

Publishing Amazon SNS Messages Privately

NAME: VIKRAM

Industry: Healthcare

Problem Statement:

How to secure patient records online and send them privately to the intended party.

In this project, a hospital's patient reporting system is deployed using AWS services to publish patient reports securely and privately through Amazon SNS. The message publication happens within a private VPC, ensuring secure communication and controlled access through IAM roles.

AWS Services Used

1. AWS CloudFormation – to automate infrastructure setup (VPC, Subnets, EC2, SNS, S3).
2. Amazon EC2 – for hosting and running the application.
3. Amazon SNS – to securely publish messages (notifications).
4. Amazon S3 – for storing reports or data securely.
5. IAM – for defining access permissions and roles.

Architecture Overview

- A VPC is created with public and private subnets.
- EC2 instance is launched inside the VPC with an IAM Role attached.
- SNS topic is created for report notifications.
- A VPC endpoint connects SNS privately within the VPC.
- Reports are uploaded to S3, and SNS notifies via email securely.

Phase 1 – Create AWS resources using CloudFormation

1. Open AWS Management Console → CloudFormation → Create Stack → With new resources.
2. Upload your CloudFormation template (YAML/JSON) defining all resources.

AWSTemplateFormatVersion: 2010-09-09

Description: Healthcare Project - Secure Patient Report Sharing with SNS and EC2 (Private VPC)

Parameters:

KeyName:

Type: String

Default: vik-87

Description: EC2 Key Pair name for SSH access

AmiId:

Type: String

Default: ami-0bdd88bd06d16ba03

Description: AMI ID for EC2 instance

S3BucketName:

Type: String

Default: vikky-s3-demo-3rd-nov

Description: S3 bucket name for storing reports

Resources:

VPC + Networking

MyVPC:

Type: AWS::EC2::VPC

Properties:

CidrBlock: 10.0.0.0/16

EnableDnsSupport: true

EnableDnsHostnames: true

Tags:

- Key: Name

Value: HealthcareVPC

InternetGateway:

Type: AWS::EC2::InternetGateway

Properties:

Tags:

- Key: Name

Value: HealthcareIGW

VPCGatewayAttachment:

Type: AWS::EC2::VPCGatewayAttachment

Properties:

InternetGatewayId: !Ref InternetGateway

VpcId: !Ref MyVPC

PublicSubnet:

Type: AWS::EC2::Subnet

Properties:

VpcId: !Ref MyVPC

CidrBlock: 10.0.1.0/24

MapPublicIpOnLaunch: true

AvailabilityZone: !Select [0, !GetAZs ""]

Tags:

- Key: Name

Value: PublicSubnet

RouteTable:

Type: AWS::EC2::RouteTable

Properties:

VpcId: !Ref MyVPC

Route:

Type: AWS::EC2::Route

DependsOn: VPCGatewayAttachment

Properties:

RouteTableId: !Ref RouteTable

DestinationCidrBlock: 0.0.0.0/0

GatewayId: !Ref InternetGateway

SubnetRouteTableAssociation:

Type: AWS::EC2::SubnetRouteTableAssociation

Properties:

SubnetId: !Ref PublicSubnet

RouteTableId: !Ref RouteTable

Security Group

InstanceSecurityGroup:

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: Enable SSH and HTTP

VpcId: !Ref MyVPC

SecurityGroupIngress:

- IpProtocol: tcp

FromPort: 22

ToPort: 22

CidrIp: 0.0.0.0/0

- IpProtocol: tcp

FromPort: 80

ToPort: 80

CidrIp: 0.0.0.0/0

EC2 Role + Policy

EC2Role:

Type: AWS::IAM::Role

Properties:

AssumeRolePolicyDocument:

Version: "2012-10-17"

Statement:

- Effect: Allow

Principal:

Service: ec2.amazonaws.com

Action: sts:AssumeRole

Policies:

- PolicyName: EC2SNSAccess

PolicyDocument:

Version: "2012-10-17"

Statement:

- Effect: Allow

Action:

- sns:Publish

- sns:ListTopics

Resource: "*"

- Effect: Allow

Action:

- s3:PutObject

- s3:GetObject

Resource: !Sub arn:aws:s3:::\${S3BucketName}/*

InstanceProfile:

Type: AWS::IAM::InstanceProfile

Properties:

Roles:

- !Ref EC2Role

EC2 Instance (Free Tier Eligible)

EC2Instance:

Type: AWS::EC2::Instance

Properties:

InstanceType: t3.micro

ImageId: !Ref AmiId

KeyName: !Ref KeyName

SubnetId: !Ref PublicSubnet

SecurityGroupIds:

- !Ref InstanceSecurityGroup

IamInstanceProfile: !Ref InstanceProfile

UserData:

Fn::Base64: !Sub |

#!/bin/bash

yum update -y

yum install -y aws-cli httpd

systemctl start httpd

systemctl enable httpd

echo "<h1>Healthcare SNS Report System is Running</h1>" >
/var/www/html/index.html

Tags:

- Key: Name

Value: HealthcareEC2

SNS Interface VPC Endpoint (PrivateLink)

SnsVpcEndpoint:

Type: AWS::EC2::VPCEndpoint

Properties:

VpcId: !Ref MyVPC

ServiceName: !Sub com.amazonaws.\${AWS::Region}.sns

VpcEndpointType: Interface

SubnetIds:

- !Ref PublicSubnet

SecurityGroupIds:

- !Ref InstanceSecurityGroup

PrivateDnsEnabled: true

Outputs:

VPCId:

Value: !Ref MyVPC

Description: ID of the VPC

EC2InstancePublicIP:

Value: !GetAtt EC2Instance.PublicIp

Description: Public IP of EC2 instance

SnsEndpointId:

Value: !Ref SnsVpcEndpoint

Description: SNS VPC Endpoint ID

3. Click Next → Next → Create Stack.

4. Wait until all resources show status “CREATE_COMPLETE”.

The screenshot shows the AWS CloudFormation console interface during the 'Prepare template' step of creating a new stack. The breadcrumb navigation at the top indicates the path: CloudFormation > Stacks > Create stack. On the left, a sidebar shows the progress of the setup steps: 'Specify stack details' (completed), 'Step 3: Configure stack options' (current step), 'Step 4: Review and create' (upcoming), and 'Review and create' (upcoming). The main content area is titled 'Prerequisite – Prepare template' and includes a link to the 'laC generator'. Below this, there are two options for preparing the template: 'Choose an existing template' (selected) and 'Build from Infrastructure Composer'. The 'Specify template' section provides information about template sources and offers three options: 'Amazon S3 URL', 'Upload a template file' (selected), and 'Sync from Git'. The 'Upload a template file' option is active, showing a file selection button and a text input field containing the filename 'hospital-sns-private-setup-vik.yml'. At the bottom, the generated S3 URL is displayed.

CloudFormation > Stacks > Create stack

Specify stack details

Step 3

Configure stack options

Step 4

Review and create

Prerequisite – Prepare template

You can also create a template by scanning your existing resources in the [laC generator](#).

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.

☒ Choose an existing template
Upload or choose an existing template.

☐ Build from Infrastructure Composer
Create a template using a visual builder.

Specify template Info

This [GitHub repository](#) contains sample CloudFormation templates that can help you get started on new infrastructure projects. [Learn more](#)

Template source

Selecting a template generates an Amazon S3 URL where it will be stored. A template is a JSON or YAML file that describes your stack's resources and properties.

☐ 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
Sync a template from your Git repository.

Upload a template file

[Choose file](#)

hospital-sns-private-setup-vik.yml

JSON or YAML formatted file

S3 URL: <https://s3.us-east-1.amazonaws.com/cf-templates-1e04a1hyej8uf-us-east-1/2025-11-03T031157.109253w-hospital-sns-private-setup-vik.yml>

CloudFormation > Stacks > Create stack

Step 3
Configure stack options

Step 4
Review and create

Stack name

sns-demo-stack

Stack name must contain only letters (a-z, A-Z), numbers (0-9) and hyphens (-) and start with a letter. Max 128 characters. Character count: 14/128.

