

Overview

In this project, we set up a Virtual Private Cloud (VPC) with public and private subnets across two Availability Zones (AZs). The infrastructure includes an Auto Scaling group, an Application Load Balancer, and NAT Gateways to manage server traffic, enhance security, and ensure high availability in a production environment.

Architecture Components

- **VPC with Public and Private Subnets:** The VPC spans two AZs, each containing both public and private subnets.
- **NAT Gateway:** Deployed in each AZ to allow instances in private subnets to access the internet securely.
- **Application Load Balancer (ALB):** Distributes incoming traffic across the instances in the private subnets.
- **Auto Scaling Group (ASG):** Automatically adjusts the number of instances in the private subnets based on demand.
- **Bastion Host:** Used to securely connect to instances in the private subnets.

Step-by-Step Implementation

1. Create the VPC and Subnets

- **Go to the VPC dashboard** in AWS.

- **Create a VPC** with a CIDR block suitable for your environment (e.g., 10.0.0.0/16).
- **Create Public and Private Subnets** in two Availability Zones (e.g., 10.0.1.0/24, 10.0.2.0/24 for public, 10.0.3.0/24, 10.0.4.0/24 for private).

[VPC](#) > [Your VPCs](#) > [Create VPC](#)

Create VPC [Info](#)

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as

VPC settings

Resources to create [Info](#)
Create only the VPC resource or the VPC and other networking resources.

☐ VPC only ☒ VPC and more

Name tag auto-generation [Info](#)
Enter a value for the Name tag. This value will be used to auto-generate Name tags for all resources in the VPC.

☒ Auto-generate

aws-prod1-example

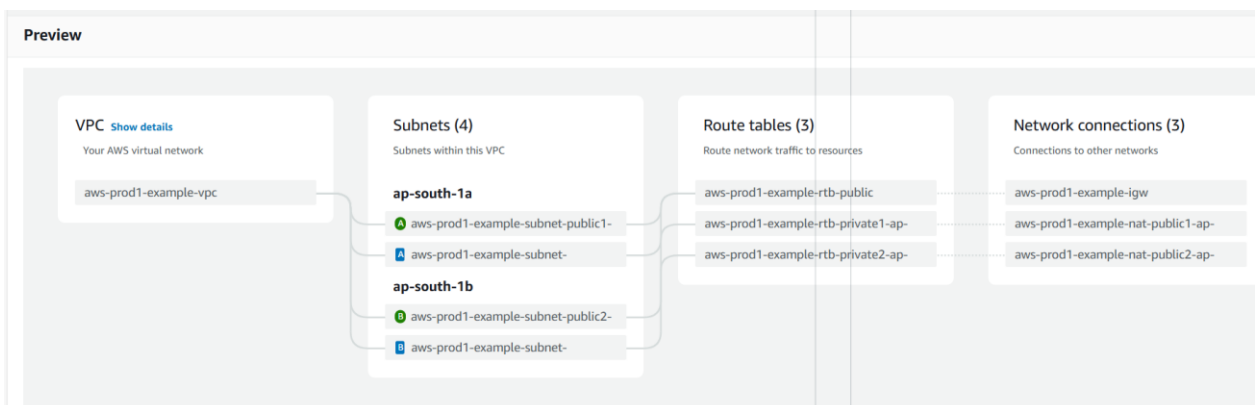
IPv4 CIDR block [Info](#)
Determine the starting IP and the size of your VPC using CIDR notation.

