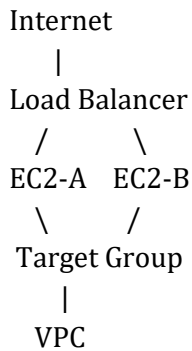


Highly Available Web Application Deployment Using AWS

Objective:

To deploy a simple web application on two EC2 instances located in different Availability Zones, fronted by an Application Load Balancer (ALB), ensuring high availability — if one EC2 instance fails, the other serves traffic seamlessly.

Architecture Diagram:



Step-by-Step Implementation:

Step 1: Create a VPC

CIDR Block: 10.0.0.0/16

The screenshot shows the AWS VPC console interface. On the left is a navigation menu with sections for 'Virtual private cloud' (containing links for Subnets, Route tables, Internet gateways, Egress-only internet gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, and Peering connections) and 'Security' (containing links for Network ACLs and Security groups). The main area is titled 'Your VPCs (1/2)' and contains a table of VPCs. The table has columns for Name, VPC ID, State, Block Public Access, and IPv4 CIDR. Two VPCs are listed: 'vpc-for-web-application-deployment' (VPC ID: vpc-01756043392ef4719, State: Available, Block Public Access: Off, IPv4 CIDR: 10.0.0.0/16) and another VPC (VPC ID: vpc-08e490d66dd663677, State: Available, Block Public Access: Off, IPv4 CIDR: 172.31.0.0/16). Below the table, the details for 'vpc-01756043392ef4719 / vpc-for-web-application-deployment' are shown. These details are organized into four columns: 'Details' (VPC ID, DNS resolution, Main network ACL), 'State' (State, Tenancy, Default VPC), 'Block Public Access' (Block Public Access, DHCP option set, IPv4 CIDR), and 'DNS hostnames' (DNS hostnames, Main route table, IPv6 pool).

Name	VPC ID	State	Block Public...	IPv4 CIDR
<input checked="" type="checkbox"/> vpc-for-web-application-deployment	vpc-01756043392ef4719	Available	Off	10.0.0.0/16
<input type="checkbox"/> -	vpc-08e490d66dd663677	Available	Off	172.31.0.0/16

vpc-01756043392ef4719 / vpc-for-web-application-deployment			
Details	State	Block Public Access	DNS hostnames
VPC ID vpc-01756043392ef4719	State Available	Block Public Access Off	DNS hostnames Disabled
DNS resolution Enabled	Tenancy default	DHCP option set dopt-01ae084900474afe9	Main route table rtb-092e4b11c7d301aae
Main network ACL acl-028884102633fad14	Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool -

Step 2: Create Two Public Subnets

Subnet-1 | us-east-1a | 10.0.1.0/24

Subnet-2 | us-east-1b | 10.0.2.0/24

Subnets (1/5) Info

Find subnets by attribute or tag

Name	Subnet ID	State	VPC
web-deployment-1a	subnet-0299aaf60ff2d8e98	Available	vpc-01756043392ef4719 vpc-...
web-deployment-1b	subnet-0cdacece13a9e660	Available	vpc-01756043392ef4719 vpc-...
-	subnet-013c9c9ef430f08da	Available	vpc-08e490d66dd663677
-	subnet-09a0a35dec79b6b2b	Available	vpc-08e490d66dd663677

subnet-0299aaf60ff2d8e98 / web-deployment-1a

Subnet ID	Subnet ARN	State	Block Public Access
subnet-0299aaf60ff2d8e98	arn:aws:ec2:ap-south-1:750311440127:subnet/subnet-0299aaf60ff2d8e98	Available	Off
IPv4 CIDR	Available IPv4 addresses	IPv6 CIDR	IPv6 CIDR association ID
10.0.1.0/24	249	-	-
Availability Zone	Availability Zone ID	Network border group	VPC
ap-south-1a	aps1-az1	ap-south-1	vpc-01756043392ef4719 vpc-for-web-application-deployment
Route table		Default subnet	
		No	

Step 3: Create and Attach an Internet Gateway

Attach it to the VPC.

Internet gateways (1/2) Info

Find internet gateways by attribute or tag

Name	Internet gateway ID	State	VPC ID
igw-web-deployment	igw-051ebd3f56c46e6fa	Attached	vpc-01756043392ef4719 vpc-...
-	igw-0622073fe5da35c7c	Attached	vpc-08e490d66dd663677

igw-051ebd3f56c46e6fa / igw-web-deployment

Details

Internet gateway ID	State	VPC ID	Owner
igw-051ebd3f56c46e6fa	Attached	vpc-01756043392ef4719 vpc-for-web-application-deployment	750311440127

Step 4: Create a Route Table and Attach to Subnets

Add route: Destination: 0.0.0.0/0 → Target: Internet Gateway

Associate with both public subnets.

