

School: CLOUDBOOSTA	Course: Cloud Computing & DevOps
Name: AKOSA SAMUEL ONYEJEKWE	Module: Networking on Cloud
	Class 2: Assignment 2a_ Custom VPC Setup & EC2 Instance Launch
Student No: CBALC0168	Issue Date: March 1, 2026

Required tasks:

1. Set up a custom VPC (Virtual Private Cloud) and its components (Subnets, NACL (Network Access Control List), Security Group, Route Table, and IGW (Internet Gateway)).
2. Provision an Amazon EC2 Instance in the VPC.
3. Launch the EC2 Instance.
4. Install Apache web server on the EC2 instance.

Information and Explanation for this demonstration:

- The Availability Region is London, eu-west-2, in Europe.
- Two subnets (1 private, and 1 public) are in 1 Availability Zone.
- 1 Internet Gateway (IGW)
- 1 EC2 instance will be provisioned on the Public Subnet.
- The VPC has IPv4 and IPv6 CIDR blocks¹.
- The VPC is assigned IPv4 CIDR block, 10.0.0.0/16, and an IPv6 CIDR ² provided by Amazon.
- The following are CIDR block sizes assigned to the Private Subnet, and Public Subnet.

Table 1: CIDR Block sizes assigned to the Private Subnet and Public Subnet

Subnet	CIDR Block size
Private	10.0.128.0/20
Public	10.0.0.0/20

¹ This VPC is a dual-stack VPC.

² In this demonstration, no NAT Gateway was required, hence the enabling of IPv6 in the EC2 instance.

- EC2 instance in Public Subnet has IPv4 and IPv6 addresses (both public).
- EC2 instance accesses the internet via IPv6.
- Every resource deployed within the Public Subnet would have a Public IP³.
- Our case study allows no egress-only IGW (Internet gateway)⁴.
- Zero EIGW was enabled in this exercise. (IPv4 always allows zero EIGW; IPv6 allows zero EIGW sometimes).
- In this demonstration, this IPv6 allows no EIGW because the IPv6 is public.
- Both outbound, and inbound internet traffic (if Security group allows), are allowed on EC2 instance.
- No NAT gateway is required since EC2 instance has an IPv6 address.
- As a result, EC2 instance is directly routable to internet, **without the instance's address translation** to public IP.
- The NACL and Security group protect EC2 instance from inbound internet traffic.
- EC2 instance would access the Amazon S3 privately without using: (i) internet; (ii) NAT gateway; (iii) going through the IGW⁵.
- EC2 instance should turn its domain name into its IP address. DNS resolution is needed for S3 access⁶.
- The following Security group rules were created (except SSH which is default) and defined for:
 - SSH (already set at default) from anywhere to custom at source 0.0.0.0/0⁷
 - HTTP and HTTPS, each at source 0.0.0.0/0

³ Auto-assign public IPv4 address, and IPv6 address was enabled.

⁴ No egress-only Internet gateway is a regular Internet gateway.

⁵ By the S3 Gateway VPC Endpoint, we keep traffic inside AWS's private network. We require secure internal access to Amazon S3. This ultimately saves cost.

⁶ If both DNS Resolution and Hostnames are enabled, then public the EC2 instance gets a public DNS name. The instance can resolve S3, RDS, API endpoints.

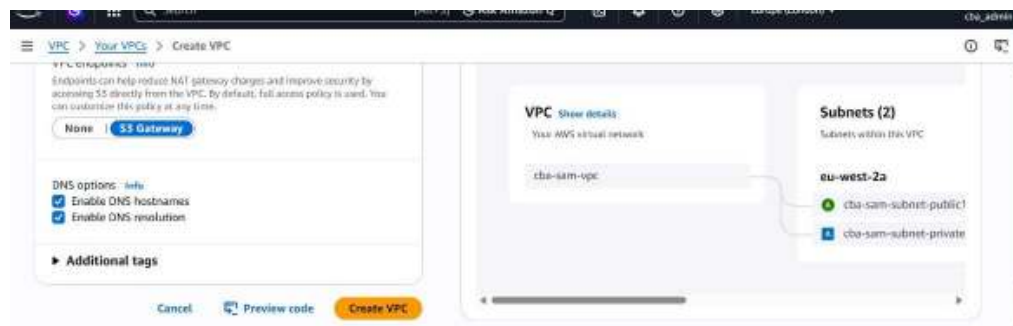
⁷ The change from anywhere to custom at 0.0.0.0/0 (which technically effect the same thing) is not common professional practice. This is because 0.0.0.0/0 will have the SSH (or Port 22) exposed to the entire internet, thereby increasing brute-force risk. This is only done for temporary testing, non-production environment, and learning labs (as is our case).

