

Securing Custom Domains with HTTPS on AWS

In this tutorial, we are going to study Domain Management with Route 53 and SSL/TLS Certificate Handling with AWS Certificate Manager.

DNS, or the Domain Name System, translates human readable domain names (for example, www.amazon.com) to machine readable IP addresses (for example, 192.0.2.44).

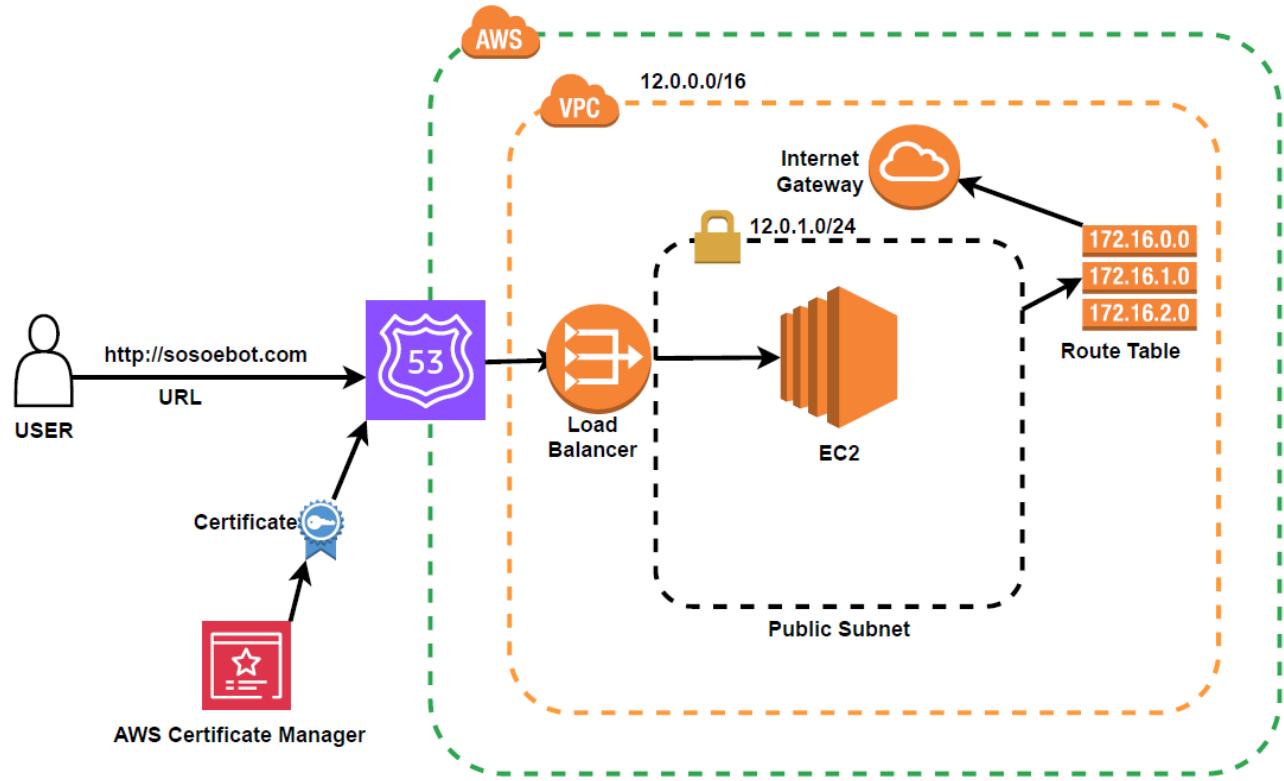
All computers on the Internet, from your smart phone or laptop to the servers that serve content for massive retail websites, find and communicate with one another by using numbers. These numbers are known as IP addresses. When you open a web browser and go to a website, you don't have to remember and enter a long number. Instead, you can enter a domain name like example.com and still end up in the right place.

A DNS service such as **Amazon Route 53** is a globally distributed service that translates human readable names like www.example.com into the numeric IP addresses like 192.0.2.1 that computers use to connect to each other. The Internet's DNS system works much like a phone book by managing the mapping between names and numbers. DNS servers translate requests for names into IP addresses, controlling which server an end user will reach when they type a domain name into their web browser. These requests are called queries.

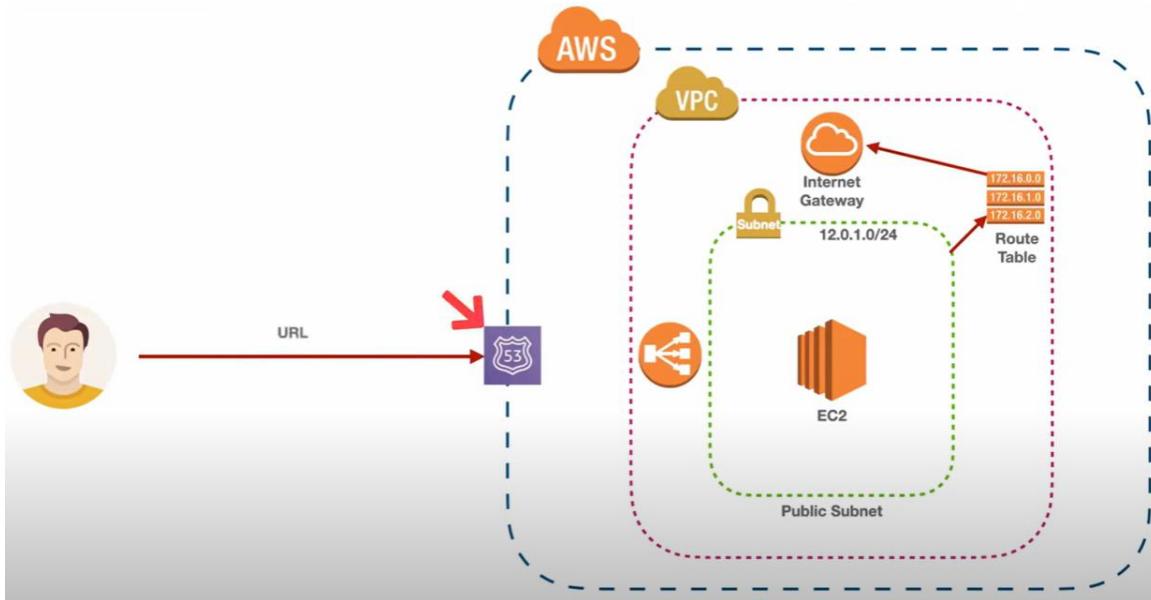
DEMO PREVIEW

In this tutorial, we are going to talk about AWS Certificate Manager. Here is what we are going to implement in our AWS Certificate Manager demo.

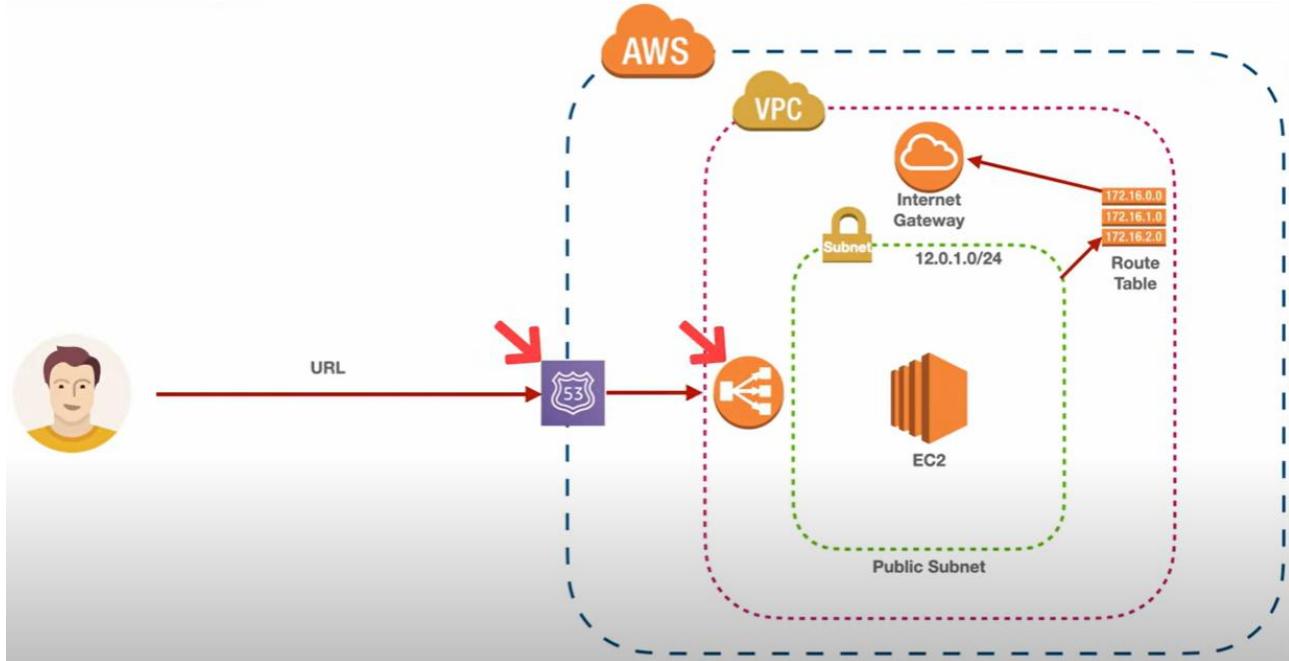
ARCHITECTURE



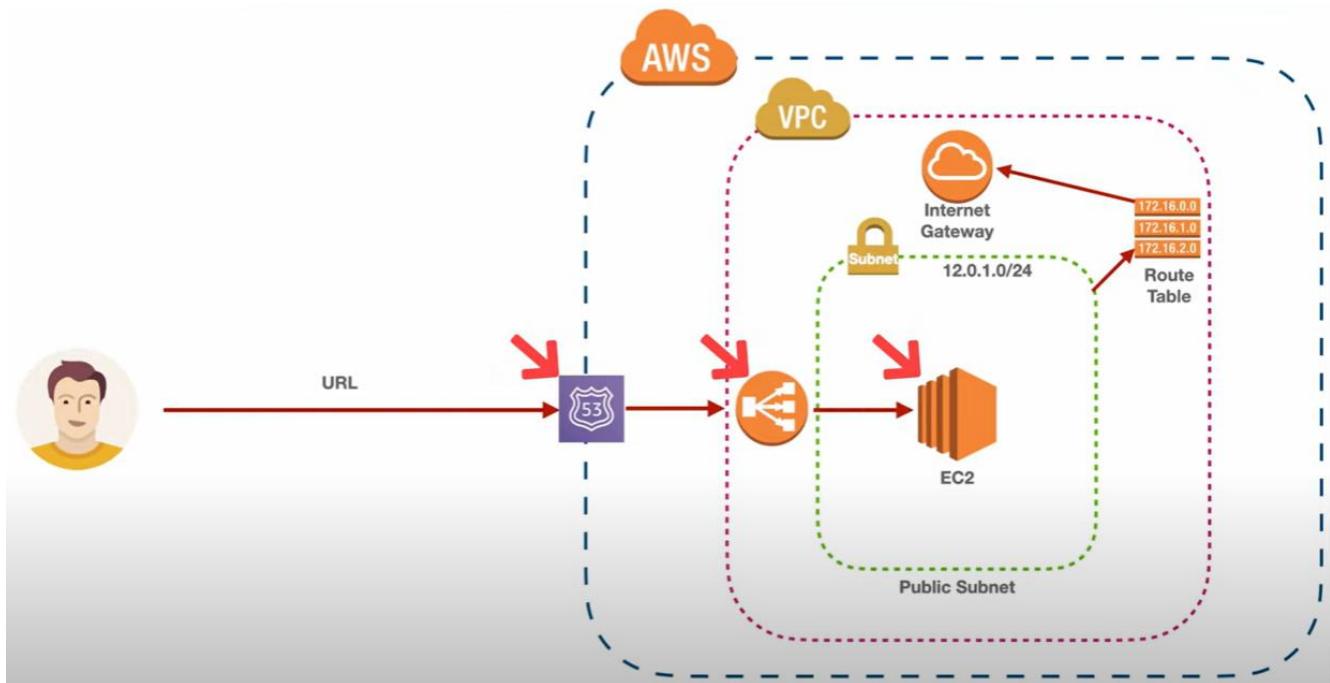
First, the user is going to make a request using the URL that request is going to land on our **AWS Route 53**.



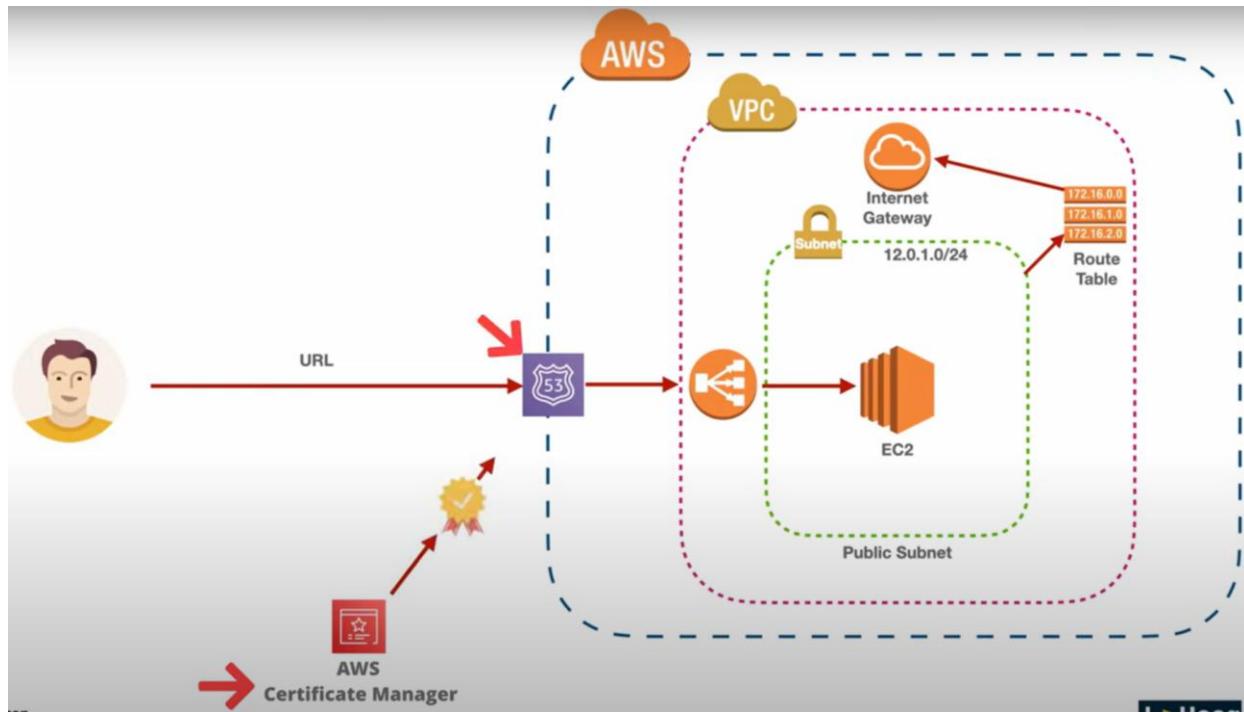
Route 53 is going to forward that request to our **application load balancer**.



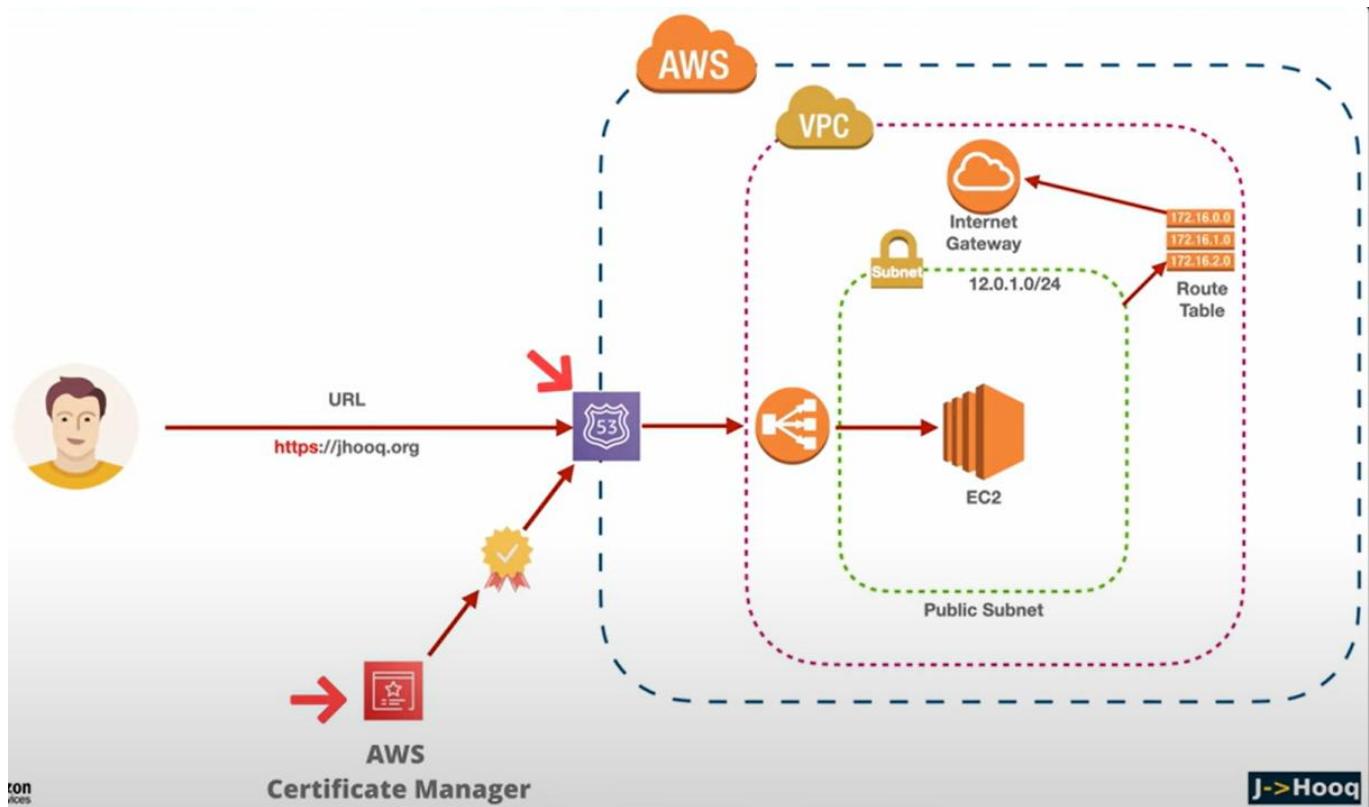
That Load Balancer is going to point to our **EC2 instance** where our application is running.



To secure that URL, we are going to use **AWS Certificate Manager**, which is going to issue the certificate to us and we are going to use those certificate records inside our AWS Route 53.

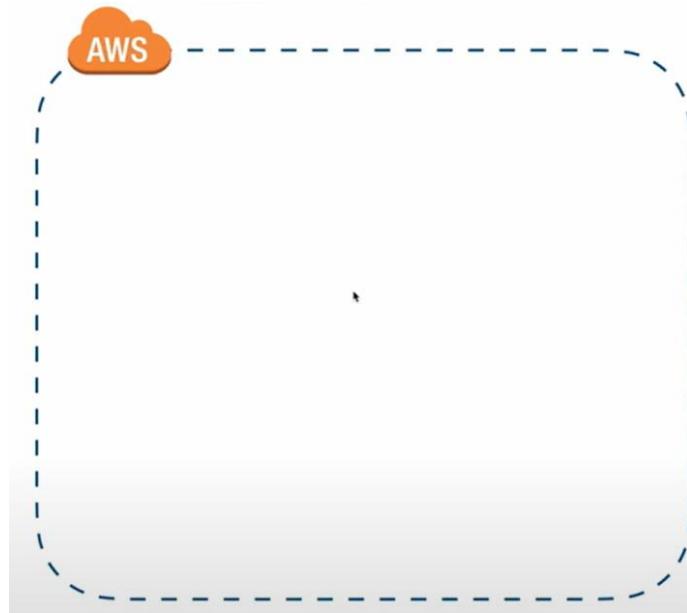


Once we have imported those records inside our Route 53, then we are going to use the same URL, but this time we are going to use the **HTTPS**.

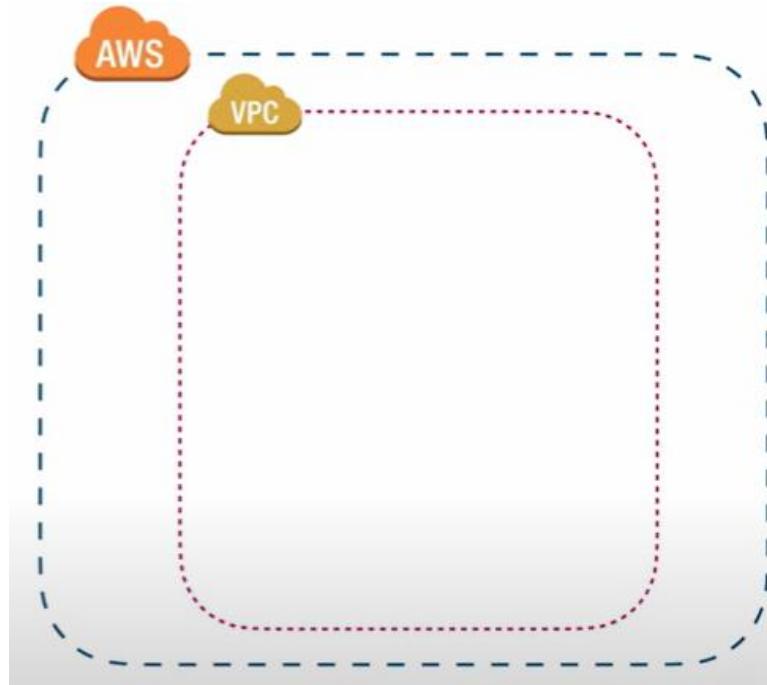


SETTING UP ARCHITECTURE COMPONENTS

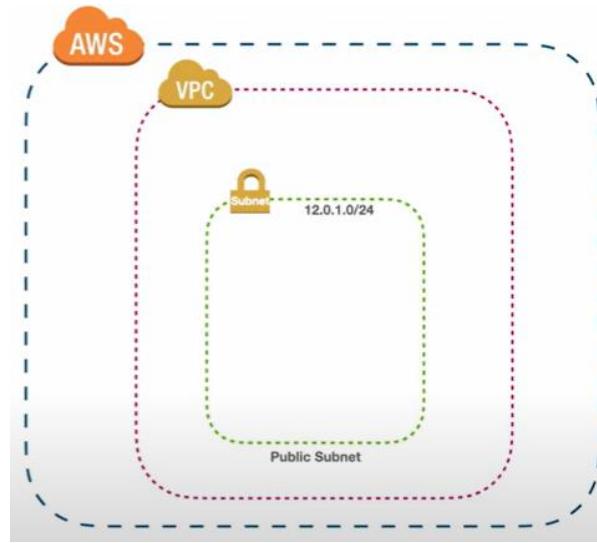
Let us break down this whole implementation into multiple steps. The first thing we will start with is our **AWS Account**.



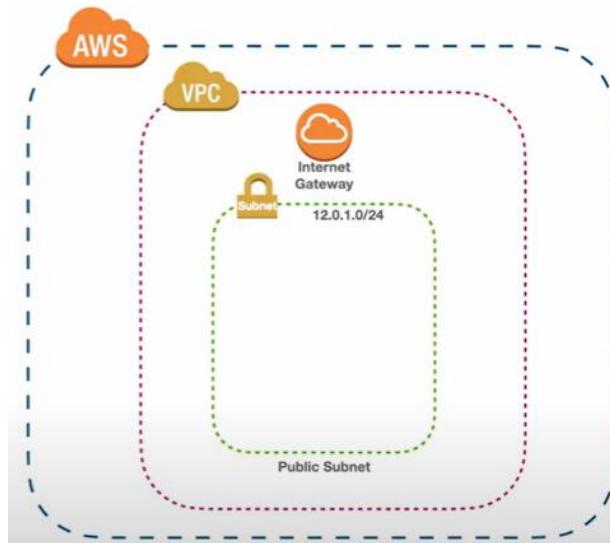
Secondly, we are going to set up our **VPC** within the AWS Account.



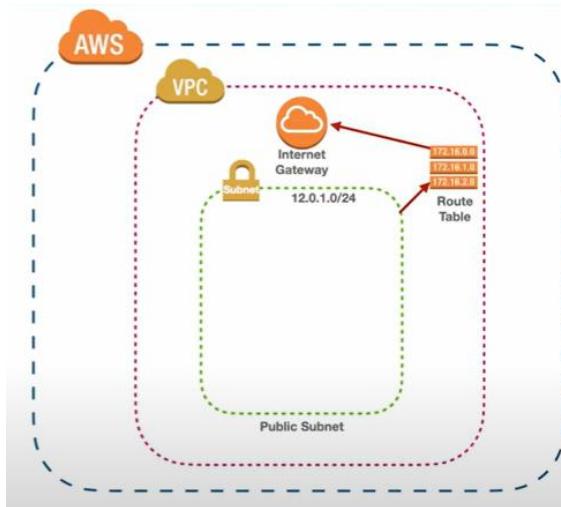
Thirdly, we are going to create a **subnet** inside that particular VPC.



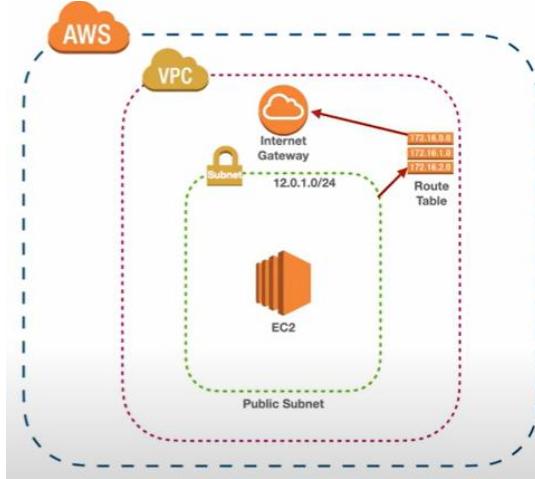
Fourthly, we are going to we are going to set up our **Internet Gateway**.



