

A Project Documentation on
“Highly Available Web Application with
Secure VPC, S3 Storage & SNS Alerts”

Project Objective

Design and deploy a secure, scalable web application on AWS using:

- EC2 for compute
- S3 for static content & backups
- VPC for networking & security
- SNS for system notifications and alerts

Steps:-

Step 1: Create a Custom VPC

- Go to VPC → Create VPC
- Select VPC only

Enter:

- Name: **webapp-vpc**
- IPv4 CIDR: **10.0.0.0/16**
- Click Create VPC

The screenshot shows the AWS VPC console interface. On the left, there's a sidebar with options like 'Virtual private cloud' (selected), 'Your VPCs', 'Subnets', 'Route tables', etc. The main area is titled 'VPCs' and shows a table of 'Your VPCs'. There are two entries:

| Name | VPC ID | State | Encryption c... | Encryption contrc |
|------------|-----------------------|-----------|-----------------|-------------------|
| - | vpc-0f98c32bed507c7b1 | Available | - | - |
| webapp-vpc | vpc-05b884418f5e47d5d | Available | - | - |

Below the table, it says 'Select a VPC above'.

Step 2: Create a Public Subnet

- Go to VPC → Subnets → Create subnet
- Select webapp-vpc

Enter:

- Subnet name: **public-subnet**
- Availability Zone: any (e.g., us-east-1a)
- CIDR: 10.0.1.0/24
- Enable Auto-assign public IPv4
- Click Create subnet

The screenshot shows the AWS VPC Subnets page. On the left, there's a sidebar with 'VPC dashboard' and a 'Virtual private cloud' section containing links for Your VPCs, Subnets, Route tables, Internet gateways, Egress-only Internet gateways, DHCP option sets, Elastic IPs, and Managed prefix lists. The main area is titled 'Subnets (4) Info'. It has a search bar and a table with columns: Name, Subnet ID, State, and VPC. The table contains the following data:

| Name | Subnet ID | State | VPC |
|---------------|--------------------------|-----------|--------------------------------|
| public subnet | subnet-04034bca7875885a6 | Available | vpc-05b884418f5e47d5d web... |
| - | subnet-0bb0dcdf11610113d | Available | vpc-0f98c32bed507c7b1 |
| - | subnet-09859e0ffbebfb80b | Available | vpc-0f98c32bed507c7b1 |
| - | subnet-09dba59bcfe32edb8 | Available | vpc-0f98c32bed507c7b1 |

Below the table, there's a section titled 'Select a subnet'.

Step 3: Create and Attach Internet Gateway

- Go to VPC → Internet Gateways
- Click Create internet gateway
- Name: **my-igw**
- Attach it to webapp-vpc

The screenshot shows the AWS VPC Internet Gateways page. The left sidebar has sections for VPC dashboard, AWS Global View, Virtual private cloud (with Your VPCs, Subnets, Route tables, and Internet gateways), and NAT gateways. The Internet gateways section is expanded, showing options like Egress-only Internet gateways, DHCP option sets, Elastic IPs, and Managed prefix lists. The main content area displays a table of Internet gateways:

| Name | Internet gateway ID | State | VPC ID |
|--------|-----------------------|----------|-----------------------------|
| - | igw-069c935f1cfdd8690 | Attached | vpc-0f98c32bed507c7b1 |
| my-igw | igw-07c2caa4c3d0a628f | Attached | vpc-05b884418f5e47d5d web |

Below the table, a message says "Select an internet gateway above". The top of the screen shows the AWS navigation bar and the URL eu-north-1.console.aws.amazon.com/vpcconsole/home?region=eu-north-1#igws:.

Step 4: Configure Route Table

- Go to VPC → Route Tables

Create a route table:

- Name: **route1**
- VPC: webapp-vpc

Add route:

- Destination: 0.0.0.0/0
- Target: Internet Gateway
- Associate the route table with Public-Subnet

The screenshot shows the AWS VPC Route Tables console. The left sidebar is titled 'VPC dashboard' and includes sections for 'AWS Global View', 'Virtual private cloud' (with 'Your VPCs' and 'Subnets' options), and 'Route tables' (which is currently selected). Under 'Route tables', there are links for 'Internet gateways', 'Egress-only Internet gateways', 'DHCP option sets', 'Elastic IPs', 'Managed prefix lists', and 'NAT gateways'. The main content area is titled 'Route tables (1/3)' and shows a table with one row for 'route1'. The table columns are: Name, Route table ID, Explicit subnet assoc..., Edge associations, and Main. The 'route1' row has a blue selection bar around it. Below the table, a detailed view for 'rtb-044b06d436ec88905 / route1' is shown, listing two routes: '0.0.0.0/0' with target 'igw-07c2caa4c3d0a6...' and status 'Active', and '10.0.0.0/16' with target 'local' and status 'Active'. The bottom of the page includes standard AWS footer links for CloudShell, Feedback, and Console Mobile App, along with copyright information and privacy terms.

