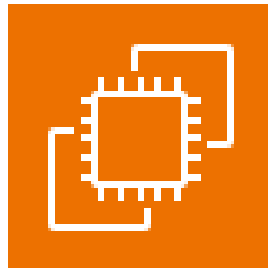
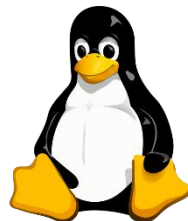




AWS Solution Architect Training with AWS Cloud Practitioner Global Certification Training

Trainer: Aravindraj.G- N minds Academy

Configure Network Load Balancer with 3 Linux Web Servers in AWS



Deepakraj R

B.E -Computer Science Engineering

Mepco Schlenk Engineering College

Sivakasi



Objective

An AWS Application Load Balancer (ALB) is a fully managed Layer 7 (HTTP/HTTPS) load balancing service provided by Amazon Web Services (AWS). It is designed to handle HTTP and HTTPS traffic and offers advanced routing capabilities, allowing you to direct traffic to different resources based on the content of the request (e.g., URL paths, hostnames, query parameters, HTTP headers, etc.).

When to Use AWS Application Load Balancer:

1. **Web Applications:** If you are building a traditional web application or microservices application, ALB can route HTTP/HTTPS traffic to different services based on URL paths, hostnames, or headers.
2. **Microservices Architecture:** ALB is a good choice for microservices, where different services are hosted on different target groups. It can route traffic to the appropriate service based on the URL or other request attributes.
3. **Content-based Routing:** If you need to route traffic based on URL paths, hostnames, HTTP headers, or query parameters, ALB's advanced routing features are very useful.
4. **SSL/TLS Termination:** If you need to offload SSL/TLS decryption from your backend servers, ALB is a great option for handling HTTPS traffic securely.
5. **WebSocket Applications:** ALB supports WebSocket connections, making it suitable for applications that require persistent, real-time, full-duplex communication.
6. **Global Applications:** If you have applications that require high availability across multiple regions or availability zones, ALB's support for cross-zone load balancing helps ensure fault tolerance.

Nomenclature and Components:

1. **Listeners:** A listener checks for connection requests, defined by a protocol (HTTP or HTTPS) and a port (typically port 80 for HTTP or port 443 for HTTPS). It forwards traffic to one or more target groups based on rules you define.
2. **Target Groups:** A target group is a set of backend resources (such as EC2 instances or containers) that the ALB forwards traffic to. Each target group can be associated with specific health check settings. Targets can be registered or deregistered as needed.



3. **Rules:** ALB allows you to define rules that determine how traffic is routed. You can configure rules to route traffic based on hostnames, path patterns, HTTP headers, query strings, or even the HTTP method (GET, POST, etc.).
4. **Health Checks:** ALB checks the health of targets by sending HTTP(S) requests to a specific path (e.g., /health). If a target is unhealthy, ALB stops routing traffic to it and forwards traffic to healthy targets.
5. **Target Types:** ALB supports three types of targets:
 - **Instance:** EC2 instances registered directly to the target group.
 - **IP:** IP addresses, such as instances in a private subnet or on-premise servers.
 - **Lambda functions:** AWS Lambda functions can also be used as targets.

Example Use Case:

Consider an e-commerce application where different services handle user authentication, product search, and order processing. You can use an ALB to route traffic as follows:

- Requests to /auth/* go to the authentication service.
- Requests to /products/* go to the product search service.
- Requests to /orders/* go to the order processing service.

This routing setup can be achieved by configuring the ALB with path-based routing rules, improving the scalability and organization of the application.

