

1. Create one VPC in N. Virginia region.

Go to n virginia region and select vpc create vpc.

The screenshot shows the 'Create VPC' wizard in the AWS VPC service. The 'VPC settings' section is active. Under 'Resources to create', 'VPC only' is selected. A 'Name tag - optional' field contains 'my-vpc'. The 'IPv4 CIDR block' is set to '10.0.0.0/16'. Under 'IPv6 CIDR block', 'No IPv6 CIDR block' is selected. The 'IPv4 CIDR' and 'IPv6 CIDR block' sections have detailed descriptions and links to 'Info'.

The screenshot shows the 'Details' page for the VPC 'my-vpc'. The VPC ID is 'vpc-0c51898d0fd83685e'. The 'State' is 'Available'. 'Block Public Access' is set to 'Off'. 'DNS hostnames' is 'Disabled'. 'Main route table' is 'rtb-0167a738f71797291'. 'IPv6 pool' is '-' and 'Owner ID' is '235351028455'. The 'Resource map' section shows 'VPC' (Your AWS virtual network), 'Subnets (2)' (Subnets within this VPC), 'Route tables (3)' (Route network traffic to resources), and 'Network Connect' (Connections to other).

2. Create two subnets: one public subnet and one private subnet.

Go to subnets and create subnet and select vpc and give the cidr range for subnet.

Screenshot of the AWS VPC Subnets creation page:

VPC ID: vpc-0c51898d0fd83685e (my-vpc)

Associated VPC CIDRs:

- IPv4 CIDRs:** 10.0.0.0/16

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

Subnet 1 of 1:

- Subnet name:** pub-subnet (Create a tag with a key of 'Name' and a value that you specify.)
- Availability Zone:** No preference (Choose the zone in which your subnet will reside, or let Amazon choose one for you.)

CloudShell Feedback © 2025, Amazon Web Services

Create public subnet and select created vpc and give subnet name and cidr range .

Screenshot of the AWS VPC Subnets list page:

Subnets (1/2) Info:

Name	Subnet ID	State	VPC	Action
pri-subnet	subnet-0e32deb8085ee322d	Available	vpc-0c51898d0fd83685e my...	Block Public Access: Off
pub-subnet	subnet-04e9b2feb7e22f4b3	Available	vpc-0c51898d0fd83685e my...	Block Public Access: Off

subnet-04e9b2feb7e22f4b3 / pub-subnet:

- Details:**
 - Subnet ID: subnet-04e9b2feb7e22f4b3
 - Subnet ARN: arn:aws:ec2:us-east-1:23535102845:subnet/subnet-04e9b2feb7e22f4b3
 - IPv4 CIDR: 10.0.0.0/20
 - Available IPv4 addresses: 4091
 - Availability Zone: us-east-1a (us-east-1a)
 - Network border group: vpc-0c51898d0fd83685e | my-vpc
- Actions:**
 - Block Public Access: Off
 - IPv6 CIDR association ID: -
 - Route table: rtb-0ae8874bb25115589 | public

The screenshot shows the AWS VPC Subnets page. On the left, there's a navigation sidebar with 'VPC dashboard' and 'Virtual private cloud' sections. The main area displays a table titled 'Subnets (1/2)'. It has columns for Name, Subnet ID, State, VPC, and Block Public Access. Two subnets are listed: 'pri-subnet' (subnet-0e32deb8085ee322d, Available, vpc-0c51898d0fd83685e, Off) and 'pub-subnet' (subnet-04e9b2feb7e22f4b3, Available, vpc-0c51898d0fd83685e, Off). Below the table, a specific subnet ('pri-subnet') is selected, showing its details: Subnet ID (subnet-0e32deb8085ee322d), Subnet ARN (arn:aws:ec2:us-east-1:23535102845:subnet/subnet-0e32deb8085ee322d), IPv4 CIDR (10.0.16.0/20), Available IPv4 addresses (4091), State (Available), and Block Public Access (Off).

3. Attach an IGW to the VPC.

Go to internet gateways and create internet gateway and attach to vpc.

The screenshot shows the 'Create internet gateway' page. The top navigation bar includes 'VPC > Internet gateways > Create internet gateway'. The main section is titled 'Create internet gateway' with a 'Name tag' input field containing 'my-internet-gateway'. Below it is a 'Tags - optional' section with a key-value pair 'Name' and 'my-internet-gateway'. At the bottom right are 'Cancel' and 'Create internet gate' buttons.

The screenshot shows the AWS VPC Internet Gateways page. The top navigation bar includes 'VPC > Internet gateways > igw-061e0cce444b0dc00'. The main area displays a table with one row for 'igw-061e0cce444b0dc00 / my-vpc-gateway'. The 'Details' section shows the Internet gateway ID (igw-061e0cce444b0dc00), State (Attached), VPC ID (vpc-0c51898d0fd83685e | my-vpc), and Owner (235351028455). The 'Tags' section shows a single tag 'Name: my-vpc-gateway'. Navigation buttons at the bottom indicate this is page 1 of 1.