Step 5: Create Security Group

- Go to EC2 → Security Groups → Create
- Name: **ec2_sg**
- VPC: webapp-vpc
- Inbound rules:
- HTTP – Port 80 – Source: 0.0.0.0/0
- SSH – Port 22 – Source: My IP
- Outbound: Allow all traffic
- Create security group

The screenshot shows the AWS EC2 Security Groups page. On the left, there's a navigation sidebar with sections like Capacity Manager, Images, Elastic Block Store, Network & Security (Security Groups selected), and others. The main area is titled "Security Groups (3)" and shows a table with three existing security groups: "default" (Security group ID: sg-07025a2b75e520017, VPC ID: vpc-05b884418f5e47d5d1), "ec2-sg" (Security group ID: sg-0e57286af8e00cc83, VPC ID: vpc-05b884418f5e47d5d1), and another "default" entry (Security group ID: sg-0127e6497af5b28b5, VPC ID: vpc-0f98c32bed507c7b1). A "Create security group" button is visible at the top right. Below the table, there's a section titled "Select a security group". At the bottom of the page, there are links for CloudShell, Feedback, Console Mobile App, and standard footer links.

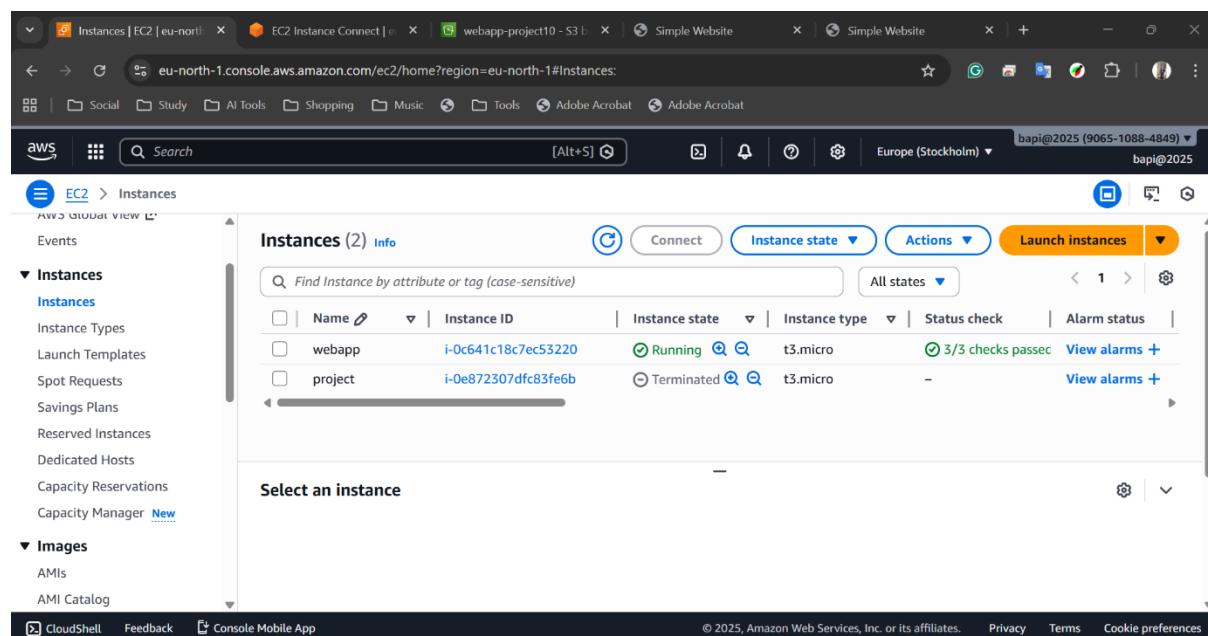
| Name | Security group ID | Security group name | VPC ID |
|------|----------------------|---------------------|------------------------|
| - | sg-07025a2b75e520017 | default | vpc-05b884418f5e47d5d1 |
| - | sg-0e57286af8e00cc83 | ec2-sg | vpc-05b884418f5e47d5d1 |
| - | sg-0127e6497af5b28b5 | default | vpc-0f98c32bed507c7b1 |