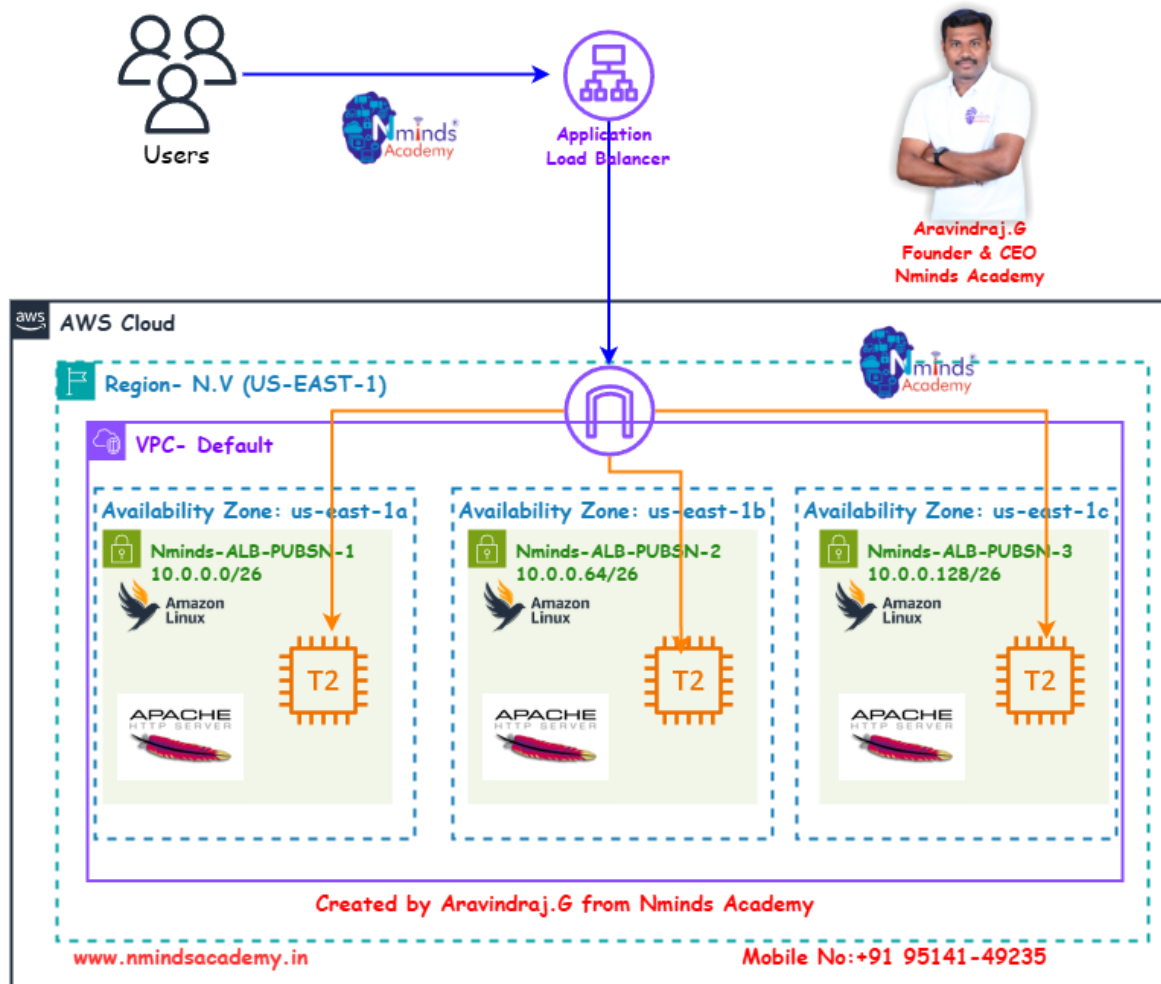
Benefits:

- **Advanced Routing:** ALB provides powerful content-based routing, making it suitable for modern applications and microservices.
- **SSL/TLS Termination:** Offload SSL/TLS decryption to the load balancer, reducing the overhead on your application servers.
- **Improved Fault Tolerance:** Health checks and automatic rerouting to healthy targets ensure high availability and reliability.
- **Ease of Use:** Fully managed and integrates easily with other AWS services like EC2, ECS, EKS, and Lambda.



Topology

Configure High Availability with 3 Webservers using Application Load Balancer in AWS