4. Create one public route table (RT) and one private route table.

Go to route table and create route table and give name and select vpc.

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
You can add 49 more tags.

VPC > Route tables > rtb-0ae8874bb25115589

rtb-0ae8874bb25115589 / public-routetable

Details Info

Route table ID rtb-0ae8874bb25115589	Main <input checked="" type="checkbox"/> No	Explicit subnet associations subnet-04e9b2feb7e22f4b3 / pub-subnet	Edge associations -
VPC vpc-0c51898d0fd83685e my-vpc	Owner ID <input checked="" type="checkbox"/> 235351028455		

Routes Subnet associations Edge associations Route propagation Tags

Routes (2)

Destination	Target	Status	Propagated	Route Origin
0.0.0.0/0	igw-061e0cce444b0dc00	Active	No	Create Route
10.0.0.0/16	local	Active	No	Create Route Tab

Create private route table and give name and select vpc.

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.
vpc-0c51898d0fd83685e (my-vpc)

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**
Q Name X pri-routetable X Remove

Add new tag
You can add 49 more tags.

Create route table

rtb-0b41bd3f28020f417 / private-routetable

Details		Explicit subnet associations		Edge associations																				
Route table ID	rtb-0b41bd3f28020f417	Main	No	-																				
VPC	vpc-0c51898d0fd83685e my-vpc	Owner ID	235351028455																					
Routes (1)																								
<table border="1"> <thead> <tr> <th colspan="2">Both</th> <th colspan="2">Edit routes</th> </tr> <tr> <th>Destination</th> <th>Target</th> <th>Status</th> <th>Propagated</th> </tr> </thead> <tbody> <tr> <td>10.0.0.0/16</td> <td>local</td> <td>Active</td> <td>No</td> </tr> <tr> <td colspan="2"></td> <td colspan="2">Route Origin</td> </tr> <tr> <td colspan="2"></td> <td colspan="2">Create Route Table</td> </tr> </tbody> </table>					Both		Edit routes		Destination	Target	Status	Propagated	10.0.0.0/16	local	Active	No			Route Origin				Create Route Table	
Both		Edit routes																						
Destination	Target	Status	Propagated																					
10.0.0.0/16	local	Active	No																					
		Route Origin																						
		Create Route Table																						

5. Deploy a NAT gateway in the public subnet and attach the NAT gateway to the private subnet.

Go to nat gateways and create nat gateway and select public subnet and allocate elastic ip.

⌚ Elastic IP address 52.206.43.157 (eipalloc-0ea1ec4217bb98d45) allocated.

Create NAT gateway Info

A highly available, managed Network Address Translation (NAT) service that instances in private subnets can use to connect to services in other VPCs, on-premises networks, or the internet.

NAT gateway settings

Name - optional

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

my-nat-gateway

The name can be up to 256 characters long.

Subnet

Select a subnet in which to create the NAT gateway.

subnet-04e9b2feb7e22f4b3 (pub-subnet)

Connectivity type

Select a connectivity type for the NAT gateway.

- Public
- Private

Elastic IP allocation ID Info

Assign an Elastic IP address to the NAT gateway.

eipalloc-0ea1ec4217bb98d45

[Allocate Elastic IP](#)

Go to route tables and go to private route table and edit routes and add nat gateway.

Destination	Target	Status	Propagated	Route Origin
10.0.0.0/16	local	Active	No	CreateRouteTable
Q. 0.0.0.0/0	NAT Gateway	-	No	CreateRoute
	Q. nat-0c067bf986b063fb3			Remove
	Use: "nat-0c067bf986b063fb3"			
	nat-0c067bf986b063fb3 (my-nat-gateway)			

6.Create two instances, one in the public subnet and one in the private subnet.

Go to instance and create instance give instance name and give the key pair

The screenshot shows the 'Launch an instance' wizard on the AWS EC2 service. The current step is 'Name and tags'. A message at the top says, 'It seems like you may be new to launching instances in EC2. Take a walkthrough to learn about EC2, how to launch instances and about best practices'. Below this, there are two buttons: 'Take a walkthrough' and 'Do not show me this message again.' The 'Name and tags' section has a 'Name' field containing 'public-instance' and a 'Add additional tags' button. To the right, a 'Summary' panel shows: Number of instances (1), Software Image (AMI) (Amazon Linux 2023 AMI 2023.9.2...), Virtual server type (instance type) (t3.micro), Firewall (security group) (New security group), and Storage (volumes) (1 volume(s) - 8 GiB). A 'Cancel' button is at the bottom right.

In network settings select your vpc and select public subnet and select existing security group or create new security group.