Step 6: Launch EC2 Instance

- Go to EC2 → Launch Instance
 - Name: **webapp**
 - AMI: Amazon Linux 2
 - Instance type: t3.micro (Free tier)
 - Key pair: created a project.pem
 - Network settings:
 - VPC: webapp-vpc
 - Subnet: public-subnet
 - Auto-assign Public IP: Enabled
 - Security group: ec2_sg
-
- User data:

```
#!/bin/bash
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
echo "Welcome to Highly Available Web App" > /var/www/html/index.html
```

- Launch instance



Step 7: Create IAM Role

- Go to IAM → Roles
- Create Role
- Trusted Entity: AWS Service
- Use case: EC2
- Permission policies: AmazonS3FullAccess
- Name: project
- Create Role

The screenshot shows the AWS IAM Roles page. The left sidebar has a 'Roles' section under 'Access management'. The main area displays a table titled 'Roles (9)'. The table columns are 'Role name', 'Trusted entities', and 'Last activity'. The rows show the following data:

| Role name | Trusted entities | Last activity |
|-----------------------------------|--|---------------|
| AWSServiceRoleForResourceExplorer | AWS Service: resource-explorer-2 (Service) | 16 minutes |
| AWSServiceRoleForSupport | AWS Service: support (Service-Linker) | - |
| AWSServiceRoleForTrustedAdvisor | AWS Service: trustedadvisor (Service) | - |
| Project | AWS Service: ec2 | 16 minutes |

At the bottom right of the table, there is a 'Manage' button.

Step 8: Create S3 Bucket

- Go to S3 → Create bucket
- Bucket name: **webapp-project10**
- Region: same as EC2
- Enable:
- Versioning
- Block public access
- Create bucket

Step 9: Upload Static Content to S3

- Open the S3 bucket
- Upload:
 - index.html
 - style.css
 - script.js

| Name | Type | Last modified | Size | Storage class |
|------------|------|---|---------|---------------|
| index.html | html | December 23, 2025, 18:19:11 (UTC+05:30) | 785.0 B | Standard |
| style.css | css | December 23, 2025, 18:19:11 (UTC+05:30) | 666.0 B | Standard |

Step 10: Create SNS Topic

- Go to SNS → Topics
- Click Create topic
- Type: Standard
- Name: **webapp-alerts**
- Create topic

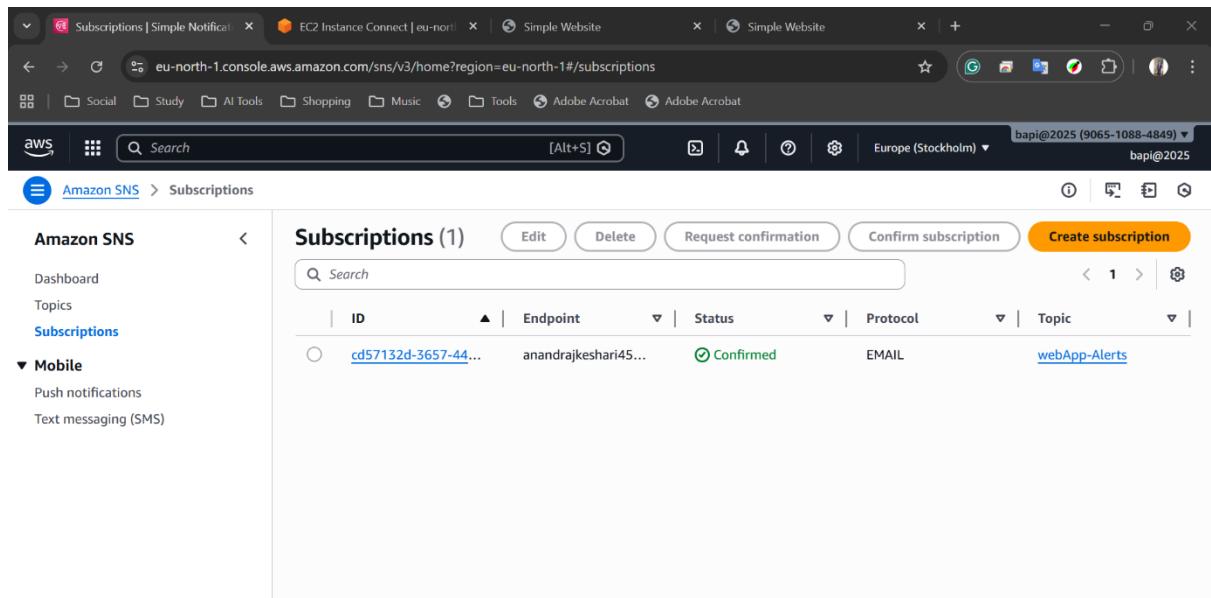
The screenshot shows the AWS SNS Topics page. The browser tab is titled "Topics | Simple Notification Ser...". The URL is "eu-north-1.console.aws.amazon.com/sns/v3/home?region=eu-north-1#/topics". The page displays a table with one row of data:

| Name | Type | ARN |
|-------------------------------|----------|--|
| webApp-Alerts | Standard | arn:aws:sns:eu-north-1:906510884849... |