Practical:

Action 1: Setting up the VPC and its components

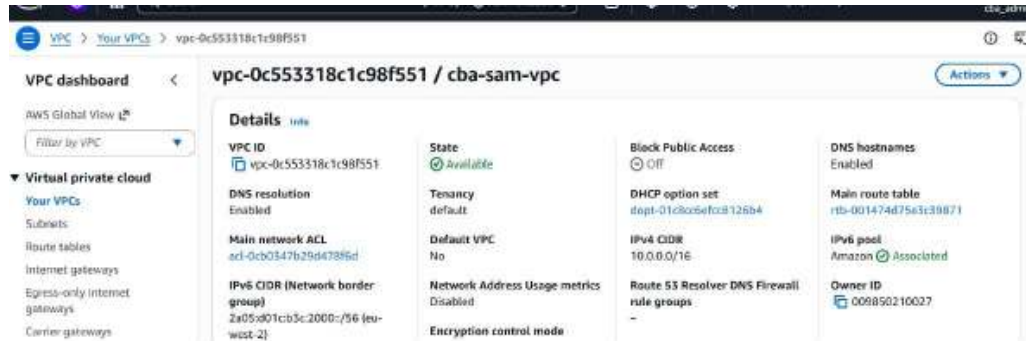
- I logged in to the AWS console as IAM Admin user.
- The Availability Region, Europe (London), eu-west-2, was selected.
- On the VPC settings, **VPC and more** button was selected after negotiating to the VPC dashboard. This is done to automatically create NACL, Route Table, and Internet Gateway.
- This name is named: “*cba-sam*”
- An IPv4 CIDR block, 10.0.0.0/16, was assigned as the size of the VPC.
- Amazon-provided IPv6 CIDR block was selected, from the console.
- Availability Zone is set to 1.
- Both Private Subnet and Public Subnet are each set to 1.
- CIDR block sizes of 10.0.0.0/20 and 10.0.128.0/20 were assigned to the public subnet and private Subnet, respectively. Please see Table 1.
- The S3 gateway VPC Endpoint was selected. See Figure 1 which presents our selections on the console.

Figure 1: Annotated panel showing a preview of the network that was created



- DNS Resolution is enabled so that our VPC can resolve domain names to IP addresses. Also, DNS Hostnames was enabled to assign public DNS name to our EC2 instance.
- The **Create VPC** button was clicked after previewing the network that I intended to create. Figure 2 shows the VPC that has been created.

Figure 2: Panel showing details of created VPC



From Figure 3, we can see that 2 subnets (1 private, and 1 public) have been created. Figure 4 also confirms the creation of 2 Route tables. From Figure 5, we can see that we have 1 IGW, newly created. Figure 6 reveals that 1 NACL has been created.

Figure 3: Panel showing our 2 Subnets

The screenshot shows the 'Subnets' panel in the AWS VPC console. It lists two subnets: 'cba-sam-subnet-private1-eu-west-2a' with ID 'subnet-04b0e4ee40959be59' and 'cba-sam-subnet-public1-eu-west-2a' with ID 'subnet-036bc011c90470bb2'. Both subnets are in an 'Available' state and are associated with the VPC 'vpc-0c553318c1c98f551'.

Name	Subnet ID	State	VPC ID
cba-sam-subnet-private1-eu-west-2a	subnet-04b0e4ee40959be59	Available	vpc-0c553318c1c98f551 cba-s...
cba-sam-subnet-public1-eu-west-2a	subnet-036bc011c90470bb2	Available	vpc-0c553318c1c98f551 cba-s...