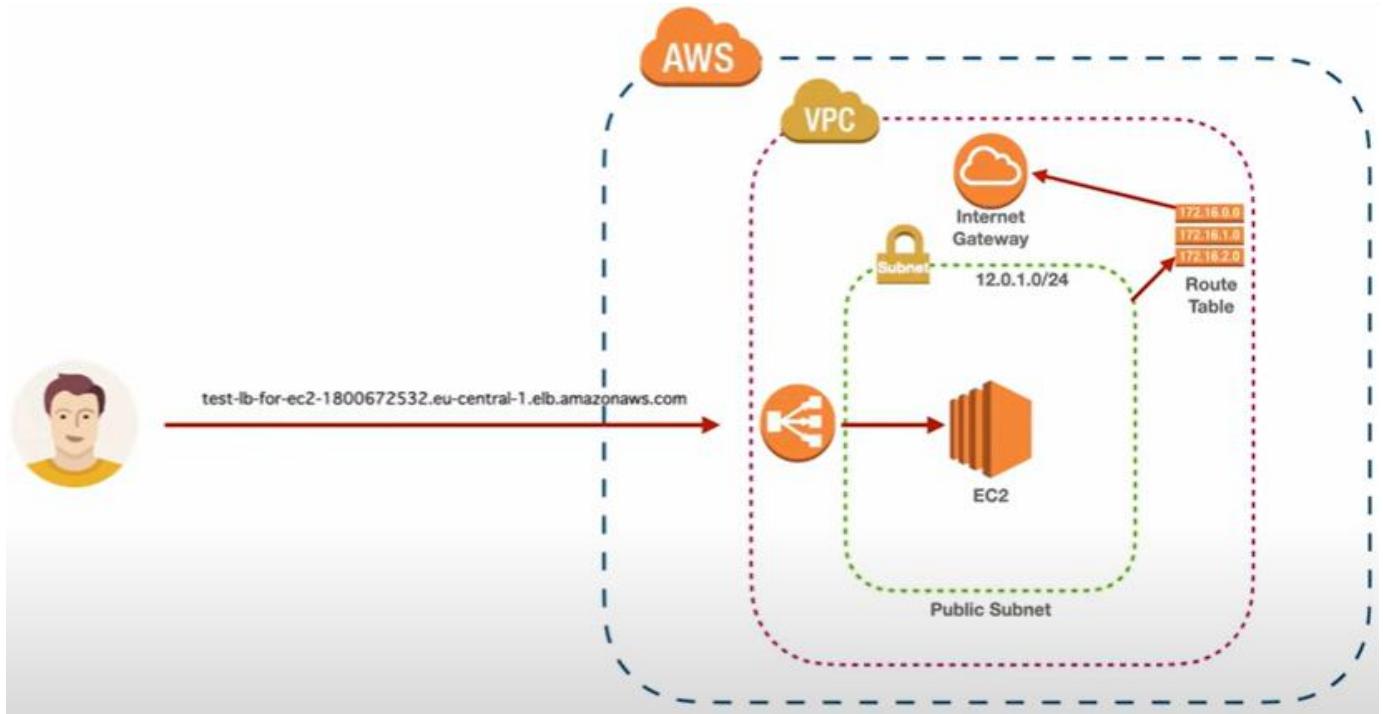
Fifth, we are going to set up our **Route Table**.



Sixth, we are going to set up our **EC2 instance**.



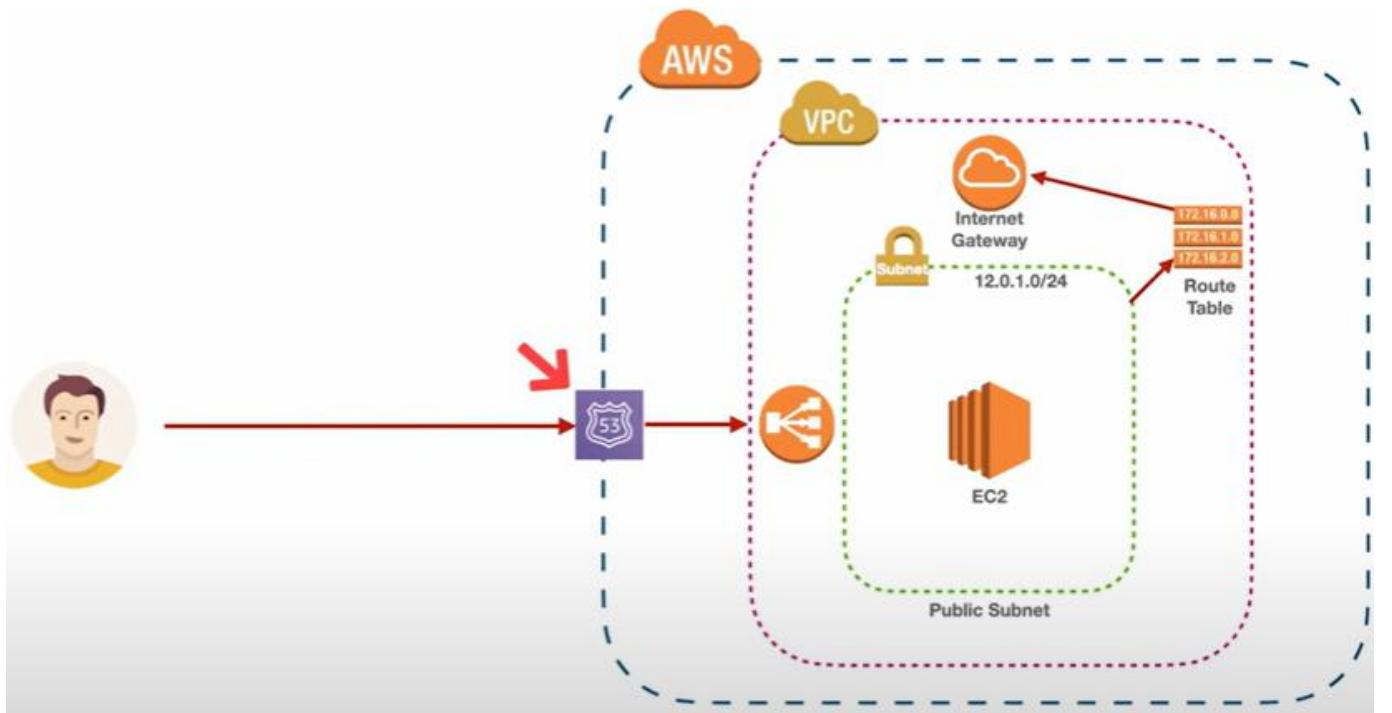
After setting up our EC2 instance, then we are going to create an **Application Load Balancer** so that a user can access our application running in the EC2 instance via the load balancer. For that we are going to use a DNS name of our Load Balancer.



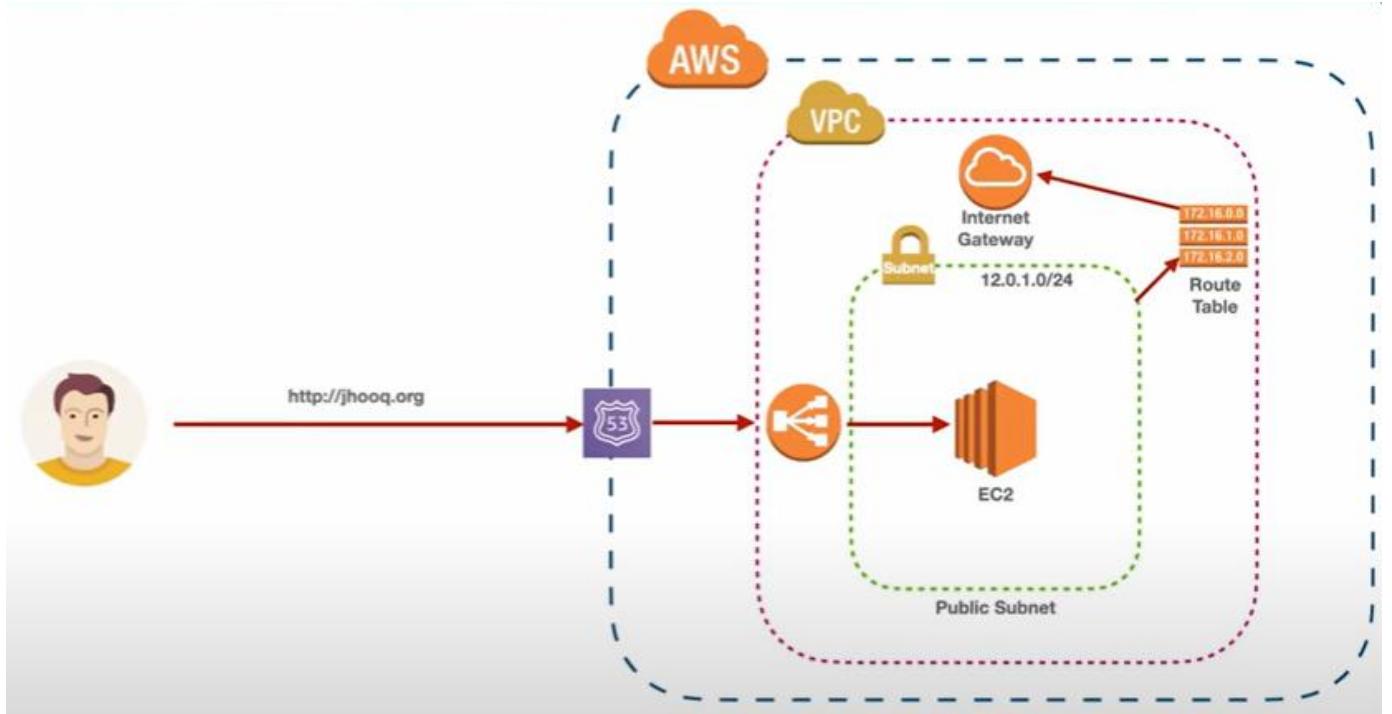
After setting up the load balancer, the next thing we are going to configure is our **AWS Route 53**.



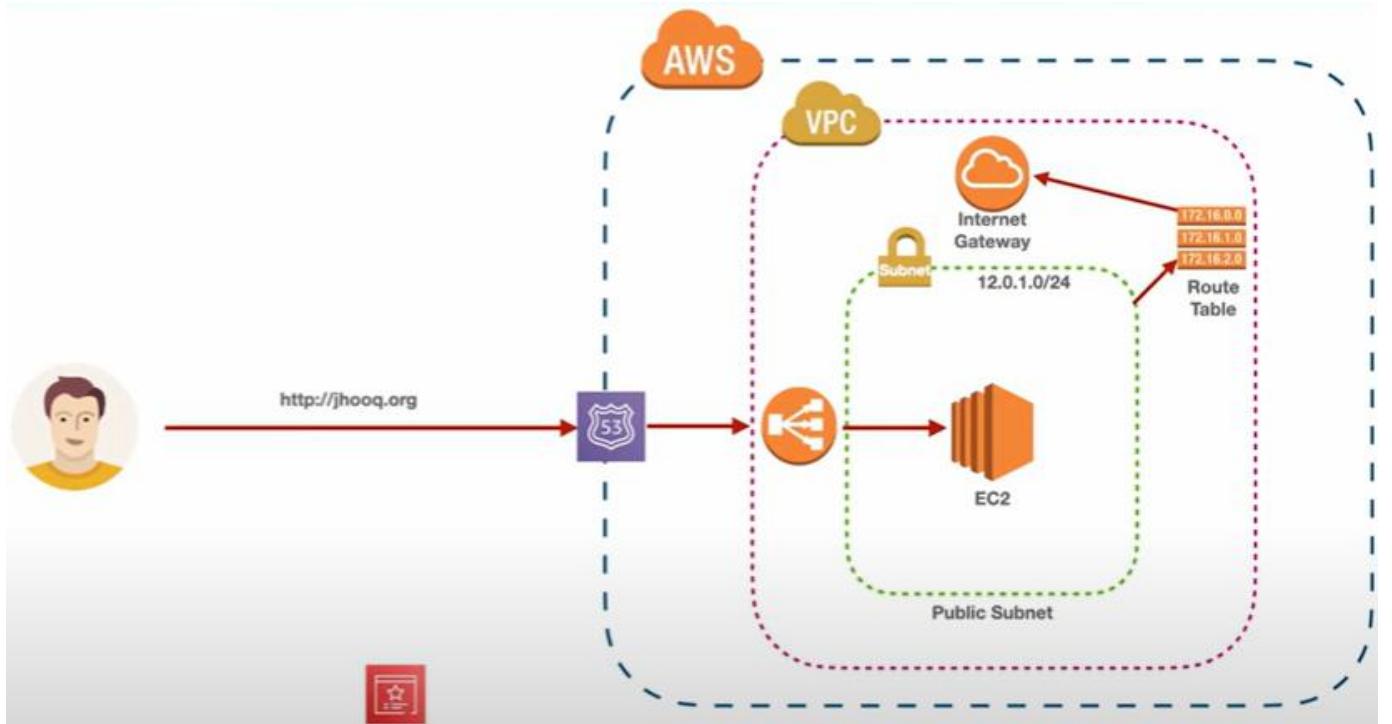
Once we configure our Route 53, then the user can make a request using the URL.



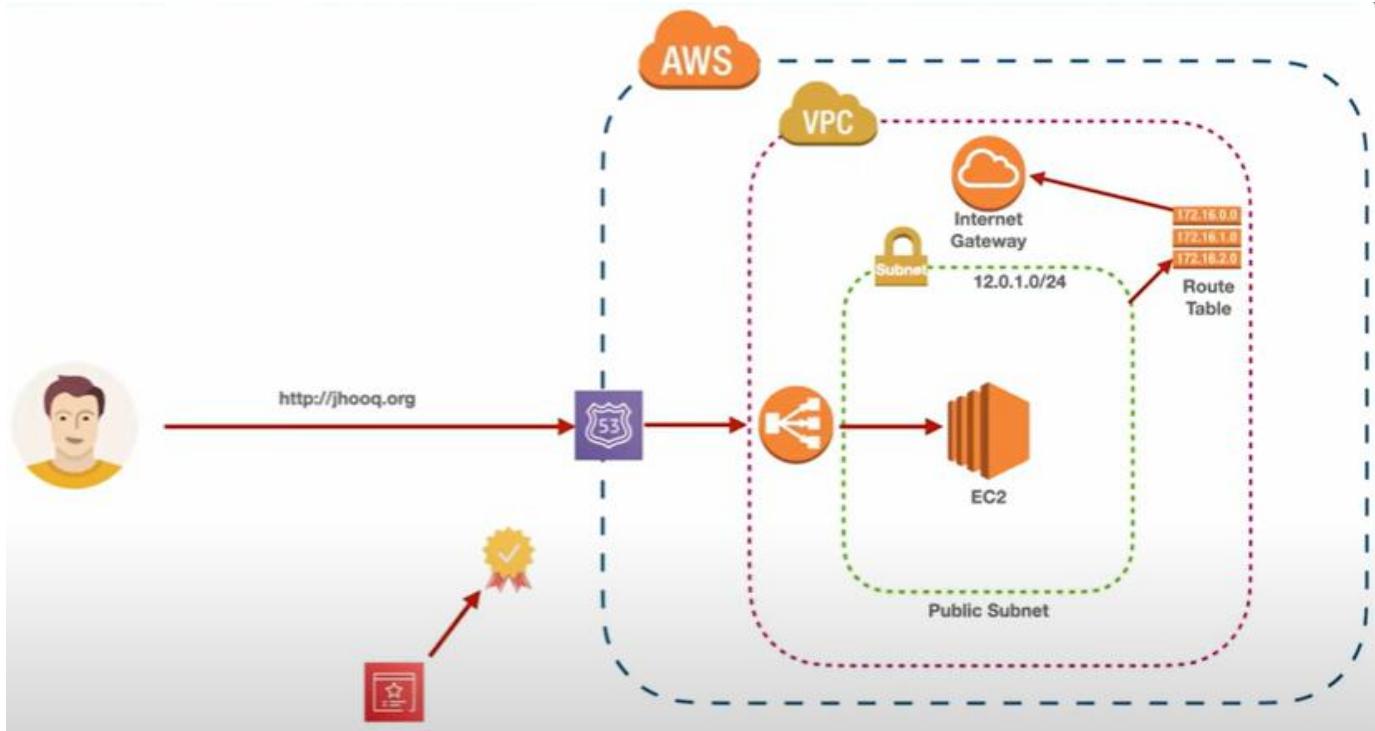
Here you can see in the diagram, we will have a Route 53 on top of our Load Balancer and that user can use the URL like <http://sosoebot.org> which is my own domain to access the particular service that is running on to my EC2 instance.



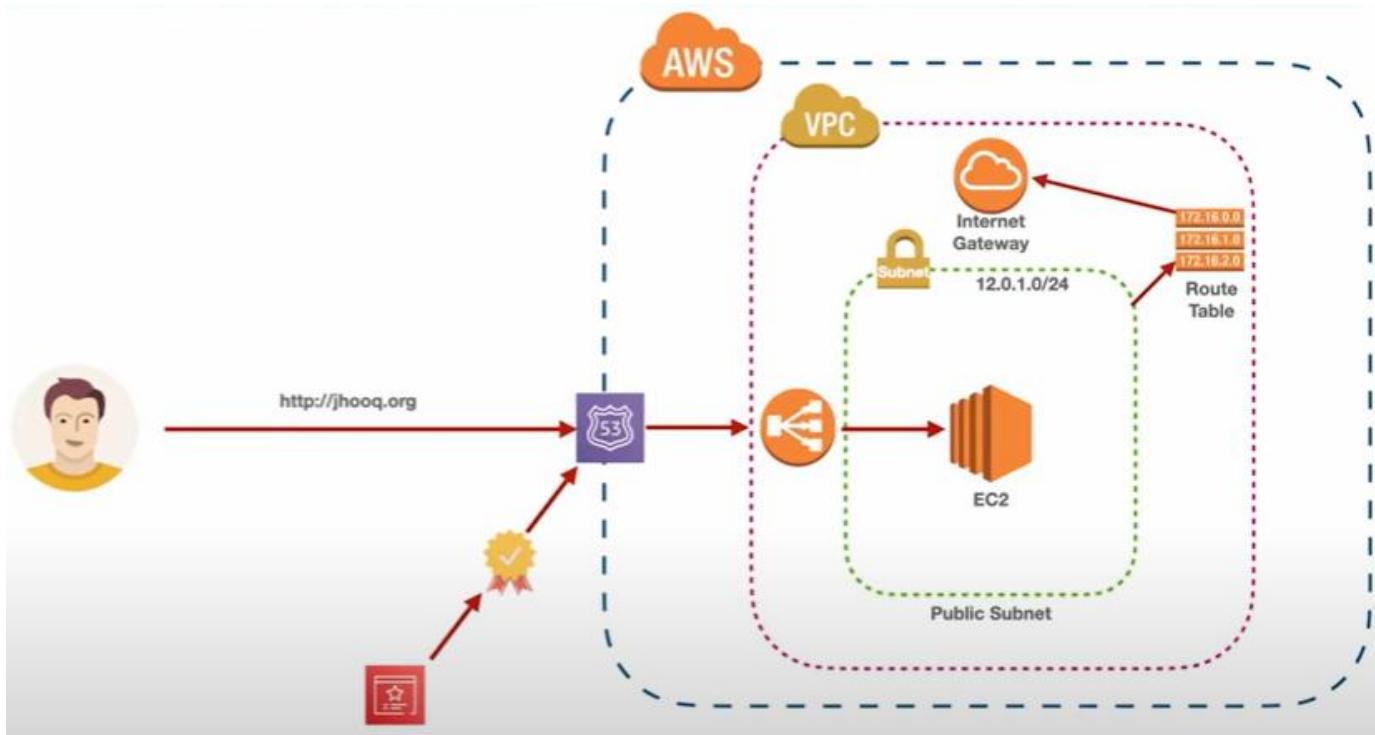
But as you have noticed in the diagram, the URL I am using is http and not https. To introduce HTTPS into the URL, then we need to use the AWS Certificate Manager.



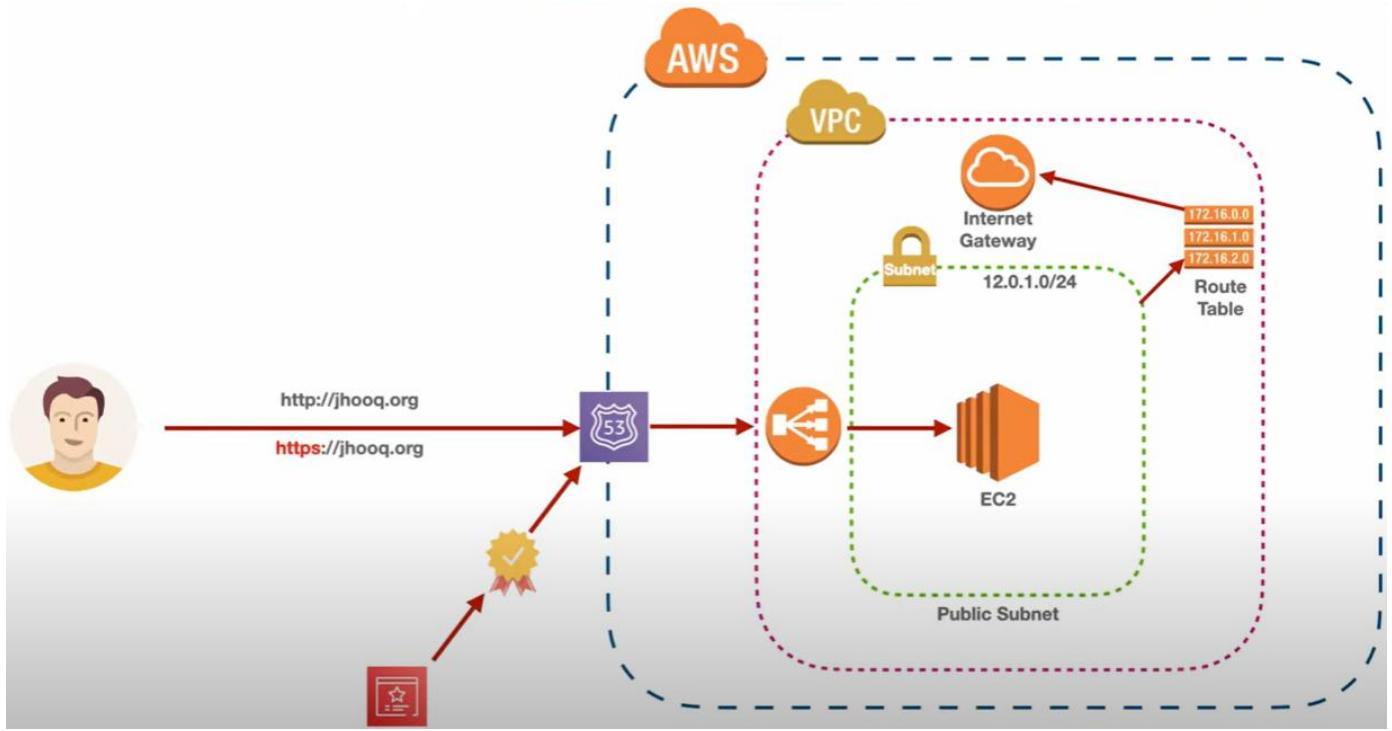
So, we are going to use the AWS Certificate Manager to generate those certificates



And we are going to use those certificate records to insert into our AWS Route 53.



So, once we create those records inside our AWS Route 53 after generating those certificates, then we can use the URL that is HTTPS along with my domain.



The entire process is broken down into four simple steps for better understanding:

- Set up our VPC
- Create a Public subnet inside the VPC
- Set up our Internet Gateway
- Set up our Route Table
- Set up our EC2 Instance
- Create an Application Load Balancer
- Configure our Route 53
- Set up AWS Certificate Manager

STEP 1: Set up our VPC

Go to AWS Management console and search for “VPC”

The screenshot shows the AWS Management Console with the search bar at the top set to "VPC". On the left, the navigation pane for EC2 is visible, with "VPC" selected under "Services". The main content area is titled "Services" and lists three items: "VPC" (Isolated Cloud Resources), "AWS Firewall Manager" (Central management of firewall rules), and "Detective" (Investigate and Analyze potential security issues). An orange arrow points to the "VPC" icon. To the right, there is a detailed view of a VPC configuration, including a table for inbound rules with columns for Protocol, Port range, and Source, and a section for "Manage tags" and "Edit inbound rules". A modal window at the bottom left asks if the results are helpful, with "Yes" and "No" buttons.

Click on “VPC” under services

The screenshot shows the "VPC dashboard" page. On the left, a sidebar lists various VPC-related services like "Virtual private cloud", "Security", and "PrivateLink and Lattice". The main area is titled "Resources by Region" and shows a grid of Amazon VPC resources across different regions. Each resource has a link to "See all regions". On the far left of the dashboard, there is a prominent orange button labeled "Create VPC". An orange arrow points to this button. To the right of the dashboard, there are sections for "Service Health", "Settings", "Additional Information", and "AWS Network Manager", each with its own set of links and descriptions.

Click on “Create VPC”

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances. Mouse over a resource to highlight the related resources.

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
project

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**

Tenancy Info
Default

Number of Availability Zones (AZs) Info
Choose the number of AZs in which to provision subnets. We recommend at least two AZs for high availability.

1 2 3

Customize AZs

Preview

VPC Show details
Your AWS virtual network

Subnets (4)
Subnets within this VPC

Route tables (3)
Route network traffic to resources

Project-vpc

- us-east-1a**
 - (A) project-subnet-public1-us-east-1a
 - (A) project-subnet-private1-us-east-1a
- us-east-1b**
 - (B) project-subnet-public2-us-east-1b
 - (B) project-subnet-private2-us-east-1b

project-rtb-public
project-rtb-private1-us-east-1a
project-rtb-private2-us-east-1b

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

Choose “VPC Only”

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

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

my-vpc-01

IPv4 CIDR block Info
 IPv4 CIDR manual input **IPAM-allocated IPv4 CIDR block**

IPv4 CIDR
10.0.0.0/24
CIDR block size must be between /16 and /28.

IPv6 CIDR block Info
 No IPv6 CIDR block **IPAM-allocated IPv6 CIDR block** **Amazon-provided IPv6 CIDR block** **IPv6 CIDR owned by me**

Tenancy Info
Default

Tags
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.