Parameters

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

AmiId

AMI ID for EC2 instance

ami-0bdd88bd06d16ba03

KeyName

EC2 Key Pair name for SSH access

vik-87

S3BucketName

S3 bucket name for storing reports

vikky-s3-demo-3rd-nov

Cancel

Previous

Next

CloudShell

Feedback

© 2025 Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

CloudFormation > Stacks

Stacks (1)

Search by stack name

Filter status

Active

View nested

Stack name	Status	Created time	Description
sns-private-stack	CREATE_COMPLETE	2025-11-03 08:53:04 UTC+0530	Healthcare Project - Secure Patient Report Sharing with SNS and EC2 (Private VPC)

CloudFormation > Stacks > sns-private-stack

Stacks (1)

Filter status

Active

View nested

sns-private-stack

2025-11-03 08:53:04 UTC+0530

CREATE_COMPLETE

sns-private-stack

Delete

Update stack

Stack actions

Create stack

Stack info

Events

Resources

Outputs

Parameters

Template

Changesets

Git sync

Overview

Stack ID

arn:aws:cloudformation:us-east-1:062250062838:stack/sns-private-stack/69101f30-b864-11f0-8fbc-0ed0135cae3f

Description

Healthcare Project - Secure Patient Report Sharing with SNS and EC2 (Private VPC)

Status

CREATE_COMPLETE

Detailed status

-

Status reason

-

Root stack

-

Parent stack

-

Created time

2025-11-03 08:53:04 UTC+0530

Updated time

-

Deleted time

-

Drift status

NOT_CHECKED

VERIFY STACK OUTPUTS

- Open AWS Console → **CloudFormation** → **Stacks**

- Select your stack → open the **Outputs** tab

You will see 3 values:

• Output Name	• Description	• Example value
• VPCId	• ID of the VPC	• vpc-0a1b2c3d4e5f6g7h
• EC2InstancePublicIP	• Public IP to access EC2	• 13.232.98.45
• SnsEndpointId	• ID of your SNS VPC Endpoint	• vpce-0a12bc34def56789

The screenshot shows the AWS CloudFormation console interface. On the left, a sidebar lists stacks, with 'sns-private-stack' selected and showing a 'CREATE_COMPLETE' status. The main panel displays the 'Outputs' tab for this stack. At the top of the main panel, there are buttons for 'Delete', 'Update stack', 'Stack actions', and 'Create stack'. Below these are tabs for 'Stack info', 'Events', 'Resources', 'Outputs' (which is active), 'Parameters', 'Template', 'Changesets', and 'Git sync'. The 'Outputs' tab shows a table with 3 outputs. The table has columns for 'Key', 'Value', 'Description', and 'Export name'. The outputs listed are 'EC2InstancePublicIP' with value '54.237.204.13', 'SnsEndpointId' with value 'vpce-0e8ed9570ce7fe63a', and 'VPCId' with value 'vpc-05e79cd01567d07e2'.

Key	Value	Description	Export name
EC2InstancePublicIP	54.237.204.13	Public IP of EC2 instance	-
SnsEndpointId	vpce-0e8ed9570ce7fe63a	SNS VPC Endpoint ID	-
VPCId	vpc-05e79cd01567d07e2	ID of the VPC	-

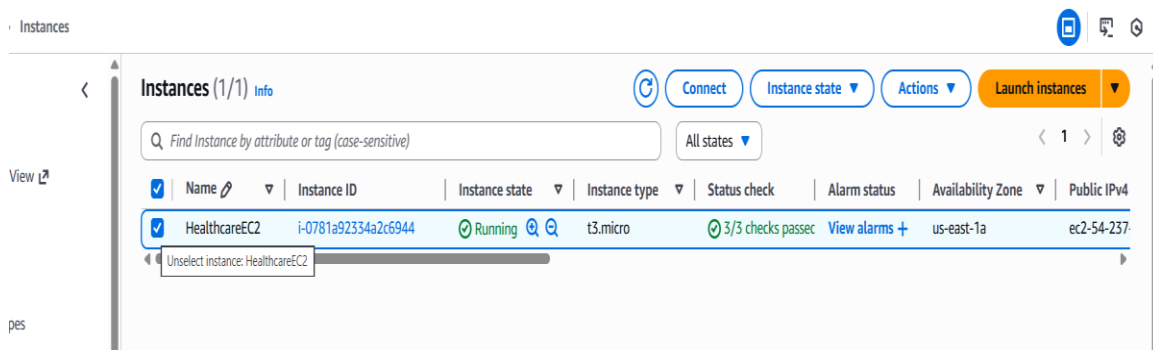
PHASE 2 — CONNECT TO EC2 INSTANCE

❑ Go to EC2 → Instances

❑ Select instance named HealthcareEC2

❑ Click Connect → EC2 Instance Connect (browser-based) → Connect

You'll now see a Linux terminal prompt.



