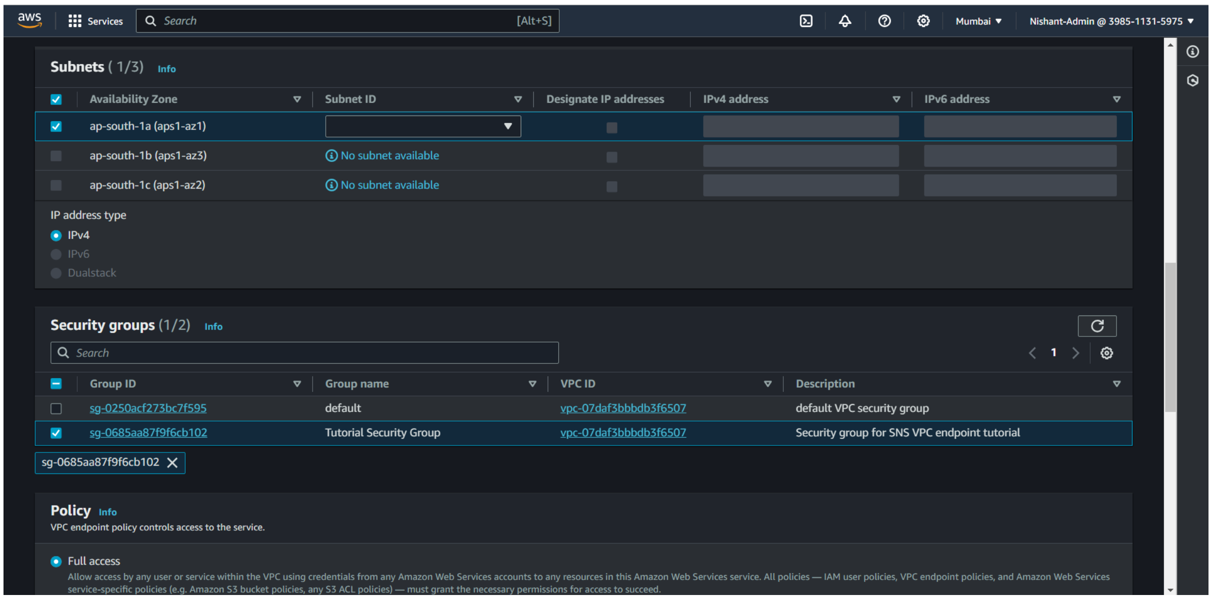
Service Name	Owner	Type
● com.amazonaws.ap-south-1.sns	amazon	Interface

VPC
Select the VPC in which to create the endpoint

VPC
The VPC in which to create your endpoint.
vpc-07daf3bbbdb3f6507 (VPCE-Tutorial-VPC)

Additional settings

- Select the private subnet and select the project security group.



Subnets (1/3) Info

✓	Availability Zone	Subnet ID	Designate IP addresses	IPv4 address	IPv6 address
✓	ap-south-1a (aps1-az1)				
	ap-south-1b (aps1-az3)	No subnet available			
	ap-south-1c (aps1-az2)	No subnet available			