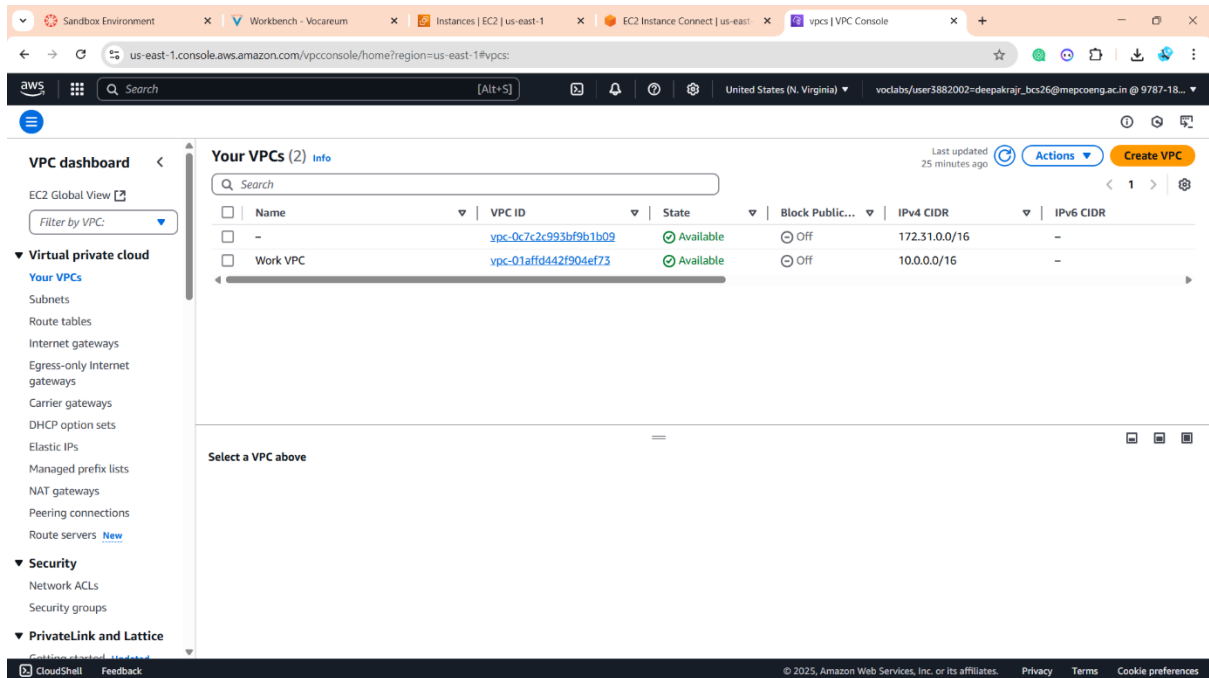


Aravindraj.G
Founder & CEO
Nminds Academy



Execution Tasks:

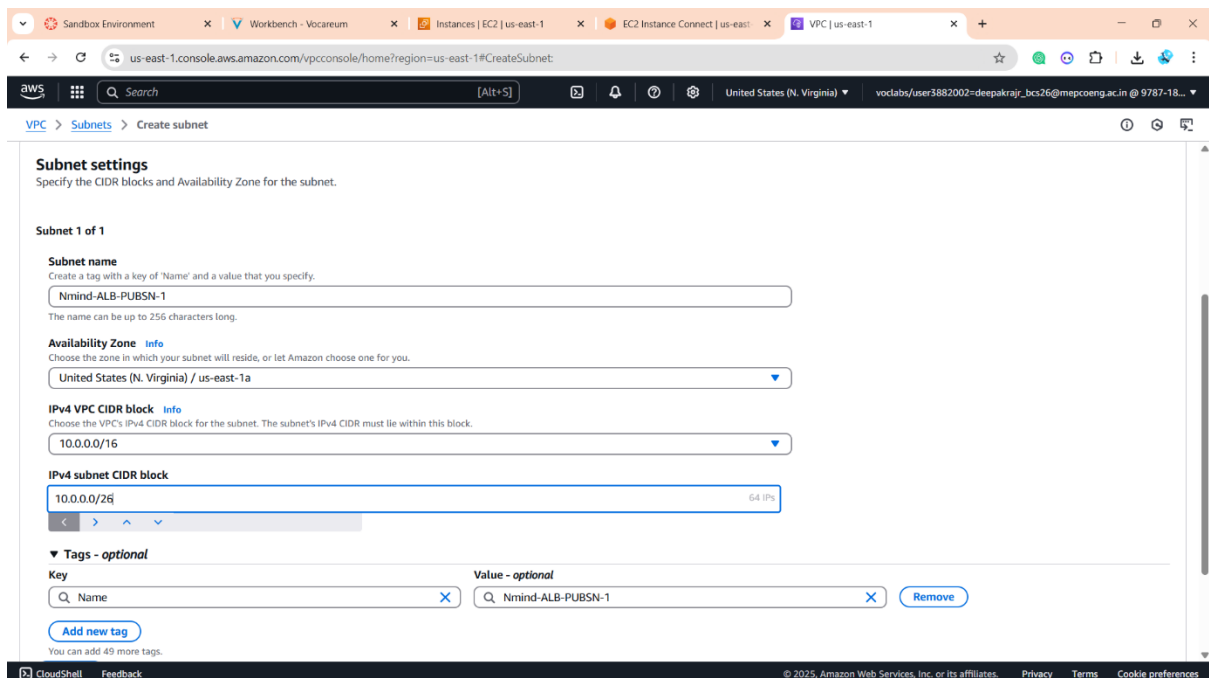
Step1: Set Up the VPC and Subnets. Use the Default VPC