Figure 4: Panel showing our 2 Route tables

The screenshot shows the 'Route tables' panel in the AWS VPC console. It lists two route tables: 'cba-sam-rtb-private1-eu-west-2a' with ID 'rtb-0c3b5ce8c33335886' and 'cba-sam-rtb-public' with ID 'rtb-0chc27c455b0bf2fe'. Both route tables are associated with the VPC 'vpc-0c553318c1c98f551' and have no explicit subnet associations.

Name	Route table ID	Explicit subnet associ...	Edge associations	Main
cba-sam-rtb-private1-eu-west-2a	rtb-0c3b5ce8c33335886	subnet-04b0e4ee40959be...	-	No
cba-sam-rtb-public	rtb-0chc27c455b0bf2fe	subnet-036bc011c90470...	-	No

Figure 5: Panel showing 1 Internet Gateway created

The screenshot shows the 'Internet gateways' panel in the AWS VPC console. It lists one Internet gateway: 'cba-sam-igw' with ID 'igw-388ec6b02800bcbdf'. The gateway is in an 'Attached' state and is associated with the VPC 'vpc-0c553318c1c98f551'.

Name	Internet gateway ID	State	VPC ID
cba-sam-igw	igw-388ec6b02800bcbdf	Attached	vpc-0c553318c1c98f551 cb

Figure 6: Panel showing 1 NACL (named: "cba-sam-NACL") created

The screenshot shows the 'Network ACLs' panel in the AWS VPC console. It lists two Network ACLs: 'cba-sam-NACL' with ID 'acl-0cb0347b29d478f6d' and an unnamed NACL with ID 'acl-03c0a3497419c4bd4'. The 'cba-sam-NACL' is associated with 2 subnets and is the default NACL for the VPC 'vpc-0c553318c1c98f551'.

Name	Network ACL ID	Associated with	Default	VPC ID
cba-sam-NACL	acl-0cb0347b29d478f6d	2 Subnets	Yes	vpc-0c553318c1c98f551 cb
-	acl-03c0a3497419c4bd4	3 Subnets	Yes	vpc-0c553318c1c98f551 cb

The Auto-assign for IPv6 and public IPv4 addresses was enabled. Please see Figure 7.

Figure 7: Panel showed enabled auto-assign setting for IPv6 and public IPv4

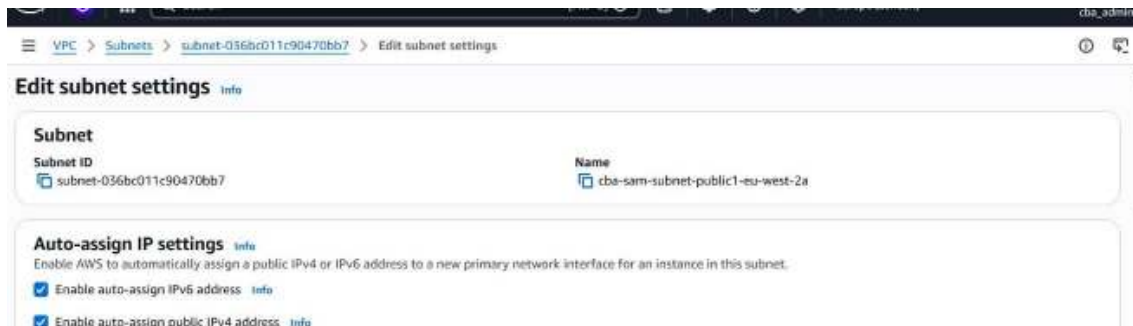


Figure 8: Panel showing successfully changed subnet settings



- The Subnet settings was saved.
- I confirmed that each of the 2 Subnets (1 Private, and 1 Public) has been explicitly associated with a Route Table. Please see Figure 9.

Figure 9: Panel showing each Subnet associated with a Route Table



I confirmed that the Public Route Table (*cba-sam-rtb-public*) has access to the Internet. Please see Figure 10.

