

Secure a Web Server using EC2, Subnets, and Firewalls

Step 1: Create a VPC and subnet

1. Sign in to the AWS Console and open the **VPC** service.
2. Create the VPC
 - Navigation: **VPC → Your VPCs → Create VPC**.
 - Settings:
 - **Name tag:** MyVPC
 - **IPv4 CIDR block:** 10.0.0.0/16
 - **Tenancy:** Default

The screenshot shows the 'Create VPC' page in the AWS console. The 'Resources to create' section has 'VPC only' selected. The 'Name tag - optional' is 'MyVPC'. The 'IPv4 CIDR block' section has 'No IPv4 CIDR manual input' selected, and the 'IPv4 CIDR' is '10.0.0.0/16'. The 'IPv6 CIDR block' section has 'No IPv6 CIDR block' selected. The 'Tenancy' is set to 'Default'. A tag is added with key 'Name' and value 'MyVPC'. At the bottom, there are buttons for 'Cancel', 'Preview code', and 'Create VPC'.

- Click **Create VPC (or Create)**.
3. Create the subnet
 - Navigation: **VPC → Subnets → Create subnet**.
 - Settings:
 - **VPC:** select MyVPC
 - **Name tag:** WebSubnet

Shaikh Ateeb Ahmed

22-08-2025

- **Availability Zone:** choose one (example: ap-south-1a)
- **IPv4 CIDR block:** 10.0.1.0/24

Create subnet [info](#)

VPC

VPC ID
Create subnets in this VPC.
vpc-09776ec9c23089eab (MyVPC)

Associated VPC CIDRs
IPv4 CIDRs
10.0.0.0/16

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
WebSubnet
The name can be up to 255 characters long.

Availability Zone [info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
Asia Pacific (Mumbai) / ap-south-1a

IPv4 VPC CIDR block [info](#)
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
10.0.0.0/16

IPv4 subnet CIDR block
10.0.1.0/24 250 IP

▼ Tags - optional

Key Value - optional
Name WebSubnet Remove
Add new tag
You can add 49 more tags.
Remove Add new subnet

Cancel Create subnet

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

- Click **Create subnet**.
4. Enable auto-assign public IPv4 on the subnet
- Navigation: **VPC** → **Subnets** → click the row for WebSubnet.
 - Actions: **Actions** → **Edit subnet settings**.
 - In the dialog: check **Enable auto-assign public IPv4** → **Save**.

Shaikh Ateeb Ahmed

22-08-2025

Edit subnet settings [info](#)

Subnet

Subnet ID: [subnet-0ba13c277433127f1](#) Name: [WebSubnet](#)

Auto-assign IP settings [info](#)

Enable AWS to automatically assign a public IPv4 or IPv6 address to a new primary network interface for an instance in this subnet.

☒ Enable auto-assign public IPv4 address [info](#)

☐ Enable auto-assign customer-owned IPv4 address [info](#)
Option disabled because no customer-owned pools found.

Resource-based name (RBN) settings [info](#)

Specify the hostname type for EC2 instances in this subnet and optional RBN DNS query settings.

☐ Enable resource name DNS A record on launch [info](#)

☐ Enable resource name DNS AAAA record on launch [info](#)

Hostname type [info](#)

☐ Resource name

☒ IP name

DNS64 settings

Enable DNS64 to allow IPv6-only services in Amazon VPC to communicate with IPv4-only services and networks.

☐ Enable DNS64 [info](#)

[Cancel](#) [Save](#)

Step 2: Create an Internet Gateway & Attach to VPC

1. Open the VPC service in the AWS Management Console.
2. Create the Internet Gateway
 - Navigation: VPC → Internet Gateways → Create internet gateway.
 - In the dialog: set Name tag to MyIGW.

