

## Part 1: VPC Architecture Design

- **Task 1.1**

### Create Custom VPC

Design and create a VPC with the following specifications:

**VPC Name:** student-vpc

**CIDR Block:** 10.0.0.0/16

**Enable DNS hostnames:** Yes

**Enable DNS resolution:** Yes

- **Task 1.2**

### Create Subnets

Create the following subnets in different Availability Zones:

- 1) Public Subnet 1: 10.0.1.0/24 (AZ: us-east-1a)
- 2) Public Subnet 2: 10.0.2.0/24 (AZ: us-east-1b)
- 3) Private Subnet 1: 10.0.3.0/24 (AZ: us-east-1a)
- 4) Private Subnet 2: 10.0.4.0/24 (AZ: us-east-1b)
- 5)

- **Task 1.3**

### Configure Internet Gateway

**Create an Internet Gateway named student-igw**

**Attach it to your VPC**

**Update route table for public subnets to route 0.0.0.0/0 to IGW**

- **Task 1.4**

### Configure NAT Gateway

**Create NAT Gateway in Public Subnet 1**

**Allocate Elastic IP for NAT Gateway**

**Update private subnet route table to route 0.0.0.0/0 to NAT Gateway**

The image shows two screenshots of the AWS VPC configuration interface.

**Create NAT gateway (Top Screenshot):**

- Name:** natgateway
- Availability mode:** Zonal (selected)
- Subnet:** subnet-01c81e53543c2e94 (student-vpc-subnet-public1-us-east-1a)
- Connectivity type:** Public (selected)
- Elastic IP allocation ID:** eipalloc-0845bdc23fb1b0a3
- Tags:** Key: Name, Value: natgateway1

**Create VPC (Bottom Screenshot):**

- Number of private subnets:** 4 (selected)
- Customize subnets CIDR blocks:**
  - Public subnet CIDR block in us-east-1a: 10.0.1.0/24 (256.0%)
  - Public subnet CIDR block in us-east-1b: 10.0.2.0/24 (256.0%)
  - Private subnet CIDR block in us-east-1a: 10.0.3.0/24 (256.0%)
  - Private subnet CIDR block in us-east-1b: 10.0.4.0/24 (256.0%)
- NAT gateways (1) - updated:** NAT gateway allows private resources to access the internet from any endpoint within your VPC, providing a single managed internet exit point for the entire region. Additional charges apply.
- VPC endpoints:** None selected.
- DNS options:** Enable DNS hostnames (checked), Enable DNS resolution (checked).
- Preview:** Shows the VPC structure with 4 subnets under us-east-1a and us-east-1b, and 3 route tables.

## Part 2: Security Configuration

- Task 2.1

### Create Security Groups

#### ALB Security Group (alb-sg):

##### Inbound Rules:

- **HTTP (80) from 0.0.0.0/0**
- **HTTPS (443) from 0.0.0.0/0**

##### Outbound Rules:

- **All traffic to 0.0.0.0/0**

**Security group (sg-09c21b512cc9e31b0 | alb-sg) was created successfully**

**sg-09c21b512cc9e31b0 - alb-sg**

**Details**

Security group name alb-sg	Security group ID sg-09c21b512cc9e31b0	Description Alb security group	VPC ID vpc-0a10980f0a0000af
Owner 033691785749	Inbound rules count 2 Permission entries	Outbound rules count 1 Permission entry	

**Inbound rules (2)**

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
-	sgr-01ee5822a5b4e153	IPv4	HTTP	TCP	80	0.0.0.0/0	-
-	sgr-0fd4654b8825d44d	IPv4	HTTPS	TCP	443	0.0.0.0/0	-

## Web Server Security Group (web-sg):

### Inbound Rules:

- **HTTP (80) from alb-sg**
- **SSH (22) from your IP**

### Outbound Rules:

- **All traffic to 0.0.0.0/0**

**Security group (sg-0926167051fc5d946 | web-sg) was created successfully**

**sg-0926167051fc5d946 - web-sg**

**Details**

Security group name web-sg	Security group ID sg-0926167051fc5d946	Description Web Server Security Group	VPC ID vpc-0a10980f0a0000af
Owner 033691785749	Inbound rules count 2 Permission entries	Outbound rules count 1 Permission entry	

**Outbound rules (2)**

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
-	sgr-06a0f53d0b2nf46b	-	HTTP	TCP	80	0.0.0.0/0	-
-	sgr-0d7f480e560fddcfa	IPv4	SSH	TCP	22	0.0.0.0/0	-