The screenshot shows the 'Launch an instance' wizard on the AWS EC2 service. The current step is 'Network settings'. It shows a dropdown for 'VPC' set to 'red', a 'Create new key pair' button, and a 'Network settings' section. Under 'VPC - required', it lists 'vpc-0c51898d0fd83685e (my-vpc)' with IP range '10.0.0.0/16'. Under 'Subnet', it shows 'subnet-04e9b2feb7e22f4b3' with details: VPC: vpc-0c51898d0fd83685e, Owner: 235351028455, Availability Zone: us-east-1a (use1-az2), Zone type: Availability Zone, IP addresses available: 4090, CIDR: 10.0.0.0/20. It also includes 'Auto-assign public IP' (set to 'Enable'), 'Firewall (security groups)' (with options to 'Create security group' or 'Select existing security group'), and a 'Security group name - required' field containing 'launch-wizard-3'. A note at the bottom states: '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:'. To the right, a 'Summary' panel shows: Number of instances (1), Software Image (AMI) (Amazon Linux 2023 AMI 2023.9.2...), Virtual server type (instance type) (t3.micro), Firewall (security group) (New security group), and Storage (volumes) (1 volume(s) - 8 GiB). A 'Cancel' button is at the bottom right.

Repeat the same process for private instance. In network settings give the private subnet.

The screenshot shows the 'Launch an instance' wizard step 4. It includes fields for a key pair name ('red'), VPC ('vpc-0c51898d0fd83685e (my-vpc)'), subnet ('subnet-0e32deb8085ee322d'), and security group ('Create security group'). The 'Auto-assign public IP' dropdown is set to 'Disable'. The 'Network settings' section is expanded, showing the selected VPC and subnet details.

The screenshot shows the EC2 Instances page with two instances listed: 'public-instance' (running, t3.micro) and 'private-instance' (running, t3.micro). Both instances are selected. The left sidebar shows the 'Instances' section is active.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
public-instance	i-0a3cf5850b7332622	Running	t3.micro	3/3 checks passed	View alarms +	us-east-1a	-
private-instance	i-0095bbdc6ca4e921e	Running	t3.micro	Initializing	View alarms +	us-east-1a	-

7. Deploy Apache server on both EC2 instances with a sample index.html file.

Connect to public instance.

```
MUJU SK@DESKTOP-LU541U4 MINGW64 ~/Downloads
$ ssh -i red.pem ec2-user@54.165.102.79
The authenticity of host '54.165.102.79 (54.165.102.79)' can't be established.
ED25519 key fingerprint is SHA256:N0Dh1opfGgj8tAg0iSUGknZpoGvXX92qeIcTIuNlo4U.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '54.165.102.79' (ED25519) to the list of known hosts.

          _#
         ~\_\_####_      Amazon Linux 2023
        ~~ \_\#####\
        ~~   \###|
        ~~     #/ ,__-
        ~~     V~ ,--> https://aws.amazon.com/linux/amazon-linux-2023
        ~~
        ~~ .-.
        ~~ / ,--/
        _/m/ ,--/ 

[ec2-user@ip-10-0-7-172 ~]$ sudo su -
[root@ip-10-0-7-172 ~]# |
```

Sudo yum update -y

Sudo yum install httpd -y

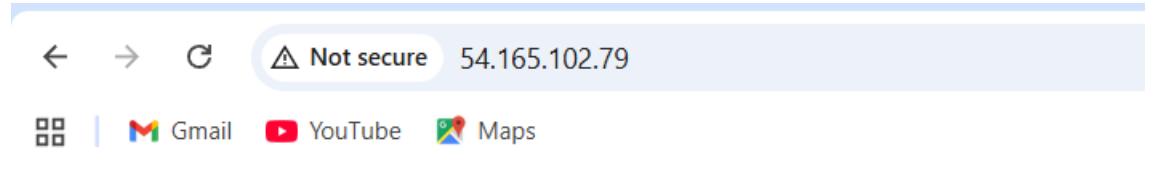
```
[root@ip-10-0-7-172 ~]# sudo yum update -y
Amazon Linux 2023 Kernel Livepatch repository
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-10-0-7-172 ~]# yum install httpd -y
Last metadata expiration check: 0:00:29 ago on Wed Oct  8 15:14:19 2025.
Dependencies resolved.
=====
 Package           Architecture      Version
=====
Installing:
 httpd            x86_64          2.4.65-1.amzn2023.
Installing dependencies:
 apr              x86_64          1.7.5-1.amzn2023.0
 apr-util         x86_64          1.6.3-1.amzn2023.0
 generic-logos-httpd noarch          18.0.0-12.amzn2023
```

Sudo systemctl start httpd

Sudo systemctl start httpd

```
[root@ip-10-0-7-172 ~]# sudo systemctl start httpd
[root@ip-10-0-7-172 ~]# sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
   Active: active (running) since wed 2025-10-08 15:15:29 UTC; 58s ago
     Docs: man:httpd.service(8)
 Main PID: 27057 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec: 0 B/s"
   Tasks: 177 (limit: 1053)
  Memory: 13.3M
    CPU: 106ms
   CGroup: /system.slice/httpd.service
           └─27057 /usr/sbin/httpd -DFOREGROUND
               ├─27084 /usr/sbin/httpd -DFOREGROUND
               ├─27088 /usr/sbin/httpd -DFOREGROUND
               ├─27089 /usr/sbin/httpd -DFOREGROUND
               └─27091 /usr/sbin/httpd -DFOREGROUND
```

Open with public ip in your browser.