The screenshot shows the AWS Management Console interface. On the left is the 'VPC dashboard' sidebar with options like 'Your VPCs', 'Subnets', 'Route tables', 'Internet gateways', etc. The main panel is titled 'Route tables (1/3)'. It contains a table of route tables. The table has columns: Name, Route table ID, Explicit subnet associations, Edge associations, and Main. The row for 'web-deployment-rt' (ID: rtb-08a0639fc9fdf48a4) is selected. Below the table, the 'Subnet associations' tab is active, showing 'Explicit subnet associations (2)'. This sub-table lists two associations: 'web-deployment-1a' (subnet: subnet-0299aaf60ff2d8e98, IPv4 CIDR: 10.0.1.0/24) and 'web-deployment-1b' (subnet: subnet-0cdacece13a9e660, IPv4 CIDR: 10.0.2.0/24).

Name	Route table ID	Explicit subnet associations	Edge associations	Main
-	rtb-093bf42c4c203d8ab	-	-	Yes
-	rtb-092e4b11c7d301aae	-	-	Yes
web-deployment-rt	rtb-08a0639fc9fdf48a4	2 subnets	-	No

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
web-deployment-1a	subnet-0299aaf60ff2d8e98	10.0.1.0/24	-
web-deployment-1b	subnet-0cdacece13a9e660	10.0.2.0/24	-

Step 5: Launch Two EC2 Instances (Web Servers) with Created VPC

AMI: Amazon Linux 2, Type: t2.micro, One in each subnet

Use this user data:

```
#!/bin/bash
```

```
sudo yum update -y
```

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

```
echo "Welcome to $(hostname) - Web Server" > /var/www/html/index.html
```

```
sudo systemctl start httpd
```

```
sudo systemctl enable httpd
```

The screenshot shows the AWS Management Console 'EC2 > Instances' page. It displays a table of instances. Two instances are listed: 'web-deployment-server-2' (ID: i-0ce4f75866ada822a, state: Running) and 'web-deployment-server-1' (ID: i-073790655b83f4e7c, state: Initializing). The details for 'web-deployment-server-1' are expanded below. It shows the instance ID, public IPv4 address (65.2.34.136), private IPv4 address (10.0.1.84), instance state (Running), private IP DNS name (ip-10-0-1-84.ap-south-1.compute.internal), and instance type (t2.micro).

Name	Instance ID	Instance state	Instance type	Status check	Alarm
web-deployment-server-2	i-0ce4f75866ada822a	Running	t2.micro	2/2 checks passed	View
web-deployment-server-1	i-073790655b83f4e7c	Initializing	t2.micro	Initializing	View

i-073790655b83f4e7c (web-deployment-server-1)

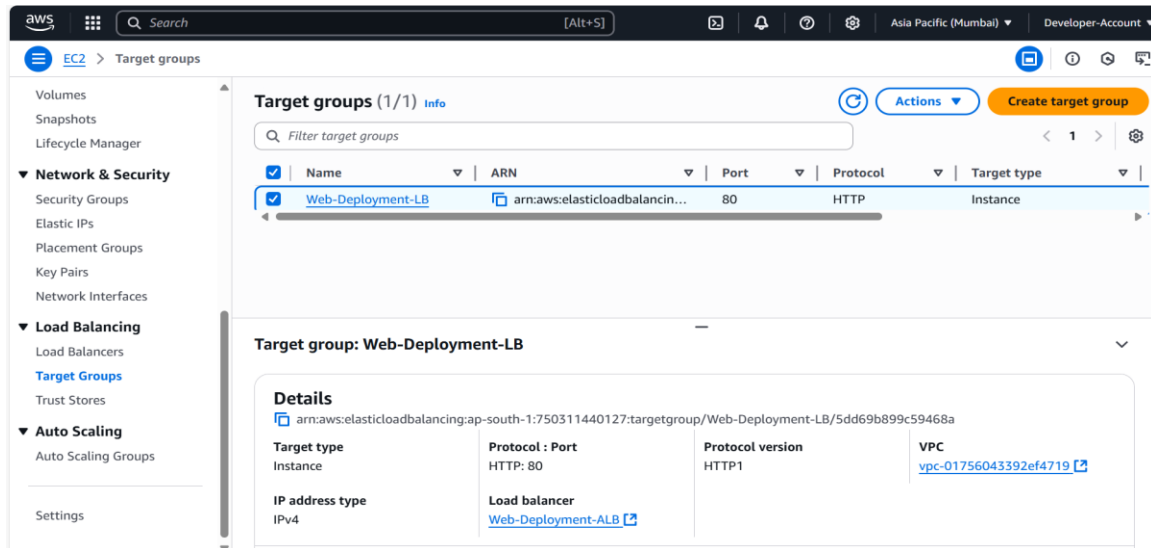
Instance ID i-073790655b83f4e7c	Public IPv4 address 65.2.34.136 open address	Private IPv4 addresses 10.0.1.84
IPv6 address -	Instance state Running	Public DNS -
Hostname type IP name: ip-10-0-1-84.ap-south-1.compute.internal	Private IP DNS name (IPv4 only) ip-10-0-1-84.ap-south-1.compute.internal	Elastic IP addresses -
Answer private resource DNS name -	Instance type t2.micro	