Figure 10: Panel confirms that both IPv4 and IPv6 are connected to IGW



Action 2: Creating and Launching EC2 instance

- The **EC2 Instances** field was accessed from the EC2 dashboard on the console.
- The **Launch instance** button was clicked.
- The EC2 instance is named “*cba-sam_instance*”
- Amazon Linux AMI was used as template for the operating system I needed to preload the EC2 instance.
- t3 micro (free tier) was used as EC2 instance type.
- A keypair was created to enable me to connect to the EC2 instance, remotely, using SSH for Linux.
- Under the **Network settings**, I changed the VPC option to *cba-sam-vpc* to enable the EC2 instance inside the custom VPC and the Subnet, which I created. See Figure 11.

Figure 11: Panel showing the change to the newly created VPC: *cba-sam-vpc*

▼ Network settings [Info](#)

VPC - required [Info](#)

vpc-0c553318c1c98f551 (cba-sam-vpc)
10.0.0.0/16
2a05:d01c:b3c:2000::/56

▼

↻

- We changed to Public Subnet so that the EC2 instance can be launched⁸. Please see Figure 12.

Figure 12: Panel shows the select Public Subnet on which the EC2 instance will be launched.

Subnet [Info](#)

subnet-036bc011c90470bb7
cba-sam-subnet-public1-eu-west-2a

VPC: vpc-0c553318c1c98f551
Owner: 009850210027

Availability Zone: eu-west-2a (euw2-az2)
Zone type: Availability Zone

IP addresses available: 4091
CIDR: 10.0.0.0/20

▼

- The Auto-assign public IP was enabled.
- A Security group were created named “*cba_sam_security_group*”⁹

⁸ To be able to remotely connect to our EC2 instance after its launch, it must have a Public IP. This means that the EC2 instance needs to be launched in a Public Subnet.

⁹ The *cba_sam_security_group* is given this description: “allow internet access to web server.”

- I proceeded to create (except SSH at default) and define 3 Security group rules. See Figure 13:
 - For SSH (at default), I changed the **Source** type from anywhere to custom defined at 0.0.0.0/0
 - For HTTP and HTTPS, each changed to Source 0.0.0.0/0

Figure 13: Panel showing custom definition of source for SSH, HTTP, and HTTPS

The image shows three identical configuration panels for AWS Security Group Rules, one for SSH, HTTP, and HTTPS. Each panel has the following fields:

- Type:** A dropdown menu with the selected value (ssh, HTTP, or HTTPS) and an 'Info' link.
- Protocol:** A text field with the value 'TCP' and an 'Info' link.
- Port range:** A text field with the port number (22, 80, or 443).
- Source type:** A dropdown menu with the value 'Custom' and an 'Info' link.
- Source:** A text field with a search icon and placeholder text 'Add CIDR, prefix list or security'. Below it, a blue box contains the value '0.0.0.0/0' with a close 'X' icon.
- Description - optional:** A text field with placeholder text 'e.g. SSH for admin desktop' and an 'Info' link.

- After this I launched the EC2 instance. Figure 14 confirms the successful launch.

Figure 14: Panel showing that EC2 instance was successfully launched



- The EC2 instance in the Public Subnet was connected after ensuring the status check passed successfully¹⁰. Figure 15 reveals that the EC2 instance is running.

Figure 15: Panel showing that the EC2 instance is running

Instance summary for i-0e11aad63bc992999 (cba-sam_instance) [Info](#)

[Connect](#) [Instance state](#) [Actions](#)