No tags associated with the resource

Add tag

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

Then we have to give the VPC a name, we will call it “**test-vpc**”

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 - optional
Creates a tag with a key of 'Name' and a value that you specify.
test-vpc

IPv4 CIDR block [Info](#)
 IPv4 CIDR manual input IPAM-allocated IPv4 CIDR block

IPv4 CIDR
10.0.0.0/24
CIDR block size must be between /16 and /28.

IPv6 CIDR block [Info](#)
 No IPv6 CIDR block IPAM-allocated IPv6 CIDR block Amazon-provided IPv6 CIDR block IPv6 CIDR owned by me

Tenancy [Info](#)
Default

Tags
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 Name Value - optional test-vpc [Remove tag](#)

[CloudShell](#) [Feedback](#)

Next, we have to give our IPv4 CIDR range. We will use “**12.0.0.0/16**”

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 - optional
Creates a tag with a key of 'Name' and a value that you specify.
test-vpc

IPv4 CIDR block [Info](#)
 IPv4 CIDR manual input IPAM-allocated IPv4 CIDR block

IPv4 CIDR
12.0.0.0/16
CIDR block size must be between /16 and /28.

IPv6 CIDR block [Info](#)
 No IPv6 CIDR block IPAM-allocated IPv6 CIDR block Amazon-provided IPv6 CIDR block IPv6 CIDR owned by me

Tenancy [Info](#)
Default

Tags
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 Name Value - optional test-vpc [Remove tag](#)

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

[Cancel](#) [Preview code](#) [Create VPC](#)

[CloudShell](#) [Feedback](#)

Click on “Create VPC”

You successfully created **vpc-0a178e5d2aec4790 / test-vpc**

VPC ID	State	Block Public Access	DNS hostnames
vpc-0a178e5d2aec4790	Available	Off	Disabled

Details [Info](#)

VPC ID: vpc-0a178e5d2aec4790
Tenancy: default
Main network ACL: acl-07bef2dcaa908b4f
IPv6 CIDR (Network border group): -
Default VPC: No
Network Address Usage metrics: Disabled
DHCP option set: dopt-067ba7e48365ea16
IPv4 CIDR: 12.0.0.0/16
Route 53 Resolver DNS Firewall rule groups: -
Main route table: rtb-0da6cd7b4e4f2f903
IPv6 pool: -
Owner ID: 524783324460

Resource map [Info](#)

VPC [Show details](#)
Your AWS virtual network
test-vpc

Subnets (0)
Subnets within this VPC

Route tables (1)
Route network traffic to resources
rtb-0da6cd7b4e4f2f903

Network connections (0)
Connections to other networks

Our “test-vpc” has been created. This can be verified by clicking on “Your VPCs”

You successfully created **vpc-0a178e5d2aec4790 / test-vpc**

Your VPCs (3) Info

Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR	DHCP op...
-	vpc-0128e9209eae1c37	Available	Off	172.31.0.0/16	-	dopt-06
Dev-VPC	vpc-0c565a10e97a2bb88	Available	Off	10.100.0.0/16	-	dopt-06
test-vpc	vpc-0a178e5d2aec4790	Available	Off	12.0.0.0/16	-	dopt-06

Select a VPC above

You can see the VPC we just created.

STEP 2: Create a Public subnet inside the VPC

We are going to create a public subnet inside our VPC with IPv4 CIDR range of “**12.0.1.0/24**”

You successfully created vpc-0a178e5d2aecc4790 / test-vpc

Your VPCs (3) Info

Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR	DHCP op...
-	vpc-0128e9209eaef1c37	Available	Off	172.31.0.0/16	-	dopt-06
Dev-VPC	vpc-0c565a10e97a2bb88	Available	Off	10.100.0.0/16	-	dopt-06
test-vpc	vpc-0a178e5d2aecc4790	Available	Off	12.0.0.0/16	-	dopt-06

Select a VPC above

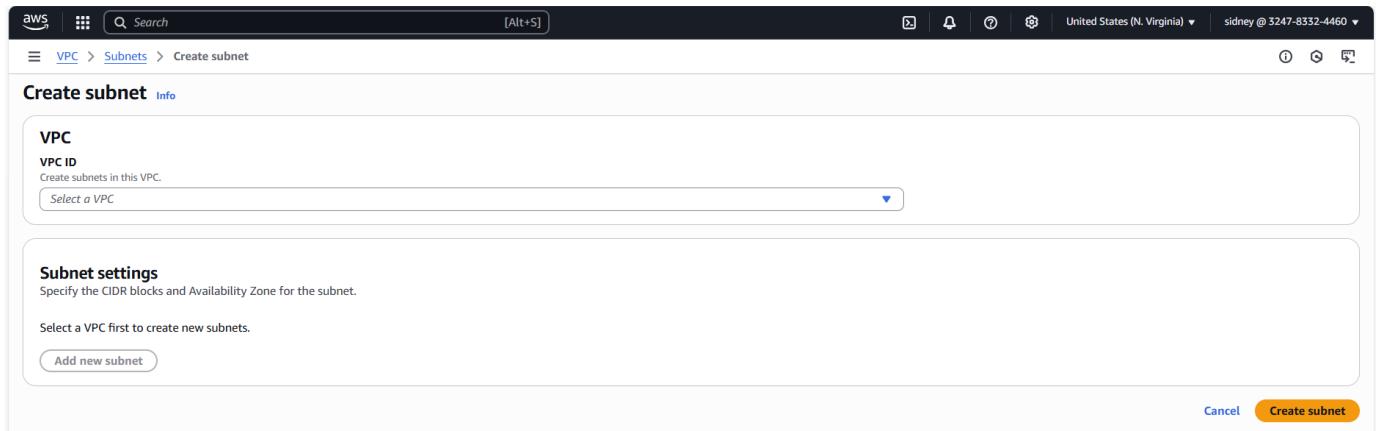
Click on “Subnets”

Subnets (14) Info

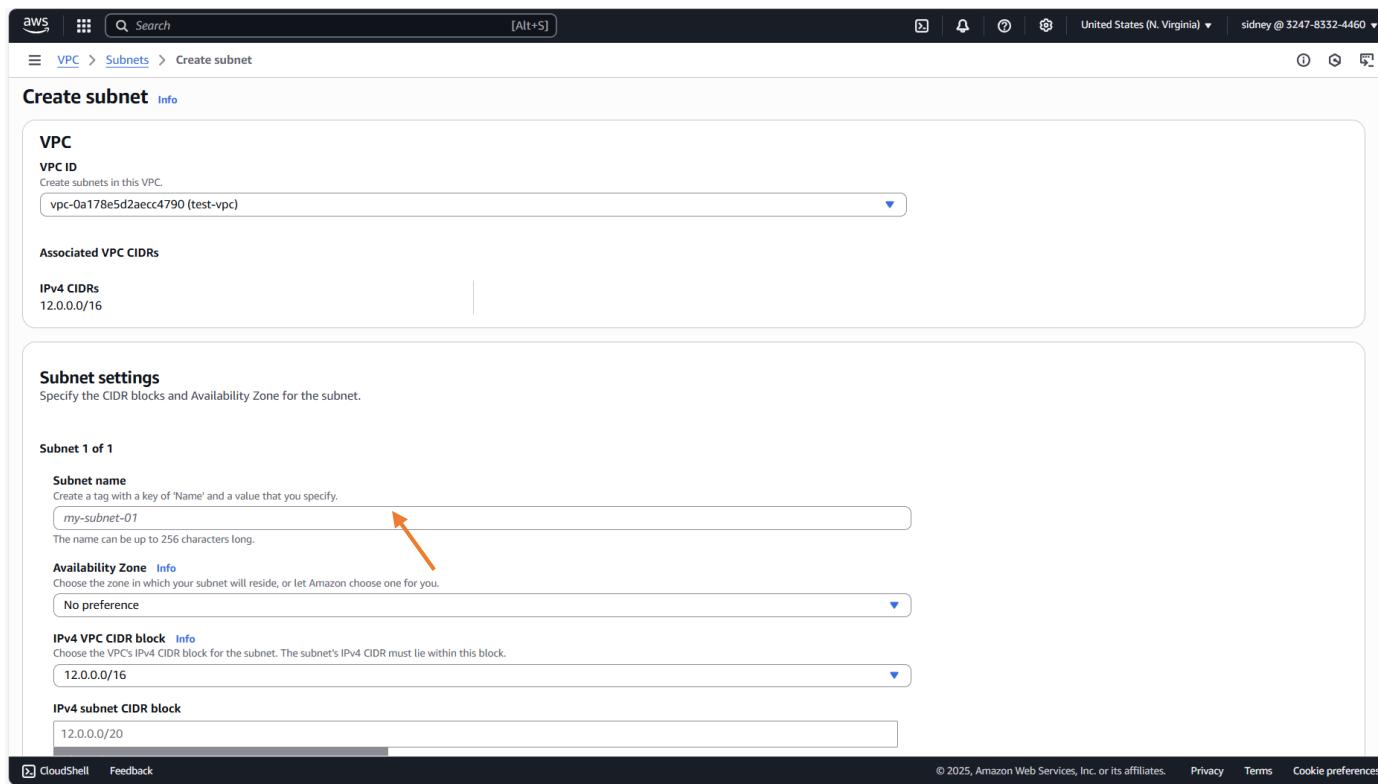
Name	Subnet ID	State	VPC	Block Public Access	IPv4 CIDR
-	subnet-0920122fe5af20950	Available	vpc-0128e9209eaef1c37	Off	172.31.0.0/20
Dev-private-subnet-1a	subnet-03a340dd0d8b6fd87	Available	vpc-0c565a10e97a2bb88 Dev-VPC	Off	10.100.80.0/21
Dev-app-subnet-1b	subnet-0ef40a6bc568a5a9d	Available	vpc-0c565a10e97a2bb88 Dev-VPC	Off	10.100.160.0/21
Dev-public-subnet-1b	subnet-0f7d7a0103ef33410	Available	vpc-0c565a10e97a2bb88 Dev-VPC	Off	10.100.16.0/21
Dev-public-subnet-1a	subnet-02a2a7b7e24f85610	Available	vpc-0c565a10e97a2bb88 Dev-VPC	Off	10.100.8.0/21
Dev-data-subnet-1a	subnet-08c5ed4635fb19a0	Available	vpc-0c565a10e97a2bb88 Dev-VPC	Off	10.100.168.0/21
-	subnet-0e978029c28591323	Available	vpc-0128e9209eaef1c37	Off	172.31.0.0/20
-	subnet-0c21087ab9e69a290	Available	vpc-0128e9209eaef1c37	Off	172.31.64.0/20
-	subnet-0410d8e91c199d290	Available	vpc-0128e9209eaef1c37	Off	172.31.32.0/20
-	subnet-09e0df53f5814abf3	Available	vpc-0128e9209eaef1c37	Off	172.31.16.0/20
-	subnet-0abe0cd1e821b6aa9	Available	vpc-0128e9209eaef1c37	Off	172.31.48.0/20
Dev-private-subnet-1b	subnet-057df62ee87416af9	Available	vpc-0c565a10e97a2bb88 Dev-VPC	Off	10.100.88.0/21
Dev-data-subnet-1b	subnet-06354ecc3e56392d0	Available	vpc-0c565a10e97a2bb88 Dev-VPC	Off	10.100.176.0/21
Dev-app-subnet-1a	subnet-0ff8868ba4edf0f0	Available	vpc-0c565a10e97a2bb88 Dev-VPC	Off	10.100.144.0/21

Select a subnet

Click on “create subnet”



Click on the drop down on “VPC ID” and select our VPC



Then we will give the subnet the name “**test-public-subnet-1-1a**”

Screenshot of the AWS VPC Create Subnet wizard, Step 2: Subnet settings.

VPC

VPC ID
Create subnets in this VPC.
vpc-0a178e5d2aecc4790 (test-vpc)

Associated VPC CIDRs

IPv4 CIDRs
12.0.0.0/16

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.
test-public-subnet-1-1a

The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
No preference

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.
12.0.0.0/16

IPv4 subnet CIDR block
12.0.0.0/20

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

Click on the drop down on “Availability Zone” and select “us-east-1a”

Screenshot of the AWS VPC Create Subnet wizard, Step 2: Subnet settings.

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.
test-public-subnet-1-1a

The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
United States (N. Virginia) / us-east-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.
12.0.0.0/16

IPv4 subnet CIDR block
12.0.0.0/20

Tags - optional

Key	Value - optional
Name	test-public-subnet-1-1a

Add new tag
You can add 49 more tags.
Remove

Add new subnet

Cancel **Create subnet**

Next is to specify the IPv4 CIDR range of the subnet. We will use “**12.0.1.0/24**”

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.

The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

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.

IPv4 subnet CIDR block
 256 IPs
◀ ▶ ⌂ ⌃ ⌄

Tags - optional

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="test-public-subnet-1-1a"/> Remove

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

[Add new subnet](#)

[Cancel](#) [Create subnet](#)

The we will add one more subnet so that we can have two subnets for more availability. Click on “**Add new subnet**”

Subnet 2 of 2

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

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.

IPv4 subnet CIDR block
 ⌂ ⌃ ⌄

Tags - optional
No tags associated with the resource.

[Add new tag](#)
You can add 50 more tags.
[Remove](#)

[Add new subnet](#)

[Cancel](#) [Create subnet](#)

We will name the subnet as “**test-public-subnet-1-1b**”

Screenshot of the AWS VPC Subnets Create subnet page. The 'Availability Zone' dropdown menu is open, showing 'No preference' as the selected option. An orange arrow points to the dropdown menu.

Click on the drop down on “Availability Zone” and select “us-east-1b”

Screenshot of the AWS VPC Subnets Create subnet page. The 'Availability Zone' dropdown menu is open, showing 'No preference' as the selected option. An orange arrow points to the dropdown menu. The 'IPv4 subnet CIDR block' input field is also highlighted with an orange arrow.

Then enter our subnet IPv4 CIDR, we will use “**12.0.2.0/24**”

Screenshot of the AWS VPC Subnets 'Create subnet' wizard.

Subnet 2 of 2

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

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.

IPv4 subnet CIDR block
 256 IPs

Tags - optional

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="test-public-subnet-1-1b"/>

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

[Add new subnet](#)

[Cancel](#) **Create subnet**

Click on “create subnet”

Screenshot of the AWS VPC Subnets dashboard showing the newly created subnets.

Subnets (2) [Info](#)

Subnet ID : subnet-04e5dcf10023b20f3 [Actions](#) **Create subnet**

Subnet ID : subnet-013af1016c64ee796 [Actions](#)

Last updated less than a minute ago

<input type="checkbox"/>	Name	Subnet ID	State	VPC	Block Public Access	IPv4 CIDR
<input type="checkbox"/>	test-public-subnet-1-1b	subnet-013af1016c64ee796	Available	vpc-0a178e5d2aecc4790 test-vpc	<input type="radio"/> Off	12.0.2.0/24
<input type="checkbox"/>	test-public-subnet-1-1a	subnet-04e5dcf10023b20f3	Available	vpc-0a178e5d2aecc4790 test-vpc	<input type="radio"/> Off	12.0.1.0/24

Select a subnet

You can see that both subnets have been created.

STEP 3: Set up Internet Gateway

Now, we have to create our internet Gateway that will help us give our public subnet internet access.

The screenshot shows the AWS VPC Subnets page. A success message at the top states: "You have successfully created 2 subnets: subnet-04e5dcf10023b20f3, subnet-013af1016c64ee796". The main table lists two subnets:

Name	Subnet ID	State	VPC	Block Public Access	IPv4 CIDR
test-public-subnet-1-1b	subnet-013af1016c64ee796	Available	vpc-0a178e5d2aecc4790 test-vpc	Off	12.0.2.0/24
test-public-subnet-1-1a	subnet-04e5dcf10023b20f3	Available	vpc-0a178e5d2aecc4790 test-vpc	Off	12.0.1.0/24

The left sidebar shows the navigation menu, and an orange arrow points to the "Internet gateways" link under the "Virtual private cloud" section.

Click on “Internet Gateways”

The screenshot shows the AWS VPC Internet Gateways page. The main table lists two existing gateways:

Name	Internet gateway ID	State	VPC ID	Owner
-	igw-0eb61fdc4add6d6ae	Attached	vpc-0128e9209eaef1c37	324783324460
Dev-VPC-IGW	igw-0f257b33b08925562	Attached	vpc-0c565a10e97a2bb88 Dev-VPC	324783324460

The left sidebar shows the navigation menu, and an orange arrow points to the "Create internet gateway" button in the top right corner.

Click on “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.

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.
No tags associated with the resource.
[Add new tag](#)
You can add 50 more tags.

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

Give the internet gateway a name. I will call it “**test-igw**”

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.

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

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

Then click on “Create Internet Gateway”

The following internet gateway was created: igw-03f3959f9436d149f - test-igw. You can now attach to a VPC to enable the VPC to communicate with the internet.

igw-03f3959f9436d149f / test-igw

Details [Info](#)

Internet gateway ID igw-03f3959f9436d149f	State Detached	VPC ID -	Owner 324783324460
--	-----------------------------------	-------------	---------------------------------------

Tags

Key	Value
Name	test-igw

[Manage tags](#)

[Actions](#)

Attach to a VPC

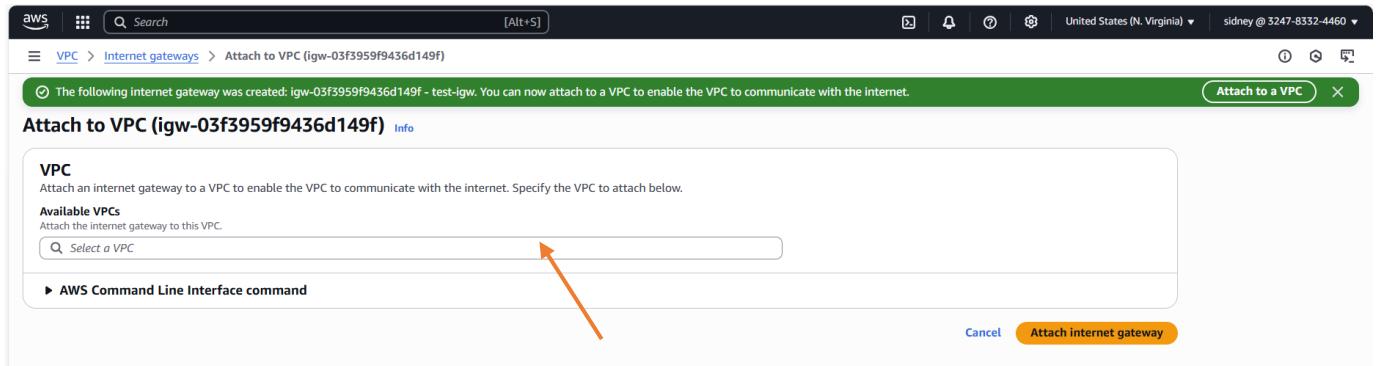
VPC dashboard

- EC2 Global View
- Filter by VPC
- Virtual private cloud**
 - Your VPCs
 - Subnets
 - Route tables
 - Internet gateways**
 - Egress-only internet gateways
 - Carrier gateways
 - DHCP option sets
 - Elastic IPs
 - Managed prefix lists
 - NAT gateways
 - Peering connections
 - Route servers [New](#)
- Security**
 - Network ACLs
 - Security groups
- PrivateLink and Lattice**
 - Getting started [Updated](#)
 - Endpoints [Updated](#)
 - Endpoint services
 - Service networks [Updated](#)
 - Lattice services

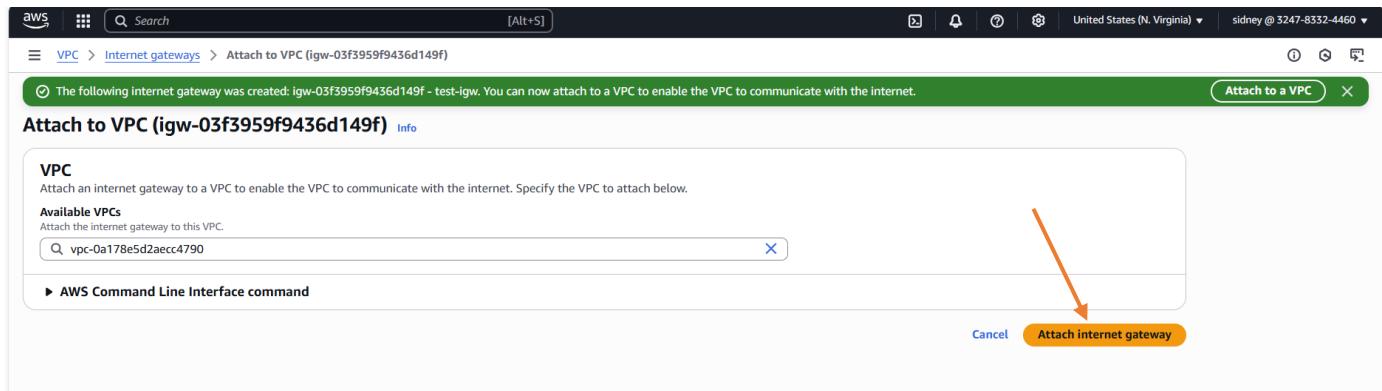
[CloudShell](#) [Feedback](#)

Our internet gateway has been created. The internet gateway is inside the VPC, so we have to associate it with our VPC.

Click on “Attach to a VPC”



Click on “Available VPCs” and select our VPC



Click on “Attach Internet Gateway”

The screenshot shows the AWS VPC dashboard with the 'Internet gateways' section selected. A success message at the top states: 'Internet gateway igw-03f3959f9436d149f successfully attached to vpc-0a178e5d2aecc4790'. The main card displays the internet gateway details: ID igw-03f3959f9436d149f, State Attached, VPC ID vpc-0a178e5d2aecc4790 | test-vpc, and Owner sidney @ 3247-8332-4460. The 'Tags' section shows a single tag named 'test-igw'. The left sidebar includes sections for Virtual private cloud (Your VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections, Route servers), Security (Network ACLs, Security groups), and PrivateLink and Lattice (Getting started, Endpoints, Endpoint services, Service networks, Lattice services).

The internet Gateway has been attached to our VPC

STEP 4: Set up our Route Table

The next resource we have to create is our Route table.

The screenshot shows the same AWS VPC dashboard as before, but with a red arrow pointing to the 'Route tables' link in the 'Virtual private cloud' sidebar. The rest of the interface is identical to the previous screenshot, showing the attached internet gateway and the VPC configuration.

Click on “Route Tables”

The screenshot shows the AWS VPC Route Tables page. On the left, there's a navigation sidebar with sections like 'Virtual private cloud', 'Route tables', and 'Security'. The main area displays a table titled 'Route tables (4) Info' with columns for Name, Route table ID, Explicit subnet assoc., Edge associations, Main, and VPC. The table lists four route tables: 'rtb-094350c8a924d963f' (Main), 'rtb-091fb3d6b19ab5ff' (Main), 'Dev-VPC-Public-RouteTable' (2 subnets), and 'Dev-VPC-Private-RouteTable' (6 subnets). At the top right of the table, there's a 'Actions' dropdown and a 'Create route table' button, which is highlighted with a red arrow. The bottom of the page has a 'Select a route table' section and some footer links.

Click on “Create Route Table”

The screenshot shows the 'Create route table' wizard. The first step, 'Route table settings', is displayed. It has fields for 'Name - optional' (containing 'my-route-table-01') and 'VPC' (with a dropdown menu labeled 'Select a VPC'). Below these, the 'Tags' section is shown, which includes an 'Add new tag' button and a note about adding up to 50 more tags. At the bottom right of the wizard, there are 'Cancel' and 'Create route table' buttons.

Give the Route Table a name, I will call it “**test-rt-public**”

AWS VPC > Route tables > Create route table

Create route table Info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - optional
Create a tag with a key of 'Name' and a value that you specify.

VPC
The VPC to use for this route table.

Tags
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
<input type="text" value="Name"/>	<input type="text" value="test-rt-public"/>

Add new tag
You can add 49 more tags.

Cancel **Create route table**

Click on the drop down on “VPC” and select our VPC

AWS VPC > Route tables > Create route table

Create route table Info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - optional
Create a tag with a key of 'Name' and a value that you specify.

VPC
The VPC to use for this route table.

Tags
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
<input type="text" value="Name"/>	<input type="text" value="test-rt-public"/>

Add new tag
You can add 49 more tags.

Cancel **Create route table**

Click on “Create Route Table”

The screenshot shows the AWS VPC dashboard with the 'Route tables' section selected. A success message at the top states: "Route table rtb-0fba8bbc017237268 | test-rt-public was created successfully." Below this, the details for the new route table are shown, including its ID, VPC association, and subnet associations. An orange arrow points from the text "Click on ‘Subnet Association’" to the 'Subnet associations' tab in the navigation bar.

The VPC has been created. The Route Table is inside our VPC, so we have to associate the public subnet with our Route Table.

Click on “Subnet Association”

The screenshot shows the same VPC Route Tables page, but now the 'Subnet associations' tab is selected. It displays a table for explicit subnet associations, which currently has zero entries. Below this, another table lists subnets without explicit associations, showing two subnets: 'test-public-subnet-1-1b' and 'test-public-subnet-1-1a'. An orange arrow points from the text "Click on ‘Edit Subnet Associations’" to the 'Edit subnet associations' button in the top right corner of the 'Explicit subnet associations' section.

Click on “Edit Subnet Associations”

Available subnets (2)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
test-public-subnet-1-1b	subnet-013af1016c64ee796	12.0.2.0/24	-	Main (rtb-0da6cd7b4e4f2f903)
test-public-subnet-1-1a	subnet-04e5dcf10023b20f3	12.0.1.0/24	-	Main (rtb-0da6cd7b4e4f2f903)

Select both public subnets

Available subnets (2/2)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
test-public-subnet-1-1b	subnet-013af1016c64ee796	12.0.2.0/24	-	Main (rtb-0da6cd7b4e4f2f903)
test-public-subnet-1-1a	subnet-04e5dcf10023b20f3	12.0.1.0/24	-	Main (rtb-0da6cd7b4e4f2f903)

Selected subnets

subnet-013af1016c64ee796 / test-public-subnet-1-1b X subnet-04e5dcf10023b20f3 / test-public-subnet-1-1a X

Click on “Save Associations”

You have successfully updated subnet associations for rtb-0fba8bbc017237268 / test-rt-public.

rtb-0fba8bbc017237268 / test-rt-public

Details

Route table ID rtb-0fba8bbc017237268	Main <input type="checkbox"/> No	Explicit subnet associations 2 Subnets	Edge associations -
VPC vpc-0a178e5d2aecc4790 test-vpc	Owner ID 324783324460		

Routes (1)

Destination	Target	Status	Propagated
12.0.0.0/16	local	Active	No

Actions

Both **Edit routes**

We have now associated our subnet with our Route Table. But we also have to attached our internet gateway to our subnet, so that the public subnet can access the internet.