Shaikh Ateeb Ahmed

22-08-2025

The screenshot shows the AWS Management Console interface for creating a new internet gateway. The breadcrumb navigation at the top indicates the path: VPC > Internet gateways > Create internet gateway. The main heading is 'Create internet gateway' with an 'Info' link. Below this, a descriptive sentence states: 'An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.' The form is divided into two main sections. The first section, 'Internet gateway settings', contains a 'Name tag' field with the value 'MyIGW' and a description: 'Creates a tag with a key of 'Name' and a value that you specify.' The second section, 'Tags - optional', explains that a tag is a label for an AWS resource and provides a table to manage tags. The table has two columns: 'Key' and 'Value - optional'. A single tag is listed with 'Name' as the key and 'MyIGW' as the value. There are buttons to 'Add new tag' and 'Remove'. At the bottom right of the form are 'Cancel' and 'Create internet gateway' buttons. The footer of the console shows 'CloudShell', 'Feedback', and copyright information for Amazon Web Services, Inc.

AWS CloudShell Feedback [Alt+S] Asia Pacific (Mumbai) Account ID: 6780-2505-6602 Shaikh Ateeb Ahmed

VPC > Internet gateways > Create internet gateway

Create internet gateway [Info](#)

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

Internet gateway settings

Name tag
Creates a tag with a key of 'Name' and a value that you specify.

MyIGW

Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
Name	MyIGW

[Add new tag](#)
You can add 49 more tags.

[Cancel](#) [Create internet gateway](#)

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

- Click Create internet gateway (or Create).
- 3. Attach the IGW to your VPC**
- Still under Internet Gateways, select the row for MyIGW.
 - Click Actions → Attach to VPC.
 - Choose MyVPC from the list and click Attach internet gateway.

The screenshot shows the 'Attach to VPC' page in the AWS Management Console. The breadcrumb navigation is: VPC > Internet gateways > Attach to VPC (igw-0b89eecb9bf640a2). The main heading is 'Attach to VPC (igw-0b89eecb9bf640a2)' with an 'Info' link. The form is titled 'VPC' and contains the instruction: 'Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.' Under the heading 'Available VPCs', there is a search bar with the text 'vpc-08776ec9c23089eab' and a dropdown arrow. Below the search bar is a section for the 'AWS Command Line Interface command'. At the bottom right are 'Cancel' and 'Attach internet gateway' buttons.

AWS CloudShell Feedback [Alt+S] Asia Pacific (Mumbai) Account ID: 6780-2505-6602 Shaikh Ateeb Ahmed

VPC > Internet gateways > Attach to VPC (igw-0b89eecb9bf640a2)

Attach to VPC (igw-0b89eecb9bf640a2) [Info](#)

Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

VPC

Available VPCs
Attach the internet gateway to this VPC.

vpc-08776ec9c23089eab

[AWS Command Line Interface command](#)

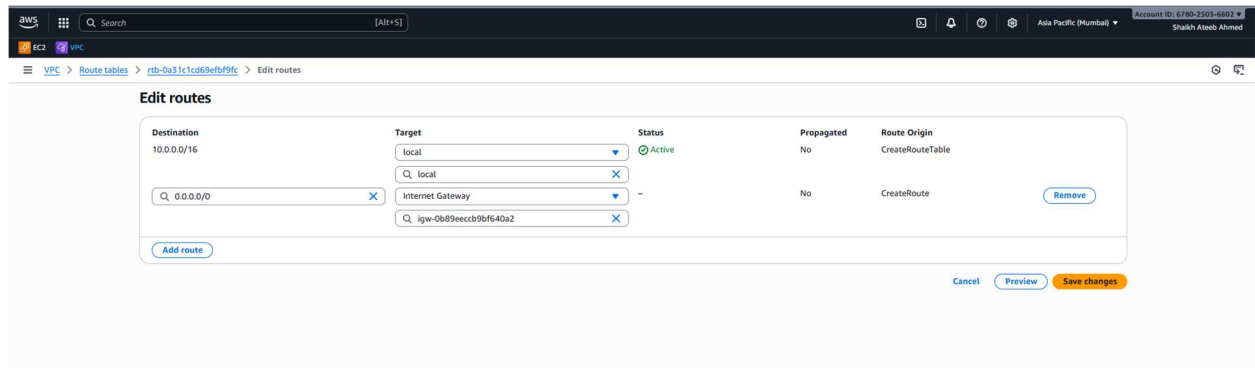
[Cancel](#) [Attach internet gateway](#)