It works!

Login with the public ip from that login with the private instance for that there is no public ip so,we use public instance ip as bastion server.

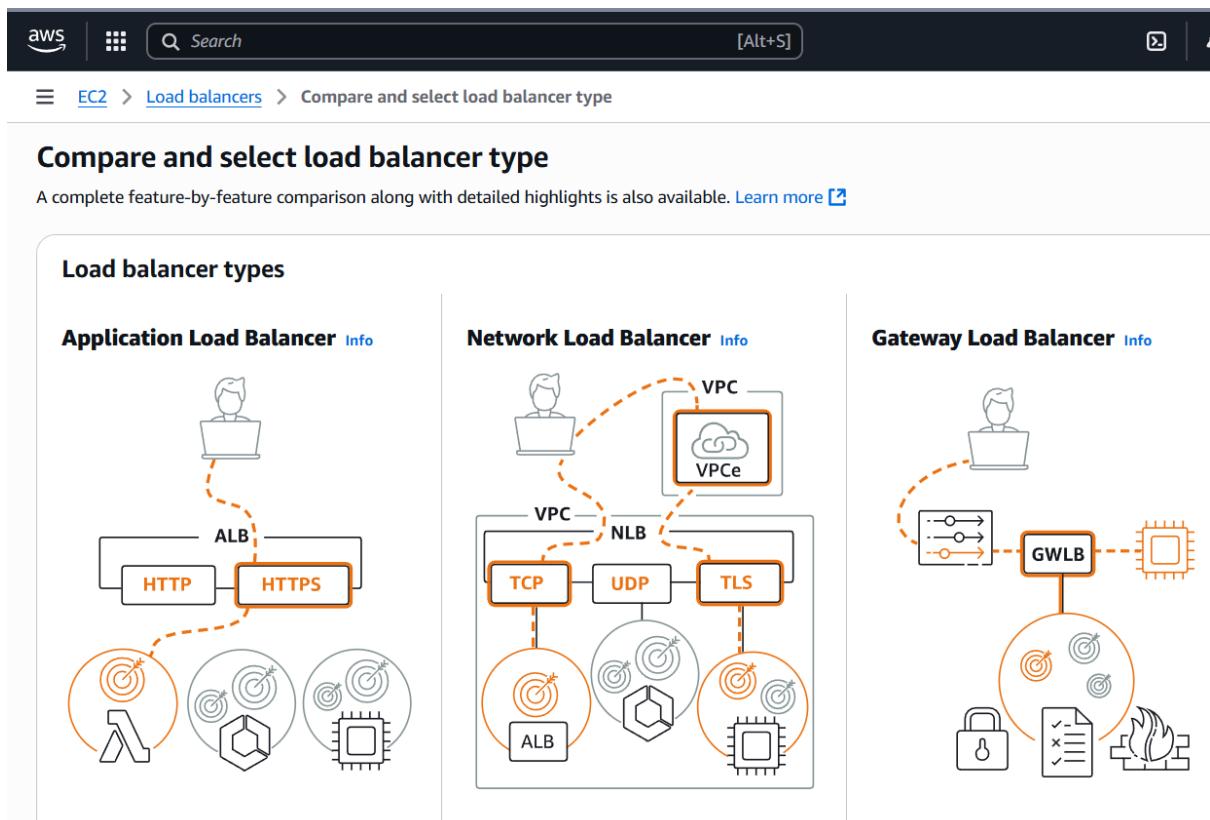
```
MUJU SK@DESKTOP-LU541U4 MINGW64 ~/Downloads
$ ssh -i red.pem ec2-user@54.165.102.79
,      #
~\_\ #####
~~ \#####`      Amazon Linux 2023
~~ \###|
~~   \#/ ,__-> https://aws.amazon.com/linux/amazon-linux-2023
~~
~~ .-' / \
~~ /_ / \
~/m/`|_
Last login: wed Oct  8 15:13:39 2025 from 103.143.169.218
[ec2-user@ip-10-0-7-172 ~]$ sudo su -
Last login: wed Oct  8 15:13:48 UTC 2025 on pts/1
[root@ip-10-0-7-172 ~]# ssh -i red.pem ec2-user@10.0.20.86
Warning: Identity file red.pem not accessible: No such file or directory.
ssh: Could not resolve hostname ec-2: Name or service not known
[root@ip-10-0-7-172 ~]# ssh -i red.pem ec2-user@10.0.20.86
Warning: Identity file red.pem not accessible: No such file or directory.
The authenticity of host '10.0.20.86 (10.0.20.86)' can't be established.
ED25519 key fingerprint is SHA256:eKoLd6V1vzbwyE306wscXsVO27Af9DSFa0ZGy9Szc1E.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.20.86' (ED25519) to the list of known hosts.
ec2-user@10.0.20.86: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[root@ip-10-0-7-172 ~]# |
```

```
[root@ip-10-0-7-172 ~]# yum install httpd -y
Last metadata expiration check: 0:22:01 ago on Wed Oct  8 15:14:19 2025.
Package httpd-2.4.65-1.amzn2023.0.1.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-10-0-7-172 ~]# sudo systemctl start httpd
[root@ip-10-0-7-172 ~]# sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: Loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
   Active: active (running) since Wed 2025-10-08 15:15:29 UTC; 21min ago
     Docs: man:httpd.service(8)
 Main PID: 27057 (httpd)
    Status: "Total requests: 2; Idle/Busy workers 100/0;Requests/sec: 0.00156; Bytes served/sec: 1 B/sec"
      Tasks: 230 (limit: 1053)
     Memory: 16.6M
        CPU: 1.567s
       CGroup: /system.slice/httpd.service
           ├─27057 /usr/sbin/httpd -DFOREGROUND
           ├─27084 /usr/sbin/httpd -DFOREGROUND
           ├─27088 /usr/sbin/httpd -DFOREGROUND
           ├─27089 /usr/sbin/httpd -DFOREGROUND
           ├─27091 /usr/sbin/httpd -DFOREGROUND
           └─27507 /usr/sbin/httpd -DFOREGROUND

Oct 08 15:15:29 ip-10-0-7-172.ec2.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Oct 08 15:15:29 ip-10-0-7-172.ec2.internal systemd[1]: Started httpd.service - The Apache HTTP server.
Oct 08 15:15:29 ip-10-0-7-172.ec2.internal httpd[27057]: Server configured, listening on: port 80
[root@ip-10-0-7-172 ~]#
```