PHASE 3 — VERIFY EC2 ENVIRONMENT

Verify Apache Web Server

Run:

```
sudo systemctl status httpd
```

You should see:

```
active (running)
```

Then open your web page:

Browser URL:

```
http://<EC2InstancePublicIP>
```

Expected Output:

Healthcare SNS Report System is Running



```
"InstanceProfileArn" : "arn:aws:iam::123456789012:instance-profile/EC2-SNS-S3-Role",  
  
"InstanceProfileId" : "AIPAXXXXXXXXXXXXXXXXXX"  
  
}
```

This confirms that the EC2 instance has a valid IAM Role attached with the correct permissions for accessing Amazon SNS and S3.

```
[ec2-user@ip-10-0-1-185 ~]$ curl http://169.254.169.254/latest/meta-data/iam/info  
{  
  "Code" : "Success",  
  "LastUpdated" : "2025-11-03T03:38:12Z",  
  "InstanceProfileArn" : "arn:aws:iam::062250062838:instance-profile/EC2-SNS-S3-Role",  
  "InstanceProfileId" : "AIPAQ47TESP3IR6JWM3WB"  
}[ec2-user@ip-10-0-1-185 ~]$
```

⚠ If the Command Shows No Output

If the command returns nothing, it usually means your EC2 instance does not yet have an IAM Role attached — so there's no IAM metadata to show.

Follow these exact steps to fix it 

Create and Attach an IAM Role

1. Go to AWS Management Console → IAM → Roles
2. Click Create role
3. Under Trusted entity type, select AWS service

4. Choose EC2 and click Next
5. In permissions, search and check:
 - [AmazonS3FullAccess](#)
 - [AmazonSNSFullAccess](#)
 - [AmazonEC2ReadOnlyAccess](#)
6. Click Next
7. Name the role: EC2-SNS-S3-Role
8. Click Create role

Now attach this role to your instance:

1. Go to EC2 Console → Instances
2. Select your instance
3. Click Actions → Security → Modify IAM role
4. Choose EC2-SNS-S3-Role → click Update IAM role

If Still No Output — Verify Metadata Options

If there's still no response from the curl command even after attaching the role, check your EC2 metadata service settings.

1. Go to EC2 → Instances → Select your instance
2. Scroll down to Metadata options

Ensure the following settings are correct:

Setting	Required Value
Access to instance metadata	Enabled
IMDSv2 required	No (optional)
Hop limit	1

If IMDSv2 is set to “Yes”, modify it:

- Click Actions → Instance settings → Edit instance metadata options
 - Set:
 - Access to instance metadata: Enable
 - IMDSv2 required: No
 - Hop limit: 1
 - Save changes
-

Re-run Verification Command

After saving the metadata changes, wait ~30 seconds and run again:

`curl http://169.254.169.254/latest/meta-data/iam/info`

Expected Result: {

A JSON response with your IAM Role details confirms proper IAM and metadata configuration.

```
[ec2-user@ip-10-0-1-185 ~]$ curl http://169.254.169.254/latest/meta-data/iam/info
{
  "Code" : "Success",
  "LastUpdated" : "2025-11-03T03:38:12Z",
  "InstanceProfileArn" : "arn:aws:iam::062250062838:instance-profile/EC2-SNS-S3-Role",
  "InstanceProfileId" : "AIPAQ47TESP3IR6JWM3WB"
}[ec2-user@ip-10-0-1-185 ~]$
```

PHASE 5 — VERIFY SNS PRIVATE CONNECTION (VPC ENDPOINT)

List all SNS Topics

Run:

`aws sns list-topics`

Expected output:

```
{  
  "Topics": []  
}
```

(Empty array is fine — means connection is successful but no topics yet.)

If no error appears → SNS PrivateLink is working.