- Verify the VPC column for MyIGW shows MyVPC (or the VPC ID).
- 4. Add a route to the IGW**
- Navigation: VPC → Route Tables.
 - Select the route table that MyVPC uses.

Shaikh Ateeb Ahmed

22-08-2025

- Click Routes → Edit routes → Add route.
- Destination: 0.0.0.0/0
- Target: choose Internet gateway and select MyIGW (it will show igw-xxxxx with the name).



- Click Save routes.

Step 3: Create a Security Group (SG)

1. Open the VPC service in the AWS Console.
2. Go to Security Groups
 - Navigation: VPC → Security Groups.
3. Create the security group
 - Click Create security group.
 - Name tag: WebServer-SG
 - Description: Allow HTTP/HTTPS; restrict SSH
 - VPC: select MyVPC
4. Add inbound rules
 - Add rule for each of the following:
 - Rule 1 — Type: HTTP
 - Protocol: TCP (auto)
 - Port range: 80
 - Source: Anywhere → enter 0.0.0.0/0

Shaikh Ateeb Ahmed

22-08-2025

- Rule 2 — Type: HTTPS
- Protocol: TCP
- Port range: 443
- Source: Anywhere → 0.0.0.0/0
- Rule 3 — Type: SSH
- Protocol: TCP
- Port range: 22
- Source: My IP (recommended) — your public IP.

Create security group

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name info

Description info

VPC info

Inbound rules

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	Anywhere	0.0.0.0/0
HTTPS	TCP	443	Anywhere	0.0.0.0/0
SSH	TCP	22	My IP	120.82.142.105/32

Outbound rules

Type	Protocol	Port range	Destination	Description - optional
All traffic	All	All	Custom	0.0.0.0/0

Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Create security group

- Click Create security group (or Create).

5. Verify outbound rules

- By default Security Groups allow all outbound traffic. Confirm Outbound rules tab shows a rule: All traffic → 0.0.0.0/0.

Step 4: Launch an EC2 instance in the subnet

1. EC2 → Launch Instance.

Shaikh Ateeb Ahmed

22-08-2025

- Name tag: MyWebServer.
- AMI: Amazon Linux 2 (free tier).
- Instance type: t3.micro.
- Network: MyVPC; Subnet: WebSubnet.
- Auto-assign Public IP: Enabled (should be set by subnet attribute).
- Security Group: choose WebServer-SG.
- Key pair: Create new key pair or select existing. Download .pem (store it safely).

Launch an instance

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags

Name: [Add additional tags](#)

Application and OS Images (Amazon Machine Image)

Search our full catalog including 1000s of application and OS images.

Amazon Linux 2023 kernel 4.14 AMI
ami-0900111111111111 / ami-0900111111111111 (64-bit x86_64, x64)
Virtualization type: x86_64-hvm

Instance type

t3.micro
Free tier eligible

Key pair (login)

Key pair name: [Create new key pair](#)

Network settings

VPC: [Select a VPC](#)
Subnet: [Select a Subnet](#)

Summary

Number of instances:

Software Image (AMI)
Amazon Linux 2023 AMI 2023.9.2... [Read more](#)
ami-0900111111111111

Virtual server type (Instance type)
t3.micro

Firewall (Security group)
WebServer-SG

Storage (optional)
1 volume(s) - 8 GiB

[Launch instance](#) [Preview costs](#)

[illegible][illegible]