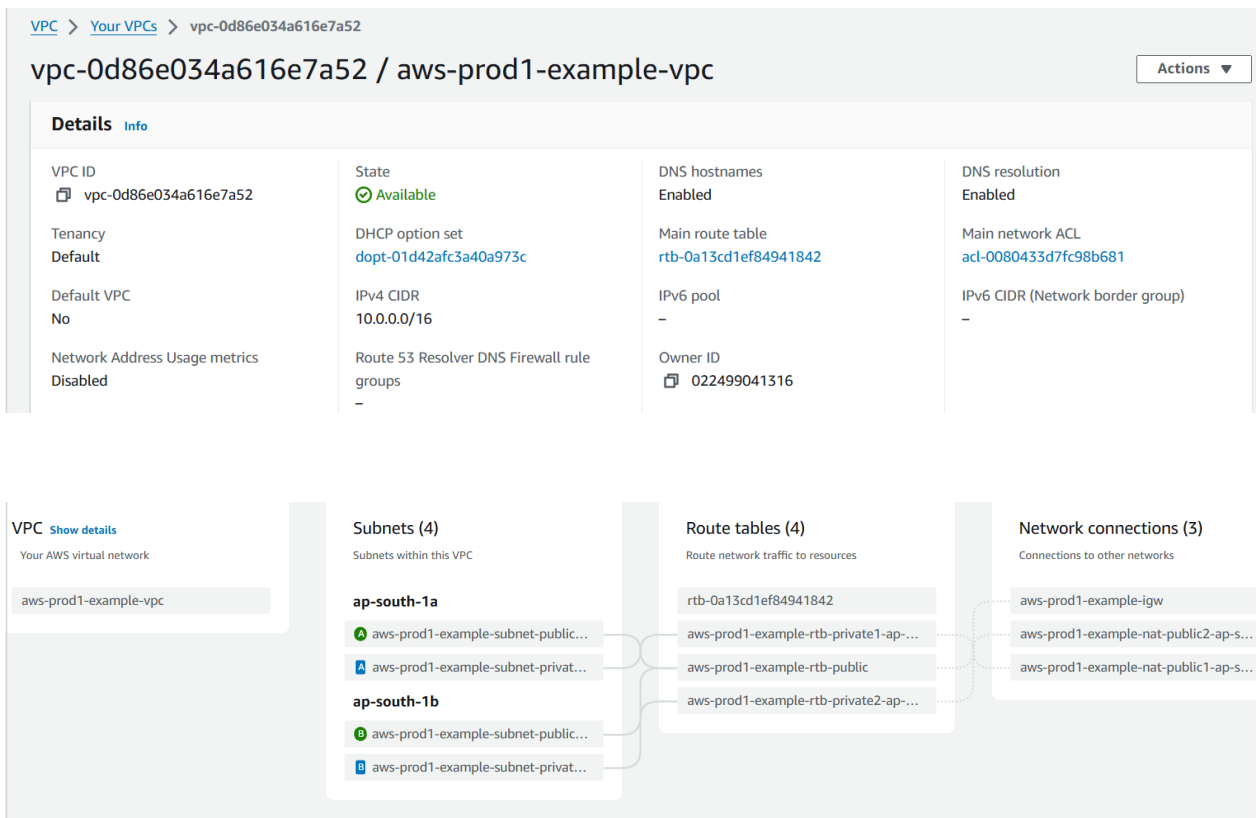
10.0.0.0/16 65,536 IPs

CIDR block size must be between /16 and /28.

IPv6 CIDR block [Info](#)

☒ No IPv6 CIDR block ☐ Amazon-provided IPv6 CIDR block





2. Configure Route Tables

- Attach an **Internet Gateway (IGW)** to the VPC.
- Create a **Route Table** for the public subnets and associate it with the subnets. Add a route to the IGW (0.0.0.0/0 -> IGW).
- For the private subnets, associate them with the default route table that has no direct internet access.

3. Create NAT Gateways

- **Go to the NAT Gateways** section in the VPC dashboard.
- Create a **NAT Gateway** in each public subnet, and associate each with an Elastic IP.
- Update the route table for the private subnets to direct internet-bound traffic to the NAT Gateway.

4. Launch Instances in Private Subnets via Auto Scaling

- **Go to EC2 -> Auto Scaling Groups.**
- **Create a Launch Template** with the following settings:
 - Select a recent AMI (Amazon Machine Image).
 - Configure the instance type (e.g., t2.micro).
 - **Create a new security group** with the following inbound rules:
 - SSH (Port 22) - Anywhere
 - Custom TCP (Port 8000) - Anywhere (for the Python app)
- **Create the Auto Scaling Group (ASG):**
 - Select the launch template created.
 - Choose the VPC and select the private subnets.
 - Set the desired capacity to 2, minimum to 1, and maximum to 4.

- Two instances will be launched without public IPs.