8.Create one application load balancer and attach it to both EC2 instances.

Go to load balancers and select application load balancer and create load balancer.



Give the name and select vpc,select internet facing and select ipv4.

The screenshot shows the 'Create Application Load Balancer' wizard. In the 'Basic configuration' section, the 'Load balancer name' is set to 'application-loadbalancer'. The 'Scheme' is set to 'Internet-facing', which is described as serving internet-facing traffic with public IP addresses. The 'Load balancer IP address type' is set to 'IPv4'. The 'Security groups' section lists a single security group named 'default'.

EC2 > Load balancers > Create 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. When a connection request is received, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target groups.

▶ 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.
application-loadbalancer

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 IP

IPv4

Select security groups and listeners port 80.

The screenshot shows the 'Create Application Load Balancer' wizard. In the 'Listeners and routing' section, a new listener is being configured for port 80 using the HTTP protocol. The 'Default action' is set to forward traffic to the default target group.

EC2 > Load balancers > Create Application Load Balancer

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-0880f310383562b69 VPC: vpc-06cf45eaab13624fe

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 is distributed between the targets.

▼ Listener HTTP:80

Protocol	Port
HTTP	80

1-65535

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

Create target group to redirect to target group which we placed.

Go to ec2 and select target groups and create.

The screenshot shows the AWS EC2 Target Groups creation interface. It consists of two main sections: Step 1 (Create target group) and Step 2 (Register targets).

Step 1: Create target group

Create target group
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

Target group name
us-east1
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.
HTTP

Port
Port number where targets receive traffic. Can be overridden for individual targets during registration.
80
1-65535

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-06cf45aab13624fe (default)
172.31.0.0/16

(default) [Create VPC](#)

aws | Search [Alt+S] United

EC2 > Target groups > Create target group

available.

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

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.
/

Up to 1024 characters allowed.

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.

Select the both instances and click create.

EC2 > Target groups > Create target group

Step 1 Create target group
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 (2/2)

Instance ID	Name	Status	Security groups
i-0b1c306f0271cf8bc	private-instance	Running	default
i-089f2e7536df47667	public-instance	Running	default

2 selected

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

EC2 > Load balancers > application-loadbalancer

Successfully created load balancer: application-loadbalancer
It might take a few minutes for your load balancer to fully set up and route traffic. Targets will also take a few minutes to complete the registration process and pass initial health checks.