The screenshot shows the AWS VPC console interface. The left sidebar contains navigation links for VPC dashboard, Virtual private cloud, Security, and PrivateLink and Lattice. The main content area displays 'Your VPCs (2)' with a table listing two VPCs: a default VPC and a 'Work VPC'. Both are in an 'Available' state. The 'Work VPC' has an IPv4 CIDR of 10.0.0.0/16. Below the table, there is a section to 'Select a VPC above'.

Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR
-	vpc-0c7c2c993bf9b1b09	Available	Off	172.31.0.0/16	-
Work VPC	vpc-01affd442f904ef73	Available	Off	10.0.0.0/16	-

Step2: create the public subnets within the default VPC:



The screenshot shows the 'Create subnet' page in the AWS VPC console. The 'Subnet settings' section is active, showing fields for 'Subnet name' (Nminds-ALB-PUBSN-1), 'Availability Zone' (United States (N. Virginia) / us-east-1a), 'IPv4 VPC CIDR block' (10.0.0.0/16), and 'IPv4 subnet CIDR block' (10.0.0.0/26). The 'Tags' section is also visible, showing a tag with key 'Name' and value 'Nminds-ALB-PUBSN-1'.

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.
Nminds-ALB-PUBSN-1
The name can be up to 256 characters long.

Availability Zone
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
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
10.0.0.0/16

IPv4 subnet CIDR block
10.0.0.0/26 (64 IPs)

Tags - optional

Key	Value - optional
Name	Nminds-ALB-PUBSN-1

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



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

Get started with PrivateLink

CloudShell Feedback

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

Step3: Select each subnet, click Actions > Edit Subnet Settings, and enable Auto-assign public IPv4 address.

Edit subnet settings

Subnet

Subnet ID: subnet-0088e5eb1cdc379d7

Name: Nimind-ALB-PUBSN-2

Auto-assign IP settings

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

☒ Enable auto-assign public IPv4 address

☐ Enable auto-assign customer-owned IPv4 address

Resource-based name (RBN) settings

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

☐ Enable resource name DNS A record on launch

☐ Enable resource name DNS AAAA record on launch

Hostname type

☐ Resource name

☒ IP name

DNS64 settings

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

CloudShell Feedback

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

Step4: Go to Route Tables, select the main route table associated with the default VPC, and ensure a route exists for 0.0.0.0/0 with a target of the Internet Gateway



us-east-1.console.aws.amazon.com/vpconsole/home?region=us-east-1#RouteTables:

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](#)

Route tables (3) Info

Last updated 41 minutes ago

[Actions](#) [Create route table](#)

Find resources by attribute or tag

<input type="checkbox"/>	Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
<input type="checkbox"/>	-	rtb-0827bfe9282b8dcfe	-	-	Yes	vpc-01affd442f904ef73 Work
<input type="checkbox"/>	Work Public Route Table	rtb-04f59da35f2146c5a	-	-	No	vpc-01affd442f904ef73 Work
<input type="checkbox"/>	-	rtb-0841cd28cfa393a2	-	-	Yes	vpc-0c7c2c993bf9b1b09

Select a route table

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

us-east-1.console.aws.amazon.com/vpconsole/home?region=us-east-1#EditRoutesRouteTableId=rtb-0827bfe9282b8dcfe

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
0.0.0.0/0	Internet Gateway	-	No

[Add route](#) [Cancel](#) [Preview](#) [Save changes](#)

CloudShell Feedback

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

Step5: Navigate to Launch Templates and click Create Launch Template.



us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateTemplate:

EC2 > Launch templates > Create launch template

▼ Instance type Info | Get advice Advanced

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

Additional costs apply for AMIs with pre-installed software

▼ 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

WebServer-Key-Pair Create new key pair

▼ Network settings Info

Subnet Info

Don't include in launch template Create new subnet

When you specify a subnet, a network interface is automatically added to your template.