IP address type
- [✓ IPv4]
- IPv6
- Dualstack

Security groups (1/2) Info

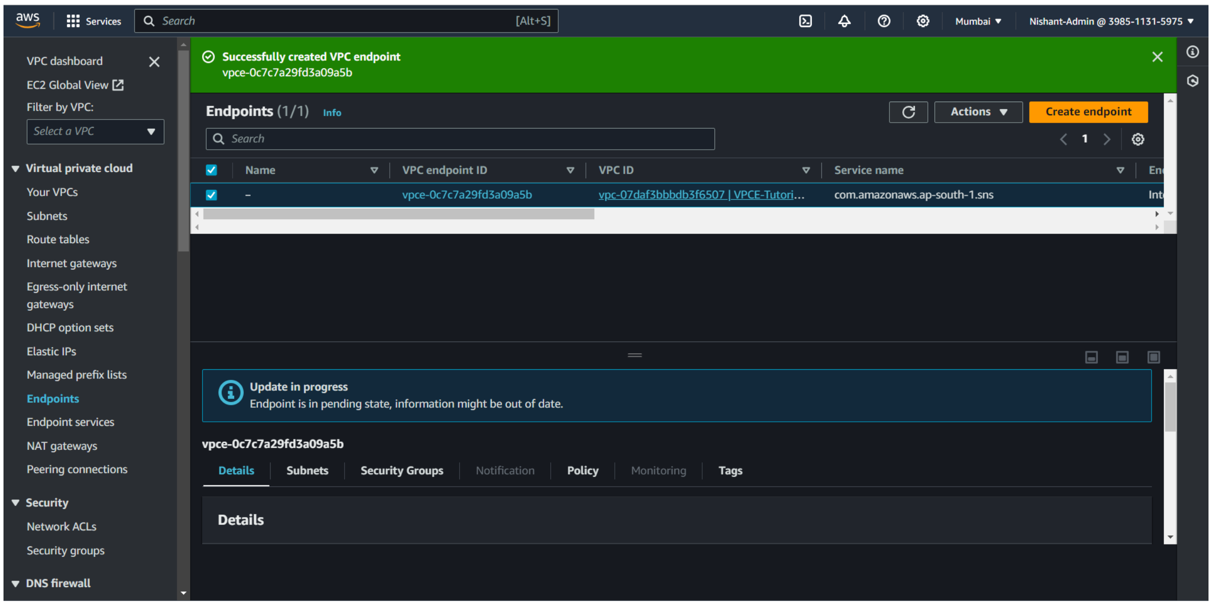
	Group ID	Group name	VPC ID	Description
	sg-0250acf273bc7f595	default	vpc-07daf3bbbdb3f6507	default VPC security group
✓	sg-0685aa87f9f6cb102	Tutorial Security Group	vpc-07daf3bbbdb3f6507	Security group for SNS VPC endpoint tutorial

sg-0685aa87f9f6cb102

Policy Info
VPC endpoint policy controls access to the service.

[✓ Full access]
Allow access by any user or service within the VPC using credentials from any Amazon Web Services accounts to any resources in this Amazon Web Services service. All policies — IAM user policies, VPC endpoint policies, and Amazon Web Services service-specific policies (e.g. Amazon S3 bucket policies, any S3 ACL policies) — must grant the necessary permissions for access to succeed.

- Click on create ENDPOINT.



Successfully created VPC endpoint
vpce-0c7c7a29fd3a09a5b

Endpoints (1/1) Info

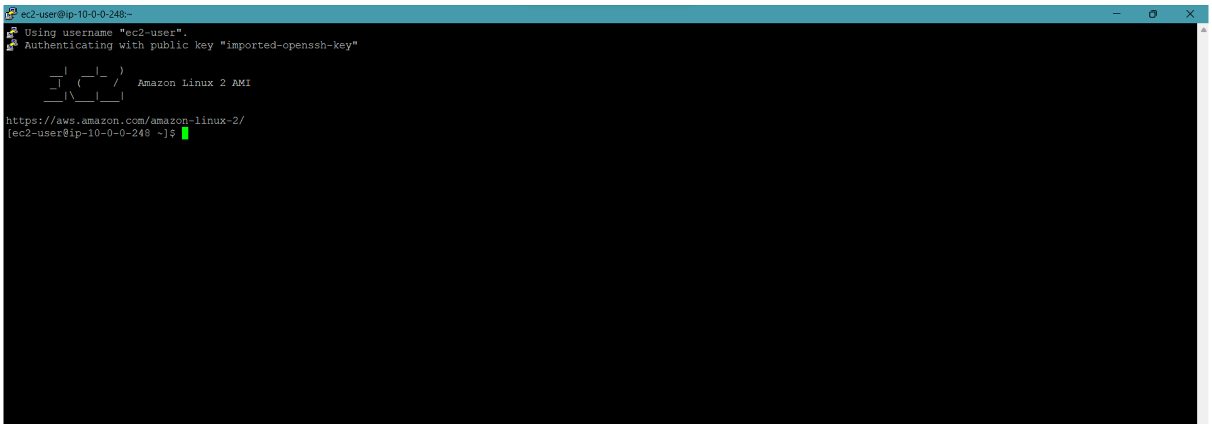
✓	Name	VPC endpoint ID	VPC ID	Service name	En
✓	–	vpce-0c7c7a29fd3a09a5b	vpc-07daf3bbbdb3f6507 | VPCE-Tutori...	com.amazonaws.ap-south-1.sns	Int