application-loadbalancer

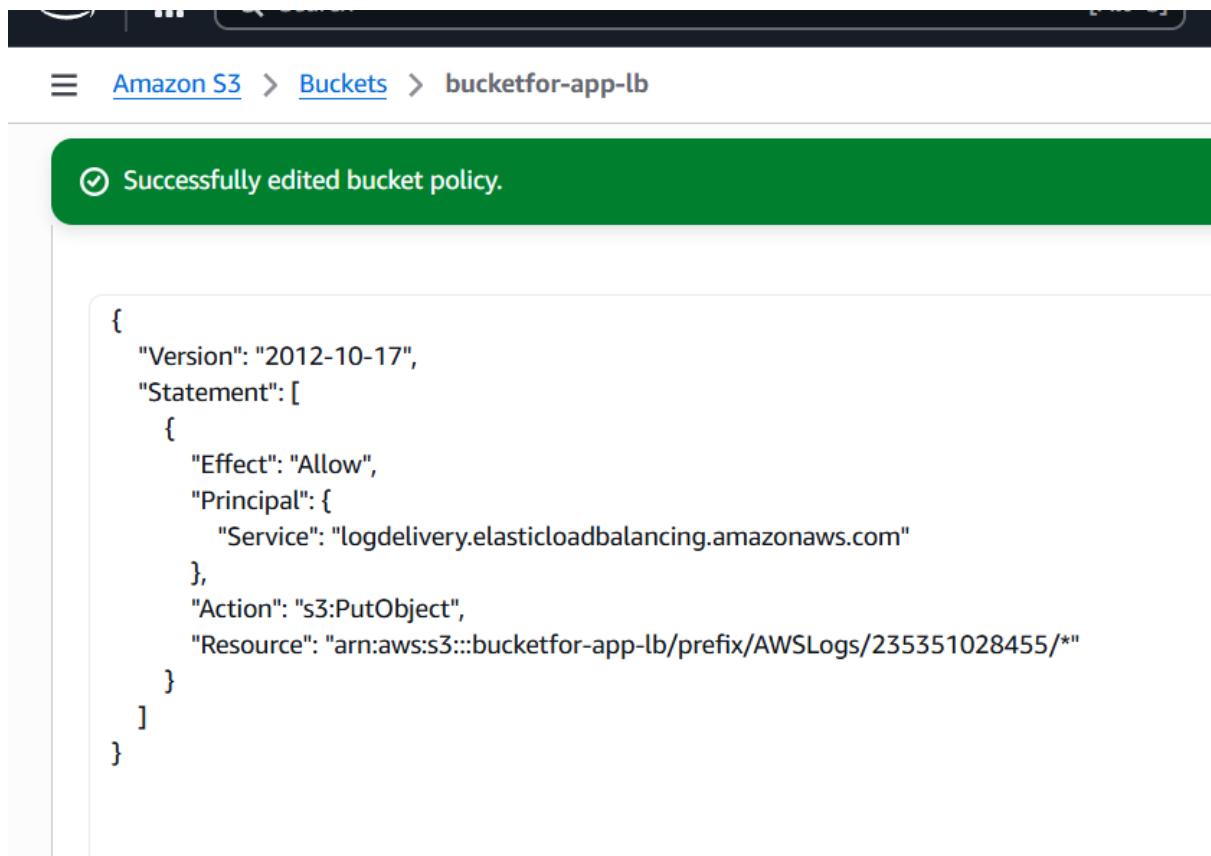
Details

Load balancer type Application	Status Provisioning	VPC vpc-06cf45eaab13624fe	Load balancer IP address type IPv4
Scheme Internet-facing	Hosted zone Z35SXDOTRQ7X7K	Availability Zones subnet-04f12a817188fcad us-east-1b (use1-az1) subnet-0a192382de0e2bf6a us-east-1a (use1-az6)	Date created October 9, 2025, 12:35 (UTC+0)
Load balancer ARN arn:aws:elasticloadbalancing:us-east-1:235351028455:loadbalancer/app/application-loadbalancer/bfa35c7473e0bf9e	DNS name application-loadbalancer-1181816658.us-east-1.elb.amazonaws.com (A record)		

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

9.Store application load balancer logs in S3.

Go to s3 bucket and create a bucket and go to that bucket and add permissions add the policy.



```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": {  
                "Service":  
                    "logdelivery.elasticloadbalancing.amazonaws.com"  
            }  
        }  
    ]  
}
```

```

    },
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::bucketfor-app-
lb/prefix/AWSLogs/235351028455/*"
}

]
}

```

Go to load balancers and edit the load balancer attribute.

The screenshot shows the AWS Management Console with the URL <https://console.aws.amazon.com/elasticloadbalancing/v2/loadbalancers>. The left sidebar navigation includes '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), and 'AWS Lambda'. The main content area displays 'Load balancers (1/1)' with a table header: Name, State, Type, Scheme, IP address type. A single row is selected for 'application-loadbalancer' (Active, application, Internet-facing, IPv4). A context menu is open over this row, with 'Edit load balancer attributes' highlighted. Other options in the menu include 'Edit IP address type', 'Edit subnets', 'Manage instances', 'Edit health check settings', 'Manage listeners', 'Edit security groups', 'Manage tags', and 'Delete load balancer'. Below the table, a section titled 'Load balancer: application-loadbalancer' contains tabs for 'Details', 'Listeners and rules', 'Network mapping', 'Resource map', 'Security', 'Monitoring', 'Integrations', 'Attributes', and 'Capacity'.

Go to monitoring and select access logs and choose your bucket.

The screenshot shows the AWS EC2 Load Balancers configuration page for an application load balancer named "application-loadbalancer". The "Edit load balancer attributes" section is open, showing two options for "Zonal shift": "Disable - Default" (selected) and "Enable". Below this is a "Protection" section with a checkbox for "Deletion protection". Under the "Monitoring" section, "Access logs" is selected, and the "S3 URI" field contains "s3://bucketfor-app-lb".

Then the flowlogs are stored in the bucket.