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.

▼ Summary

Software Image (AMI)

Amazon Linux 2023 AMI 2023.7.2...read more

ami-00a929b66ed6e0de6

Virtual server type (instance type)

t2.micro

Firewall (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 Create launch template

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateTemplate:

EC2 > Launch templates > Create launch template

Inbound Security Group Rules

▼ Security group rule 1 (TCP; 22, 42.104.211.153/32) Remove

Type Info Protocol Info Port range Info

ssh TCP 22

Source type Info Name Info Description - optional Info

My IP 42.104.211.153/32 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

Custom 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

▼ Summary

Software Image (AMI)

Amazon Linux 2023 AMI 2023.7.2...read more

ami-00a929b66ed6e0de6

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 Create launch template



us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateTemplate:

Success
Successfully created Nmind-WebServer-Template(lt-09fb281e1e2423ceb).

► Actions log

Next Steps

Launch an instance
With On-Demand Instances, you pay for compute capacity by the second (for Linux, with a minimum of 60 seconds) or by the hour (for all other operating systems) with no long-term commitments or upfront payments. Launch an On-Demand Instance from your launch template.

[Launch instance from this template](#)

Create an Auto Scaling group from your template
Amazon EC2 Auto Scaling helps you maintain application availability and allows you to scale your Amazon EC2 capacity up or down automatically according to conditions you define. You can use Auto Scaling to help ensure that you are running your desired number of Amazon EC2 instances during demand spikes to maintain performance and decrease capacity during lulls to reduce costs.

[Create Auto Scaling group](#)

Create Spot Fleet
A Spot Instance is an unused EC2 instance that is available for less than the On-Demand price. Because Spot Instances enable you to request unused EC2 instances at steep discounts, you can lower your Amazon EC2 costs significantly. The hourly price for a Spot Instance (of each instance type in each Availability Zone) is set by Amazon EC2, and adjusted gradually based on the long-term supply of and demand for Spot Instances. Spot instances are well-suited for data-analysis, batch jobs, background processing, and optional tasks.

[Create Spot Fleet](#)

[View launch templates](#)

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

Step6: Create an Application Load Balancer (ALB)

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#SelectCreateELBWizard:

EC2 > Load balancers > Compare and select load balancer type

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](#)

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

Step7: Configure ALB



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](#). For a new VPC, [create a VPC](#).

Work VPC
vpc-01affd442f904ef73
IPv4 VPC CIDR: 10.0.0.0/16

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 [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-az2)**
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-04aab8f5cb08bf2d3
IPv4 subnet CIDR: 10.0.0.0/26
Nmind-ALB-PUBSN-1

☒ **us-east-1b (use1-az4)**
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-0088e5eb1cdc379d7
IPv4 subnet CIDR: 10.0.64.0/26
Nmind-ALB-PUBSN-2

☒ **us-east-1c (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-027659d6be7958d7c
IPv4 subnet CIDR: 10.0.128.0/26
Nmind-ALB-PUBSN-3

Step8: Create Target Group

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.



us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateTargetGroup:protocol=HTTP;vpc=vpc-01affd442f904ef73

EC2 > Target groups > Create target group

Health check port
The port the load balancer uses when performing health checks on targets. By default, the health check port is the same as the target group's traffic port. However, you can specify a different port as an override.

☒ Traffic port
☐ Override

Healthy threshold
The number of consecutive health check successes required before considering an unhealthy target healthy.

2-10

Unhealthy threshold
The number of consecutive health check failures required before considering a target unhealthy.

2-10

Timeout
The amount of time, in seconds, during which no response means a failed health check.

seconds

2-120

Interval
The approximate amount of time between health checks of an individual target

seconds

5-300

Success codes
The HTTP codes to use when checking for a successful response from a target. You can specify multiple values (for example, "200,202") or a range of values (for example, "200-299").

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

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#TargetGroup:targetGroupArn=arn:aws:elasticloadbalancing:us-east-1:978718986665:targetgroup/Nmind-TG/...