2. On the EC2 instance, update and install Apache:

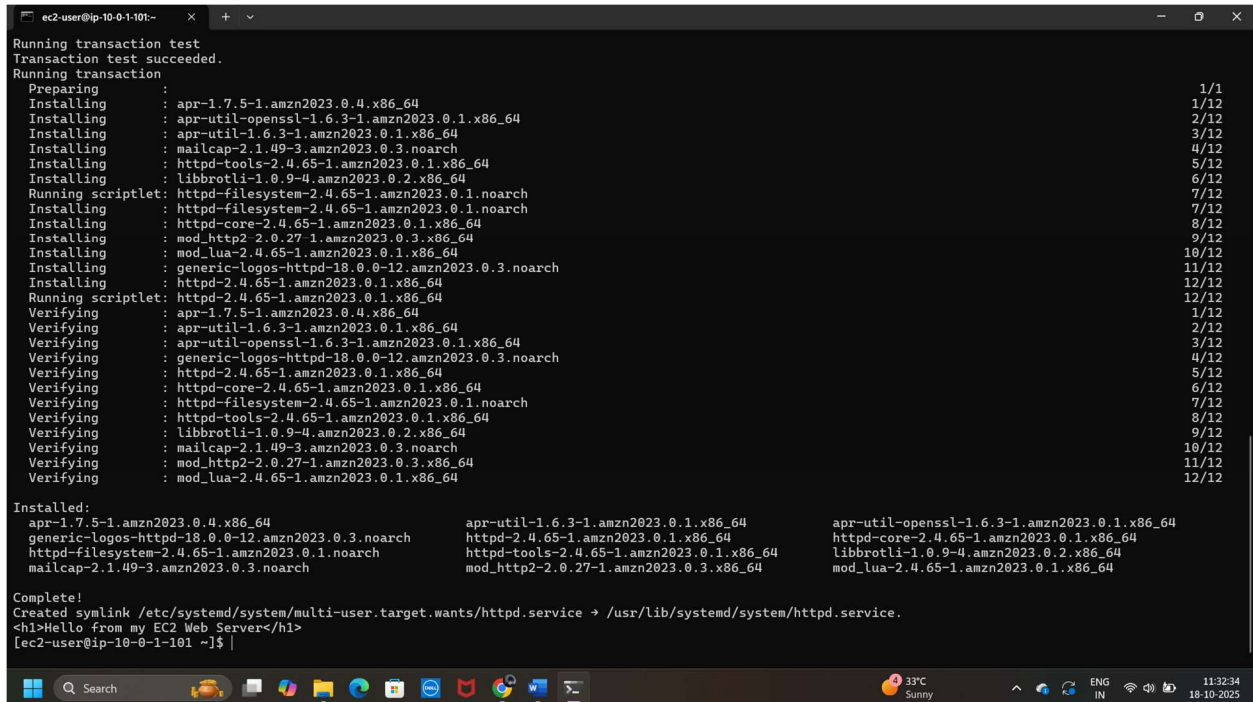
```
sudo yum update -y
```

```
sudo yum install -y httpd
```

```
sudo systemctl start httpd
```

```
sudo systemctl enable httpd
```

```
echo "<h1>Hello from my EC2 Web Server</h1>" | sudo tee /var/www/html/index.html
```



```
Running transaction test
Transaction test succeeded.
Running transaction
Preparing : apr-1.7.5-1.amzn2023.0.4.x86_64 1/1
Installing : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64 1/12
Installing : apr-util-1.6.3-1.amzn2023.0.1.x86_64 2/12
Installing : mailcap-2.1.49-3.amzn2023.0.3.noarch 3/12
Installing : httpd-tools-2.4.65-1.amzn2023.0.1.x86_64 4/12
Installing : libbrotli-1.0.9-4.amzn2023.0.2.x86_64 5/12
Installing : httpd-filesystem-2.4.65-1.amzn2023.0.1.noarch 6/12
Running scriptlet: httpd-filesystem-2.4.65-1.amzn2023.0.1.noarch 7/12
Installing : httpd-filesystem-2.4.65-1.amzn2023.0.1.noarch 7/12
Installing : httpd-core-2.4.65-1.amzn2023.0.1.x86_64 8/12
Installing : mod_http2-2.0.27-1.amzn2023.0.3.x86_64 9/12
Installing : mod_lua-2.4.65-1.amzn2023.0.1.x86_64 10/12
Installing : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch 11/12
Installing : httpd-2.4.65-1.amzn2023.0.1.x86_64 12/12
Running scriptlet: httpd-2.4.65-1.amzn2023.0.1.x86_64 12/12