[EC2](#) > [Auto Scaling groups](#) > aws-prod1-example

aws-prod1-example

[Details](#) | [Activity](#) | [Automatic scaling](#) | [Instance management](#) | [Monitoring](#) | [Instance refresh](#)

Group details

Edit

Auto Scaling group name aws-prod1-example	Desired capacity 2	Desired capacity type Units (number of instances)	Amazon Resource Name (ARN) arn:aws:autoscaling:ap-south-1:022499041316:autoScalingGroup:de321437-d2e9-4347-998a-e54045ea2bc1:autoScalingGroupName/aws-prod1-example
Date created Sat Aug 31 2024 18:33:39 GMT+0530 (India Standard Time)	Minimum capacity 1	Status -	
	Maximum capacity 4		

Launch template

Edit

Launch template lt-08287f91ad430b447 aws-prod1-example	AMI ID ami-0ad21ae1d0696ad58	Instance type t2.micro	Owner arn:aws:iam::022499041316:root
Version Default	Security groups -	Security group IDs sg-043adc8482d868121	Create time Sat Aug 31 2024 18:29:26 GMT+0530 (India Standard Time)
Description proof of concept for app deploy in aws private subnet	Storage (volumes) -	Key pair name awsdemo	Request Spot Instances No

[View details in the launch template console](#)

5. Set Up a Bastion Host

- **Launch a new EC2 instance** in the public subnet (AZ 1a).
- Enable the public IP and select the VPC created.
- Use the default AMI or your preferred Linux distribution.
- Once launched, **copy the key pair** from your local machine to the Bastion Host

Terminus is a popular terminal emulator and SSH client that allows users to securely connect to remote servers from their local machine. It provides a graphical interface for users to access command-line environments on remote servers over a network using SSH (Secure Shell) protocol.

- **Install Terminus:** Download and install Terminus from its official website.
- **Open Terminus:** Launch the Terminus application.
- **Create a New SSH Connection:** Click "New SSH Connection" and enter the host, port, username, and authentication details like key.
- **Connect to the Server:** Click "Connect" and, if prompted, accept the server's SSH

New Host

Personal vault ▾



Address



43.204.38.116

General

Label



Parent Group



Tags



Backspace

Default

SSH on port

Credentials from Personal vault



ubuntu



Password



awsdemo.pem

Connect

key fingerprint.



Vaults



SFTP



43.204.38.116



Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

* Documentation: <https://help.ubuntu.com>
* Management: <https://landscape.canonical.com>
* Support: <https://ubuntu.com/pro>

System information as of Sat Aug 31 14:26:14 UTC 2024

System load:	0.0	Processes:	105
Usage of /:	22.9% of 6.71GB	Users logged in:	0
Memory usage:	20%	IPv4 address for enX0:	10.0.8.181
Swap usage:	0%		

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See <https://ubuntu.com/esm> or run: `sudo pro status`

The list of available updates is more than a week old.
To check for new updates run: `sudo apt update`

Last login: Sat Aug 31 14:24:40 2024 from 103.10.226.150
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-8-181:~\$ █

```

ubuntu@ip-10-0-8-181:~$ pwd
/home/ubuntu
ubuntu@ip-10-0-8-181:~$ ls
awsdemo.pem
ubuntu@ip-10-0-8-181:~$ sudo cp awsdemo.pem/home/ubuntu/.ssh
cp: missing destination file operand after 'awsdemo.pem/home/ubuntu/.ssh'
Try 'cp --help' for more information.
ubuntu@ip-10-0-8-181:~$ chmod 400 awsdemo.pem
ubuntu@ip-10-0-8-181:~$ ssh -i awsdemo.pem ubuntu@43.204.38.116
The authenticity of host '43.204.38.116 (43.204.38.116)' can't be established.
ED25519 key fingerprint is SHA256:TpEd1fA8Q6ffEr+LKBRVlFXiQZ2MRytBLJPi+hfb7Ss.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '43.204.38.116' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Sat Aug 31 14:37:01 UTC 2024

System load:  0.0               Processes:            110
Usage of /:   22.9% of 6.71GB   Users logged in:     1
Memory usage: 21%               IPv4 address for enX0: 10.0.8.181
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Sat Aug 31 14:26:15 2024 from 103.10.226.150
ubuntu@ip-10-0-8-181:~$ █