EC2 > Target groups > Nmind-TG

EC2

- Dashboard
- EC2 Global View
- Events
- Instances
 - 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

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

Nmind-TG Actions

Details

arn:aws:elasticloadbalancing:us-east-1:978718986665:targetgroup/Nmind-TG/9c1c78eb19f6f05e

Target type Instance	Protocol : Port HTTP: 80	Protocol version HTTP1	VPC vpc-01affd442f904ef73
IP address type IPv4	Load balancer None associated		

Targets Monitoring Health checks Attributes Tags

Registered targets (0) Info Anomaly mitigation: Not applicable Deregister Register targets

Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.

Filter targets

Instance ID	Name	Port	Zone	Health status	Health status details	Admin...	Over
Loading...							

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

Step9: Attach Target Group



The top screenshot shows the 'Create Application Load Balancer' wizard in the AWS Management Console. The 'Security groups' section has 'WebServer-SG' (sg-09a871cd58c5caa9f) selected. The 'Listeners and routing' section shows a listener for HTTP on port 80, forwarding to 'Nmind-TG' (Target type: Instance, IPv4). The bottom screenshot shows the 'Load balancers' page with a table listing one active load balancer: 'Application-Load-Balancer...' with VPC ID 'vpc-01affd442f904ef73' and 3 Availability Zones.

Name	DNS name	State	VPC ID	Availability Zones	Type	Date create
Application-Load-Balancer...	Application-Load-Balancer...	Active	vpc-01affd442f904ef73	3 Availability Zones	application	April 17, 20...

Step10: Create an Auto Scaling Group (ASG)



Step11: Configure ASG

Step12: Attach Application Load Balancer and associate with Target group.



us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateAutoScalingGroup:

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 3 - optional
Integrate with other services
 Step 4 - optional
 Configure group size and scaling
 Step 5 - optional
 Add notifications
 Step 6 - optional
 Add tags
 Step 7
 Review

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.

☐ 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.

Attach to an existing load balancer

Select the load balancers that you want to attach to your Auto Scaling group.

☒ Choose from your load balancer target groups
 This option allows you to attach Application, Network, or Gateway Load Balancers.

☐ Choose from Classic Load Balancers

Existing load balancer target groups

Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

Select target groups

Nmind-TG | HTTP
 Application Load Balancer: Application-Load-Balancer

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.

☐ Attach to VPC Lattice service
 An Amazon VPC Lattice service is required to use VPC Lattice.

CloudShell Feedback

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

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#AutoScalingGroups:

EC2 > Auto Scaling groups

Auto Scaling groups (1) Info

<input type="checkbox"/>	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
<input type="checkbox"/>	WebServer-ASG	Nmind-WebServer-Template Version Del	3	-	3	1	6	us-east-1a, us-east-1b,...

0 Auto Scaling groups selected

CloudShell Feedback

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

Step13: Test the Configuration



us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#instances?v=3;\$case=tags:true%5C,client:false;\$regex=tags:false%5C,client:false

EC2 > Instances

Instances (1/5) Info

Find Instance by attribute or tag (case-sensitive)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public
Bastion Host	i-0e7145ba0c7c1fc5e	Terminated	t2.micro	-	View alarms +	us-east-1a	-
	i-00b230736ac3c2e7b	Running	t2.micro	Initializing	View alarms +	us-east-1a	ec2-5
Linux-Web-Server	i-05a95f133fc3bff1f	Terminated	t2.micro	-	View alarms +	us-east-1a	-
	i-0229e37005ee979c9	Running	t2.micro	Initializing	View alarms +	us-east-1b	ec2-5
	i-0b36d751e754aa910	Running	t2.micro	Initializing	View alarms +	us-east-1c	ec2-3

i-0229e37005ee979c9

Details | Status and alarms | Monitoring | Security | Networking | Storage | Tags

Instance summary Info

Instance ID: i-0229e37005ee979c9

IPv6 address: -

Hostname type: IP name: ip-10-0-64-30.ec2.internal

Public IPv4 address: 54.172.50.198 | open address

Instance state: Running

Private IP DNS name (IPv4 only): ip-10-0-64-30.ec2.internal

Private IPv4 addresses: 10.0.64.30

Public IPv4 DNS: ec2-54-172-50-198.compute-1.amazonaws.com | open address