Update in progress
Endpoint is in pending state, information might be out of date.

vpce-0c7c7a29fd3a09a5b

Details

- Now we will publish the message to the SNS topic to do that we will first ssh into the ec2 instance.

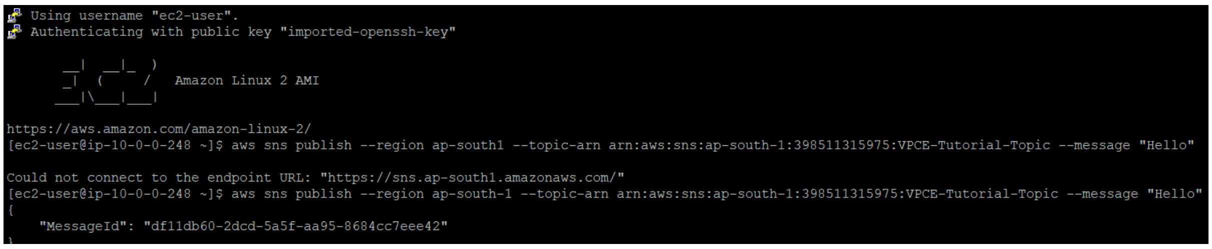


```
ec2-user@ip-10-0-0-248:~
Using username "ec2-user".
Authenticating with public key "imported-openssh-key"

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-0-248 ~]$
```

- Now we will try to send SNS message from the EC2 instance.

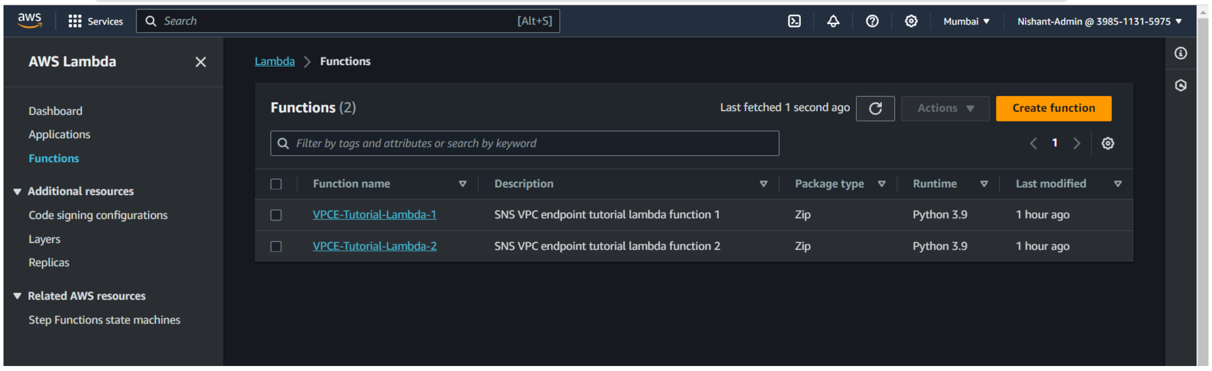
- We have to use this instance.