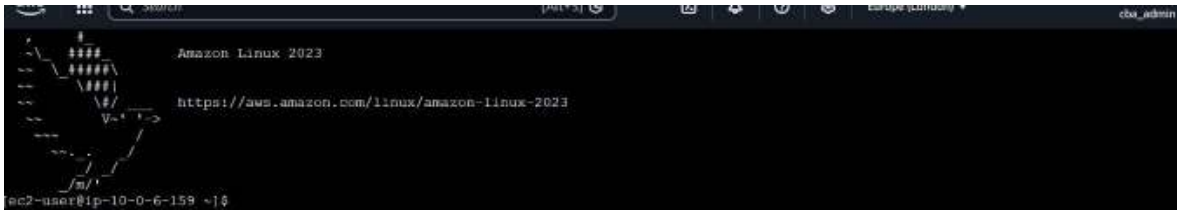
Updated 1 minute ago

Instance ID i-0e11aad63bc992999	Public IPv4 address 35.179.140.75 open address	Private IPv4 addresses 10.0.6.159
IPv6 address 2a05:d01c:b3c:2000:1369:8c01:7950:a90b	Instance state Running	Public DNS ec2-35-179-140-75.eu-west-2.compute.amazonaws.com open address
Hostname type IP name: ip-10-0-6-159.eu-west-2.compute.internal	Private IP DNS name (IPv4 only) ip-10-0-6-159.eu-west-2.compute.internal	Private resource DNS name -
Answer private resource DNS name -	Instance type t3.micro	Elastic IP addresses -
Auto-assigned IP address 35.179.140.75 [Public IP]	VPC ID vpc-0c553318c1c98f551 (cba-sam-vpc)	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. Learn more
Auto-assigned IP address 35.179.140.75 [Public IP]	VPC ID vpc-0c553318c1c98f551 (cba-sam-vpc)	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. Learn more
IAM role -	Subnet ID subnet-036bc011c90470bb7 (cba-sam-subnet-public1-eu-west-2a)	Auto Scaling Group name -
IMDSv2 Required	Instance ARN arn:aws:ec2:eu-west-2:009850210027:instance/i-0e11aad63bc992999	Managed false

- The SSH terminal opens in a new tab. Please see Figure 16.

¹⁰ The connection was made using Public IP address (public IPv4 or IPv6 address).

Figure 16: Panel shows SSH terminal opened in a new tab



Action 3: Installing Apache Webserver (httpd) on the EC2 instance

Screenshots of each step are attached as appendix to this report (please see pages 12 to 14).

The following actions were performed:

- Changed to root user type by typing the command: **sudo su**
- Ran an update on the operating system packages by typing the command: **yum update -y**
- Installed the Apache by typing the command: **yum install httpd -y**
- Checked that the web server has been installed by typing the command:
cd /var/www/html
- Created an index file for the website within the HTML folder by typing the command:
touch index.html
- Added content to index file by typing the command: **vi index.html**
- Typed the letter **i** to make the document editable.
- Copied the content of this:
<https://github.com/SafeEHA/simple-portfolio/blob/main/index.html>
- Pasted the content within the vi editor (using mouse right click).
- Pressed the keyboard Esc key once done editing the document.
- Saved and exited the file by typing the command **:wq**
- Listed the files in the html folder. The **index.html** was seen.
- Viewed the content of the index.html file by typing the command: **cat index.html**
- Turned on Apache by typing the command: **service httpd start**
- Confirmed that the Apache server is running by typing the: **service httpd status**
- Typed the command to ensure the Apache service comes up whenever my server reboots:
chkconfig on