10. Store the VPC flow logs in a CloudWatch log group.

Go to cloud watch and select flowlog groups and create flowlogs

The screenshot shows the AWS CloudWatch interface. The left sidebar has a navigation tree: CloudWatch (selected), Favorites and recents, Dashboards, AI Operations (New), Alarms (0), Logs (selected), Log groups (selected), Log Anomalies, Live Tail, Logs Insights, Contributor Insights, Metrics (0), and All metrics (New). The main content area is titled 'Create log group'. It contains a 'Log group details' section with a note about log classes, a 'Log group name' input field containing 'vpc flowlogs', a 'Retention setting' dropdown set to '1 day', a 'Log class' dropdown set to 'Standard', and a 'KMS key ARN - optional' input field which is empty. At the bottom right of the content area, it says '© 2025, Amazon Web Services, Inc.'

Go to vpc and select flowlogs and create flowlogs.

aws | Search [Alt+S]

VPC > Your VPCs > Create flow logs

Flow log settings

Name - *optional*

flowlogs-01

Filter

The type of traffic to capture (accepted traffic only, rejected traffic only, or all traffic).

Accept
 Reject
 All

Maximum aggregation interval | [Info](#)

The maximum interval of time during which a flow of packets is captured and aggregated into a flow log record.

10 minutes
 1 minute

Destination

The destination to which to publish the flow log data.

Send to CloudWatch Logs
 Send to an Amazon S3 bucket
 Send to Amazon Data Firehose in the same account
 Send to Amazon Data Firehose in a different account

Use: logs

logs

aws | Search [Alt+S] Account ID: 2353-5102-8455 root

CloudWatch > Log groups United States (N. Virginia)

The following log group(s) have been deleted:
logs

Log groups (1)

By default, we only load up to 10000 log groups.

Filter log groups or try pattern search Exact match

Log group	Log class	Anomaly d...	Data pr...	Sensitiv...	Retention	Metric fi...
logs	Standard	Configure	-	-	Never expire	-

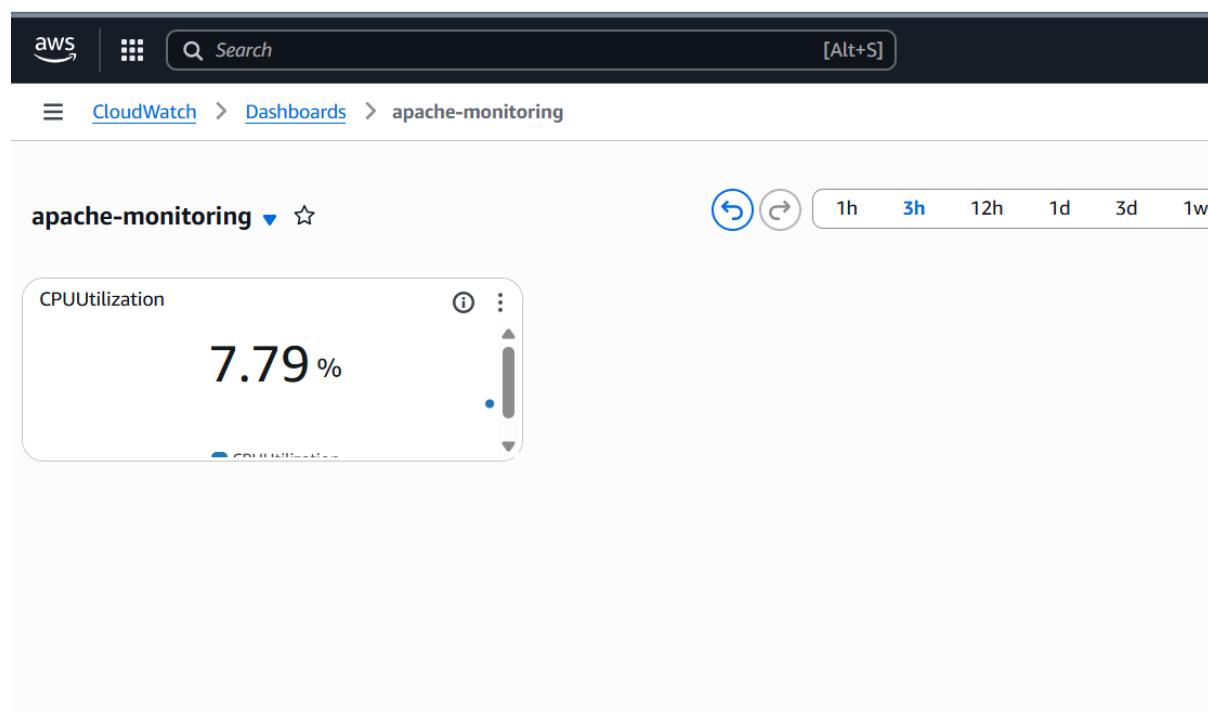
11.Create monitoring dashboards to monitor CPU utilization and to monitor the Apache service.