```

Now we have to login in one of the instance to install python application using

command

`ssh -i keyname.pem ubuntu@private_ip_address_of_one_instance`

in this case

we used

`ssh -i awsdemo.pem ubuntu@10.0.139.23`

```
ubuntu@ip-10-0-8-181:~$ ls
awsdemo.pem
ubuntu@ip-10-0-8-181:~$ ssh -i awsdemo.pem ubuntu@10.0
.139.23
The authenticity of host '10.0.139.23 (10.0.139.23)' c
an't be established.
ED25519 key fingerprint is SHA256:A0nAZxA4frhPJY0SfLUA
7MYC5tZXFtgdv3K5mY7g6Bg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[
fingerprint])? yes
Warning: Permanently added '10.0.139.23' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1009-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Sat Aug 31 14:43:20 UTC 2024

System load:  0.0                      Processes:            103
Usage of /:   22.7% of 6.71GB          Users logged in:     0
Memory usage: 19%                     IPv4 address for enX0: 10.0.139.23
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
```

Now we successfully login in one instance in private subnet

6. Deploy a Simple Web Application

- **Create a basic HTML file** on one of the private instances:

vim index.html

```
ubuntu@ip-10-0-139-23:~$ vim index.html
ubuntu@ip-10-0-139-23:~$ cat index.html
<!DOCTYPE html>
<html>
<body>
<h1>My First AWS PROJECT to demonstrate apps in private subnet</h1>

</body>
</html>

ubuntu@ip-10-0-139-23:~$
```

- **Run the application** using Python's built-in HTTP server:

python3 -m http.server 8000

```
ubuntu@ip-10-0-139-23:~$ vim index.html
ubuntu@ip-10-0-139-23:~$ cat index.html
<!DOCTYPE html>
<html>
<body>
<h1>My First AWS PROJECT to demonstrate apps in private subnet</h1>

</body>
</html>

ubuntu@ip-10-0-139-23:~$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

6. Create an Application Load Balancer

- **Go to the EC2 -> Load Balancers** section and create an Application Load Balancer.
- Select the VPC and choose the public subnets.
- Assign the security group created earlier to the Load Balancer.
- **Create a Target Group:**
 - Choose HTTP as the protocol and set the port to 8000.

- Register the private instances in the target group.
- **Associate the Target Group** with the Load Balancer and finish the setup.

EC2 > Load balancers > aws-prod1-example

aws-prod1-example

↻
Actions ▼

▼ Details

Load balancer type Application	Status ✔ Active	VPC vpc-0d86e034a616e7a52	Load balancer IP address type IPv4
Scheme Internet-facing	Hosted zone ZP97RAFLXTNZK	Availability Zones subnet-0dc9d31b60e01ef5c ap-south-1b (aps1-az3) subnet-04743b776969adb85 ap-south-1a (aps1-az1)	Date created August 31, 2024, 20:39 (UTC+05:30)
Load balancer ARN arn:aws:elasticloadbalancing:ap-south-1:022499041316:loadbalancer/app/aws-prod1-example/93909505e6cca3a5		DNS name Info aws-prod1-example-97533889.ap-south-1.elb.amazonaws.com (A Record)	

EC2 > Target groups > aws-prod1-example

aws-prod1-example

Actions ▼

Details

[arn:aws:elasticloadbalancing:ap-south-1:022499041316:targetgroup/aws-prod1-example/909d246f2ed7e157](#)

Target type Instance	Protocol : Port HTTP: 8000	Protocol version HTTP1	VPC vpc-0d86e034a616e7a52
IP address type IPv4	Load balancer aws-prod1-example		

But in load balancer there is error

Listeners and rules | Network mapping | Resource map - new | Security | Monitoring | Integrations | Attributes | Tags

Listeners and rules (1) [Info](#)

↻
Manage rules ▼

A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.

<input type="checkbox"/>	Protocol:Port ▼	Default action ▼	Rules ▼	ARN ▼	Security policy ▼
<input type="checkbox"/>	HTTP:80 ⚠ Not reachable	Forward to target group Listener port unreachable The security groups for your load balancer don't allow traffic on this listener port. Manage your security groups in Security tab.		ARN	Not applicable

- **Update the Load Balancer's Security Group** to allow inbound traffic on HTTP (Port 80).

Now error must be resolve.

7. Test the Setup

- Obtain the **DNS name** of the Load Balancer from the AWS console.
- **Open the DNS name in a web browser.**
- The application should be accessible, and traffic will be directed to the healthiest instance in the private subnets.