Action 4: Validating the website is working fine

- On the AWS console, I clicked on the EC2 instance-ID

Figure 17: Panel shows that EC2 instance ID from the AWS console



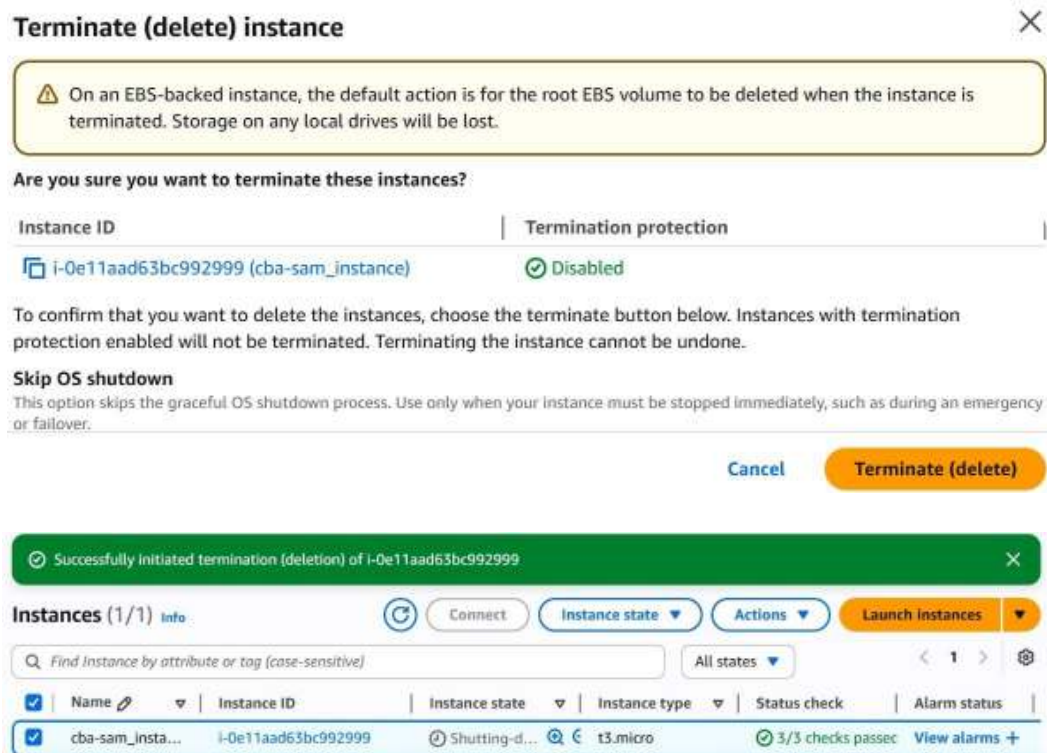
- Copied the Public IPv4 url and ran it on a new tab of my browser. The following webpage appeared with the content in the index.html. Please see Figure 18.

Figure 18: Panel showing the webpage



- The EC2 instance was later terminated after ensuring that everything was fine. Please see Figure 19.

Figure 19: Panel confirming that EC2 instance has been terminated



Summary:

- The public subnet is now connected to internet.
- The Public Subnet now has 1 EC2 instance provisioned on it.
- Three inbound security group rules allowing SSH, HTTP, and HTTPS were configured for the EC2 instance.


```
[root@ip-10-0-6-159 html]# ls
index.html
[root@ip-10-0-6-159 html]#
```

```
<p>GitHub: <a href="https://github.com/your handle" target="_blank">github.com/your handle</a></p>
<p>LinkedIn: <a href="https://www.linkedin.com" target="_blank">Add your LinkedIn URL</a></p>
</div>

<div class="section title">Status</div>
<p class="chip">
  Open to Remote • Cloud / DevOps / DevRel
</p>
</aside>
</div>
</main>

<footer>
  Served from <code>/var/www/html/index.html</code> on an Apache web server running on Amazon EC2.
</footer>
</body>
</html>
[root@ip-10-0-6-159 html]#
```

```
Redirecting to /bin/systemctl start httpd.service
[root@ip-10-0-6-159 html]# service httpd status
Redirecting to /bin/systemctl status httpd.service
• httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Active: active (running) since Sat 2026-02-28 17:16:03 UTC; 1min 51s ago
     Docs: man:httpd.service(8)
   Main PID: 26953 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served/sec: 0 B/sec"
    Tasks: 177 (limit: 1009)
  Memory: 13.6M
     CPU: 162ms
   CGroup: /system.slice/httpd.service
           └─26953 /usr/sbin/httpd -DFOREGROUND
             └─26954 /usr/sbin/httpd -DFOREGROUND
               └─26955 /usr/sbin/httpd -DFOREGROUND
                 └─26956 /usr/sbin/httpd -DFOREGROUND
```



Sign in

SafeEHA / simple-portfolio Public

Notifications Fork 0 Star 0

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#)

 main simple-portfolio / index.html 

Go to file

 SafeEHA Remove 'Safe' from job title in index.html

599870a · 4 days ago

264 lines (254 loc) · 6.94 KB

CodeBlame

Raw

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>Cloud & DevOps Engineer Portfolio</title>
7    <style>
8      :root {
9        --bg: #020617;
10       --card: #020617;
11       --accent: #38bdf8;
12       --accent-soft: #0ea5e9;
13       --text: #e5e7eb;
14       --muted: #9ca3af;
15       --chip-bg: #22c55e;
```