Click on “Edit Route”

The screenshot shows the AWS VPC Edit routes interface. A single route entry is listed:

Destination	Target	Status	Propagated
12.0.0.0/16	local	Active	No

Buttons at the bottom include "Add route", "Cancel", "Preview", and "Save changes".

Click on “Add Route”

The screenshot shows the AWS VPC Edit routes interface. An empty search bar is visible, and an orange arrow points to the "Add route" button.

Click and select “0.0.0.0/0”

The screenshot shows the AWS VPC Edit routes interface with a new route entry:

Destination	Target	Status	Propagated
12.0.0.0/16	local	Active	No
0.0.0.0/0	local	-	No

An orange arrow points to the "local" dropdown for the destination "0.0.0.0/0".

Click on the drop down and select “Internet Gateway”

The screenshot shows the AWS VPC Edit routes interface with a new route entry:

Destination	Target	Status	Propagated
12.0.0.0/16	local	Active	No
0.0.0.0/0	Internet Gateway	-	No

An orange arrow points to the "Internet Gateway" dropdown for the destination "0.0.0.0/0".

Click on “igw” and select our Internet Gateway

Edit routes

Destination	Target	Status	Propagated
12.0.0.0/16	local	Active	No
0.0.0.0/0	Internet Gateway igw-03f3959f9436d149f	-	No

Add route Remove Cancel Preview Save changes

Click on “Save Changes”

Updated routes for rtb-0fba8bbc017237268 / test-rt-public successfully

rtb-0fba8bbc017237268 / test-rt-public

Details Info

Route table ID rtb-0fba8bbc017237268	Main No	Explicit subnet associations 2 subnets	Edge associations -
VPC vpc-0a178e5d2aec4790 test-vpc	Owner ID 324783324460		

Routes Subnet associations Edge associations Route propagation Tags

Routes (2)

Destination	Target	Status	Propagated
0.0.0.0/0	igw-03f3959f9436d149f	Active	No
12.0.0.0/16	local	Active	No

Both Edit routes

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

You can see that our internet gateway has been attached to our route table.

STEP 5: Launch the EC2 Instance

The next resource we have to create is the EC2 instance. Go to EC2 dashboard on AWS Management Console.

AWS Management Console - EC2 Instances

Instances (1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
cluster-server	i-02cf0bf1114013caf	Stopped	t2.micro	-	-	us-east-1a

Select an instance

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

Click on “Launch Instance”

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: e.g. My Web Server [Add additional tags](#)

Application and OS Images (Amazon Machine Image)

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

[Search our full catalog including 1000s of application and OS images](#)

Recents **Quick Start**

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Linux Debian

Amazon Machine Image (AMI)

Amazon Linux 2023 kernel-6.1 AMI ami-08a6ef1d148b1f7504 (64-bit (x86), uefi-preferred) / ami-0aaaf509a1ebd95e61 (64-bit (Arm), uefi) Free tier eligible

Description

Amazon Linux 2023 (kernel-6.1) is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Amazon Linux 2023 AMI 2023.8.20250721.2 x86_64 HVM kernel-6.1

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.8.2... [read more](#)

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

[Cancel](#) [Launch instance](#) [Preview code](#)

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

We will call the Instance “**test-instance**”

Launch an instance Info

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 Info

Name

test-instance

Add additional tags

On “**AMI**”, we will use “**Ubuntu**”

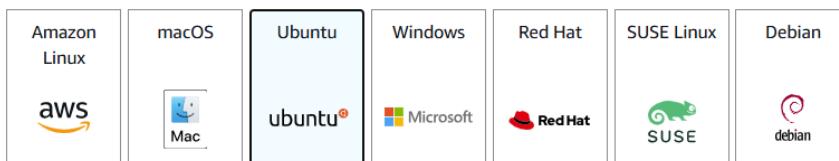
▼ Application and OS Images (Amazon Machine Image) Info

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

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

Recents

Quick Start



[Browse more AMIs](#)

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
ami-020cba7c55df1f615 (64-bit (x86)) / ami-07041441b708acbd6 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

Description

Ubuntu Server 24.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Canonical, Ubuntu, 24.04, amd64 noble image

Architecture

64-bit (x86)

AMI ID

ami-020cba7c55df1f615

Publish Date

2025-06-10

Username

ubuntu

Verified provider

On “**Instance Type**”, select “**t2.micro**”

▼ Instance type Info | Get advice

Instance type

t2.micro

Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0162 USD per Hour On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour On-Demand RHEL base pricing: 0.026 USD per Hour
On-Demand Linux base pricing: 0.0116 USD per Hour

All generations

[Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

On “Key Pair”, I will select a key pair I had created previously. Click on the drop down and select the key.

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

dev

[Create new key pair](#)

On “Network Settings”, click on “Edit”

▼ Network settings [Info](#)

VPC - required [Info](#)

vpc-0128e9209eaef1c37
172.31.0.0/16

(default)

[Create new subnet](#)

Subnet [Info](#)

No preference

[Create new subnet](#)

Availability Zone [Info](#)

No preference

[Enable additional zones](#)

Auto-assign public IP [Info](#)

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group

Select existing security group

Security group name - required

launch-wizard-3

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and _-./()#,@[]+=&;{}\$*

Description - required [Info](#)

launch-wizard-3 created 2025-08-03T05:24:07.558Z

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

[Remove](#)

Type Info	Protocol Info	Port range Info
ssh	TCP	22

Click on the drop down and select our VPC

Network settings

VPC - required | Info
vpc-0a178e5d2aec4790 (test-vpc)
12.0.0.0/16

Subnet | Info
subnet-013af1016c64ee796 test-public-subnet-1-1b
VPC: vpc-0a178e5d2aec4790 Owner: 324783324460 Availability Zone: us-east-1b
Zone type: Availability Zone IP addresses available: 251 CIDR: 12.0.2.0/24

Create new subnet

Auto-assign public IP | Info
Disable

Firewall (security groups) | Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.
 Create security group Select existing security group

Description - required | Info
launch-wizard-3 created 2025-08-03T05:24:07.558Z

Inbound Security Group Rules
▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type	Protocol	Port range
ssh	TCP	22

Source type | Info
Anywhere

Description - optional | Info
e.g. SSH for admin desktop

Summary
Number of instances | Info
1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-020cba7c55df1f615

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

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

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Launch instance

On “**Subnet**”, select one of the public subnets we created

Network settings

VPC - required | Info
vpc-0a178e5d2aec4790 (test-vpc)
12.0.0.0/16

Subnet | Info
subnet-013af1016c64ee796 test-public-subnet-1-1b
VPC: vpc-0a178e5d2aec4790 Owner: 324783324460 Availability Zone: us-east-1b
Zone type: Availability Zone IP addresses available: 251 CIDR: 12.0.2.0/24

Create new subnet

Auto-assign public IP | Info
Disable

Firewall (security groups) | Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.
 Create security group Select existing security group

Description - required | Info
launch-wizard-3 created 2025-08-03T05:24:07.558Z

Inbound Security Group Rules
▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type	Protocol	Port range
ssh	TCP	22

Source type | Info
Anywhere

Description - optional | Info
e.g. SSH for admin desktop

Summary
Number of instances | Info
1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-020cba7c55df1f615

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

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

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Launch instance

Click on the drop down on “**Auto-assign public IP**” and select “**Enable**”

Auto-assign public IP | [Info](#)

Enable ▼

Additional charges apply when outside of free tier allowance

Firewall (security groups) | [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

Security group name - required

launch-wizard-3 ↑

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and _-./()#@+=;&{}\$*

Description - required | [Info](#)

launch-wizard-3 created 2025-08-03T05:24:07.558Z

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0) [Remove](#)

Type	Protocol	Port range
ssh	TCP	22

Type | [Info](#) **Protocol** | [Info](#) **Port range** | [Info](#)

Source type | [Info](#) **Source** | [Info](#) **Description - optional** | [Info](#)

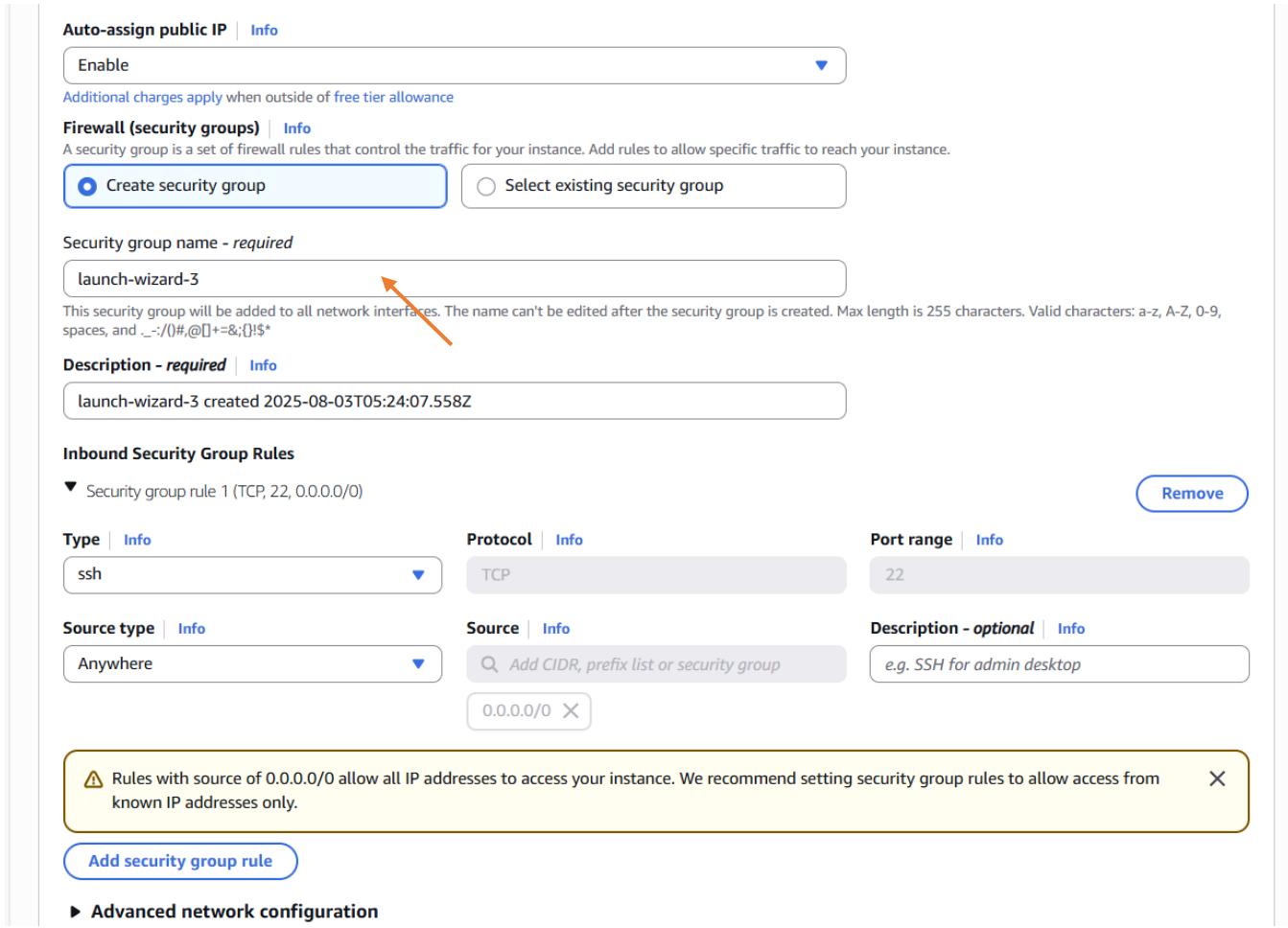
Anywhere ▼ e.g. SSH for admin desktop

0.0.0.0/0 X

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. X

[Add security group rule](#)

► Advanced network configuration



On “**Security Group Name**”, we will call it “**test-sg**”

Security group name - *required*

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and _-./()#@+=;&{}\$*

Description - *required* | [Info](#)

launch-wizard-3 created 2025-08-03T05:24:07.558Z

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type | [Info](#) Protocol | [Info](#) Port range | [Info](#)

ssh	TCP	22
-----	-----	----

Source type | [Info](#) Source | [Info](#) Description - *optional* | [Info](#)

Anywhere	<input type="text" value="0.0.0.0/0"/> X	e.g. SSH for admin desktop
----------	--	----------------------------

▼ Security group rule 2 (TCP, 80, 0.0.0.0/0)

Type | [Info](#) Protocol | [Info](#) Port range | [Info](#)

HTTP	TCP	80
------	-----	----

Source type | [Info](#) Source | [Info](#) Description - *optional* | [Info](#)

Anywhere	<input type="text" value="0.0.0.0/0"/> X	e.g. SSH for admin desktop
----------	--	----------------------------

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. X

[Add security group rule](#) ←

Then we have to create a security group with some rules. Click on “**Add security group rule**”

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type | Info Protocol | Info Port range | Info
ssh TCP 22

Source type | Info Source | Info Description - optional | Info
Anywhere Add CIDR, prefix list or security group e.g. SSH for admin desktop
0.0.0.0/0 X

Remove

▼ Security group rule 2 (TCP, 0)

Type | Info Protocol | Info Port range | Info
Custom TCP TCP 0

Source type | Info Source | Info Description - optional | Info
Custom Add CIDR, prefix list or security group e.g. SSH for admin desktop

Remove

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. X

Add security group rule

► Advanced network configuration

Click on the drop down on “Type” and select “HTTP”

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type | Info Protocol | Info Port range | Info
ssh TCP 22

Source type | Info Source | Info Description - optional | Info
Anywhere Add CIDR, prefix list or security group e.g. SSH for admin desktop
0.0.0.0/0 X

Remove

▼ Security group rule 2 (TCP, 80)

Type | Info Protocol | Info Port range | Info
HTTP TCP 80

Source type | Info Source | Info Description - optional | Info
Custom Add CIDR, prefix list or security group e.g. SSH for admin desktop

Remove

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. X

Add security group rule

► Advanced network configuration

On “Source Type”, select “Anywhere”

Inbound Security Group Rules

- ▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

[Remove](#)

Type Info	Protocol Info	Port range Info
ssh	TCP	22
Source type Info	Source Info	Description - optional Info
Anywhere	<input type="text" value="Add CIDR, prefix list or security group"/> 0.0.0.0/0	e.g. SSH for admin desktop

- ▼ Security group rule 2 (TCP, 80, 0.0.0.0/0)

[Remove](#)

Type Info	Protocol Info	Port range Info
HTTP	TCP	80
Source type Info	Source Info	Description - optional Info
Anywhere	<input type="text" value="Add CIDR, prefix list or security group"/> 0.0.0.0/0	e.g. SSH for admin desktop

⚠️ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

[Add security group rule](#)

► Advanced network configuration

Scroll down to “Advanced Details” and click on it

▼ Advanced details [Info](#)

Domain join directory | [Info](#)
 [Create new directory](#)

IAM instance profile | [Info](#)
 [Create new IAM profile](#)

Hostname type | [Info](#)

DNS Hostname | [Info](#)
 Enable IP name IPv4 (A record) DNS requests
 Enable resource-based IPv4 (A record) DNS requests
 Enable resource-based IPv6 (AAAA record) DNS requests

Instance auto-recovery | [Info](#)

Shutdown behavior | [Info](#)

Stop - Hibernate behavior | [Info](#)

Termination protection | [Info](#)

Stop protection | [Info](#)

► Summary

Number of instances | [Info](#)

Software Image (AMI)
 Canonical, Ubuntu, 24.04, amd64... [read more](#)
 ami-020cba7c55df1f615

Virtual server type (instance type)
 t2.micro

Firewall (security group)
 New security group

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

ⓘ Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

[Cancel](#)
[Launch instance](#)
[Preview code](#)

Scroll down to the end

⚠ For V2 requests, you must include a session token in all instance metadata requests.
Applications or agents that use V1 for instance metadata access will break.

Metadata response hop limit | [Info](#)
2

Allow tags in metadata | [Info](#)
Select ▾

User data - optional | [Info](#)
Upload a file with your user data or enter it in the field.
[Choose file](#)

User data has already been base64 encoded

Number of instances | [Info](#)
1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd64... [read more](#)
ami-020cba7c55df1f615

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

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

ⓘ **Free tier:** In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel
[Launch instance](#)
[Preview code](#)

On “user data” paste the code below to install Apache

```
#!/bin/bash
yes | sudo apt update
yes | sudo apt install apache2
echo "<h1>Server Details</h1><p><strong>Hostname:</strong> $(hostname)<p><strong>IP Address:</strong> $(hostname -I | cut -d" " -f1)</p>" > /var/www/html/index.html
sudo systemctl restart apache2
```

⚠ For V2 requests, you must include a session token in all instance metadata requests.
Applications or agents that use V1 for instance metadata access will break.

Metadata response hop limit | [Info](#)
2

Allow tags in metadata | [Info](#)
Select ▾

User data - optional | [Info](#)
Upload a file with your user data or enter it in the field.
[Choose file](#)

User data has already been base64 encoded

Number of instances | [Info](#)
1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd64... [read more](#)
ami-020cba7c55df1f615

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

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

ⓘ **Free tier:** In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel
[Launch instance](#)
[Preview code](#)

Click on “Launch Instance”

The screenshot shows the AWS EC2 'Launch an instance' success page. At the top, there's a green success banner stating 'Successfully initiated launch of instance (i-01da05ddfa6b69a11)'. Below it, a 'Launch log' button is visible. A 'Next Steps' section contains several cards:

- Create billing and free tier usage alerts**: To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds. Includes a 'Create billing alerts' button.
- Connect to your instance**: Once your instance is running, log into it from your local computer. Includes a 'Connect to instance' button and a 'Learn more' link.
- Connect an RDS database**: Configure the connection between an EC2 instance and a database to allow traffic flow between them. Includes a 'Connect an RDS database' button and a 'Create a new RDS database' link.
- Create EBS snapshot policy**: Create a policy that automates the creation, retention, and deletion of EBS snapshots. Includes a 'Create EBS snapshot policy' button.
- Manage detailed monitoring**: Enable or disable detailed monitoring for the instance. If you enable detailed monitoring, the Amazon EC2 console displays monitoring graphs with a 1-minute period. Includes a 'Manage detailed monitoring' button.
- Create Load Balancer**: Create a application, network gateway or classic Elastic Load Balancer. Includes a 'Create Load Balancer' button.
- Create AWS budget**: AWS Budgets allows you to create budgets, forecast spend, and take action on your costs and usage from a single location. Includes a 'Create AWS budget' button.
- Manage CloudWatch alarms**: Create or update Amazon CloudWatch alarms for the instance. Includes a 'Manage CloudWatch alarms' button.

At the bottom right, there's a 'View all instances' button. The footer includes standard AWS links like CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

The instance is being created. Click on “View All Instances”

The screenshot shows the AWS EC2 Instances page. The left sidebar has a tree view of services: EC2 (selected), Dashboard, EC2 Global View, Events, Instances (selected), Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (AMIs, AMI Catalog), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces). The main area shows a table titled 'Instances (1) Info' with one row:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
cluster-server	i-02cf0bf1114013caf	Stopped	t2.micro	-	View alarms +	us-east-1a

A 'Select an instance' dropdown is open below the table. The footer includes standard AWS links like CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

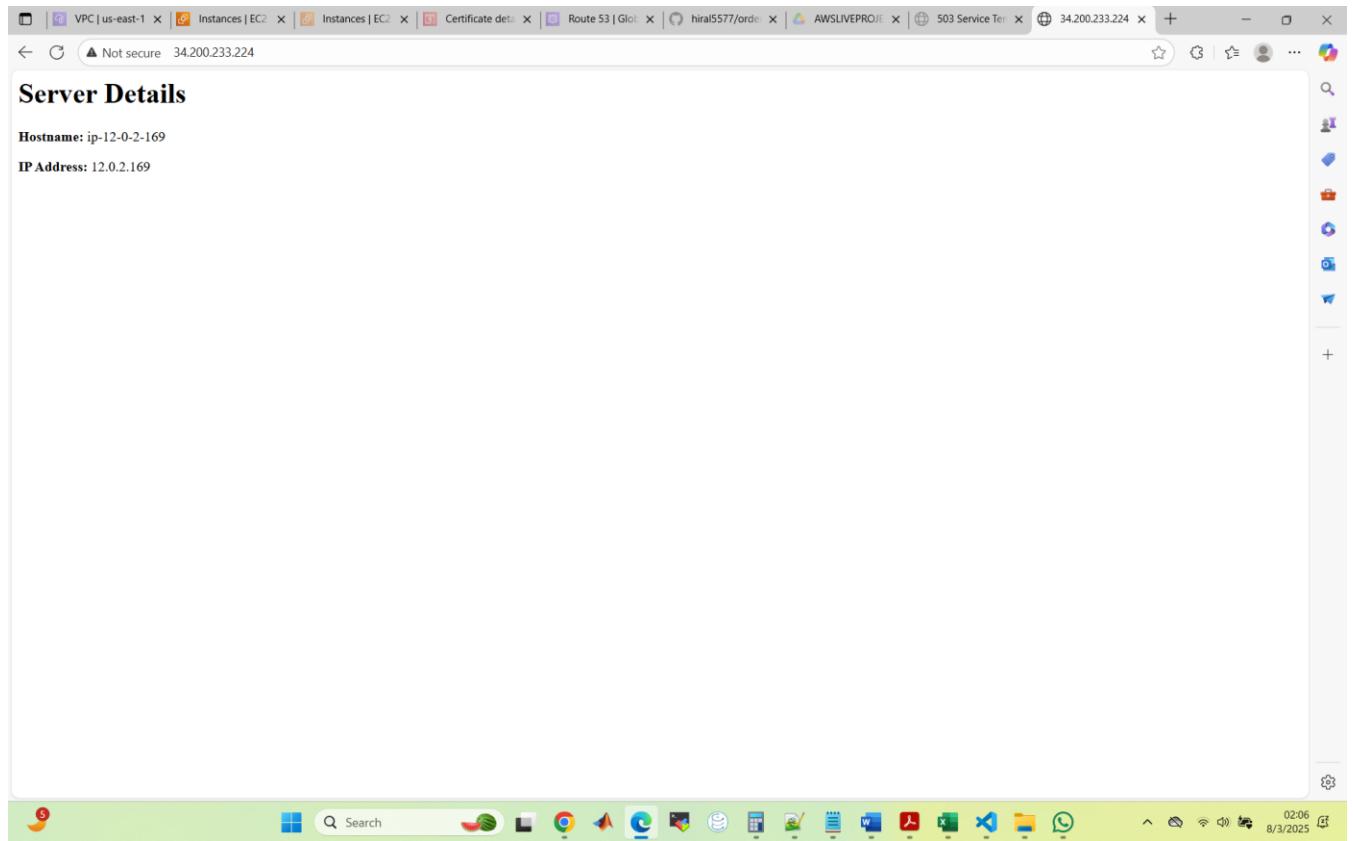
Wait for the instance to pass the “2/2 check”

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with sections like EC2, Instances, Images, Elastic Block Store, and Network & Security. The main area displays a table of instances. The first instance, "test-instance", has an Instance ID of i-01da05ddfa6b69a11, is running, and has passed 2/2 checks. The second instance, "cluster-server", has an Instance ID of i-02cf0bf1114013caf, is stopped. There are buttons for Connect, Instance state, Actions, and Launch instances.

The instance has passed the 2/2 check. Select the instance

The screenshot shows the same EC2 Instances page, but now the "test-instance" row is selected, indicated by a blue border. In the bottom half of the screen, a detailed view for "i-01da05ddfa6b69a11 (test-instance)" is displayed. The "Details" tab is active. Under the "Instance summary" section, the "Public IPv4 address" field contains "34.200.233.224 | open address". An orange arrow points from the text "Copy the ‘Public IPv4 address’ and test on the browser." to this specific field. Other details shown include Instance state (Running), Instance type (t2.micro), and VPC ID (vpc-0a178e5d2aecc4790 (test-vpc)).

Copy the “Public IPv4 address” and test on the browser.



You can see that the supposed homepage of the EC2 instance is displayed. So far everything is working.

Remember we are not going to use the IP address. Our aim is to create a load balancer then point that domain to this particular IP address.

STEP 6: Set up Load Balancer

The next resource to create is the load balancer but before we create a load balancer, we need to create a “**Target Group**” for the load balancer.

Go to the EC2 dashboard

This screenshot shows the AWS EC2 Instances dashboard. On the left, there's a navigation sidebar with sections like Spot Requests, Images, Elastic Block Store, Network & Security, Load Balancing, Auto Scaling, and Settings. Under the Load Balancing section, the 'Target Groups' link is highlighted with an orange arrow. The main pane displays a table of instances with columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. Two instances are listed: 'test-instance' (Running, t2.micro, 2/2 checks passed, us-east-1b) and 'cluster-server' (Stopped, t2.micro, us-east-1a). Below the table, a dropdown menu says 'Select an instance'.

Click on “Target Groups”.

This screenshot shows the AWS Target groups dashboard. The left sidebar includes the same navigation categories as the previous screen. The main area is titled 'Target groups' and shows a table with columns for Name, ARN, Port, Protocol, Target type, Load balancer, and VPC ID. A message at the top states 'No target groups' and 'You don't have any target groups in us-east-1'. A blue 'Create target group' button is prominently displayed. An orange arrow points from the 'Create target group' button back up towards the 'Load Balancing' section in the sidebar.

Click on “Create Target Group”. Target group is just a logical group where we will put our EC2 instance. For example, we have created one EC2 instance, so we can add that EC2 instance to one target group. One target group can contain multiple EC2 instances, so that our Load Balancer can point to the target group and the target group can have a bunch of EC2 instances running. Load balancer can direct the traffic between those EC2 instances.

Step 1
Specify group details

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

Basic configuration
Settings in this section can't be changed after the target group is created.