Verifying : apr-1.7.5-1.amzn2023.0.4.x86_64 1/12
Verifying : apr-util-1.6.3-1.amzn2023.0.1.x86_64 2/12
Verifying : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64 3/12
Verifying : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch 4/12
Verifying : httpd-2.4.65-1.amzn2023.0.1.x86_64 5/12
Verifying : httpd-core-2.4.65-1.amzn2023.0.1.x86_64 6/12
Verifying : httpd-filesystem-2.4.65-1.amzn2023.0.1.noarch 7/12
Verifying : httpd-tools-2.4.65-1.amzn2023.0.1.x86_64 8/12
Verifying : libbrotli-1.0.9-4.amzn2023.0.2.x86_64 9/12
Verifying : mailcap-2.1.49-3.amzn2023.0.3.noarch 10/12
Verifying : mod_http2-2.0.27-1.amzn2023.0.3.x86_64 11/12
Verifying : mod_lua-2.4.65-1.amzn2023.0.1.x86_64 12/12

Installed:
apr-1.7.5-1.amzn2023.0.4.x86_64          apr-util-1.6.3-1.amzn2023.0.1.x86_64          apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch  httpd-2.4.65-1.amzn2023.0.1.x86_64          httpd-core-2.4.65-1.amzn2023.0.1.x86_64
httpd-filesystem-2.4.65-1.amzn2023.0.1.noarch  httpd-tools-2.4.65-1.amzn2023.0.1.x86_64    libbrotli-1.0.9-4.amzn2023.0.2.x86_64
mailcap-2.1.49-3.amzn2023.0.3.noarch          mod_http2-2.0.27-1.amzn2023.0.3.x86_64      mod_lua-2.4.65-1.amzn2023.0.1.x86_64

Complete!
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
<h1>Hello from my EC2 Web Server</h1>
[ec2-user@ip-10-0-1-101 ~]$
```