PHASE 6 — CREATE SNS TOPIC

Create a Private SNS Topic

Run:

```
aws sns create-topic --name health-topic
```

Output example:

```
{  
  "TopicArn": "arn:aws:sns:us-east-1:062250062838:health-topic"  
}
```

Copy that **Topic ARN** — we'll use it in the next step.

```
[ec2-user@ip-10-0-1-185 ~]$ aws sns list-topics  
{  
  "Topics": []  
}  
[ec2-user@ip-10-0-1-185 ~]$ aws sns create-topic --name health-topic  
{  
  "TopicArn": "arn:aws:sns:us-east-1:062250062838:health-topic"  
}  
[ec2-user@ip-10-0-1-185 ~]$ █
```

PHASE 7 — PUBLISH A TEST MESSAGE

Send a Message to the Topic

Run the following command inside your EC2 terminal:

```
aws sns publish \  
  
--topic-arn arn:aws:sns:us-east-1:062250062838:health-topic \  
  
--message "Patient report uploaded successfully" \  
  
--subject "Hospital Report Update"
```

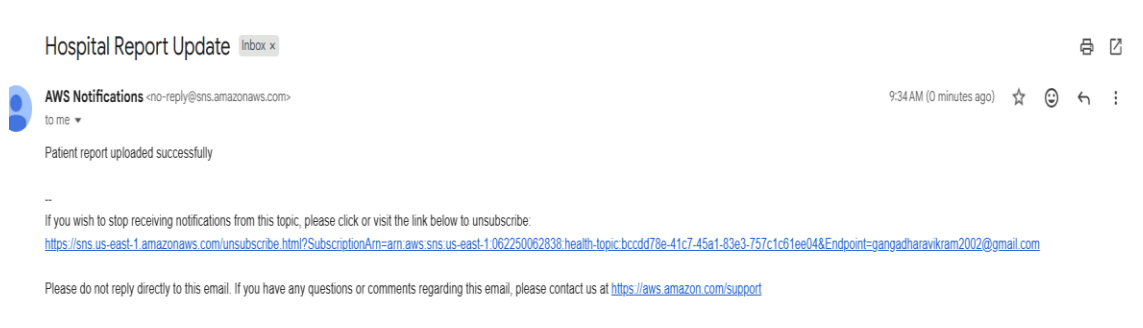
Example Output:

```
{  
  
  "MessageId": "b1a2c3d4-5678-9012-3456-7890abcd1234"  
  
}
```

If you see a MessageId, your message was successfully published privately through the VPC Endpoint.

Within a few seconds, you should receive an email notification from Amazon SNS containing the message.

```
[ec2-user@ip-10-0-1-185 ~]$ aws sns publish \  
> --topic-arn arn:aws:sns:us-east-1:062250062838:health-topic \  
> --message "Patient report uploaded successfully" \  
> --subject "Hospital Report Update"  
{  
  "MessageId": "fcddd7e7-f03b-5260-b6f8-b665db23356e"  
}  
[ec2-user@ip-10-0-1-185 ~]$
```



If You Do Not Receive the Email

If no email arrives after running the publish command, it means your SNS topic subscription is not yet confirmed.

Go to AWS Console → SNS → Topics → **health-topic**

- If you see your subscription status as “Pending confirmation”, open your email inbox and click the confirmation link sent by AWS.
- If you don’t see any subscription listed, create one and then confirm it through the email you receive.

