

# **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 'Create stack' wizard at Step 3: Prerequisite - Prepare template. On the left, a sidebar lists steps: Step 3 (Configure stack options), Step 4 (Review and create). The main area has a heading 'Prerequisite - Prepare template'. It says 'You can also create a template by scanning your existing resources in the IaC generator.' Below is a 'Prepare template' section with the note '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.' There are three options: 1) 'Choose an existing template' (selected, highlighted in blue), which says 'Upload or choose an existing template.' 2) 'Build from Infrastructure Composer' (disabled, greyed out), which says 'Create a template using a visual builder.' 3) 'Sync from Git' (disabled, greyed out), which says 'Sync a template from your Git repository.' Below this is a 'Specify template' section with a note 'This GitHub repository' containing sample CloudFormation templates. It has sections for 'Template source' (Amazon S3 URL, Upload a template file, Sync from Git), 'Upload a template file' (Choose file, hospital-sns-private-setup-vik.yaml), and 'S3 URL' (https://s3.us-east-1.amazonaws.com/cf-templates-1e04a1hyej8uf-us-east-1/2025-11-03T031157.109Z53w-hospital-sns-private-setup-vik.yaml).

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)**

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)**

Stack name	Status
sns-private-stack	CREATE_COMPLETE

**sns-private-stack**

Stack info	Events	Resources	Outputs	Parameters	Template	Changesets	Git sync
<b>Overview</b>							
<b>Stack ID</b> arn:aws:cloudformation:us-east-1:062250062838:stack/sns-private-stack/69101f30-b864-11f0-8fbc-0ed0135cae3f	<b>Description</b> Healthcare Project - Secure Patient Report Sharing with SNS and EC2 (Private VPC)						
<b>Status</b> CREATE_COMPLETE	<b>Detailed status</b>						
<b>Status reason</b> -	<b>Root stack</b>						
<b>Parent stack</b> -	<b>Created time</b> 2025-11-03 08:53:04 UTC+0530						
	<b>Updated time</b>						
<b>Deleted time</b> -	<b>Drift status</b> 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 Stacks page. On the left, there is a sidebar with a search bar and a 'Stacks (1)' section containing a single stack named 'sns-private-stack' with a status of 'CREATE\_COMPLETE'. On the right, the main area is titled 'sns-private-stack' and shows the 'Outputs' tab selected. Below the tabs, there is a search bar and a table titled 'Outputs (3)'. The table has columns for 'Key', 'Value', 'Description', and 'Export name'. The data in the table is as follows:

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.

The screenshot shows the AWS EC2 Instances page. At the top, there are buttons for 'Connect', 'Instance state', 'Actions', and 'Launch instances'. Below the header, there's a search bar and a dropdown menu set to 'All states'. A table lists one instance: 'HealthcareEC2' (Instance ID: i-0781a92334a2c6944), which is 'Running' (Status check: 3/3 checks passed), of type 't3.micro', in 'us-east-1a' availability zone, with a public IPv4 address 'ec2-54-237-'. There are buttons for 'View alarms' and 'Unselect instance: HealthcareEC2'.

## 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

```

   ,      #
  ~\_ ####_      Amazon Linux 2023
  ~~ \_\#\#\#\\
  ~~   \|##|
  ~~     \|/|____ https://aws.amazon.com/linux/amazon-linux-2023
  ~~       V~'`->
  ~~     /
  ~~..  /
  ~~ /_/
  _/m'_

[ec2-user@ip-10-0-1-185 ~]$ sudo yum update
Last metadata expiration check: 0:05:17 ago on Mon Nov  3 03:24:57 2025.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-10-0-1-185 ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
    Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
    Active: active (running) since Mon 2025-11-03 03:25:16 UTC; 5min ago
      Docs: man:httpd.service(8).