4. On your browser open: <http://<public-ip>> → You should see the Hello message.



Hello from my EC2 Web Server

Step 6: Add NACL Rules

1. VPC → Network ACLs → Create network ACL.

- Name: WebNACL.
- VPC: MyVPC.

The screenshot shows the 'Create network ACL' page in the AWS Management Console. The page has a dark header with the AWS logo and navigation icons. The breadcrumb trail is 'VPC > Network ACLs > Create network ACL'. The main content area is titled 'Create network ACL' with a subtitle 'A network ACL is an optional layer of security that acts as a firewall for controlling traffic in and out of a subnet.' Below this, there are two sections: 'Network ACL settings' and 'Tags'. In the 'Network ACL settings' section, the 'Name' field is 'WebNACL' and the 'VPC' dropdown is set to 'vpc-08776ec9c23089eab (MyVPC)'. In the 'Tags' section, there is a table with columns 'Key' and 'Value - optional'. A tag is added with 'Key' as 'Name' and 'Value' as 'WebNACL'. At the bottom right, there are 'Cancel' and 'Create network ACL' buttons.

2. Associate the NACL with WebSubnet (Subnet Associations → Edit → add WebSubnet).

The screenshot shows the 'Edit subnet associations' page in the AWS Management Console. The breadcrumb trail is 'VPC > Network ACLs > acl-083599ff4a3775635 / WebNACL > Edit subnet associations'. The main content area is titled 'Edit subnet associations' with a subtitle 'Change which subnets are associated with this network ACL.' Below this, there are two sections: 'Available subnets (1/1)' and 'Selected subnets'. The 'Available subnets' section has a table with columns: 'Name', 'Subnet ID', 'Associated with', 'Availability Zone', 'IPv4 CIDR', and 'IPv6 CIDR'. One subnet is listed: 'WebSubnet' with Subnet ID 'subnet-0ba13c277453127f1', Associated with 'acl-014ac3d06381d6d116', Availability Zone 'aps1-az1 (ap-south-1a)', IPv4 CIDR '10.0.1.0/24', and IPv6 CIDR '-'. The 'Selected subnets' section shows 'subnet-0ba13c277453127f1 / WebSubnet' with a close button. At the bottom right, there are 'Cancel' and 'Save changes' buttons.

3. Add entries (NACL uses rule numbers; processed lowest → highest):

- Inbound:
 - Rule 100: Allow TCP 80 (HTTP 80) (Source 0.0.0.0/0)
 - Rule 110: Allow TCP 443 (HTTPS 443) (Source 0.0.0.0/0)
 - Rule 120: Allow TCP 22 (SSH 22) (Source YOUR_IP/32) (120.62.143.103/32)

Shaikh Ateeb Ahmed

22-08-2025

The screenshot shows the AWS Management Console interface for editing inbound rules. The breadcrumb navigation is: VPC > Network ACLs > acl-08356d8ff4a377635 / WebNACL > Edit inbound rules. The page title is "Edit inbound rules" with a sub-header "Inbound rules control the incoming traffic that's allowed to reach the VPC." Below this is a table with columns: Rule number, Type, Protocol, Port range, Source, and Allow/Deny. There are four rules listed: Rule 100 (HTTP 80, TCP, 80, 0.0.0.0/0, Allow), Rule 110 (HTTPS 443, TCP, 443, 0.0.0.0/0, Allow), Rule 120 (SSH 22, TCP, 22, 120.62.143.103/32, Allow), and Rule * (All traffic, All, All, 0.0.0.0/0, Deny). At the bottom right are buttons for "Cancel", "Preview changes", and "Save changes".

Rule number	Type	Protocol	Port range	Source	Allow/Deny
100	HTTP (80)	TCP (80)	80	0.0.0.0/0	Allow
110	HTTPS (443)	TCP (443)	443	0.0.0.0/0	Allow
120	SSH (22)	TCP (22)	22	120.62.143.103/32	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

- **Outbound:**
- **Rule 100: Allow All TCP/UDP/ICMP to 0.0.0.0/0**

The screenshot shows the AWS Management Console interface for editing outbound rules. The breadcrumb navigation is: VPC > Network ACLs > acl-08356d8ff4a377635 / WebNACL > Edit outbound rules. The page title is "Edit outbound rules" with a sub-header "Outbound rules control the outgoing traffic that's allowed to leave the VPC." Below this is a table with columns: Rule number, Type, Protocol, Port range, Destination, and Allow/Deny. There are two rules listed: Rule 100 (All traffic, All, All, 0.0.0.0/0, Allow) and Rule * (All traffic, All, All, 0.0.0.0/0, Deny). At the bottom right are buttons for "Cancel", "Preview changes", and "Save changes".

Rule number	Type	Protocol	Port range	Destination	Allow/Deny
100	All traffic	All	All	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

4. **Save.**
5. **Test again → Website should still be reachable only on allowed ports.**

The screenshot shows a web browser window with the address bar displaying "65.2.38.33" and a "Not secure" warning. The main content of the page is the text "Hello from my EC2 Web Server".

Hello from my EC2 Web Server