Step 6: Configure Security Group for EC2s

Allow SSH (22) from your IP and HTTP (80) from 0.0.0.0/0

Step 7: Create a Target Group

Type: Instance, Protocol: HTTP, Port: 80, Health check path: /



The screenshot shows the AWS Management Console interface for the 'Target groups' page. The left sidebar contains navigation links for Volumes, Snapshots, Lifecycle Manager, Network & Security, Load Balancing, and Auto Scaling. The main content area displays the 'Target groups (1/1)' list with a table containing one entry: 'Web-Deployment-LB'. Below the table, the 'Details' section for 'Web-Deployment-LB' is shown, including its ARN, Target type (Instance), Protocol (HTTP), Port (80), Protocol version (HTTP1), and VPC ID.

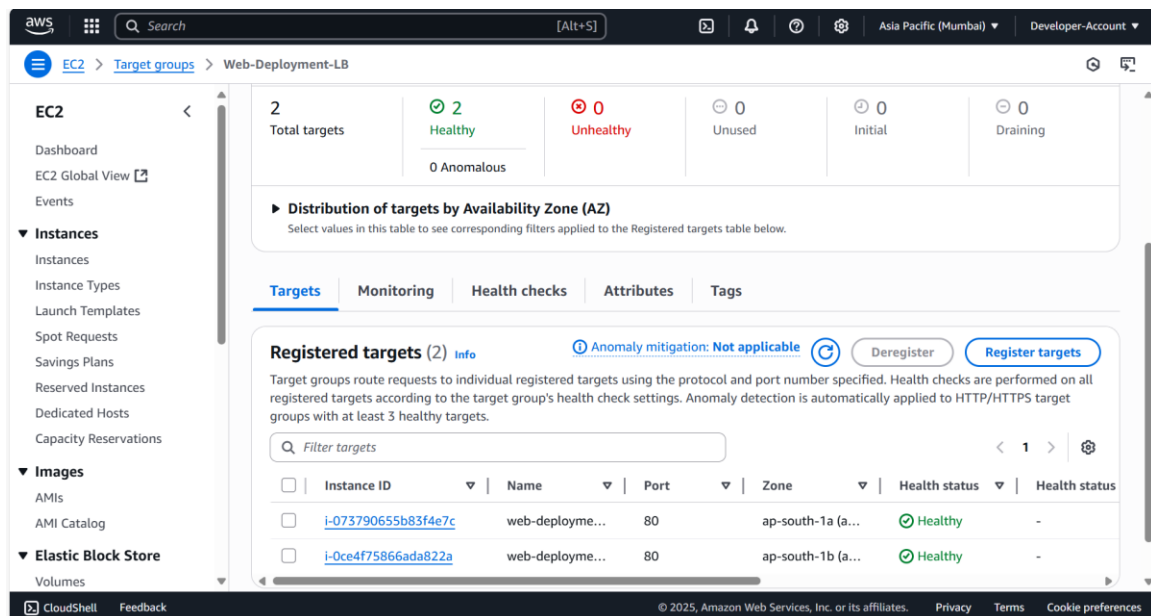
Name	ARN	Port	Protocol	Target type
Web-Deployment-LB	arn:aws:elasticloadbalancing:ap-south-1:750311440127:targetgroup/Web-Deployment-LB/5dd69b899c59468a	80	HTTP	Instance

Target group: Web-Deployment-LB

Details

arn:aws:elasticloadbalancing:ap-south-1:750311440127:targetgroup/Web-Deployment-LB/5dd69b899c59468a

Target type	Protocol : Port	Protocol version	VPC
Instance	HTTP: 80	HTTP1	vpc-01756043392ef4719
IP address type	Load balancer		
IPv4	Web-Deployment-ALB		



The screenshot shows the AWS Management Console interface for the 'Web-Deployment-LB' target group. The left sidebar contains navigation links for EC2, Instances, Images, and Elastic Block Store. The main content area displays the 'Web-Deployment-LB' target group details. The 'Targets' tab is selected, showing a summary of 2 total targets, all healthy, and a table of registered targets with their Instance IDs, Names, Ports, Zones, and Health Status.