Launch a instance and install apache in that.

```
[root@ip-172-31-83-40 ~]# yum install httpd
Last metadata expiration check: 0:00:17 ago on Thu Oct  9 09:43:54 2025.
Dependencies resolved.
=====
 Package                               Architecture      Version
=====
Installing:
 httpd                                x86_64          2.4.65-1.amzn2023.0.1
Installing dependencies:
 apr                                  x86_64          1.7.5-1.amzn2023.0.4
 apr-util                             x86_64          1.6.3-1.amzn2023.0.1
 generic-logos-httdp                  noarch         18.0.0-12.amzn2023.0.3
 httpd-core                           x86_64          2.4.65-1.amzn2023.0.1
 httpd-filesystem                     noarch         2.4.65-1.amzn2023.0.1
 httpd-tools                          x86_64          2.4.65-1.amzn2023.0.1
 libbrotli                           x86_64          1.0.9-4.amzn2023.0.2
 mailcap                             noarch         2.1.49-3.amzn2023.0.3
Installing weak dependencies:
 apr-util-openssl                   x86_64          1.6.3-1.amzn2023.0.1
 mod_http2                           x86_64          2.0.27-1.amzn2023.0.3
 mod_lua                            x86_64          2.4.65-1.amzn2023.0.1
```

Go to cloudwatch and create a dashboard.

Select the metrics as number,ec2,copy the instance id and paste there and select apache service cpu utilization.



12.If CPU utilization is more than 70%, then it should trigger auto scaling and launch new instance.

Go to templates launch create template

The screenshot shows the 'Create launch template' wizard. Step 1: Launch template name and description. The 'Launch template name - required' field contains 'my-template'. A note below says 'Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '*', '@'.' The 'Template version description' field contains 'v1'. A note below says 'Max 255 chars'. Under 'Auto Scaling guidance', there is an info link and a checkbox for 'Provide guidance to help me set up a template that I can use with EC2 Auto Scaling'. Step 2: Launch template contents. It includes sections for 'Instance type', 'Key pair (login)', 'Network settings', and a summary on the right.

The screenshot shows the 'Launch template contents' step of the wizard. It includes sections for 'Instance type' (selected: t3.micro), 'Key pair (login)' (selected: red), 'Network settings' (selected: Don't include in launch template), and a 'Summary' section on the right. The summary includes details about the software image (Amazon Linux 2023 kernel-6.1...), virtual server type (t3.micro), firewall (default), and storage (1 volume(s) - 8 GiB). At the bottom, there are 'Cancel' and 'Next Step' buttons.

EC2		Launch Templates (1) <small>Info</small>						
		<small>Select a launch template</small>						
		Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time	Created By	
		lt-07620cc86076ddcc8	my-template	1	1	2025-10-09T10:08:19.000Z	arn:aws:iam::2353	

Go to auto scaling groups and create auto scaling groups.

Give the name and select subnets, and give the security groups.

The screenshot shows the AWS EC2 Auto Scaling Groups creation wizard at Step 7: Review. On the left, a sidebar lists optional steps: Add notifications, Add tags, and Review. The main area shows the selected launch template (my-template, ID lt-07620cc86076ddcc8), instance type (t3.micro), and network settings. Under Network, it says: "For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across suitable for getting started quickly." The VPC section shows "vpc-06cf45aab13624fe (default)" selected. The Availability Zones and subnets section lists two subnets: "use1-az6 (us-east-1a) | subnet-0a192382de0e2bf6a (default-public-subnet)" and "use1-az1 (us-east-1b) | subnet-04f12a817188fcadc (default-private-subnet)".

Integrate with other services - *optional* Info

Use a load balancer to distribute network traffic across multiple servers. Enable service-to-service communications with VPC Lattice. Shift resources away from impaired Availability Zones with zonal shift. You can also customize health check replacements and monitoring.

Load balancing Info

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

Select Load balancing options

- No load balancer** Traffic to your Auto Scaling group will not be fronted by a load balancer.
- Attach to an existing load balancer** Choose from your existing load balancers.
- Attach to a new load balancer** Quickly create a basic load balancer to attach to your Auto Scaling group.

VPC Lattice integration options Info

To improve networking capabilities and scalability, integrate your Auto Scaling group with VPC Lattice. VPC Lattice facilitates communications between AWS services and helps you connect and manage your applications across compute services in AWS.

Select VPC Lattice service to attach

- No VPC Lattice service** VPC Lattice will not manage your Auto Scaling group's network access and connectivity with other services.
- Attach to VPC Lattice service** Incoming requests associated with specified VPC Lattice target groups will be routed to your Auto Scaling group.

[Create new VPC Lattice service](#)

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

Two instances were created.

Activity history (2)

Status	Description	Cause	Start time
Successful	Launching a new EC2 instance: i-01272fdb4825d907a	At 2025-10-09T10:27:12Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 2. At 2025-10-09T10:30:10Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 2.	2025 October 09, 04:00:12 PM +05:30
Successful	Launching a new EC2 instance: i-0ded17e4a9157e40	At 2025-10-09T10:27:12Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 2. At 2025-10-09T10:30:10Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 2.	2025 October 09, 04:00:12 PM +05:30