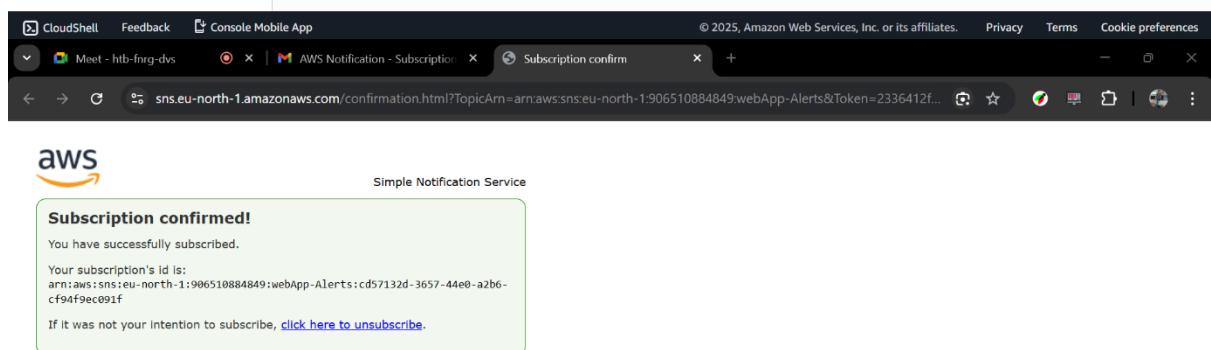
The left sidebar shows navigation links: Dashboard, Topics (which is selected), Subscriptions, Mobile (Push notifications, Text messaging (SMS)), CloudShell, Feedback, and Console Mobile App. The bottom of the page includes copyright information: © 2025, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

Step 11: Subscribe to SNS Topic

- Open webapp-alerts
- Click Create subscription
- Protocol: Email
- Endpoint: your email address → dishahota23@gmail.com
- Confirm subscription from email



The screenshot shows the AWS SNS Subscriptions page. On the left, there's a sidebar with links for Amazon SNS, Dashboard, Topics, Subscriptions, Mobile (Push notifications, Text messaging (SMS)), and a CloudShell tab. The main area has a title "Subscriptions (1)". Below it is a search bar and a table with columns: ID, Endpoint, Status, Protocol, and Topic. One row is visible, showing an ID starting with "cd57132d-3657-44...", an endpoint "anandrajkeshari45...", a status "Confirmed" with a green checkmark, a protocol "EMAIL", and a topic "webApp-Alerts". There are buttons for Edit, Delete, Request confirmation, Confirm subscription, and Create subscription.



The screenshot shows the AWS Subscription confirmation page. It features the AWS logo and the Simple Notification Service header. A green box contains the message "Subscription confirmed! You have successfully subscribed. Your subscription's ID is: arn:aws:sns:eu-north-1:906510884849:webApp-Alerts:cd57132d-3657-44e0-a2b6-cf94f9ec091f. If it was not your intention to subscribe, [click here to unsubscribe](#)." The URL in the browser is "sns.eu-north-1.amazonaws.com/confirmation.html?TopicArn=arn:aws:sns:eu-north-1:906510884849:webApp-Alerts&Token=2336412f...".

Step 12: Connect SNS with Monitoring

- Go to CloudWatch
- Create alarm (example):
- EC2 CPU utilization > 80%
- Set alarm action → SNS topic
- Save alarm

The screenshot shows the AWS CloudWatch Alarms interface. On the left, there's a sidebar with navigation links like Dashboards, Alarms (selected), AI Operations, Application Signals, and Infrastructure Monitoring. The main area displays a green success message: "Successfully created alarm webapp-alarma." Below this, a table lists the created alarm:

| Name | State | Last state update (UTC) | Conditions |
|---------------|-------|-------------------------|--|
| webapp-alarma | OK | 2025-12-23 13:45:01 | CPUUtilization > 10000 for 1 datapoints within 5 minutes |

Step 13: Test the Application

- Copy EC2 public IP → **16.170.223.116**
 - Paste into browser → Web page loads
 - Stop EC2 instance → verify SNS alert (if alarm configured)
 - Check S3 uploads