2 Total targets

2 Healthy

0 Unhealthy

0 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 (2)

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
i-073790655b83f4e7c	web-deploye...	80	ap-south-1a (a...	Healthy	-
i-0ce4f75866ada822a	web-deploye...	80	ap-south-1b (a...	Healthy	-

Step 8: Create an Application Load Balancer

Internet-facing, Listener on HTTP 80, Subnet-1 and Subnet-2, Attach target group

aws [Search] [Alt+S] Asia Pacific (Mumbai) Developer-Account

EC2 > Load balancers

Load balancers (1/1)

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

Filter load balancers

<input checked="" type="checkbox"/>	Name	State	Type	Scheme	IP address type	VPC ID
<input checked="" type="checkbox"/>	Web-Deployment-ALB	Active	application	Internet-facing	IPv4	vpc-0175i

Load balancer: Web-Deployment-ALB

Load balancer type Application	Status Active	VPC vpc-01756043392ef4719	Load balancer IP address type IPv4
Scheme Internet-facing	Hosted zone ZP97RAFLXTNZK	Availability Zones subnet-0299aaf60ff2d8e98 ap-south-1a (aps1-az1) subnet-0cdaceece13a9e660 ap-south-1b (aps1-az3)	Date created August 5, 2025, 17:36 (UTC+05:30)

aws [Search] [Alt+S] Asia Pacific (Mumbai) Developer-Account

EC2 > Load balancers > Web-Deployment-ALB

Web-Deployment-ALB

Details

Load balancer type Application	Status Active	VPC vpc-01756043392ef4719	Load balancer IP address type IPv4
Scheme Internet-facing	Hosted zone ZP97RAFLXTNZK	Availability Zones subnet-0299aaf60ff2d8e98 ap-south-1a (aps1-az1) subnet-0cdaceece13a9e660 ap-south-1b (aps1-az3)	Date created August 5, 2025, 17:36 (UTC+05:30)
Load balancer ARN arn:aws:elasticloadbalancing:ap-south-1:750311440127:loadbalancer/app/Web-Deployment-ALB/2aaf3b5bb37fb863		DNS name info Web-Deployment-ALB-1839859024.ap-south-1.elb.amazonaws.com (A Record)	

Step 9: Test the ALB in Browser

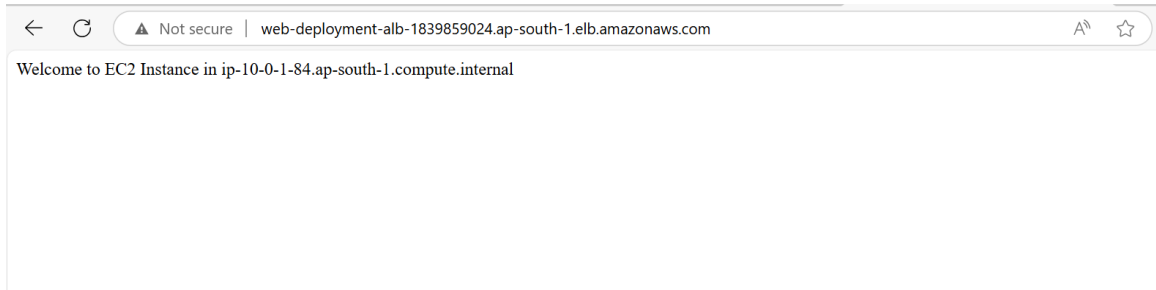
Find ALB DNS in EC2 → Load Balancers.