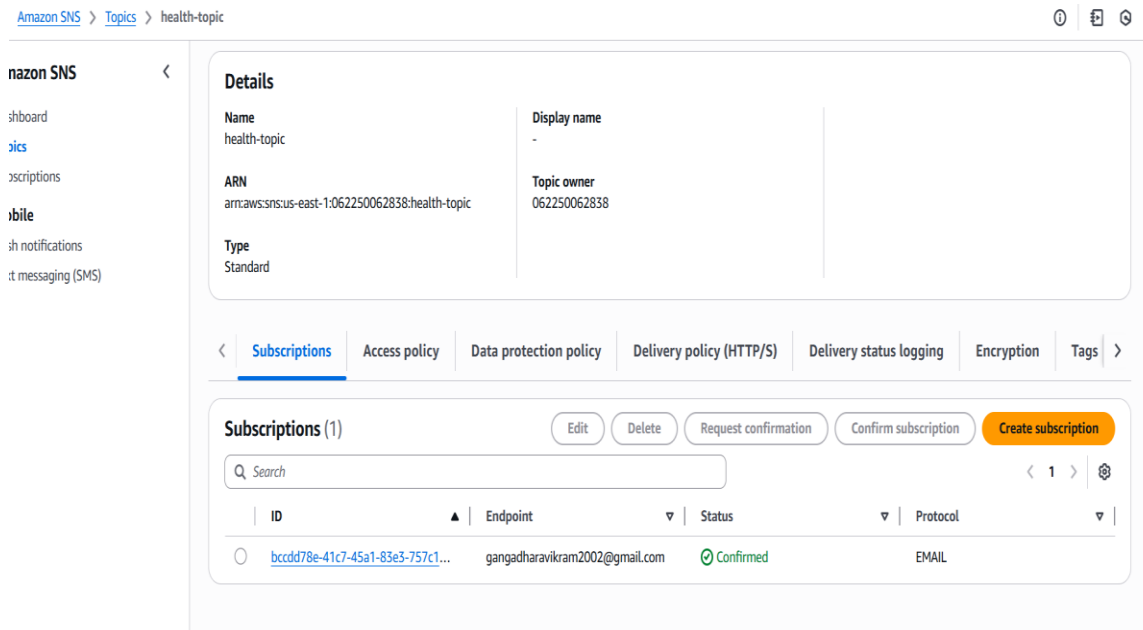
After confirming the subscription, re-run the same publish command.
You will then receive the email successfully.

PHASE 8 — VERIFY IN SNS CONSOLE

Check Message Delivery

- Go to AWS Console → SNS → Topics
- Click on health-topic
- Under Details, verify:
 - Topic ARN matches your CLI output
 - The topic exists (no red errors)

This confirms the EC2 instance published securely through your private SNS VPC Endpoint.



PHASE 9 — VERIFY S3 ACCESS (SECURE FILE UPLOAD)

Upload a Sample Report to S3 from EC2

Run:

```
echo "Patient Report - Private Upload Test" > report.txt
```

```
aws s3 cp report.txt s3://bucket name/
```

Expected output:

```
upload: ./report.txt to s3://s3-demo-3rd-nov/report.txt
```

Then verify:

```
aws s3 ls s3:// s3-demo-3rd-nov //
```

You should see:

```
2025-11-03 12:45:01    37 report.txt
```

```
[ec2-user@ip-10-0-1-185 ~]$ aws s3 ls s3://s3-demo-3rd-nov/
[ec2-user@ip-10-0-1-185 ~]$ echo "Patient Report - Private Upload Test" > report.txt
[ec2-user@ip-10-0-1-185 ~]$ aws s3 cp report.txt s3://s3-demo-3rd-nov/
upload: ./report.txt to s3://s3-demo-3rd-nov/report.txt
[ec2-user@ip-10-0-1-185 ~]$ aws s3 ls s3://s3-demo-3rd-nov/

2025-11-03 04:16:10    37 report.txt
[ec2-user@ip-10-0-1-185 ~]$
```

Amazon S3 > Buckets > s3-demo-3rd-nov

s3-demo-3rd-nov Info

Objects | Metadata | Properties | Permissions | Metrics | Management | Access Points

Objects (1) Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	report.txt	txt	November 3, 2025, 09:46:10 (UTC+05:30)	37.0 B	Standard

PHASE 10 — SECURE PRIVATE SNS COMMUNICATION (INSIDE VPC)

Go to the VPC Console and Verify the Endpoint

1. Open AWS Console → VPC → Endpoints
2. Locate your SNS VPC Endpoint in the list.
3. Click on it to open its details page.
4. Under the details section, copy your VPC Endpoint ID
5. Confirm that the Status shows Available and the Type is Interface.

VPC > Endpoints

Endpoints (1/1) Info Actions Create endpoint

Find endpoints by attribute or tag

<input checked="" type="checkbox"/>	Name	VPC endpoint ID	Endpoint type	Status	Service name
<input checked="" type="checkbox"/>	sns-vpc-endpoint	vpce-0e8ed9570ce7fe63a	Interface	Available	com.amazonaws.us-east-

Verify the Endpoint from EC2

From your EC2 terminal, verify the SNS VPC Endpoint status using the copied ID:

```
aws ec2 describe-vpc-endpoints --vpc-endpoint-ids vpce-0a1b2c3d4e5f6g7h8
```