Choose a target type

- Instances
 - Supports load balancing to instances within a specific VPC.
 - Facilitates the use of [Amazon EC2 Auto Scaling](#) to manage and scale your EC2 capacity.
- IP addresses
 - Supports load balancing to VPC and on-premises resources.
 - Facilitates routing to multiple IP addresses and network interfaces on the same instance.
 - Offers flexibility with microservice based architectures, simplifying inter-application communication.
 - Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.
- Lambda function
 - Facilitates routing to a single Lambda function.
 - Accessible to Application Load Balancers only.
- Application Load Balancer
 - Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
 - Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

Target group name

test-tg

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol

Default protocol for load balancing requests to this target group. You can change it later.

Port

Default port number for incoming traffic to this target group. You can change it later.

CloudShell Feedback

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

For the “Target Type”, we will use “Instances”. Then for “Target Group Name”, we will use “test-tg”

Step 1
Specify group details

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

Basic configuration
Settings in this section can't be changed after the target group is created.

Choose a target type

- Instances
 - Supports load balancing to instances within a specific VPC.
 - Facilitates the use of [Amazon EC2 Auto Scaling](#) to manage and scale your EC2 capacity.
- IP addresses
 - Supports load balancing to VPC and on-premises resources.
 - Facilitates routing to multiple IP addresses and network interfaces on the same instance.
 - Offers flexibility with microservice based architectures, simplifying inter-application communication.
 - Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.
- Lambda function
 - Facilitates routing to a single Lambda function.
 - Accessible to Application Load Balancers only.
- Application Load Balancer
 - Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
 - Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

Target group name

test-tg

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol

Default protocol for load balancing requests to this target group. You can change it later.

Port

Default port number for incoming traffic to this target group. You can change it later.

CloudShell Feedback

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

Scroll down to “VPC”

Target group name
test-tg

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol
Protocol for load balancer-to-target communication. Can't be modified after creation.

Port
Port number where targets receive traffic. Can be overridden for individual targets during registration.

IP address type
Only targets with the indicated IP address type can be registered to this target group.

IPv4
Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.

IPv6
Each instance you register must have an assigned primary IPv6 address. This is configured on the instance's default network interface (eth0). [Learn more](#)

VPC
Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.

vpc-0128e9209eaef1c37
172.31.0.0/16 (default)

Protocol version
 HTTP1
Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.

HTTP2
Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.

gRPC
Send requests to targets using gRPC. Supported when the request protocol is gRPC.

Click on the drop down on VPC and select our VPC that is “test-vpc”

Target group name
test-tg

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol
Protocol for load balancer-to-target communication. Can't be modified after creation.

Port
Port number where targets receive traffic. Can be overridden for individual targets during registration.

IP address type
Only targets with the indicated IP address type can be registered to this target group.

IPv4
Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.

IPv6
Each instance you register must have an assigned primary IPv6 address. This is configured on the instance's default network interface (eth0). [Learn more](#)

VPC
Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.

vpc-0a178e5d2aec4790 (test-vpc)
12.0.0.0/16

Protocol version
 HTTP1
Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.

HTTP2
Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.

gRPC
Send requests to targets using gRPC. Supported when the request protocol is gRPC.

Then scroll down to the end

Health checks
The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol
HTTP

Health check path
Use the default path of "/" to perform health checks on the root, or specify a custom path if preferred.
/

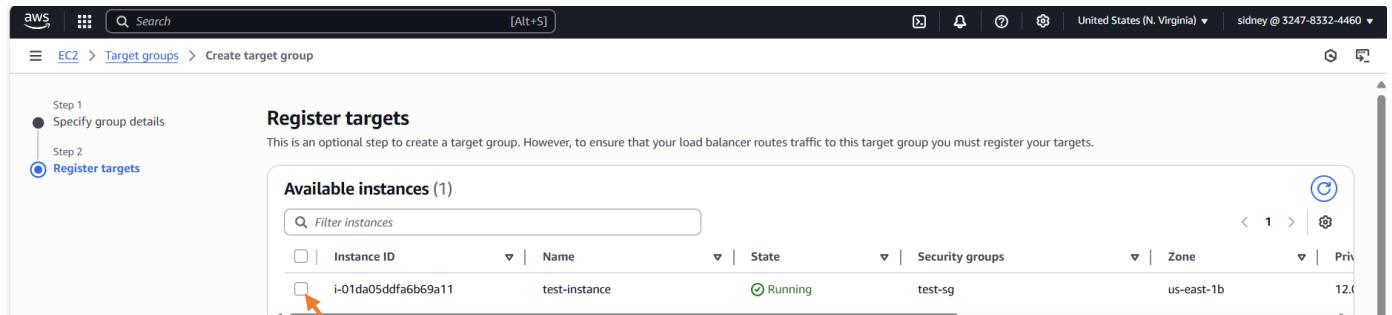
Advanced health check settings

Attributes
Certain default attributes will be applied to your target group. You can view and edit them after creating the target group.

Tags - optional
Consider adding tags to your target group. Tags enable you to categorize your AWS resources so you can more easily manage them.

[Cancel](#) **Next**

Click on “Next”



Step 1
 Specify group details
 Step 2
 Register targets

Register targets

This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.

Available instances (1)

Instance ID	Name	State	Security groups	Zone	Private IP
i-01da05ddfa6b69a11	test-instance	Running	test-sg	us-east-1b	12.0.0.1

Ports for the selected instances
Ports for routing traffic to the selected instances.
80
1-65535 (separate multiple ports with commas)
[Include as pending below](#)

Review targets

Targets (0)

Instance ID	Name	Port	State	Security groups	Zone	Private IPv4 address	Subnet ID	Launch time
-------------	------	------	-------	-----------------	------	----------------------	-----------	-------------

No instances added yet
Specify instances above, or leave the group empty if you prefer to add targets later.

[Remove all pending](#)

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

Now, we have to add our instance to the target group. Select the instance

Screenshot of the AWS EC2 Target Groups "Create target group" wizard Step 2: Register targets.

Available instances (1/1)

Instance ID	Name	State	Security groups	Zone	Private IP
i-01da05ddfa6b69a11	test-instance	Running	test-sg	us-east-1b	12.0.2.169

Ports for the selected instances
Ports for routing traffic to the selected instances.
80
1-65555 (separate multiple ports with commas)

Review targets

Targets (0)

Include as pending below (button highlighted with an orange arrow)

Click on “Include as pending below”

Screenshot of the AWS EC2 Target Groups "Create target group" wizard Step 2: Register targets, after clicking "Include as pending below".

Available instances (1)

Instance ID	Name	State	Security groups	Zone	Private IP
i-01da05ddfa6b69a11	test-instance	Running	test-sg	us-east-1b	12.0.2.169

Ports for the selected instances
Ports for routing traffic to the selected instances.
80
1-65555 (separate multiple ports with commas)

Review targets

Targets (1)

Create target group (button highlighted with an orange arrow)

Then click on “create target group”

Successfully created the target group: test-tg. Anomaly detection is automatically applied to all registered targets. Results can be viewed in the Targets tab.

test-tg

Details

Target type Instance	Protocol : Port HTTP: 80	Protocol version HTTP1
IP address type IPv4	Load balancer None associated	VPC vpc-0a178e5d2aecc4790

1 Total targets	0 Healthy	0 Unhealthy	1 Unused	0 Initial	0 Draining
0 Anomalous					

Distribution of targets by Availability Zone (AZ)
Select values in this table to see corresponding filters applied to the Registered targets table below.

Targets | Monitoring | Health checks | Attributes | Tags

Registered targets (1) [Info](#)

Anomaly mitigation: Not applicable							Deregister	Register targets
Filter targets	Instance ID	Name	Port	Zone	Health status	Health status details	Administrative o...	Override details
	i-01da05ddfa6b69a11	test-instance	80	us-east-1b (us...)	Unused	Target group is not co...	-	-

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

The target group has been created and our instance has been added to this target group.

Now, we will create our load balancer. Click on “Load Balancers”

EC2

Load balancers

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Name	State	Type	Scheme	IP address type	VPC ID	Availability Zones	Security
No load balancers You don't have any load balancers in us-east-1							
Create load balancer							

0 load balancers selected

Select a load balancer above.

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

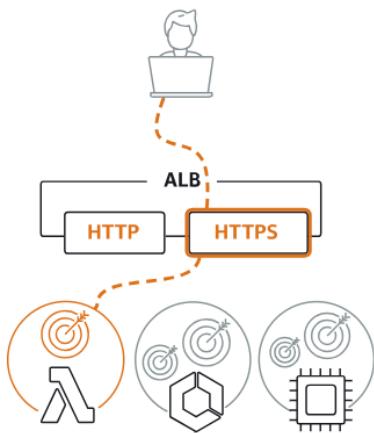
Click on “create load balancer”

Compare and select load balancer type

A complete feature-by-feature comparison along with detailed highlights is also available. [Learn more](#)

Load balancer types

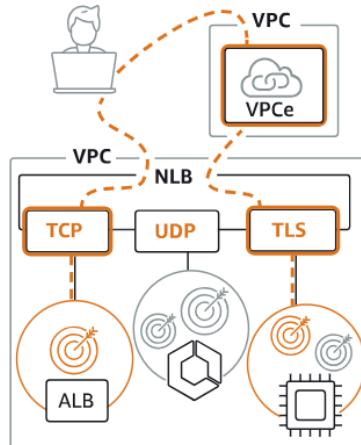
Application Load Balancer [Info](#)



Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.

[Create](#)

Network Load Balancer [Info](#)



Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.

[Create](#)

Gateway Load Balancer [Info](#)



Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls.

[Create](#)

► [Classic Load Balancer - previous generation](#)

[Close](#)

We are going to use application load balancer, so click on “create” under “Application Load Balancer”

Create Application Load Balancer Info

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

How Application Load Balancers work

Basic configuration

Load balancer name
Name must be unique within your AWS account and can't be changed after the load balancer is created.

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme Info
Scheme can't be changed after the load balancer is created.

Internet-facing

- Serves internet-facing traffic.
- Has public IP addresses.
- DNS name resolves to public IPs.
- Requires a public subnet.

Internal

- Serves internal traffic.
- Has private IP addresses.
- DNS name resolves to private IPs.
- Compatible with the **IPv4** and **Dualstack** IP address types.

Load balancer IP address type Info
Select the front-end IP address type to assign to the load balancer. The VPC and subnets mapped to this load balancer must include the selected IP address types. Public IPv4 addresses have an additional cost.

IPv4
Includes only IPv4 addresses.

Dualstack
Includes IPv4 and IPv6 addresses.

Dualstack without public IPv4
Includes a public IPv6 address, and private IPv4 and IPv6 addresses. Compatible with **internet-facing** load balancers only.

Network mapping

VPC Info

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

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

We have to give the Load Balancer a name, I will call it “**test-lb**”

Create Application Load Balancer Info

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

How Application Load Balancers work

Basic configuration

Load balancer name
Name must be unique within your AWS account and can't be changed after the load balancer is created.

test-lb

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme Info
Scheme can't be changed after the load balancer is created.

Internet-facing

- Serves internet-facing traffic.
- Has public IP addresses.
- DNS name resolves to public IPs.
- Requires a public subnet.

Internal

- Serves internal traffic.
- Has private IP addresses.
- DNS name resolves to private IPs.
- Compatible with the **IPv4** and **Dualstack** IP address types.

Load balancer IP address type Info
Select the front-end IP address type to assign to the load balancer. The VPC and subnets mapped to this load balancer must include the selected IP address types. Public IPv4 addresses have an additional cost.

IPv4
Includes only IPv4 addresses.

Dualstack
Includes IPv4 and IPv6 addresses.

Dualstack without public IPv4
Includes a public IPv6 address, and private IPv4 and IPv6 addresses. Compatible with **internet-facing** load balancers only.

Network mapping

VPC Info

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

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

The scroll down to “**Network Mapping**”

Network mapping [Info](#)
The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC | [Info](#)
The load balancer will exist and scale within the selected VPC. The selected VPC is also where the load balancer targets must be hosted unless routing to Lambda or on-premises targets, or if using VPC peering. To confirm the VPC for your targets, view [target groups](#).

vpc-0128e9209eaef1c37
172.31.0.0/16

IP pools - new | [Info](#)
You can optionally choose to configure an IPAM pool as the preferred source for your load balancers IP addresses. Create or view [Pools](#) in the [Amazon VPC IP Address Manager console](#).

Use IPAM pool for public IPv4 addresses
The IPAM pool you choose will be the preferred source of public IPv4 addresses. If the pool is depleted IPv4 addresses will be assigned by AWS.

Availability Zones and subnets | [Info](#)
Select at least two Availability Zones and a subnet for each zone. A load balancer node will be placed in each selected zone and will automatically scale in response to traffic. The load balancer routes traffic to targets in the selected Availability Zones only.

us-east-1a (use1-az6)

us-east-1b (use1-az1)

us-east-1c (use1-az2)

us-east-1d (use1-az4)

us-east-1e (use1-az3)

us-east-1f (use1-az5)

Click on the drop down on “VPC” and select our VPC that is “test-vpc”

Network mapping [Info](#)
The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC | [Info](#)
The load balancer will exist and scale within the selected VPC. The selected VPC is also where the load balancer targets must be hosted unless routing to Lambda or on-premises targets, or if using VPC peering. To confirm the VPC for your targets, view [target groups](#).

vpc-0a178e5d2aecc4790 (test-vpc)
12.0.0.0/16

IP pools - new | [Info](#)
You can optionally choose to configure an IPAM pool as the preferred source for your load balancers IP addresses. Create or view [Pools](#) in the [Amazon VPC IP Address Manager console](#).

Use IPAM pool for public IPv4 addresses
The IPAM pool you choose will be the preferred source of public IPv4 addresses. If the pool is depleted IPv4 addresses will be assigned by AWS.

Availability Zones and subnets | [Info](#)
Select at least two Availability Zones and a subnet for each zone. A load balancer node will be placed in each selected zone and will automatically scale in response to traffic. The load balancer routes traffic to targets in the selected Availability Zones only.

us-east-1a (use1-az6)

us-east-1b (use1-az1)

On “Availability Zones and subnets”, select the two availability zones and their subnets

Network mapping [Info](#)
The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC | [Info](#)
The load balancer will exist and scale within the selected VPC. The selected VPC is also where the load balancer targets must be hosted unless routing to Lambda or on-premises targets, or if using VPC peering. To confirm the VPC for your targets, view [target groups](#).

vpc-0a178e5d2aecc4790 (test-vpc)
12.0.0.0/16

IP pools - new | [Info](#)
You can optionally choose to configure an IPAM pool as the preferred source for your load balancers IP addresses. Create or view [Pools](#) in the [Amazon VPC IP Address Manager console](#).

Use IPAM pool for public IPv4 addresses
The IPAM pool you choose will be the preferred source of public IPv4 addresses. If the pool is depleted IPv4 addresses will be assigned by AWS.

Availability Zones and subnets | [Info](#)
Select at least two Availability zones and a subnet for each zone. A load balancer node will be placed in each selected zone and will automatically scale in response to traffic. The load balancer routes traffic to targets in the selected Availability Zones only.

us-east-1a (use1-az6)
Subnet
Only CIDR blocks corresponding to the load balancer IP address type are used. At least 8 available IP addresses are required for your load balancer to scale efficiently.
subnet-04e5dcf10023b20f3
IPv4 subnet CIDR: 12.0.1.0/24

us-east-1b (use1-az1)
Subnet
Only CIDR blocks corresponding to the load balancer IP address type are used. At least 8 available IP addresses are required for your load balancer to scale efficiently.
subnet-013af1016c64ee796
IPv4 subnet CIDR: 12.0.2.0/24

In the load balancer we also need to enable the same port 22 for SSH so that we will be able to connect to our EC2 instance. We need to also enable port 80 for http and also enable **port 443**. For now, we will enable the **security group for Port 22 and 80**.

Scroll down to “Security Group”

Security groups [Info](#)
A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can [create a new security group](#).

Security groups

Select up to 5 security groups

default
sg-0ffb0ddfd008e8ede VPC: vpc-0a178e5d2aecc4790

Remove the “default” security group

Security groups [Info](#)
A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can [create a new security group](#).

Security groups

Select up to 5 security groups

⚠ Application Load Balancers require at least one security group. If none are selected, the VPC's default security group will be applied.

Click on “create a new security group”

Create security group [Info](#)
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](#)
MyWebServerGroup

Name cannot be edited after creation.

Description [Info](#)
Allows SSH access to developers

VPC [Info](#)
vpc-0128e9209eaef1c37

Inbound rules [Info](#)
This security group has no inbound rules.

Add rule

Outbound rules [Info](#)

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

Add rule

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

We will call the security group “**test-sg-for-lb**”. For description, we will type “Allow ssh port 22 and 80”

A screenshot of the AWS EC2 'Create security group' page. The 'Basic details' section shows a security group name 'test-sg-for-lb' and a description 'Allow ssh port 22 and 80'. The 'VPC Info' section shows a dropdown menu with 'vpc-0128e9209eaef1c37' selected. An orange arrow points to the dropdown arrow icon. The 'Inbound rules' section indicates no inbound rules, with an 'Add rule' button. The 'Outbound rules' section shows a table with columns for Type, Protocol, Port range, Destination, and Description - optional. The 'Type' column has 'All traffic' selected. The 'Protocol' column has 'All' selected. The 'Port range' column has 'All' selected. The 'Destination' column has 'Custom' selected and a '0.0.0.0/0' entry. The 'Description - optional' column is empty. The bottom of the page includes standard AWS navigation links like CloudShell, Feedback, and cookie preferences.

Then for “**VPC**”, click on the drop down and select our VPC, that is “**test-vpc**”

A screenshot of the AWS EC2 'Create security group' page. The 'Basic details' section shows a security group name 'test-sg-for-lb' and a description 'Allow ssh port 22 and 80'. The 'VPC Info' section shows a dropdown menu with 'vpc-0a178e5d2aecc4790 (test-vpc)' selected. The 'Inbound rules' section indicates no inbound rules, with an 'Add rule' button. The 'Outbound rules' section shows a table with columns for Type, Protocol, Port range, Destination, and Description - optional. The 'Type' column has 'All traffic' selected. The 'Protocol' column has 'All' selected. The 'Port range' column has 'All' selected. The 'Destination' column has 'Custom' selected and a '0.0.0.0/0' entry. The 'Description - optional' column is empty. The bottom of the page includes standard AWS navigation links like CloudShell, Feedback, and cookie preferences.

Then, under “**Inbound Rules**”, click on “**Add rule**”

Create security group Info

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
test-sg-for-lb
Name cannot be edited after creation.

Description Info
Allow ssh port 22 and 80

VPC Info
vpc-0a178e5d2aec4790 (test-vpc)

Inbound rules Info

Type	Protocol	Port range	Source	Description - optional
Custom TCP	TCP	0	Custom	Allow ssh port 22 and 80

[Add rule](#) [Delete](#)

Outbound rules Info

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

[Delete](#)

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

Now, we will add HTTP and SSH. Click on the drop down on “Type” and select “HTTP”

Create security group Info

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
test-sg-for-lb
Name cannot be edited after creation.

Description Info
Allow ssh port 22 and 80

VPC Info
vpc-0a178e5d2aec4790 (test-vpc)

Inbound rules Info

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	Custom	Allow ssh port 22 and 80

[Add rule](#) [Delete](#)

Outbound rules Info

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

[Delete](#)

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

Click on the drop down on “Source” and select “Anywhere-IPv4”

The screenshot shows the AWS EC2 Security Groups 'Create security group' interface. It includes sections for Inbound rules, Outbound rules, and Tags - optional. The Inbound rules section has a rule for HTTP (TCP port 80) from anywhere. The Outbound rules section has a rule for all traffic (Custom TCP port 0) to 0.0.0.0/0. The Tags section is optional, showing no tags associated with the resource.

Inbound rules

Type: HTTP
Protocol: TCP
Port range: 80
Source: Anywhere
Description: 0.0.0.0/0

Add rule

Outbound rules

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

Add rule

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.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags

Cancel Create security group

Now, let us add SSH. Click on “Add rule”

The screenshot shows the same AWS EC2 Security Groups 'Create security group' interface. An orange arrow points to the 'Type' dropdown menu in the Inbound rules section, which currently displays 'HTTP'. This indicates where the user needs to click to change the rule type to SSH.

Inbound rules

Type: HTTP
Protocol: TCP
Port range: 80
Source: Anywhere
Description: 0.0.0.0/0

Custom TCP
Add rule

Outbound rules

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

Add rule

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.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags

Cancel Create security group

Click on the drop down on “Type” and select “SSH”

Inbound rules

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	Anywhere... <input type="button" value="Custom"/>	0.0.0.0/0 <input type="button" value="Delete"/>
SSH	TCP	22	Custom <input type="button" value="Anywhere..."/>	0.0.0.0/0 <input type="button" value="Delete"/>

Add rule

Outbound rules

Type	Protocol	Port range	Destination	Description - optional
All traffic	All	All	Custom <input type="button" value="Anywhere..."/>	0.0.0.0/0 <input type="button" value="Delete"/>

Add rule

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.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags

Cancel Create security group

Then click on the drop down on “Source” and select “Anywhere-IPv4”

Inbound rules

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	Anywhere... <input type="button" value="Custom"/>	0.0.0.0/0 <input type="button" value="Delete"/>
SSH	TCP	22	Anywhere-IPv4 <input type="button" value="Custom"/>	0.0.0.0/0 <input type="button" value="Delete"/>

Add rule

Outbound rules

Type	Protocol	Port range	Destination	Description - optional
All traffic	All	All	Custom <input type="button" value="Anywhere..."/>	0.0.0.0/0 <input type="button" value="Delete"/>

Add rule

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.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags

Cancel Create security group

Click on “Create Security Group”

EC2 > Security Groups > sg-0eb32582090cab331 - test-sg-for-lb

Details

Security group name test-sg-for-lb	Security group ID sg-0eb32582090cab331	Description Allow ssh port 22 and 80	VPC ID vpc-0a178e5d2aec4790
Owner 324783324460	Inbound rules count 2 Permission entries	Outbound rules count 1 Permission entry	

Inbound rules | Outbound rules | Sharing - new | VPC associations - new | Tags

Inbound rules (2)

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
-	sgr-03de90b7b26862f6f	IPv4	HTTP	TCP	80	0.0.0.0/0
-	sgr-0b7ae0aeb243c754a	IPv4	SSH	TCP	22	0.0.0.0/0

The security group has been created. Head back to “**security Group**” in our “**Application Load Balancer**” tab

Security groups [Info](#)

A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can [create a new security group](#).

Security groups

Select up to 5 security groups

⚠ Application Load Balancers require at least one security group. If none are selected, the VPC's default security group will be applied.

Refresh and select the security group we just created

Security groups [Info](#)

A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can [create a new security group](#).

Security groups

Select up to 5 security groups

test-sg-for-lb
sg-0eb32582090cab331 VPC: vpc-0a178e5d2aec4790

Scroll down to “**Listeners and Routing**”

Listeners and routing Info

A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

▼ Listener HTTP:80

Protocol	Port	Default action	Remove
HTTP	80 1-65535	Forward to Select a target group Create target group	

Listener tags - optional
Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

[Add listener tag](#)
You can add up to 50 more tags.

[Add listener](#)
You can add up to 49 more listeners.

Click on the drop down on “Target Group” and select the target group we created

Listeners and routing Info

A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

▼ Listener HTTP:80

Protocol	Port	Default action	Remove
HTTP	80 1-65535	Forward to test-tg Target type: Instance, IPv4	

Listener tags - optional
Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

[Add listener tag](#)
You can add up to 50 more tags.

[Add listener](#)
You can add up to 49 more listeners.

The scroll down to the end

aws Search [Alt+S] United States (N. Virginia) sidney @ 3247-8332-4460

EC2 > Load balancers > Create Application Load Balancer

Review the load balancer configurations and make changes if needed. After you finish reviewing the configurations, choose [Create load balancer](#).

Summary
Review and confirm your configurations. [Estimate cost](#)

Basic configuration [Edit](#)
Name: test-lb
Scheme: Internet-facing
IP address type: IPv4

Network mapping [Edit](#)
VPC: vpc-0a178e5d2aec4790
Public IPv4 IPAM pool: -
Availability Zones and subnets:

- us-east-1a
 - subnet-04e5dcf10023b20f3
 - test-public-subnet-1-1a
- us-east-1b
 - subnet-013af1016c64ee796
 - test-public-subnet-1-1b

Security groups [Edit](#)
test-sg-for-lb
sg-0eb32582090cab331

Listeners and routing [Edit](#)
HTTP:80 | Target group: [test-tg](#)

Service integrations [Edit](#)
Amazon CloudFront + AWS Web Application Firewall (WAF): -
AWS WAF: -
AWS Global Accelerator: -

Tags [Edit](#)
-

Attributes
Certain default attributes will be applied to your load balancer. You can view and edit them after creating the load balancer.

Creation workflow and status

► **Server-side tasks and status**
After completing and submitting the above steps, all server-side tasks and their statuses become available for monitoring.

[Cancel](#) [Create load balancer](#)

Click on “create load balancer”

The screenshot shows the AWS EC2 Load Balancers console. In the top navigation bar, there is a green banner message: "Successfully created load balancer: test-lb". Below the banner, the load balancer details are displayed, including its name, type (Application), status (Provisioning), VPC (vpc-0a178e5d2aecc4790), and various availability zones. The DNS name is listed as test-lb-270db.us-east-1.elb.amazonaws.com (A Record). The "Listeners and rules" tab is selected, showing one rule for port 80 that forwards traffic to a target group named "test-tg".

Our load balancer has been created. Click on “Load Balancers”

The screenshot shows the AWS EC2 Load Balancers console with a single load balancer named "test-lb" listed. The "test-lb" entry is highlighted with a red arrow pointing to its "State" column, which shows "Provisioning...". The "Create load balancer" button is visible in the top right corner.

You can see that the load balancer is provisioning. It will take a couple of minutes for it to be provisioned.