aws sns publish --region ap-south-1 --topic-arn arn:aws:sns:ap-south-1:398511315975:VPCE-Tutorial-Topic --message "Hello"



```
Using username "ec2-user".
Authenticating with public key "imported-openssh-key"

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-0-248 ~]$ aws sns publish --region ap-south1 --topic-arn arn:aws:sns:ap-south-1:398511315975:VPCE-Tutorial-Topic --message "Hello"

Could not connect to the endpoint URL: "https://sns.ap-south1.amazonaws.com/"
[ec2-user@ip-10-0-0-248 ~]$ aws sns publish --region ap-south-1 --topic-arn arn:aws:sns:ap-south-1:398511315975:VPCE-Tutorial-Topic --message "Hello"
{
    "MessageId": "df11db60-2dcd-5a5f-aa95-8684cc7eee42"
```

- Once we have sent the message we can check the and verify if the message has been delivered.
- Open the AWS Lambda console at https://console.aws.amazon.com/lambda/.

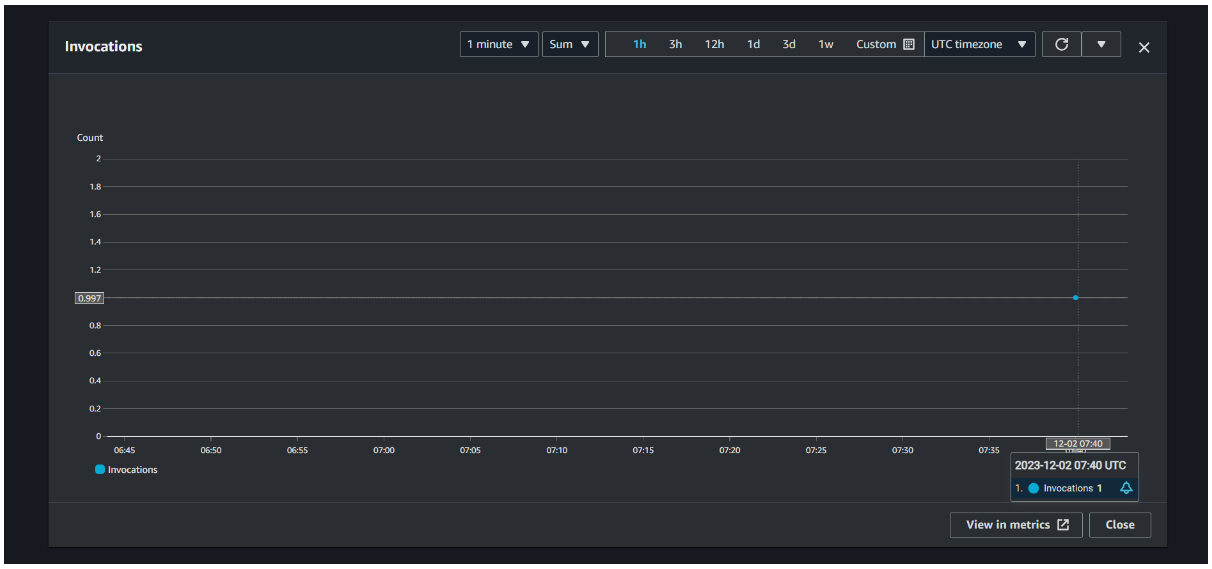


Functions (2)

Function name	Description	Package type	Runtime	Last modified
VPCE-Tutorial-Lambda-1	SNS VPC endpoint tutorial lambda function 1	Zip	Python 3.9	1 hour ago
VPCE-Tutorial-Lambda-2	SNS VPC endpoint tutorial lambda function 2	Zip	Python 3.9	1 hour ago

- On the Functions page, choose VPCE-Tutorial-Lambda-1.
- Choose Monitoring.
- Check the Invocation count graph. This graph shows the number of times that the Lambda function has been run.

  The invocation count matches the number of times you published a message to the topic.



- Our message has successfully invoked the graph once, that means we have received the SNS message successfully.