- **Task 2.2**

## Configure Network ACLs

Create custom NACL for public subnets

Allow HTTP/HTTPS inbound and outbound

Allow ephemeral ports (1024-65535)

## Part 3: Deploy Web Servers

- **Task 3.1**

### Launch EC2 Instances

Launch 2 EC2 instances with the following configuration:

**AMI: Amazon Linux 2**

**Instance Type: t2.micro**

**Network: student-vpc**

**Subnet: Private Subnet 1 and Private Subnet 2**

**Security Group: web-sg**

**Auto-assign Public IP: Disable**

- **Task 3.2**

### Configure Web Server

Use the following User Data script to install Apache web server:

```
#!/bin/bash
```

```
yum update -y
```

```
yum install -y httpd
```

```
systemctl start httpd
```

```
systemctl enable httpd
```

```
# Create custom page
```

```
INSTANCE_ID=$(curl -s http://169.254.169.254/latest/meta-data/instance-id)
```

```
AZ=$(curl -s http://169.254.169.254/latest/meta-data/placement/availability-zone)
```

```
cat <<EOF > /var/www/html/index.html
```

```
<!DOCTYPE html>
```

```
<html>
```

```

<head><title>Student Project</title></head>
<body style="font-family: Arial; text-align: center; padding: 50px;">
<h1>VPC & ALB Project</h1>
<h2>Instance ID: $INSTANCE_ID</h2>
<h3>Availability Zone: $AZ</h3>
<p>Student Name: [Your Name]</p>
</body>
</html>
EOF

```

The screenshot shows the AWS EC2 Instances page. At the top, there is a green success message: "Successfully initiated termination (deletion) of i-0dab3005424f59393". Below this, the "Instances" table lists two instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 IP	Elastic IP	IPv6 IPs	Monitoring	Security Groups
ec21	i-0e2841382b271eddc	running	t3.micro	Initializing	<a href="#">View alarms</a>	us-east-1a	ec2-3-237-92-106.com...	3.237.92.106	-	-	disabled	web-sg
ec22	i-06022fb1ba7c1ec5e	running	t3.micro	Initializing	<a href="#">View alarms</a>	us-east-1b	ec2-44-206-253-209.co...	44.206.253.209	-	-	disabled	web-sg

## Part 4: Application Load Balancer Setup

- Task 4.1

### Create Target Group

Name: student-tg

Target type: Instances

Protocol: HTTP

Port: 80

VPC: student-vpc

Health check path: /

Health check interval: 30 seconds

Healthy threshold: 2

Unhealthy threshold: 2

Register both EC2 instances to this target group.

- Task 4.2

### Create Application Load Balancer

Name: student-alb

Scheme: Internet-facing

IP address type: IPv4

VPC: student-vpc

Subnets: Select both public subnets

Security Group: alb-sg

- Task 4.3

### Configure Listener

**Protocol: HTTP**  
**Port: 80**  
**Default action: Forward to student-tg**

The screenshots show the configuration of an Application Load Balancer (ALB) with the following details:

- Load balancer name:** student-alb
- Scheme:** Internet-facing (selected)
- Internet:** Selects public IP addresses.
- Local balancer IP address type:** IPv4 (selected)
- VPC:** student-vpc (selected)
- IP pools:** student-vpc-subnet-public1-us-east-1a (selected)
- Availability Zones and subnets:** us-east-1a (us-east-1a1) and us-east-1b (us-east-1a2) selected.
- Security groups:** alb-sg selected.
- Listeners and routing:**
  - Protocol:** HTTP
  - Port:** 80
  - Default action:** Forward to target groups (selected)
  - Routing action:** Forward to target group (selected)
- Target group:** student-tg (selected)
- Weight:** 1
- Percent:** 100%
- Listener tags - optional:** None specified.

**Final output**

Screenshot of the AWS CloudWatch Metrics console showing the CloudWatch Metrics Metrics Insights page. The search bar at the top contains the query "student-alb". The results list shows a single metric named "student-alb" with a timestamp of "2025-01-01T00:00:00Z". The details pane shows the metric value is 1000.0000000000001.

Screenshot of a web browser displaying the URL [student-alb-700450806.us-east-1.elb.amazonaws.com](https://student-alb-700450806.us-east-1.elb.amazonaws.com). The page content includes a navigation bar with links like Home, About, Contact, and Log In, along with a main content area featuring a large image of a person and some descriptive text.

## VPC & ALB Project

**Instance ID:**

**Availability Zone:**

Student Name: [Your Name]