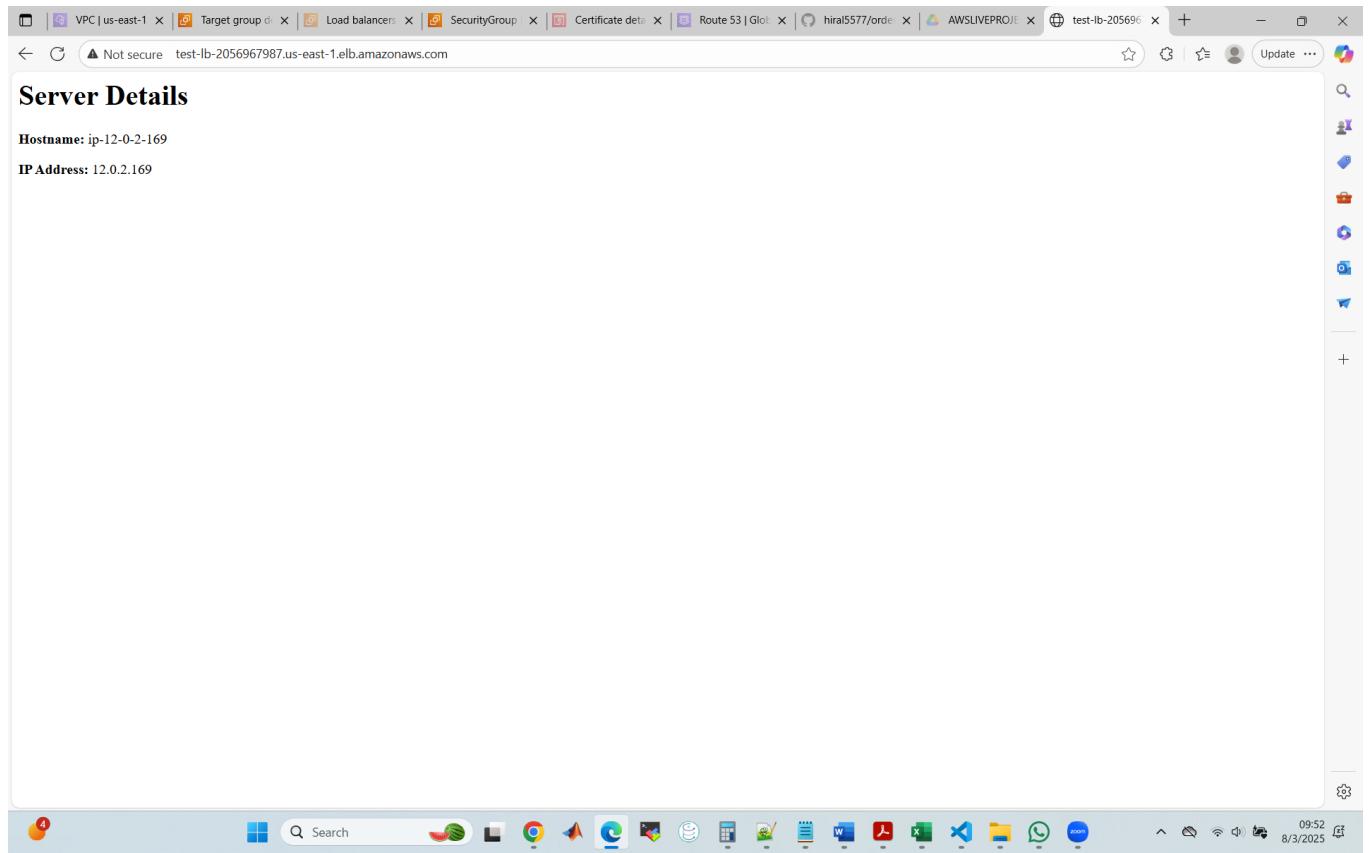
The screenshot shows the AWS EC2 Load Balancers page. On the left, there's a navigation sidebar with sections like EC2, Instances, Images, Elastic Block Store, Network & Security, and more. The main area is titled "Load balancers (1)" and shows a table with one row for "test-lb". The table columns include Name, State, Type, Scheme, IP address type, VPC ID, Availability Zones, and Security. The "test-lb" row has "Active" in the State column, "application" in Type, "Internet-facing" in Scheme, "IPv4" in IP address type, and "vpc-0a178e5d2aecc4790" in VPC ID. It also lists "2 Availability Zones" and "sg-0eb32" in Security. Below the table, it says "0 load balancers selected" and "Select a load balancer above."

Now, the load balancer is “**Active**” and running. Select the load balancer.

This screenshot shows the same EC2 Load Balancers page, but now the "test-lb" load balancer is selected. In the main content area, the title changes to "Load balancer: test-lb". Below it, there's a "Details" tab and several other tabs: Listeners and rules, Network mapping, Resource map, Security, Monitoring, Integrations, Attributes, Capacity, and Tags. The "Details" tab is active and displays information about the load balancer. It includes fields for Load balancer type (Application), Status (Active), VPC (vpc-0a178e5d2aecc4790), Scheme (Internet-facing), Hosted zone (Z355XDOTRQ7X7K), Availability Zones (subnet-04e5dcf10023b20f3 us-east-1a (use1-az6) and subnet-013af1016c64ee796 us-east-1b (use1-az1)), Load balancer ARN (arn:aws:elasticloadbalancing:us-east-1:324783324460:loadbalancer/app/test-lb/e37466de634270d8), and DNS name (test-lb-2056967987.us-east-1.elb.amazonaws.com (A Record)). An orange arrow points to the DNS name field. At the bottom of the page, there are links for CloudShell and Feedback, and a footer with copyright information.

Copy the DNS name and paste on your browser to test if the application load balancer is working.

test-lb-2056967987.us-east-1.elb.amazonaws.com



You can see that the load balancer is pointing to the EC2 instance which we have processed using the public IP. The DNS name has been used for this load balancer is pointing to our EC2 instance.

STEP 7: Setting up Amazon Route 53

In this step we are going to create a Hosted zone using Amazon Route 53. We do not want to use the Load Balancer DNS name; we want to use our own domain name. So, we need to introduce our Route 53.

Route 53 does not reside in the VPC, it is a global resource. Whenever you create a Route 53 or a hosted zone in Route 53, then it does not belong to any region or VPC. At last, we will have our domain name which will be used.

Once we create our Route 53, the user is going to use our URL and the URL is going to point to the Route 53 and then it will go to the Load Balancer and then the load balancer will point to our EC2 instance.

We start by searching for Route 53 on the AWS Management Console.

The screenshot shows the AWS Management Console search interface. The search bar at the top contains the text "Route 53". Below the search bar, there is a search result summary: "Search results for 'Route'" and a note "Try searching with longer queries for more relevant results". The main results section is titled "Services" and contains four items:

- Route 53** ★ Scalable DNS and Domain Name Registration
- Route 53 Resolver** Resolve DNS queries in your Amazon VPC and on-premises network.
- Route 53 Application Recovery Controller** ★ Monitor application recovery readiness and manage failovers
- Amazon Location Service** ★ Securely and easily add location data to applications.

Below the services section, there is a "Features" section with one item:

- Transit Gateway route tables** ★ VPC feature

On the left side of the search results, there is a sidebar with various navigation links and a "See all 31 results" link.

Click on “Route 53” under services

What is a Hosted Zone?

A hosted zone represents a set of records that belongs to a specific domain

Click on “Hosted Zones” on the Left-hand side.

Route 53 Dashboard

DNS management
A hosted zone tells Route 53 how to respond to DNS queries for a domain such as example.com.

Traffic management
A visual tool that lets you easily create policies for multiple endpoints in complex configurations.

Availability monitoring
Health checks monitor your applications and web resources, and direct DNS queries to healthy resources.

Domain registration
1 Domain

Register domain
Find and register an available domain, or transfer your existing domains to Route 53.

Notifications 13

Resource	Status	Last update
sosoebot.com	Update transfer lock successful	2025-08-03 11:54:24
sosoebot.com	Update transfer lock successful	2025-08-03 11:48:56
sosoebot.com	Update transfer lock successful	2025-08-03 11:48:17
sosoebot.com	Domain ownership change successful	2025-08-03 11:36:03
sosoebot.com	Update domain contact successful	2025-08-03 11:29:46

Click on “Hosted Zones”

Route 53 Hosted zones

Hosted zones (0)

No hosted zones
There are no hosted zones created for this account.

Create hosted zone

Click on “create Hosted Zone”

Screenshot of the AWS Route 53 'Create hosted zone' configuration page. The 'Domain name' field is highlighted with a red arrow pointing to it. The 'Domain name' input field contains 'example.com'. Below the domain name, there is a description field labeled 'The hosted zone is used for...' and a type selection section for 'Public hosted zone' and 'Private hosted zone'. The 'Tags' section shows no tags associated with the resource.

[Cancel](#) [Create hosted zone](#)

Enter our domain name “**sosoebot.org**”

Screenshot of the AWS Route 53 'Create hosted zone' configuration page. The 'Domain name' field is highlighted with a red arrow pointing to it. The 'Domain name' input field contains 'sosoebot.org'. Below the domain name, there is a description field labeled 'The hosted zone is used for...' and a type selection section for 'Public hosted zone' and 'Private hosted zone'. The 'Tags' section shows no tags associated with the resource.

[Cancel](#) [Create hosted zone](#)

Click on “create hosted zone”

The screenshot shows the AWS Route 53 Hosted Zones interface. On the left, there's a navigation sidebar with sections like Route 53, IP-based routing, Traffic flow, Domains, and Resolver. The main area shows a success message: "sosoebot.org was successfully created. Now you can create records in the hosted zone to specify how you want Route 53 to route traffic for your domain." Below this, the "Hosted zone details" section is shown for the domain "sosoebot.org". Under the "Records" tab, there are two entries:

Type	Name	Value/Route traffic to
NS	sosoebot...	ns-1363.awsdns-42.org. ns-612.awsdns-12.net. ns-22.awsdns-02.com. ns-1952.awsdns-52.co.uk.
SOA	sosoebot...	ns-1363.awsdns-42.org. aw...

You can see that our hosted zone has two records with types NS (Name server) and SOA in it.

NAME SERVER

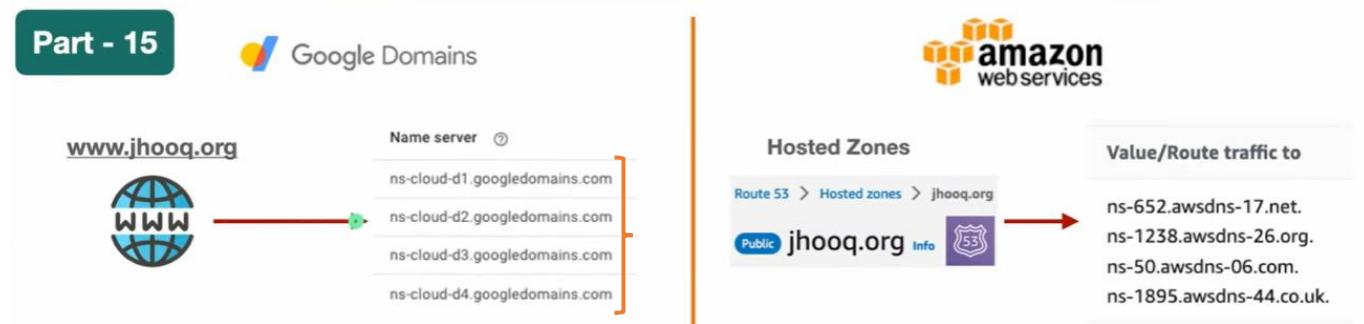
I would like to explain the concept of name server which is name server setting concept. If you purchase a domain name from a vendor other than Amazon, for example Google domains. You will have to do some name server settings.



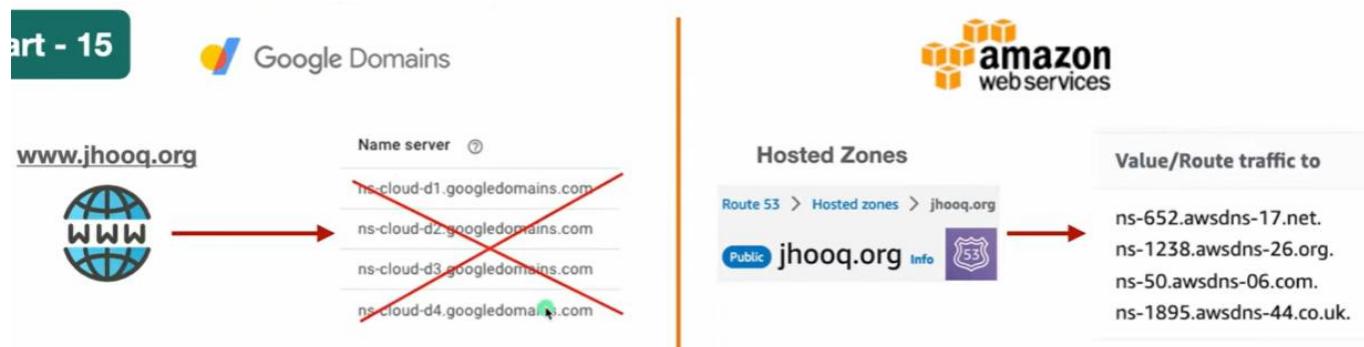
We have created a hosted zone. Once we create our hosted zone, we get our name server record which is from AWS, given below:

ns-1363.awsdns-42.org.
 ns-612.awsdns-12.net.
 ns-22.awsdns-02.com.
 ns-1952.awsdns-52.co.uk.

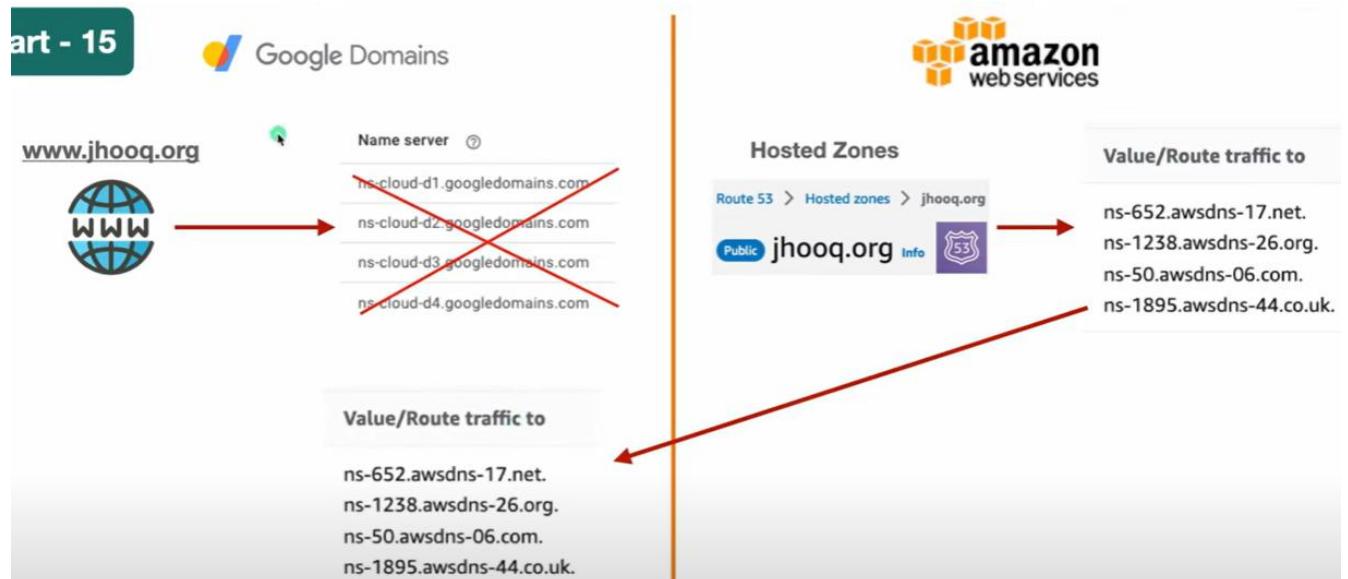
Also, if you purchase a domain name from any vendor other than AWS, for example: Google domains. The domain name will also have their own name servers.



As you can see above, these are records from Google domain. To point the Google domain to AWS services. We will not be able to use the Google Domain name servers.

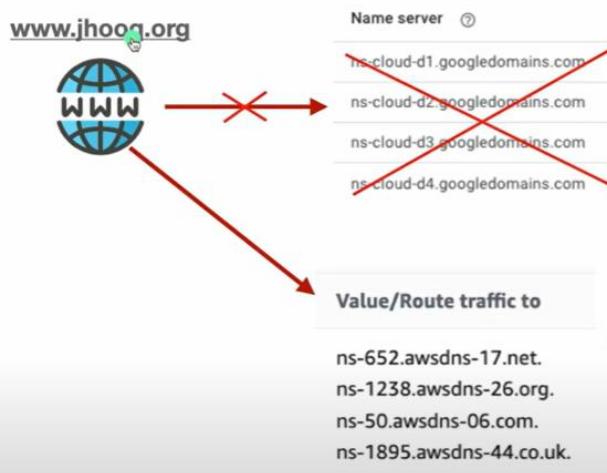


So, we need to reply on the name server record of AWS. We will copy the records from AWS and paste the record in the Google Domain's name server record.



So, the Google Domain will be pointing to the Domain name server of AWS instead of the Google name server.

Part - 15



Hosted Zones

Route 53 > Hosted zones > jhooq.org
Public jhooq.org Info

Value/Route traffic to

ns-652.awsdns-17.net.
ns-1238.awsdns-26.org.
ns-50.awsdns-06.com.
ns-1895.awsdns-44.co.uk.

This is how the name server settings works when you purchase the domain name from another service provider.

So, we have to modify the Name servers in our third-party Domain name

The screenshot shows the Squarespace DNS settings page. On the left, a sidebar menu includes "Overview", "DNS", "DNS Settings", "Domain Nameservers" (which is underlined), "Nameserver Registration", and "DNSSEC". The main content area is titled "Domain Nameservers" and contains the text: "Nameservers determine the way your website is found using your domain name. [Learn more about nameservers.](#)" Below this, a note says: "You are using Squarespace's nameservers for DNS resolution. Using our nameservers enables the most functionality for your domain. If you would like to use different nameservers instead, select Use Custom Nameservers above." To the right of this note is a button labeled "USE CUSTOM NAMESERVERS". An orange arrow points from the bottom of the slide towards this button.

Click on “Use Custom Nameservers”

The screenshot shows the 'Domain Nameservers' section of the Squarespace dashboard. On the left sidebar, under 'DNS Settings', 'Domain Nameservers' is selected. The main content area displays the 'Update nameservers' form. It contains two input fields labeled 'NAME SERVER 1' and 'NAME SERVER 2', both currently empty. Below these fields is a button labeled 'ADD NAMESERVER'. At the bottom right of the form are 'SAVE' and 'CANCEL' buttons. A large, semi-transparent modal window titled 'USE CUSTOM NAMESERVERS' is overlaid on the page, containing text about the benefits of using custom nameservers.

Paste the four copied Name servers we copied from AWS here leaving out the **point (.)** at the end

This screenshot shows the same 'Domain Nameservers' update page as the previous one, but with four name servers listed in the 'NAME SERVER 1' field. The servers are: ns-1363.awsdns-42.org, ns-612.awsdns-12.net, ns-22.awsdns-02.com, and ns-1952.awsdns-52.co.uk. Each server entry includes a small red trash can icon to its right. An orange arrow points from the bottom of the page towards the 'SAVE' button, which is highlighted in red. The rest of the interface and the 'USE CUSTOM NAMESERVERS' modal are identical to the first screenshot.

Click on “Save”

The screenshot shows the Squarespace DNS settings for the domain sosoebot.org. On the left, a sidebar lists various domain management options like Overview, DNS Settings, Domain Nameservers, and Website. The 'Domain Nameservers' section is selected. The main content area is titled 'Domain Nameservers' and contains a note: 'Nameservers determine the way your website is found using your domain name. [Learn more about nameservers.](#)' A button labeled 'USE SQUARESPACE NAMESERVERS' is visible. Below this, a list of four custom nameservers is shown: ns-1363.awsdns-42.org, ns-1952.awsdns-52.co.uk, ns-22.awsdns-02.com, and ns-612.awsdns-12.net. An orange bracket on the right side of the screen groups these four entries. At the bottom of the list is a link 'UPDATE NAMESERVERS'.

You can see that we have modified the name servers in our third-party DNS to match those in AWS.

Let us head back to our Hosted Zone and create an “A -record”

The screenshot shows the AWS Route 53 Hosted Zone details for the domain sosoebot.org. The left sidebar includes sections for Route 53, IP-based routing, Traffic flow, Domains, and Resolver. The 'Hosted zones' section is active. The main area displays the 'Hosted zone details' for sosoebot.org. Under the 'Records' tab, there are two existing NS records listed. A new record is being created, indicated by a yellow arrow pointing to the 'Create record' button at the top right of the table. The table columns include Record, Type, Value/Route traffic to, TTL, Health, and Evaluate.

Record	Type	Value/Route traffic to	TTL	Health	Evaluate
sosoebot...	NS	ns-1363.awsdns-42.org, ns-612.awsdns-12.net, ns-22.awsdns-02.com, ns-1952.awsdns-52.co.uk	172800	-	-
sosoebot...	SOA	ns-1363.awsdns-42.org, aws...	900	-	-

Now, we need to create an “A record” so that it points to our Load Balancer.

Click on “Create Record”

AWS Search [Alt+S] Global sidney @ 3247-8332-4460 ▾

Route 53 > Hosted zones > sosoebot.org > Create record

Create record Info

Quick create record

▼ Record 1

Record name Info **sosoebot.org** **Record type** Info

Keep blank to create a record for the root domain.

Alias

Value Info

Enter multiple values on separate lines.

TTL (seconds) Info **Routing policy** Info

1m 1h 1d Recommended values: 60 to 172800 (two days)

Add another record

Cancel **Create records**

switch to wizard

► View existing records

The following table lists the existing records in sosoebot.org.

CloudShell Feedback

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

Change the view by clicking on “Switch to Wizard”

AWS Search [Alt+S] Global sidney @ 3247-8332-4460 ▾

Route 53 > Hosted zones > sosoebot.org > Create record

Simple routing Use if you want all of your clients to receive the same response(s).

Weighted Use when you have multiple resources that do the same job, and you want to specify the proportion of traffic that goes to each resource. For example: two or more EC2 instances.

Geolocation Use when you want to route traffic based on the location of your users.

Latency Use when you have resources in multiple AWS Regions and you want to route traffic to the Region that provides the best latency.

Failover Use to route traffic to a resource when the resource is healthy, or to a different resource when the first resource is unhealthy.

Multivalue answer Use when you want Route 53 to respond to DNS queries with up to eight healthy records selected at random.

IP-based Use to route traffic to locations of IP address ranges in CIDR notation.

Geoproximity Use when you want Route 53 to route DNS queries to a certain geographic location.

Next

Cancel

Select “Simple Routing” and click on “Next”

Step 1
Choose routing policy
Step 2
Configure records

Configure records Info
You can create multiple records at a time that have the same routing policy.

Simple routing records to add to sosoebot.org Info
Use if you want all of your clients to receive the same response(s).

Record name	Type	Value/Route traffic to	TTL (seconds)
Define simple records to this list, then choose Create records .			

Existing records

Define simple record

Cancel Previous Create records

Click on “Define simple record”

Define simple record

Record name Info
To route traffic to a subdomain, enter the subdomain name. For example, to route traffic to blog.example.com, enter *blog*. If you leave this field blank, the default record name is the name of the domain.

sosoebot.org

Keep blank to create a record for the root domain.

Record type Info
The DNS type of the record determines the format of the value that Route 53 returns in response to DNS queries.

Choose when routing traffic to AWS resources for EC2, API Gateway, Amazon VPC, CloudFront, Elastic Beanstalk, ELB, or S3. For example: 192.0.2.44.

Value/Route traffic to Info
The option that you choose determines how Route 53 responds to DNS queries. For most options, you specify where you want to route internet traffic.

192.0.2.235

Enter multiple values on separate lines.

TTL (seconds) Info
The amount of time, in seconds, that DNS resolvers and web browsers cache the settings in this record. (“TTL” means “time to live.”). This value does not apply to alias records. [Learn more](#)

1m **1h** **1d**

Recommended values: 60 to 172800 (two days)

Cancel Define simple record

Click on the drop down on “Route Traffic to” and select “Application and Classic Load Balancer”

Define simple record

Record name | [Info](#)
To route traffic to a subdomain, enter the subdomain name. For example, to route traffic to blog.example.com, enter *blog*. If you leave this field blank, the default record name is the name of the domain.

sosoebot.org

Keep blank to create a record for the root domain.

Record type | [Info](#)
The DNS type of the record determines the format of the value that Route 53 returns in response to DNS queries.

Choose when routing traffic to AWS resources for EC2, API Gateway, Amazon VPC, CloudFront, Elastic Beanstalk, ELB, or S3. For example: 192.0.2.44.

Value/Route traffic to | [Info](#)
The option that you choose determines how Route 53 responds to DNS queries. For most options, you specify where you want to route internet traffic.





Evaluate target health
Select Yes if you want Route 53 to use this record to respond to DNS queries only if the specified AWS resource is healthy.

Yes

[Cancel](#) **Define simple record**

Click on the drop down on “Region” and select “us-east”

Define simple record

Record name | [Info](#)
To route traffic to a subdomain, enter the subdomain name. For example, to route traffic to blog.example.com, enter *blog*. If you leave this field blank, the default record name is the name of the domain.

sosoebot.org

Keep blank to create a record for the root domain.

Record type | [Info](#)
The DNS type of the record determines the format of the value that Route 53 returns in response to DNS queries.

Choose when routing traffic to AWS resources for EC2, API Gateway, Amazon VPC, CloudFront, Elastic Beanstalk, ELB, or S3. For example: 192.0.2.44.

Value/Route traffic to | [Info](#)
The option that you choose determines how Route 53 responds to DNS queries. For most options, you specify where you want to route internet traffic.







Evaluate target health
Select Yes if you want Route 53 to use this record to respond to DNS queries only if the specified AWS resource is healthy.

Yes

[Cancel](#) **Define simple record**

Then click on “Choose Load Balancer” and select the application load balancer we created.

Define simple record

Record name [Info](#)
 To route traffic to a subdomain, enter the subdomain name. For example, to route traffic to blog.example.com, enter blog. If you leave this field blank, the default record name is the name of the domain.

sosoebot.org

Keep blank to create a record for the root domain.

Record type [Info](#)
 The DNS type of the record determines the format of the value that Route 53 returns in response to DNS queries.

Choose when routing traffic to AWS resources for EC2, API Gateway, Amazon VPC, CloudFront, Elastic Beanstalk, ELB, or S3. For example: 192.0.2.44.

Value/Route traffic to [Info](#)
 The option that you choose determines how Route 53 responds to DNS queries. For most options, you specify where you want to route internet traffic.

[X](#)

Alias hosted zone ID: Z35SXDOTRQ7X7K

Evaluate target health
 Select Yes if you want Route 53 to use this record to respond to DNS queries only if the specified AWS resource is healthy.