Expected Output:

```
{
  "VpcEndpoints": [
    {
      "VpcEndpointId": "vpce-0a1b2c3d4e5f6g7h8",
      "VpcEndpointType": "Interface",
      "ServiceName": "com.amazonaws.us-east-1.sns",
      "State": "available"
    }
  ]
}
```

If "State": "available", the endpoint is active and ready for use.

```
[ec2-user@ip-10-0-1-185 ~]$ aws ec2 describe-vpc-endpoints --vpc-endpoint-ids vpce-0e8ed9570ce7fe63a
{
  "VpcEndpoints": [
    {
      "VpcEndpointId": "vpce-0e8ed9570ce7fe63a",
      "VpcEndpointType": "Interface",
      "VpcId": "vpc-05e79cd01567d07e2",
      "ServiceName": "com.amazonaws.us-east-1.sns",
      "State": "available",
      "PolicyDocument": "{\n  \"Statement\": [\n    {\n      \"Action\": \"**\", \n      \"Effect\": \"Allow\", \n      \"Principal\": \"**\", \n      \"Resource\": \"**\"\n    }\n  ]\n}",
      "RouteTableIds": [],
      "SubnetIds": [
        "subnet-09c9303125967a9b1"
      ],
      "Groups": [
        {
          "GroupId": "sg-07dd551537f384bd8",
          "GroupName": "sns-private-stack-InstanceSecurityGroup-8tzgR0bNKGWN"
        }
      ],
      "IpAddressType": "ipv4",
      "DnsOptions": {
        "DnsRecordIpType": "ipv4"
      },
      "PrivateDnsEnabled": true,
      "RequesterManaged": false,
    }
  ]
}
```

Publish a Private SNS Message

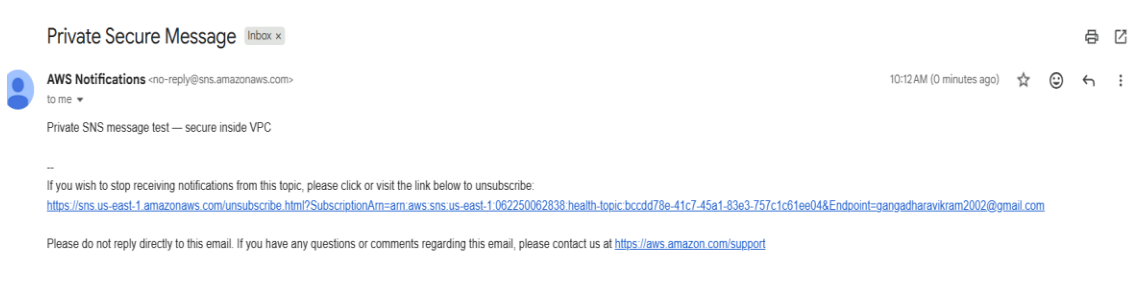
Now publish a private message from your EC2 instance through the endpoint:

```
aws sns publish \  
  
--topic-arn arn:aws:sns:us-east-1:<your-account-id>:health-topic \  
  
--message "Private SNS message test — secure inside VPC" \  
  
--subject "Private Secure Message"
```

Verification:

- The command returns a MessageId
- You receive the email instantly
- The message is transmitted privately via the VPC Endpoint, not over the public internet

```
[ec2-user@ip-10-0-1-185 ~]$ aws sns publish \  
> --topic-arn arn:aws:sns:us-east-1:062250062838:health-topic \  
> --message "Private SNS message test — secure inside VPC" \  
> --subject "Private Secure Message"  
{  
  "MessageId": "ad9d1610-8bb4-59b3-8785-7792bf4ed8d6"  
}  
[ec2-user@ip-10-0-1-185 ~]$
```



Confirm Endpoint Configuration in the Console

- 1. Go to VPC → Endpoints → Select your SNS endpoint
- 2. Verify these details:

Setting Expected Value

Type Interface

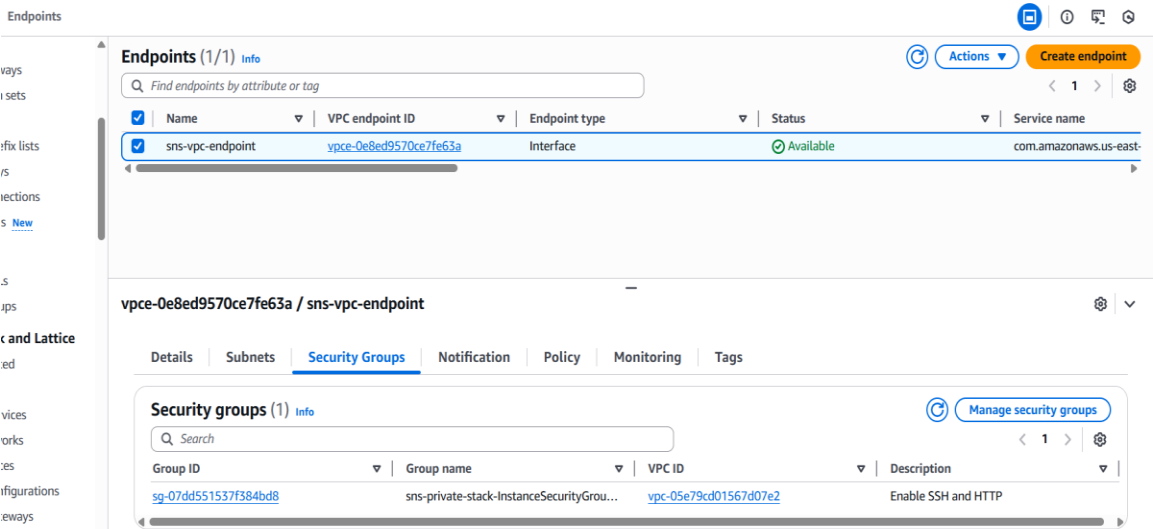
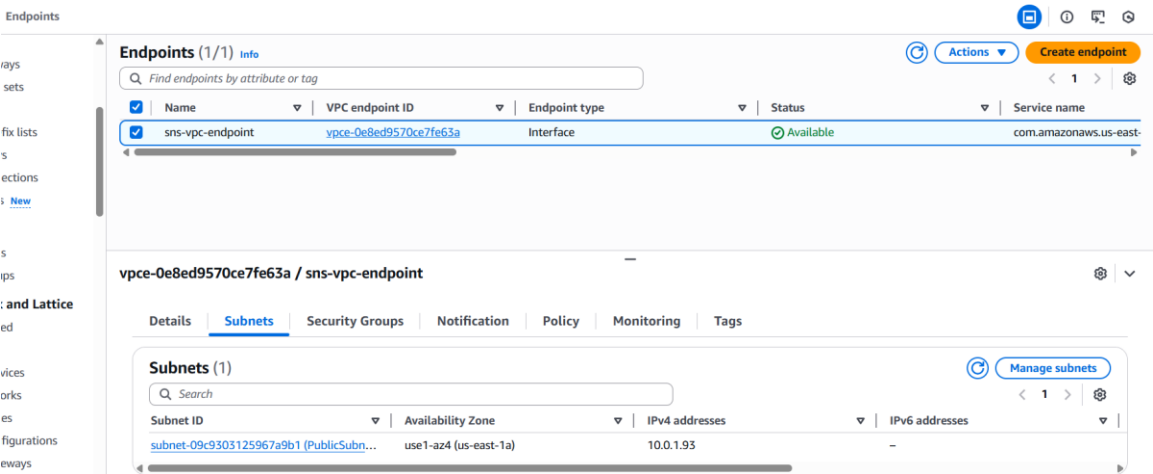
Status Available

DNS Enabled

Subnet Your VPC Subnet ID

Security Group InstanceSecurityGroup

If all these configurations are correct, it confirms that SNS communication between EC2 and SNS is private, secure, and routed entirely through AWS’s internal network.



VERIFICATION CHECKLIST

- Web Server Running — Returns “Healthcare SNS Report System is Running”
- SNS Access — Executes successfully — no errors
- Topic Created — Returns a valid Topic ARN
- Message Published — Returns a valid MessageId
- S3 Upload — File uploaded successfully to S3
- Endpoint Status — Status shows Available

FINAL CONFIRMATION

If all verifications above are successful:

The Healthcare SNS Project is fully functional, private, and secure.

All messages and data transfers occur within the AWS VPC, ensuring end-to-end protection through PrivateLink (Interface Endpoint).