```



## PHASE 4 — VERIFY AWS CLI & IAM ROLE ACCESS

### Check IAM Role Attachment

Run this command inside your EC2 terminal:

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

### Expected Output:

```
{
  "Code" : "Success",
  "LastUpdated" : "2025-11-03T12:30:45Z",
```

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

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 ~]$ █
```

The screenshot shows an email from AWS Notifications (no-reply@sns.amazonaws.com) to the user. The subject is "Hospital Report Update". The email body contains the message: "Patient report uploaded successfully". It also includes unsubscribe instructions: "If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe: <https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-1:062250062838:health-topic:bccdd78e-41c7-45a1-83e3-757c1c61ee04&Endpoint=gangadharavikram2002@gmail.com>". A note at the bottom says: "Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at <https://aws.amazon.com/support>".

## 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.

The screenshot shows the AWS SNS console. On the left, there's a sidebar with navigation links: 'Amazon SNS', 'Topics', 'health-topic', 'Dashboard', 'Topics', 'Subscriptions', 'Mobile', 'SMS notifications', and 'Text messaging (SMS)'. The main area has a title 'Details' and a table with columns: Name (health-topic), Display name (-), ARN (arn:aws:sns:us-east-1:062250062838:health-topic), Topic owner (062250062838), and Type (Standard). Below this is a navigation bar with tabs: Subscriptions (selected), Access policy, Data protection policy, Delivery policy (HTTP/S), Delivery status logging, Encryption, and Tags. Under the Subscriptions tab, there's a sub-table titled 'Subscriptions (1)' with columns: ID, Endpoint, Status, and Protocol. One row is shown: ID 'bccdd78e-41c7-45a1-83e3-757c1...', Endpoint 'gangadharavikram2002@gmail.com', Status 'Confirmed', and Protocol 'EMAIL'.

## 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 ~]$
```

The screenshot shows the AWS S3 console interface. At the top, the navigation bar indicates 'Amazon S3 > Buckets > s3-demo-3rd-nov'. Below the navigation, there are tabs for 'Objects', 'Metadata', 'Properties', 'Permissions', 'Metrics', 'Management', and 'Access Points'. The 'Objects' tab is selected. The main area displays a table titled 'Objects (1)'. The table has columns for Name, Type, Last modified, Size, and Storage class. A single object, 'report.txt', is listed. It is a 'txt' file last modified on November 3, 2025, at 09:46:10 (UTC+05:30), with a size of 37.0 B and a storage class of Standard. Action buttons for 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Actions', 'Create folder', and 'Upload' are located at the top of the table.

## 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.

The screenshot shows the AWS VPC console interface. On the left, a sidebar lists various VPC components: Gateways, Option sets, IP ranges, Prefix lists, Tunnels, Peering connections, Route tables, Security groups, and Network ACLs. The 'Endpoints' option is selected. The main content area is titled 'Endpoints (1/1)'. A table lists the endpoint details. The columns are Name, VPC endpoint ID, Endpoint type, Status, and Service name. One endpoint is listed: 'sns-vpc-endpoint' with VPC endpoint ID 'vpce-08ed9570ce7fe63a', Endpoint type 'Interface', Status 'Available', and Service name 'com.amazonaws.us-east-1.sns'. Action buttons for 'Actions' and 'Create endpoint' are located at the top right of the table.

## 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\",  
                        \"Principal\": \"*\",\n                        \"Resource\": \"*\n                    }\n                ]\n            }",  
            "RouteTableIds": [],  
            "SubnetIds": [  
                "subnet-09c9303125967a9b1"  
            ],  
            "Groups": [  
                {  
                    "GroupId": "sg-07dd551537f384bd0",  
                    "GroupName": "sns-private-stack-InstanceSecurityGroup-2tzgR0bNkGWN"  
                }  
            ],  
            "IpAddressType": "ipv4",  
            "DnsOptions": {  
                "DnsRecordType": "ipv4"  
            },  
            "PrivateDnsEnabled": true,  
            "RequesterManaged": false,  
            "Status": "available",  
            "LastModified": "2023-09-01T10:00:00Z",  
            "CreationTime": "2023-09-01T10:00:00Z",  
            "Tags": []  
        }  
    ]  
}
```

## Publish a Private SNS Message

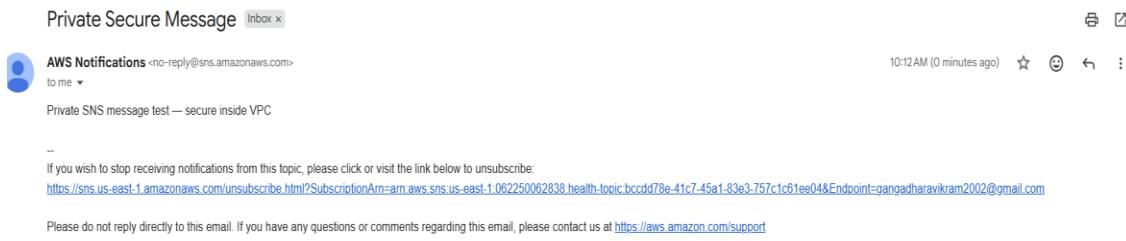
Now publish a private message from your EC2 instance through the endpoint:

```
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 ~]$
```



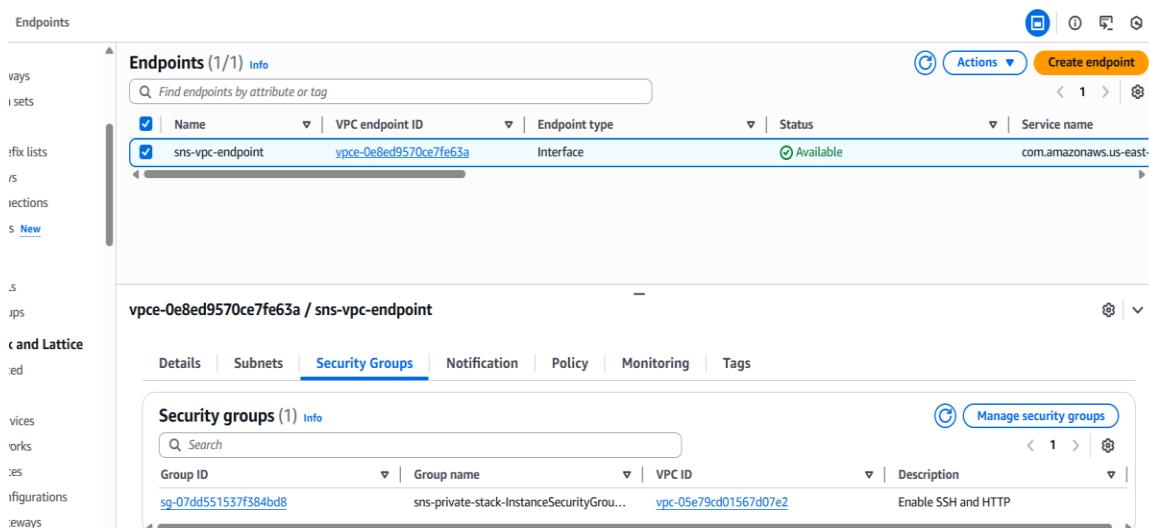
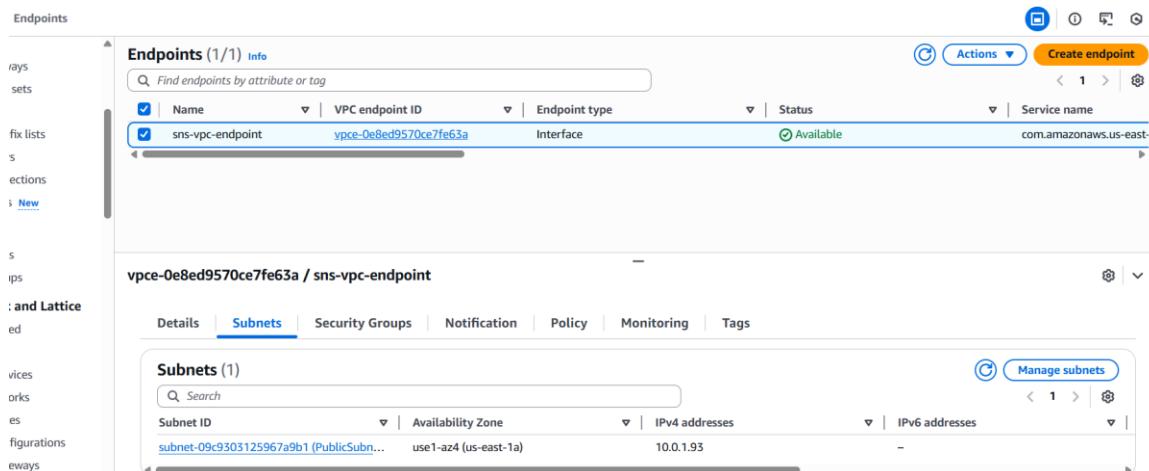
## Confirm Endpoint Configuration in the Console

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).