Yes

[Cancel](#) [Define simple record](#)

Click on “Define simple record”

[aws](#) [Search](#) [Alt+S]

Route 53 > Hosted zones > sosoebot.org > Create record

Step 1 Choose routing policy
 Step 2 Configure records

Configure records [Info](#)
 You can create multiple records at a time that have the same routing policy.

Simple routing records to add to sosoebot.org [Info](#)
 Use if you want all of your clients to receive the same response(s).

<input type="checkbox"/>	Record name	Type	Value/Route traffic to	TTL (seconds)
<input type="checkbox"/>	sosoebot.org	A	dualstack.test-lb-2056967987.us-east-1...	-

[Edit](#) [Delete](#) [Define simple record](#)

[Existing records](#)

[Cancel](#) [Previous](#) [Create records](#)

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

Select the Record name

Step 1
Choose routing policy

Step 2
Configure records

Configure records Info

You can create multiple records at a time that have the same routing policy.

Simple routing records to add to sosoebot.org Info

Use if you want all of your clients to receive the same response(s).

Record name	Type	Value/Route traffic to	TTL (seconds)
sosoebot.org	A	dualstack.test-lb-2056967987.us-east-1...	-

Existing records

Create records

click on “Create Record”

Route 53

- Dashboard
- Hosted zones
- Health checks
- Profiles New
- IP-based routing
- Traffic flow
- Domains
- Resolver
- VPCs
- Inbound endpoints
- Outbound endpoints
- Rules
- Query logging
- Outposts

Hosted zones

sosoebot.org Info

Hosted zone details

Records (3) **DNSSEC signing** **Hosted zone tags (0)**

Records (3) Info

Automatic mode is the current search behavior optimized for best filter results. [To change modes go to settings.](#)

Record ...	Type	Routin...	Differ...	Alias	Value/Route traffic to	TTL (s...)	Health ...	Evaluat...	Recor...
sosoebot....	A	Simple	-	Yes	dualstack.test-lb-205696798...	-	-	Yes	-
sosoebot....	NS	Simple	-	No	ns-1363.awsdns-42.org. ns-612.awsdns-12.net. ns-22.awsdns-02.com. ns-1952.awsdns-52.co.uk.	172800	-	-	-
sosoebot....	SOA	Simple	-	No	ns-1363.awsdns-42.org. aws...	900	-	-	-

Create record

You can see that we have one more record that is pointing to our application load balancer.

Route 53 < Record for sosoebot.org was successfully created. Route 53 propagates your changes to all of the Route 53 authoritative DNS servers within 60 seconds. Use "View status" button to check propagation status.

sosoebot.org Info

Hosted zone details

Records (3) DNSSEC signing Hosted zone tags (0)

Records (3) Info

	Type	Routine...	Differ...	Alias	Value/Route traffic to	TTL (s...)	Health ...	Evaluat...	Record...
<input type="checkbox"/> sosoebot...	A	Simple	-	Yes	dualstack.test-lb-205696798...	-	-	Yes	-
<input type="checkbox"/> sosoebot...	NS	Simple	-	No	ns-1363.awsdns-42.org, ns-612.awsdns-12.net, ns-22.awsdns-02.com, ns-1952.awsdns-52.co.uk	172800	-	-	-
<input type="checkbox"/> sosoebot...	SOA	Simple	-	No	ns-1363.awsdns-42.org. aw...	900	-	-	-

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

Click on “check status”

Route 53 < Hosted zones > sosoebot.org > Change Info

C0856880L6S1DZ463731 Info

Change info details

ID /change/C0856880L6S1DZ463731 Submitted at August 06, 2025, 00:40 (UTC-04:00)

Status **INSYNC**

Comment -

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

You can see that the status is “**INSYNC**”.

Let us go back to our hosted zone by clicking on “**Hosted Zones**”

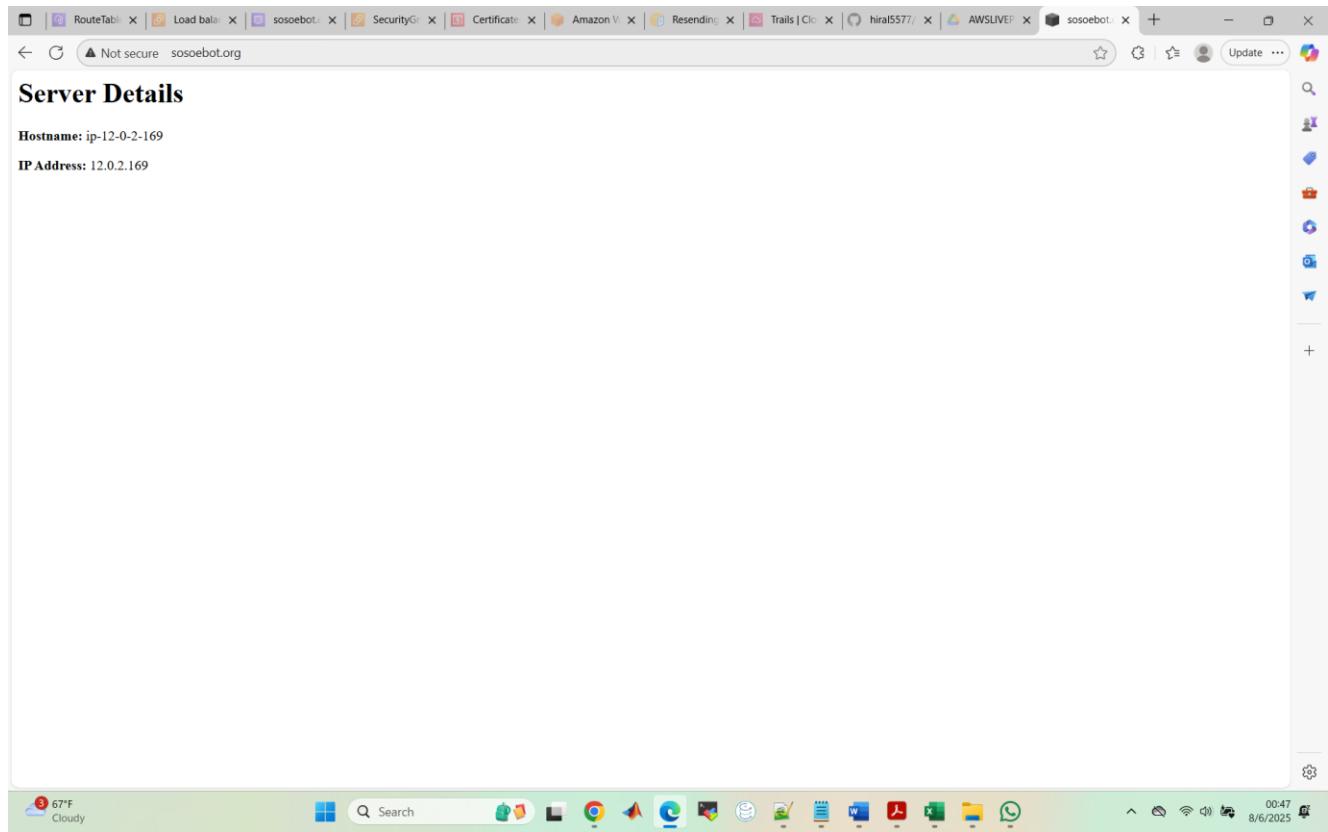
The screenshot shows the AWS Route 53 console. On the left, there's a navigation sidebar with various services like Dashboard, Hosted zones, Health checks, Profiles, IP-based routing, Traffic flow, Domains, Resolver, DNS Firewall, and Application Recovery Controller. The 'Hosted zones' section is selected. The main area is titled 'Hosted zones (1)' and shows a table with one row for 'sosoebot.org'. The table columns include Hosted zone name, Type, Created by, Record count, Description, and Hosted zone ID. The 'sosoebot.org' row has a Public type, created by 'Route 53', 3 records, and a long Hosted zone ID.

Click on the “Hosted Zone Name” that is “**sosoebot.org**”

This screenshot shows the 'Hosted zone details' page for 'sosoebot.org'. It includes sections for 'Hosted zone details', 'Records (3)', 'DNSSEC signing', and 'Hosted zone tags (0)'. The 'Records (3)' section lists three records: an A record pointing to 'dualstack.test-lb-205696798...', an NS record pointing to 'ns-1363.awsdns-42.org.', and an SOA record. An orange arrow points to the 'sosoebot.org' entry in the SOA record list.

Record name	Type	Value/Route traffic to	TTL (s...)	Health ...	Evaluat...
sosoebot.org	A	Simple	-	Yes	dualstack.test-lb-205696798...
sosoebot.org	NS	Simple	-	No	ns-1363.awsdns-42.org. ns-612.awsdns-12.net. ns-22.awsdns-02.com. ns-1952.awsdns-52.co.uk.
sosoebot.org	SOA	Simple	-	No	ns-1363.awsdns-42.org.awsd...

Copy the record name (**sosoebot.org**) and test on your browser



You can see that I am able to access the home page of my EC2 instance using my domain name "**sosoebot.org**"

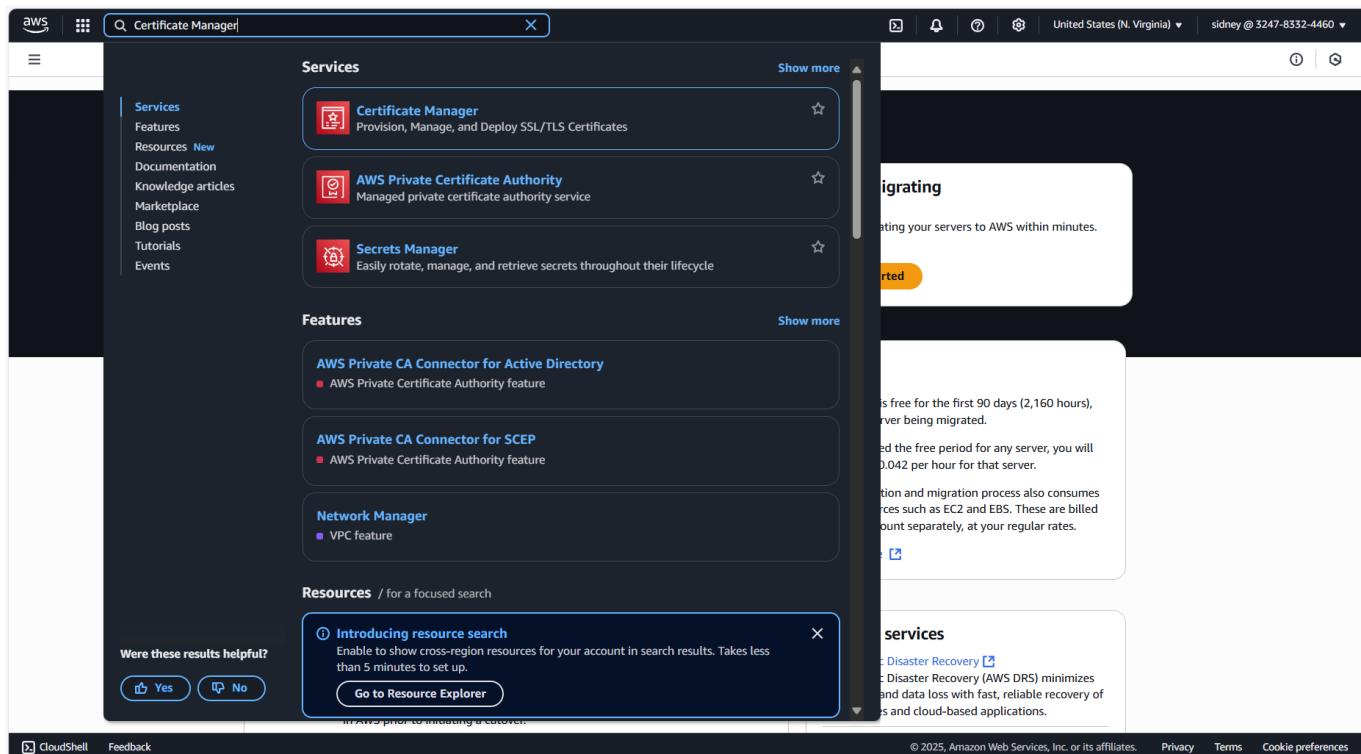
We have completed the set up and the user is able to access the URL without HTTPS. The HTTP is pointing to the Route 53 and the Route 53 is going through the Load Balancer which is accessing the EC2 instance.

STEP 8: Setting up General Certificate

The next thing we have to set up is the HTTPS and SSL. For that we need to have our AWS Certificate Manager. We are going to use the AWS Certificate Manager to request the certificate and once we get those certificates, we are going to create those records inside our AWS Route 53. Once we import or create those records inside our Route 53, then we will be able to use the HTTPS URL along with our domain.

Enabling HTTPS with AWS Certificate Manager (ACM), this ensures that secure connections are established between your visitors and your website.

Search for “Certificate Manager” on AWS Management Console



Click on “Certificate Manager”

AWS Certificate Manager (ACM)

Security, Identity & Compliance

AWS Certificate Manager

Easily provision, manage, deploy, and renew SSL/TLS certificates

New ACM managed certificate

Request a public certificate from Amazon or a private certificate from your organization's certificate authority (CA).

Request a certificate

Import certificates that you obtained outside of AWS

Import a certificate

Create private certificate authority (CA) hierarchies for your organization.

Create a private CA

How it works

- 1 Request or import a TLS/SSL certificate you would like to use into your AWS account.
- 2 Validate domain ownership for your requested certificate using Domain Name System (DNS) or email validation to complete certificate issuance.
- 3 Use your newly issued or imported certificates in various AWS services like Elastic Load Balancing (ELB), Amazon CloudFront etc.

Pricing (US)

Public SSL/TLS certificates provisioned through AWS Certificate Manager are free. You pay only for the AWS resources you create to run your application. [Learn more](#)

Getting started

What is Certificate Manager?

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

Click on “Request a Certificate”

Certificate type [Info](#)

ACM certificates can be used to establish secure communications access across the internet or within an internal network. Choose the type of certificate for ACM to provide.

Request a public certificate
Request a public SSL/TLS certificate from Amazon. By default, public certificates are trusted by browsers and operating systems.

Request a private certificate
No private CAs available for issuance.

Requesting a private certificate requires the creation of a private certificate authority (CA). To create a private CA, visit [AWS Private Certificate Authority](#)

Cancel Next

Click on “Next”

Screenshot of the AWS Certificate Manager 'Request public certificate' page. The 'Fully qualified domain name' field is highlighted with a red arrow. The 'Allow export' section shows 'Disable export' selected. The 'Validation method' section shows 'DNS validation - recommended' selected. The 'Key algorithm' section shows 'RSA 2048' selected.

Enter your domain name here. My domain name is “**sosoebot.org**”

Screenshot of the AWS Certificate Manager 'Request public certificate' page. The 'Fully qualified domain name' field contains 'sosoebot.org'. The 'Allow export' section shows 'Enable export' selected. The 'Validation method' section shows 'DNS validation - recommended' selected. The 'Key algorithm' section shows 'RSA 2048' selected.

Scroll down

Screenshot of the AWS Certificate Manager 'Request public certificate' wizard page 2.

Disable export
Use this certificate only with integrated AWS services. The private key for this certificate will be disallowed for exporting from AWS.

Enable export
Export this certificate and private key for use with any TLS workflow. ACM will charge your account based on the requested domains when the certificate is issued for the first time and for each renewal.

Validation method Info
Select a method for validating domain ownership.

- DNS validation - recommended**
Choose this option if you are authorized to modify the DNS configuration for the domains in your certificate request.
- Email validation**
Choose this option if you do not have permission or cannot obtain permission to modify the DNS configuration for the domains in your certificate request.

Key algorithm Info
Select an encryption algorithm. Some algorithms may not be supported by all AWS services.

- RSA 2048**
RSA is the most widely used key type.
- ECDSA P 256**
Equivalent in cryptographic strength to RSA 3072.
- ECDSA P 384**
Equivalent in cryptographic strength to RSA 7680.

Tags Info
No tags associated with the resource.

[Add new tag](#)
You can add up to 50 tags.

[Cancel](#) [Previous](#) **Request**

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

Click on "Request"

Screenshot of the AWS Certificate Manager '307fe2d8-4689-4cdd-ad9d-019da57208d9' certificate details page.

Successfully requested certificate with ID 307fe2d8-4689-4cdd-ad9d-019da57208d9
A certificate request with a status of pending validation has been created. Further action is needed to complete the validation and approval of the certificate.

View certificate

Certificate status

Identifier	Status
307fe2d8-4689-4cdd-ad9d-019da57208d9	Pending validation Info

ARN
[arn:aws:acm:us-east-1:524783324460:certificate/307fe2d8-4689-4cdd-ad9d-019da57208d9](#)

Type
Amazon Issued

Domains (0)

Domain	Status
Loading	

Details

In use	Serial number	Requested at	Renewal eligibility
No	N/A	August 06, 2025, 00:56:58 (UTC-04:00)	Ineligible
Domain name	Public key info	Issued at	Export option
-	Signature algorithm	N/A	Disabled
Number of additional names	Can be used with	Not before	
0		N/A	

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

Click on "View Certificate"

AWS Certificate Manager (ACM)

Certificate status

Identifier	307fe2d8-4689-4cdd-ad9d-019da57208d9	Status	Pending validation Info
ARN	arn:aws:acm:us-east-1:324783324460:certificate/307fe2d8-4689-4cdd-ad9d-019da57208d9	Type	Amazon Issued

Domains (1)

Domain	Status	Renewal status	Type	CNAME name
sosoebot.org	Pending validation	-	CNAME	_Oe55f88c845eef31d6cd057887aa5aa8.sosoebot.org.

Details

In use	Serial number	Requested at	Renewal eligibility
No	N/A	August 06, 2025, 00:56:58 (UTC-04:00)	Ineligible
Domain name	Public key info	Issued at	Export option
sosoebot.org	RSA 2048	N/A	Disabled
Number of additional names	Signature algorithm	Not before	
0	SHA-256 with RSA	N/A	
Can be used with	Not after		

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

Our certificate has been created but the status is “Pending”

Now we have to create records on Route 53

AWS Certificate Manager (ACM)

Certificate status

Identifier	307fe2d8-4689-4cdd-ad9d-019da57208d9	Status	Pending validation Info
ARN	arn:aws:acm:us-east-1:324783324460:certificate/307fe2d8-4689-4cdd-ad9d-019da57208d9	Type	Amazon Issued

Domains (1)

Domain	Status	Renewal status	Type	CNAME name
sosoebot.org	Pending validation	-	CNAME	_Oe55f88c845eef31d6cd057887aa5aa8.sosoebot.org.

Details

In use	Serial number	Requested at	Renewal eligibility
No	N/A	August 06, 2025, 00:56:58 (UTC-04:00)	Ineligible
Domain name	Public key info	Issued at	Export option
sosoebot.org	RSA 2048	N/A	Disabled
Number of additional names	Signature algorithm	Not before	
0	SHA-256 with RSA	N/A	
Can be used with	Not after		

[Create records in Route 53](#) Export to CSV

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

Now, click on “Create Records in Route 53”. This is automatically going to create some records in AWS Route 53.

Create DNS records in Amazon Route 53 (1/1)

Validation status = Pending validation | Validation status = Failed | Is domain in Route 53? = Yes

Domain: sosoebot.org | Validation status: Pending validation | Is domain in Route 53?: Yes

Create records

Click on “Create Records”

Successfully created DNS records
Successfully created DNS records in Amazon Route 53 for certificate with ID 307fe2d8-4689-4cdd-ad9d-019da57208d9.

307fe2d8-4689-4cdd-ad9d-019da57208d9

Certificate status

Identifier 307fe2d8-4689-4cdd-ad9d-019da57208d9	Status <input checked="" type="checkbox"/> Pending validation Info
ARN arn:aws:acm:us-east-1:324783324460:certificate/307fe2d8-4689-4cdd-ad9d-019da57208d9	Type Amazon Issued

Domains (1)

Domain	Status	Renewal status	Type	CNAME name
sosoebot.org	<input checked="" type="checkbox"/> Pending validation	-	CNAME	_0e55f88c845eeef31d6cd057887aa5aa8.sosoebot.org

Details

In use No	Serial number N/A	Requested at August 06, 2025, 00:56:58 (UTC-04:00)	Renewal eligibility Ineligible
Domain name sosoebot.org	Public key info RSA 2048	Issued at N/A	Export option Disabled
Number of additional names 0	Signature algorithm SHA-256	Not before N/A	

Head back to Route 53 and click on “Hosted Zones”

Route 53 < Hosted zones (1)

Automatic mode is the current search behavior optimized for best filter results. To change modes go to settings.

Hosted zone name	Type	Created by	Record count	Description	Hosted zone ID
sosoebot.org	Public	Route 53	4	-	Z06682941KBRW1BCA...

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

Click on our Hosted Zone name

Route 53 < Hosted zones > sosoebot.org

Public sosoebot.org Info Delete zone Test record Configure query logging

Hosted zone details Edit hosted zone

Records (4) DNSSEC signing Hosted zone tags (0)

Records (4) Info

Record name	Type	Routing p...	Differ...	Alias	Value/Route traffic to	TTL (s...)	Health ...	Evaluat...
sosoebot.org	A	Simple	-	Yes	dualstack.test-lb-205696798...	-	-	Yes
sosoebot.org	NS	Simple	-	No	ns-1363.awsdns-42.org. ns-612.awsdns-12.net. ns-22.awsdns-02.com. ns-1952.awsdns-52.co.uk.	172800	-	-
sosoebot.org	SOA	Simple	-	No	ns-1363.awsdns-42.org. aw...	900	-	-
_0e55f88c845eef31d6cd0578...	CNAME	Simple	-	No	_bccf75c8964d660b38b1bb...	300	-	-

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

You can see that we now have an additional record of type “**CNAME**”

Now, go back and check the status of the created certificate

The screenshot shows the AWS Certificate Manager (ACM) interface. A newly issued certificate for the domain `sosoebot.org` is displayed. The status is shown as "Issued". An orange arrow points to the "Status" section. Below it, the "Domains" section lists the single domain `sosoebot.org` with a status of "Success". The "Details" section provides comprehensive information about the certificate, including its serial number, public key info, and signature algorithm.

Now, the certificate has been issued. We now move to the next step,

Before we access the URL with HTTPS, we still need to work on the load balancer. We have to give access to HTTPS port. Go back to the load balancer.

The screenshot shows the AWS EC2 Load Balancers page. A load balancer named `test-lb` is selected, indicated by an orange arrow. The "Load balancers (1/1)" section shows the details of the selected load balancer. The "Details" tab is active, displaying information such as the load balancer type (Application), status (Active), VPC (vpc-0a178e5d2aecc4790), and availability zones (us-east-1a, us-east-1b). The "DNS name Info" section shows the DNS name `test-lb-2056967987.us-east-1.elb.amazonaws.com` (A Record).

Click on the Load Balancer name

The screenshot shows the AWS EC2 Load Balancers console for a load balancer named 'test-lb'. The 'Listeners and rules' tab is selected. A red arrow points to the 'HTTP:80' entry in the list of listeners. Another red arrow points to the 'Add listener' button at the top right of the table.

You can see that in the “Listener and rules”, the protocol port is only for HTTP, so we have to add HTTPS. Click on “Add Listener”

The screenshot shows the 'Add listener' configuration page for the 'test-lb' load balancer. The 'Protocol' dropdown is set to 'HTTP'. A red arrow points to this dropdown. The 'Port' dropdown is set to '81'. Another red arrow points to this dropdown.

Click on the drop down and select “HTTPS”

Add listener [Info](#)

Add a listener to your Application Load Balancer (ALB) to define how client requests and network traffic are routed within your application. Every listener is made up of a default action that's required and can only be edited. Additional rules can be added, edited and deleted from the listener.

Load balancer details: test-lb

Listener: HTTPS:443

A listener checks for connection requests using the protocol and port that you configure. The default action and any additional rules that you create determine how the Application Load Balancer routes requests to its registered targets.

Protocol Used for connections from clients to the load balancer. **Port** The port on which the load balancer is listening for connections.

HTTPS **443**
1-65535

Default action [Info](#)
The default action is used if no other rules apply. Choose the default action for traffic on this listener.

Authentication action - optional [Info](#)
Authentication requires IPv4 connectivity to authentication endpoints. [Learn more](#)

Authenticate users
Configure user authentication through either OpenID Connect (OIDC) or Amazon Cognito.

Routing action

Forward to target groups [Info](#)
Choose a target group and specify routing weight or [create target group](#)

Redirect to URL
 Return fixed response

Forward to target group [Info](#)
Choose a target group and specify routing weight or [create target group](#)

Target group

Select a target group	Weight	Percent
<input type="button" value="Select a target group"/>	1	100%
	0-999	

[+ Add target group](#)
You can add up to 4 more target groups.

Target group stickiness [Info](#)

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

Click on the drop down on Target group and select the target group we created, that is “test-tg”

Add listener [Info](#)

Add a listener to your Application Load Balancer (ALB) to define how client requests and network traffic are routed within your application. Every listener is made up of a default action that's required and can only be edited. Additional rules can be added, edited and deleted from the listener.

Load balancer details: test-lb

Listener: HTTPS:443

A listener checks for connection requests using the protocol and port that you configure. The default action and any additional rules that you create determine how the Application Load Balancer routes requests to its registered targets.

Protocol Used for connections from clients to the load balancer. **Port** The port on which the load balancer is listening for connections.

HTTPS **443**
1-65535

Default action [Info](#)
The default action is used if no other rules apply. Choose the default action for traffic on this listener.

Authentication action - optional [Info](#)
Authentication requires IPv4 connectivity to authentication endpoints. [Learn more](#)

Authenticate users
Configure user authentication through either OpenID Connect (OIDC) or Amazon Cognito.

Routing action

Forward to target groups [Info](#)
Choose a target group and specify routing weight or [create target group](#)

Redirect to URL
 Return fixed response

Forward to target group [Info](#)
Choose a target group and specify routing weight or [create target group](#)

Target group

test-tg	HTTP	Weight	Percent
Target type: Instance, IPv4 Target stickiness: Off	<input type="button" value="Select a target group"/>	1	100%
	0-999		

[+ Add target group](#)
You can add up to 4 more target groups.

Target group stickiness [Info](#)

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

Scroll down to “Secure listener settings”

Screenshot of the AWS EC2 Load Balancers "Add listener" configuration page.