Open in browser: `http://<ALB-DNS-Name>`

web-deployment-alb-1839859024.ap-south-1.elb.amazonaws.com

← ↻ Not secure | web-deployment-alb-1839859024.ap-south-1.elb.amazonaws.com

Welcome to EC2 Instance in ip-10-0-2-140.ap-south-1.compute.internal



aws Search [Alt+S] Asia Pacific (Mumbai) Developer-Account

EC2 > Target groups > Web-Deployment-LB

Web-Deployment-LB

Details

arn:aws:elasticloadbalancing:ap-south-1:750311440127:targetgroup/Web-Deployment-LB/5dd69b899c59468a

Target type Instance	Protocol : Port HTTP: 80	Protocol version HTTP1	VPC vpc-01756043392ef4719
IP address type IPv4	Load balancer Web-Deployment-ALB		

2	2 Healthy	0 Unhealthy	0 Unused	0 Initial	0 Draining
Total targets	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

Step 10: Test High Availability

Stop EC2-A. Refresh ALB DNS. Traffic should route to EC2-B.

Target group: Healthy: 1, Unused: 1

aws Search [Alt+S] Asia Pacific (Mumbai) Developer-Account

EC2 > Instances

Instances (1/2)

Find Instance by attribute or tag (case-sensitive) All states

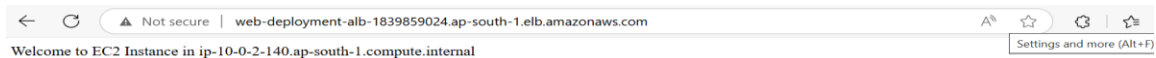
Name	Instance ID	Instance state	Instance type	Status check	Alarm
web-deployment-server-2	i-0ce4f75866ada822a	Running	t2.micro	2/2 checks passed	View
web-deployment-server-1	i-073790655b83f4e7c	Stopped	t2.micro	-	View

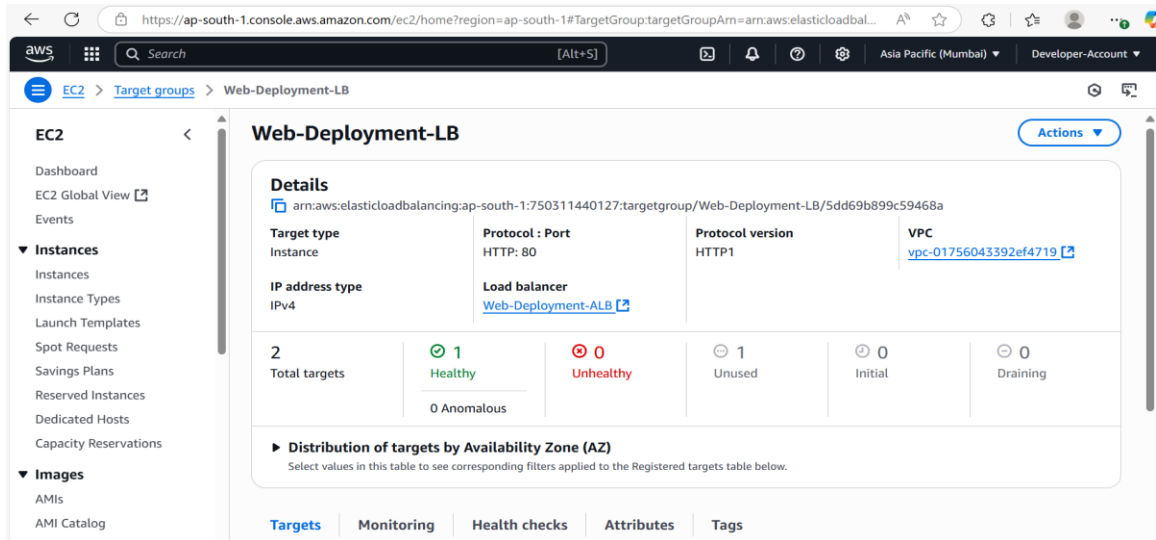
i-073790655b83f4e7c (web-deployment-server-1)

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

Instance summary

Instance ID i-073790655b83f4e7c	Public IPv4 address -	Private IPv4 addresses 10.0.1.84
IPv6 address -	Instance state Stopped	Public DNS -
Hostname type IP name: ip-10-0-1-84.ap-south-	Private IP DNS name (IPv4 only) ip-10-0-1-84.ap-south-1.compute.internal	





Routing Table Summary:

Destination: 0.0.0.0/0 → Target: Internet Gateway

Final Output Verification:

Check	Result
ALB URL reachable	Yes
EC2 servers show web page	Yes
Stopping one EC2 works	Traffic goes to other EC2
Target group health status	One Healthy, One Unused

Conclusion:

Successfully deployed a highly available web application in AWS using:

- Two EC2s in different AZs
- Application Load Balancer
- Proper routing tables and health checks

This setup ensures traffic is served even if one instance fails.