Target group stickiness Info
 Enables the load balancer to bind a user's session to a specific target group. To use stickiness the client must support cookies. If you want to bind a user's session to a specific target, turn on the Target Group attribute Stickiness.
 Turn on target group stickiness

Secure listener settings Info

Security policy Info
 Your load balancer uses a Secure Socket Layer (SSL) negotiation configuration called a security policy to manage SSL connections with clients. [Compare security policies](#)

Security category

Default SSL/TLS server certificate
 The certificate used if a client connects without SNI protocol, or if there are no matching certificates. You can source this certificate from AWS Certificate Manager (ACM), Amazon Identity and Access Management (IAM), or import a certificate. This certificate will automatically be added to your listener certificate list.

Certificate source
 From ACM From IAM Import certificate

Certificate (from ACM)
 The selected certificate will be applied as the default SSL/TLS server certificate for this load balancer's secure listeners.
 
[Request new ACM certificate](#)

Client certificate handling Info
 Client certificates are used to make authenticated requests to remote servers. [Learn more](#)

Mutual authentication (mTLS)
 Mutual TLS (Transport Layer Security) authentication offers two-way peer authentication. It adds a layer of security over TLS and allows your services to verify the client that's making the connection.

 Default attributes are applied when adding a listener. You can edit them after adding the listener.

Listener tags - optional
 Tags can help you manage, identify, organize, search for and filter resources.

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

Click on the drop down and select our certificate for “**sosoebot.org**”

Screenshot of the AWS EC2 Load Balancers "Add listener" configuration page.

Security category

Default SSL/TLS server certificate
 The certificate used if a client connects without SNI protocol, or if there are no matching certificates. You can source this certificate from AWS Certificate Manager (ACM), Amazon Identity and Access Management (IAM), or import a certificate. This certificate will automatically be added to your listener certificate list.

Certificate source
 From ACM From IAM Import certificate

Certificate (from ACM)
 The selected certificate will be applied as the default SSL/TLS server certificate for this load balancer's secure listeners.
 
 307fe2db-4689-4cdd-ad9d-019da57208d9
[Request new ACM certificate](#)

Client certificate handling Info
 Client certificates are used to make authenticated requests to remote servers. [Learn more](#)

Mutual authentication (mTLS)
 Mutual TLS (Transport Layer Security) authentication offers two-way peer authentication. It adds a layer of security over TLS and allows your services to verify the client that's making the connection.

 Default attributes are applied when adding a listener. You can edit them after adding the listener.

Listener tags - optional
 Tags can help you manage, identify, organize, search for and filter resources.

Server-side tasks and status
 After completing and submitting the above steps, all server-side tasks and their statuses become available for monitoring.

[Cancel](#)  **Add listener**

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

Click on “**Add Listener**”

The screenshot shows the AWS EC2 Load Balancers console for a load balancer named "test-lb". A green success message at the top says "Successfully created listener." Below it, the "Details" section shows the load balancer is active, has an Internet-facing scheme, and is associated with a VPC and specific availability zones. The "Listeners and rules" tab is selected, displaying two listeners: one for HTTP:80 and one for HTTPS:443, each with a single rule forwarding traffic to a target group.

Scroll down

This screenshot shows the same "test-lb" load balancer configuration as above, but with the "Security" tab highlighted in blue. An orange arrow points to the "Security" tab in the navigation bar. The "Listeners and rules" section remains the same, showing the two configured listeners and their rules.

Now you can see two listeners and rules. Now, click on the “Security Tab”

EC2

test-lb

Details

Status: Active

VPC: vpc-0a178e5d2aecc4790

Load balancer IP address type: IPv4

Load balancer ARN: arn:aws:elasticloadbalancing:us-east-1:324783324460:loadbalancer/app/test-lb/e37466de6542

Availability Zones: subnet-04e5dcf10023b20f3 us-east-1a (use1-az6), subnet-013af1016c64ee796 us-east-1b (use1-az1)

DNS name info: test-lb-2056967987.us-east-1.elb.amazonaws.com (A Record)

Listeners and rules | **Network mapping** | **Resource map** | **Security** | **Monitoring** | **Integrations** | **Attributes** | **Capacity** | **Tags**

Security groups (2)

A security group is a set of firewall rules that control the traffic to your load balancer.

Security Group ID	Name	Description
sg-0eb32582090cab331	test-sg-for-lb	Allow ssh port 22 and 80
sg-0ffbd0ddfd008e8ede	default	default VPC security group

CloudShell **Feedback**

We have to add inbound rule for **HTTPS**. Click on the security group we created

EC2

sg-0eb32582090cab331 - test-sg-for-lb

Details

Security group name: test-sg-for-lb

Owner: 524783324460

Security group ID: sg-0eb32582090cab331

Description: Allow ssh port 22 and 80

Inbound rules count: 2 Permission entries

Outbound rules count: 1 Permission entry

VPC ID: vpc-0a178e5d2aecc4790

Inbound rules | **Outbound rules** | **Sharing - new** | **VPC associations - new** | **Tags**

Inbound rules (2)

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
-	sgr-03de90b7b26862f6f	IPv4	HTTP	TCP	80	0.0.0.0/0
-	sgr-0b7ae0aeb243c754a	IPv4	SSH	TCP	22	0.0.0.0/0

Actions

CloudShell **Feedback**

Click on “**Edit Inbound Rules**”

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-03de90b7b26862f6f	HTTP	TCP	80	Custom	0.0.0.0/0
sgr-0b7ae0aeb243c754a	SSH	TCP	22	Custom	0.0.0.0/0

Add rule

Cancel Preview changes Save rules

Click on “Add Rule”

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-03de90b7b26862f6f	HTTP	TCP	80	Custom	0.0.0.0/0
sgr-0b7ae0aeb243c754a	SSH	TCP	22	Custom	0.0.0.0/0
-	Custom TCP	TCP	0	Custom	0.0.0.0/0

Add rule

Cancel Preview changes Save rules

Click on the drop down and select “HTTPS”

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-03de90b7b26862f6f	HTTP	TCP	80	Custom	0.0.0.0/0
sgr-0b7ae0aeb243c754a	SSH	TCP	22	Custom	0.0.0.0/0
-	HTTPS	TCP	443	Custom	0.0.0.0/0

Add rule

Cancel Preview changes Save rules

Click on the drop down on “Source” and select “Anywhere-IPv4”

Inbound rules

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-03de90b7b26862f6f	HTTP	TCP	80	Custom	0.0.0.0/0
sgr-0b7ae0aeb243c754a	SSH	TCP	22	Custom	0.0.0.0/0
-	HTTPS	TCP	443	Anywhere	0.0.0.0/0

Add rule

Save rules

Click on “Save Rules”

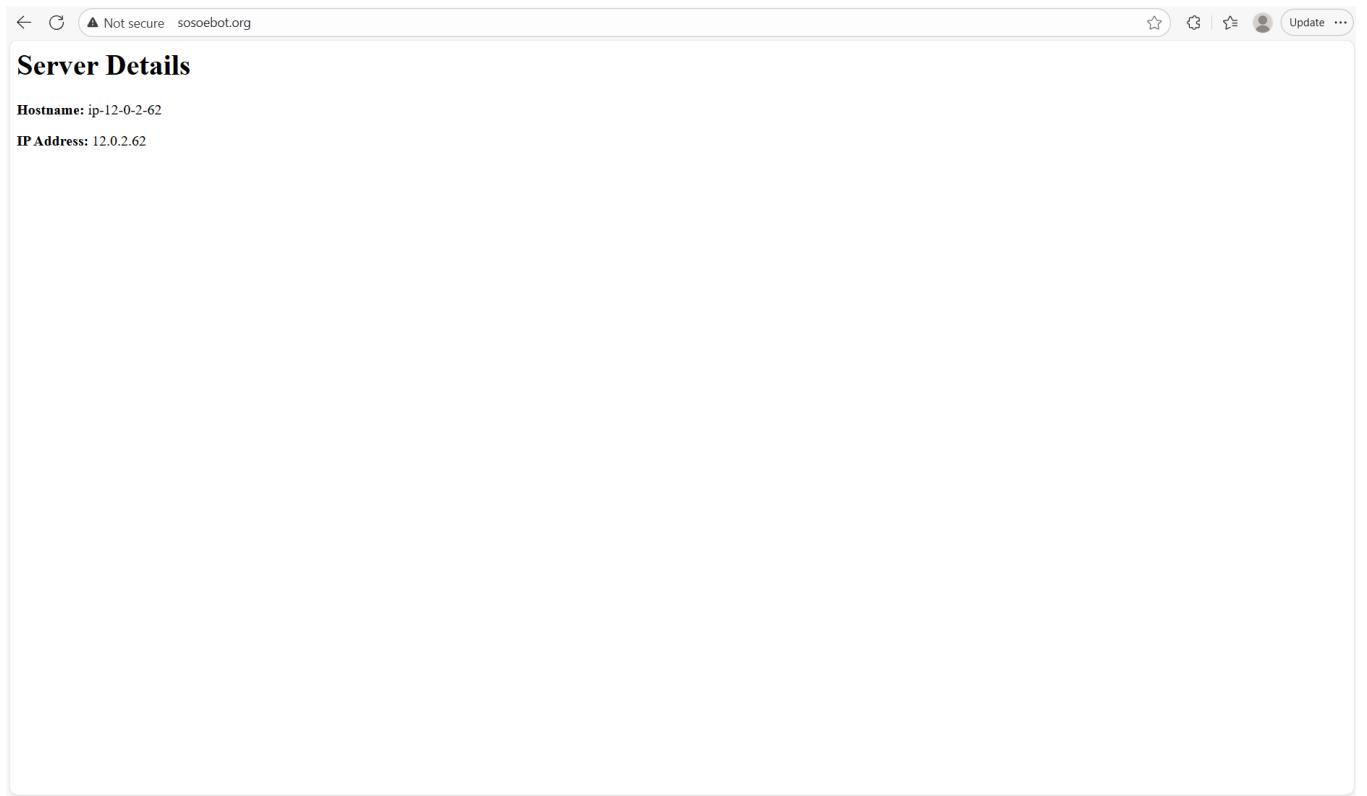
Details

Security group name	Security group ID	Description	VPC ID
test-sg-for-lb	sg-0eb32582090cab331	Allow ssh port 22 and 80	vpc-0a178e5d2aecc4790
Owner	Inbound rules count	Outbound rules count	
324783324460	3 Permission entries	1 Permission entry	

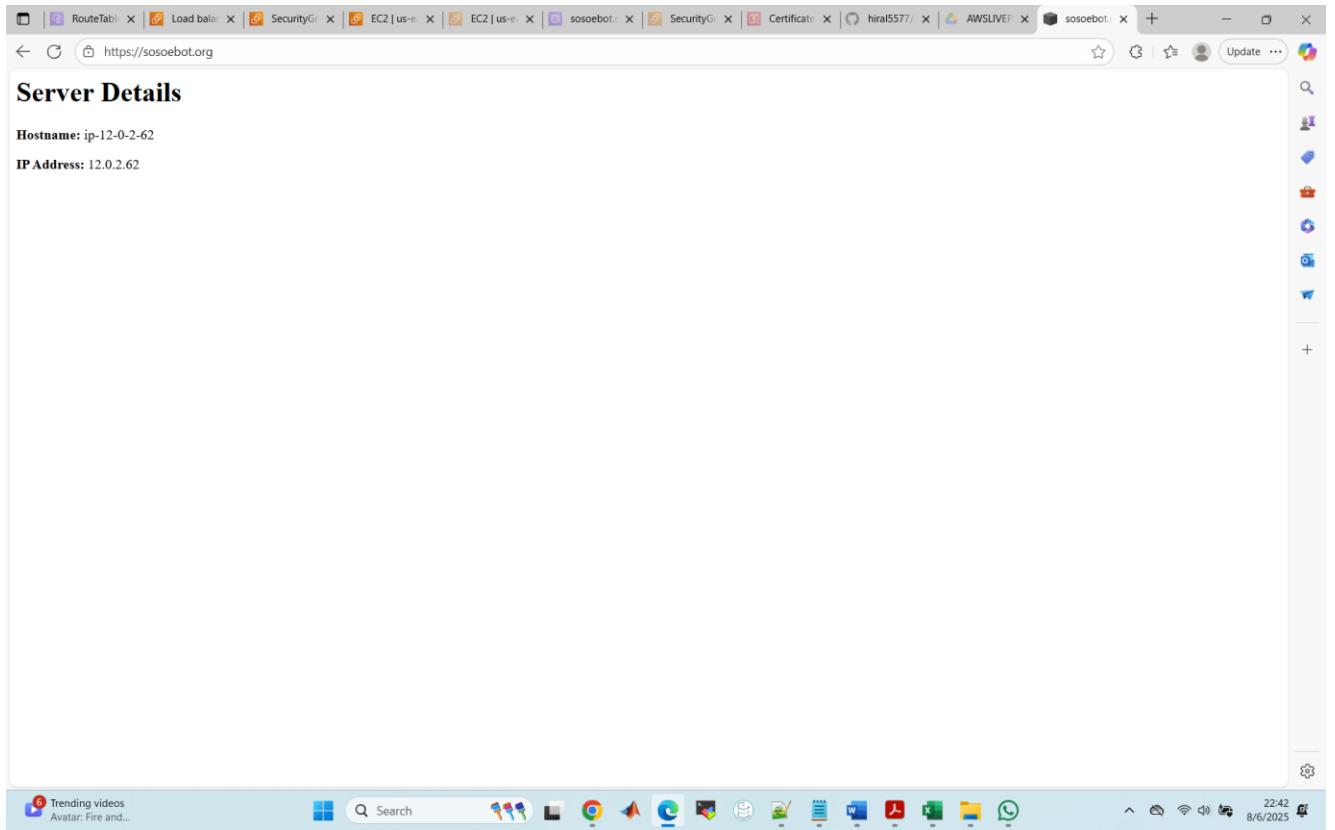
Inbound rules

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
-	sgr-03de90b7b26862f6f	IPv4	HTTP	TCP	80	0.0.0.0/0
-	sgr-0b7ae0aeb243c754a	IPv4	SSH	TCP	22	0.0.0.0/0
-	sgr-034d0eecd67b6814e5	IPv4	HTTPS	TCP	443	0.0.0.0/0

Let us first verify by using <http://sosoebot.org> on our browser



Now, let us verify by using <https://sosoebot.org> on our browser



It works. We are able to use HTTPS inside our URL. Our both URL patterns are working but the thing which we need to notice here is we need to stop the HTTP request or at least we need to redirect this particular request which is coming without HTTPS to our HTTPS part.

The next thing we are going to fix is to redirect this particular request where we don't put HTTPS and we should redirect this request to land on this HTTP URL.

How do we fix the redirection issue?

Go to your EC2 dashboard

The screenshot shows the AWS EC2 Dashboard. On the left sidebar, under the 'Load Balancing' section, the 'Load Balancers' item is highlighted with a red arrow pointing to it. The main content area displays various EC2 resources and metrics. The 'Resources' section shows counts for Instances (running), Auto Scaling Groups, Capacity Reservations, Dedicated Hosts, Elastic IPs, Instances, Key pairs, Load balancers, Placement groups, Security groups, Snapshots, and Volumes. The 'Launch instance' section includes a 'Launch instance' button and a note about launching in the N. Virginia Region. The 'Service health' section shows the AWS Health Dashboard and indicates that the service is operating normally. The 'Zones' section lists availability zones: us-east-1a, us-east-1b, us-east-1c, us-east-1d, and us-east-1e, each associated with a specific zone ID. The 'Account attributes' section shows the Default VPC (vpc-0128e9209eaef1c37) and other settings like Data protection and security, Allowed AMIs, Zones, EC2 Serial Console, and Default credit specification. The bottom of the page includes standard AWS footer links for CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

Click on "Load Balancer"

The screenshot shows the AWS EC2 Load Balancers page. On the left, there's a navigation sidebar with various services like Launch Templates, Spot Requests, and Load Balancing. The 'Load Balancing' section is expanded, showing 'Load Balancers' (1). The main pane displays a table titled 'Load balancers (1)'. The table has columns for Name, State, Type, Scheme, IP address type, VPC ID, Availability Zones, and Security. A single row is selected, showing 'test-lb' in the Name column, which is highlighted with a blue arrow. The table also shows 'Active' status, 'application' type, 'Internet-facing' scheme, 'IPv4' IP address type, 'vpc-0a178e5d2aecc479...' VPC ID, '2 Availability Zones', and 'sg-0eb52...' Security Group.

Click on the Load Balancer name, that is “**test-lb**”

The screenshot shows the details page for the 'test-lb' load balancer. The left sidebar is identical to the previous screenshot. The main content area shows the 'Details' tab for 'test-lb'. It includes fields for Load balancer type (Application), Status (Active), Scheme (Internet-facing), VPC (vpc-0a178e5d2aecc479...), Availability Zones (us-east-1a, us-east-1b), Load balancer ARN (arn:aws:elasticloadbalancing:us-east-1:324783324460:loadbalancer/app/test-lb/6d5cef74184aabb), and DNS name (test-lb-1360045039.us-east-1.elb.amazonaws.com). Below this, the 'Listeners and rules' tab is selected, showing two listeners: 'HTTPS:443' and 'HTTP:80'. Each listener has a 'Forward to target group' rule with a target group named 'test-tg'.

Go to “Listener and Rules”

Details

Load balancer ARN: arn:aws:elasticloadbalancing:us-east-1:324783324460:loadbalancer/app/test-lb/6d5cef74184a4aab

DNS name: test-lb-1360045039.us-east-1.elb.amazonaws.com (A Record)

Listeners and rules (2) Info

Protocol:Port	Default action	Rules	ARN	Security policy	Default SSL/TLS certificate
HTTPS:443	Forward to target group test-tg [1] (100%) Target group stickiness: Off	1 rule	ARN	ELBSecurityPolicy-TLS13-1-2...	sosobot.org (Certificate ID: 3f...
HTTP:80	Forward to target group test-tg [1] (100%) Target group stickiness: Off	1 rule	ARN	Not applicable	Not applicable

Now, we are interested into the request only for HTTP. We want to let go that request and redirect it to the HTTPS. Click on “HTTP”

HTTP:80 Info

Details

Protocol:Port: HTTP:80

Load balancer: test-lb

Default actions:

- Forward to target group
test-tg [1] (100%)
Target group stickiness: Off

Listener ARN: arn:aws:elasticloadbalancing:us-east-1:324783324460:listener/app/test-lb/6d5cef74184a4aab/bb39bb03a714256

Listener rules (1) Info

Priority	Name tag	Conditions (If)	Actions (Then)	ARN	Tags	Actions
Last (default)	Default	If no other rule applies	Forward to target group test-tg [1] (100%) Target group stickiness: Off	ARN	0 tags	

Click on “Default” rule

The screenshot shows the AWS CloudFront console with the path: EC2 > Load balancers > test-lb > HTTP:80 listener > Default rule. The left sidebar lists various AWS services. The main panel displays the 'Default rule' configuration. The 'Actions' button in the top right corner is highlighted with an orange arrow.

Then click on the drop down on “Action”

The screenshot shows the AWS CloudFront console with the same path and configuration as the previous screenshot. The 'Actions' dropdown menu is open, and the 'Edit rule' option is highlighted with an orange arrow.

Select “Edit Rule”

Screenshot of the AWS CloudFront console showing the "Edit listener" page for a load balancer named "test-lb".

Listener details:

- Protocol:** HTTP
- Port:** 80

Default action: Forward to target groups

Target group: test-tg (HTTP, weight 1, percent 100%)

Default Actions: Redirect to URL

Scroll down to “Default Actions”

Screenshot of the AWS CloudFront console showing the "Edit listener" page for a load balancer named "test-lb".

Listener ARN: arn:aws:elasticloadbalancing:us-east-1:324783324460:listener/app/test-lb/6d5cef74184a4aab/bb39bb03a714256

Protocol: HTTP

Port: 80

Default action: Redirect to URL

Target group: test-tg (HTTP, weight 1, percent 100%)

Default Actions: Redirect to URL

Select “Redirect to URL”

Screenshot of the AWS CloudFront Listener configuration page for an HTTP listener on port 80.

Protocol: HTTP

Port: 80

Default action: Info

Routing action: Redirect to URL

Redirect to URL: Info

URI parts: Full URL

Protocol: HTTPS

Port: 443

Custom host, path, query: Unchecked

Status code: 301 - Permanently moved

Server-side tasks and status:

After completing and submitting the above steps, all server-side tasks and their statuses become available for monitoring.

Save changes button highlighted.

On PORT enter “443” and make sure “Custom host, path, query” is unchecked.

Screenshot of the AWS CloudFront Listener configuration page for an HTTPS listener on port 443.

Protocol: HTTPS

Port: 443

Default action: Info

Routing action: Redirect to URL

Redirect to URL: Info

URI parts: Full URL

Protocol: HTTPS

Port: 443

Custom host, path, query: Unchecked

Status code: 301 - Permanently moved

Server-side tasks and status:

After completing and submitting the above steps, all server-side tasks and their statuses become available for monitoring.

Save changes button highlighted.

Click on “Save Changes”

The screenshot shows the AWS CloudFront console. In the left sidebar, under the 'Load Balancing' section, 'Load Balancers' is selected. The main area displays a 'HTTP:80' listener configuration for a load balancer named 'test-lb'. A green success message at the top states 'Successfully modified listener.' An orange arrow points from this message to the 'Actions' button in the top right corner of the main panel. The 'Rules' tab is active, showing one rule: 'Last (default) Default If no other rule applies' with the action 'Redirect to HTTPS://#[host]:443/#[path]#[query] Status code: HTTP_301'. The ARN of the listener is listed as 'arn:aws:elasticloadbalancing:us-east-1:324783324460:listener/app/test-lb/6d5cef74184a4aab/bb39bb03a714256'.

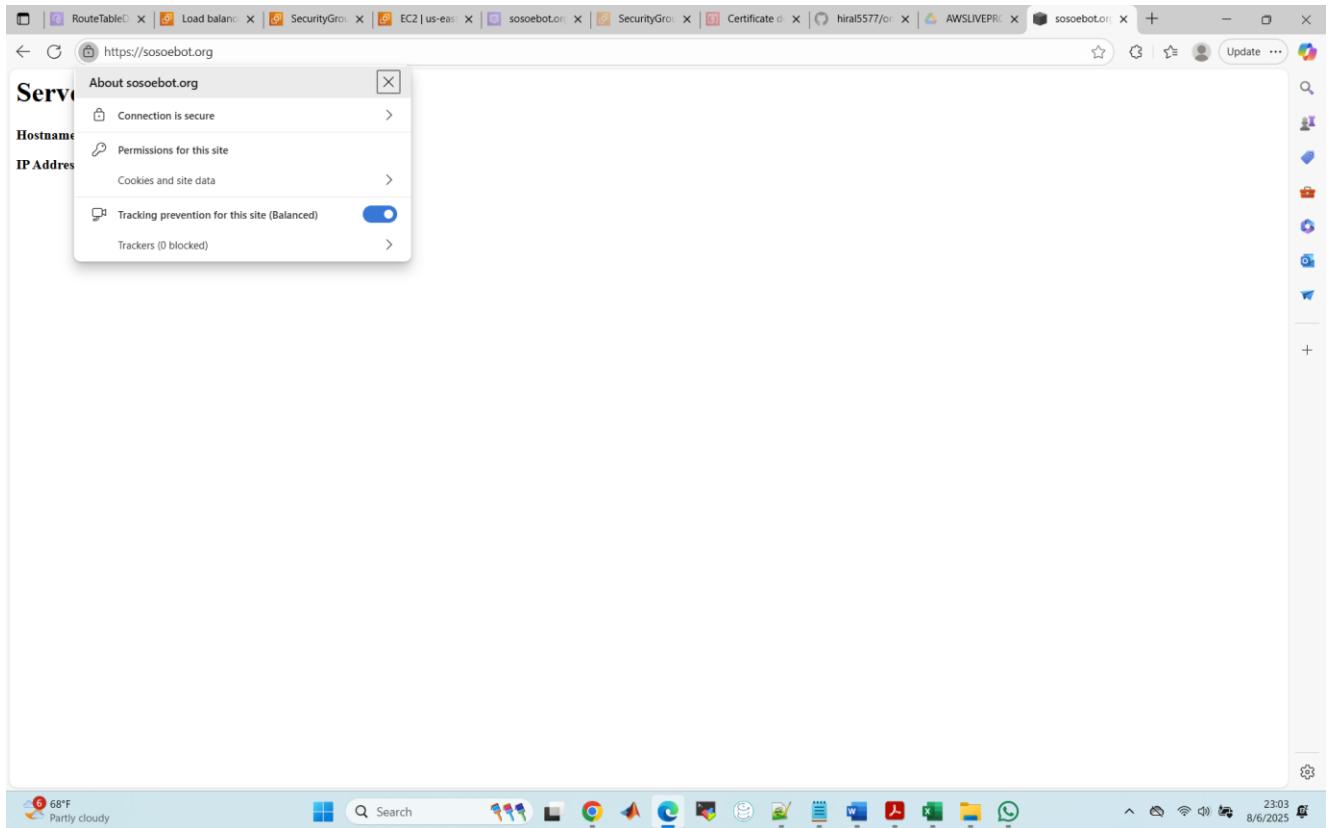
You can see “successfully modified the listener”. You can wait for a couple of minutes and then it will start redirecting the normal HTTP request to the HTTPS.

Let us test it now. We will enter <http://sosoebot.org> on our browser and hit Enter. It will redirect us to <https://sosoebot.org>.

The screenshot shows a web browser window with multiple tabs open. The address bar shows 'https://sosoebot.org'. An orange arrow points from the browser's address bar to the URL. The main content area displays 'Server Details' with the following information: Hostname: ip-12-0-2-62 and IP Address: 12.0.2.62. The browser interface includes a search bar, a toolbar with various icons, and a status bar at the bottom showing weather information (68°F Partly cloudy), system icons, and the date/time (23:01 8/6/2025).

You can see it has redirected me to <https://sosoebot.org>

You can verify



You can see that the connection is secured.

So, this is how you are going to set up your SSL and you are going to redirect the normal http request to https.

In case you are using some other domain, let us say for example www.sosoebot.org. Then this is a different name, then you need to request a certificate for this particular domain name as well.

So, you need to create a new certificate and there you need to enter the root name as www.sosoebot.org to request a new certificate for this domain name. Then push the records again to our Route 53, after that you will be able to access the URL or home page using the www URL pattern as well.

I give credit to Rahul Wagn for using images from his tutorial to put up this document

