

Auto Scaling Web Infrastructure with Elastic Load Balancer

Objective:

To deploy a scalable and highly available web server infrastructure by creating a custom VPC with two public subnets across different Availability Zones, launching two EC2 instances (one in each subnet), and configuring an Application Load Balancer (ELB) within the same VPC. The setup ensures high availability and fault tolerance, with Auto Scaling triggered based on CPU utilization, using the same Load Balancer for traffic distribution and instance health monitoring.

Launch VPC:

The screenshot shows the AWS Management Console interface for the 'Your VPCs' page. The top navigation bar includes the AWS logo, a search bar, and the current region (United States (Oregon)) and user (Aathira V Sajive). The left sidebar shows the 'VPC dashboard' with a 'Filter by VPC' dropdown and a list of VPC-related services. The main content area displays a table of VPCs with the following columns: Name, VPC ID, State, Block Public Access, IPv4 CIDR, and IPv6 CIDR. Two VPCs are listed: 'ProjectVPC' (vpc-036308989081c6ba5) and 'vpc-04ee060841da01271'. Below the table, detailed information for the selected VPC is shown, including its ID, state, DNS resolution, main network ACL, IPv6 CIDR, block public access, DHCP option set, IPv4 CIDR, route 53 resolver DNS firewall rule groups, DNS hostnames, main route table, IPv6 pool, and owner ID.

Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR
ProjectVPC	vpc-036308989081c6ba5	Available	Off	10.0.0.0/16	-
-	vpc-04ee060841da01271	Available	Off	172.31.0.0/16	-

VPC ID
vpc-036308989081c6ba5

DNS resolution
Enabled

Main network ACL
acl-097375031cf5495b4

IPv6 CIDR (Network border group)
-

State
Available

Tenancy
default

Default VPC
No

Network Address Usage metrics
Disabled

Block Public Access
Off

DHCP option set
dopt-007470053b86aa83b

IPv4 CIDR
10.0.0.0/16

Route 53 Resolver DNS Firewall rule groups
-

DNS hostnames
Disabled

Main route table
rtb-05a8e248a4dedce11

IPv6 pool
-

Owner ID
615299759622

- Log in to AWS Console
- Choose the Region
- Navigate to VPC Dashboard > Your VPCs > Create VPC
- Choose:
 - Name tag: **ProjectVPC**
 - IPv4 CIDR block: **10.0.0.0/16**
- Click Create VPC

Create Subnets

The screenshot shows the AWS Management Console interface for the 'Subnets' page. The top navigation bar includes the AWS logo, a search bar, and the current region 'United States (Oregon)'. The left sidebar shows the 'VPC dashboard' with a 'Subnets' link highlighted. The main content area displays a table of subnets for the 'ProjectVPC' VPC. The table has columns for Name, Subnet ID, State, VPC, Block size, and IPv4 CIDR. Two subnets are visible: 'PublicSubnet2' and 'PublicSubnet1'. The 'PublicSubnet2' row shows a Subnet ID of 'subnet-0cd17c723c98b44b7', State of 'Available', VPC of 'vpc-036308989081c6ba5 | ProjectVPC', Block size of 'Off', and IPv4 CIDR of '10.0.99.0/24'. The 'PublicSubnet1' row shows a Subnet ID of 'subnet-06e1cd5d58a6a9fbc', State of 'Available', VPC of 'vpc-036308989081c6ba5 | ProjectVPC', Block size of 'Off', and IPv4 CIDR of '10.0.100.0/24'. The bottom of the page shows a summary of the subnets: 'Subnets: subnet-0cd17c723c98b44b7, subnet-06e1cd5d58a6a9fbc'.

Name	Subnet ID	State	VPC	Block size	IPv4 CIDR
PublicSubnet2	subnet-0cd17c723c98b44b7	Available	vpc-036308989081c6ba5 ProjectVPC	Off	10.0.99.0/24
-	subnet-0208a69e5c1bf43ea	Available	vpc-04ee060841da01271	Off	172.31.48.0/20
-	subnet-0478eb70ae90d1173	Available	vpc-04ee060841da01271	Off	172.31.32.0/20
-	subnet-059bfa6b5c69ed836	Available	vpc-04ee060841da01271	Off	172.31.0.0/20
PublicSubnet1	subnet-06e1cd5d58a6a9fbc	Available	vpc-036308989081c6ba5 ProjectVPC	Off	10.0.100.0/24
-	subnet-037c75bf81cda14ee	Available	vpc-04ee060841da01271	Off	172.31.16.0/20

- Go to VPC Dashboard > Subnets > Create subnet
- Select your ProjectVPC
- Name:
 - **PublicSubnet1**
 - **PublicSubnet2**
- Availability Zones:
 - Choose different AZs: **us-west-2a** and **us-west-2b**

- IPv4 CIDR blocks:
 - **10.0.100.0/24** → for Public Subnet 1
 - **10.0.99.0/24** → for Public Subnet 2
- Launch both Subnets

Create Internet Gateway

The screenshot shows the AWS Management Console interface for Internet Gateways. On the left, the 'VPC dashboard' sidebar is visible with options like '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', and 'Route servers'. The main content area is titled 'Internet gateways (1/2)' and contains a table with two entries:

Name	Internet gateway ID	State	VPC ID	Owner
-	igw-0a4727ec93446ea89	Attached	vpc-04ee060841da01271	615299759622
<input checked="" type="checkbox"/> ProjectIGW	igw-048684532787a5170	Attached	vpc-036308989081c6ba5 ProjectVPC	615299759622

Below the table, the details for the selected gateway 'igw-048684532787a5170 / ProjectIGW' are shown. The 'Details' tab is active, displaying the following information:

Internet gateway ID	State	VPC ID	Owner
igw-048684532787a5170	Attached	vpc-036308989081c6ba5 ProjectVPC	615299759622

- Go to VPC Dashboard > Internet Gateways > Create internet gateway
- Name: **ProjectIGW**
- Click Create Internet Gateway
- Select ProjectIGW, then choose Actions > Attach to VPC and attach it to your ProjectVPC.

Create Route Tables

Route table rtb-0ebc52576217de612 | PublicRouteTable was created successfully.

Route tables (1/3) Info

Find resources by attribute or tag

	Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
<input type="checkbox"/>	-	rtb-0a5dc26feb39c942e	-	-	Yes	vpc-04ee060841da01271
<input checked="" type="checkbox"/>	PublicRouteTable	rtb-0ebc52576217de612	-	-	No	vpc-036308989081c6ba5
<input type="checkbox"/>	-	rtb-05a8e248a4dedce11	-	-	Yes	vpc-036308989081c6ba5

Details

Route table ID rtb-0ebc52576217de612	Main No	Explicit subnet associations -	Edge associations -
VPC vpc-036308989081c6ba5 ProjectVPC	Owner ID 615299759622		

- Go to VPC Dashboard > Route Tables > Create route table
- Name tag: **PublicRouteTable**
- VPC: Choose your VPC
- Click Create route table

Add Route to the Internet Gateway

Edit routes

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

Add route

Use: "igw-048684532787a5170"

igw-048684532787a5170 (ProjectIGW)

Cancel Preview Save changes

- Select the **PublicRouteTable**
- Go to the "Routes" tab > click "Edit routes"
- Click Add route:
 - Destination: **0.0.0.0/0**
 - Target: Select your ProjectIGW
- Click Save changes

Associate Both Public Subnets

Edit subnet associations
Change which subnets are associated with this route table.

Available subnets (2/2)

<input checked="" type="checkbox"/>	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input checked="" type="checkbox"/>	PublicSubnet1	subnet-06e1cd5d58a6a9fbe	10.0.100.0/24	-	Main (rtb-05a8e248a4dedce11)
<input checked="" type="checkbox"/>	PublicSubnet2	subnet-0cd17c723c98b44b7	10.0.99.0/24	-	Main (rtb-05a8e248a4dedce11)

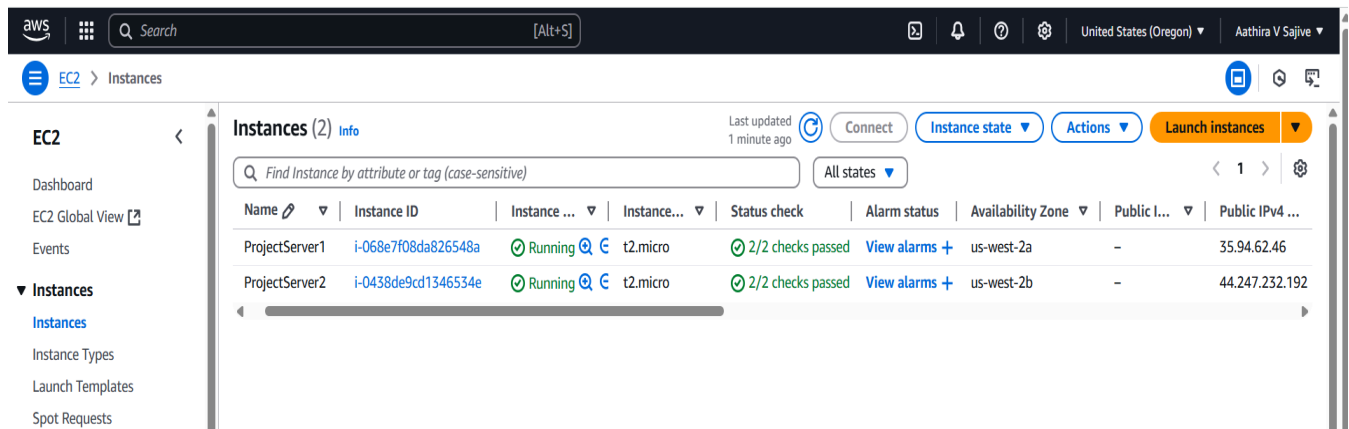
Selected subnets

subnet-06e1cd5d58a6a9fbe / PublicSubnet1 subnet-0cd17c723c98b44b7 / PublicSubnet2

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

- In the same RouteTable > Go to the "Subnet associations" tab
- Click Edit subnet associations
- Select both:
 - **PublicSubnet1** → **10.0.100.0/24**
 - **PublicSubnet2** → **10.0.99.0/24**
- Save

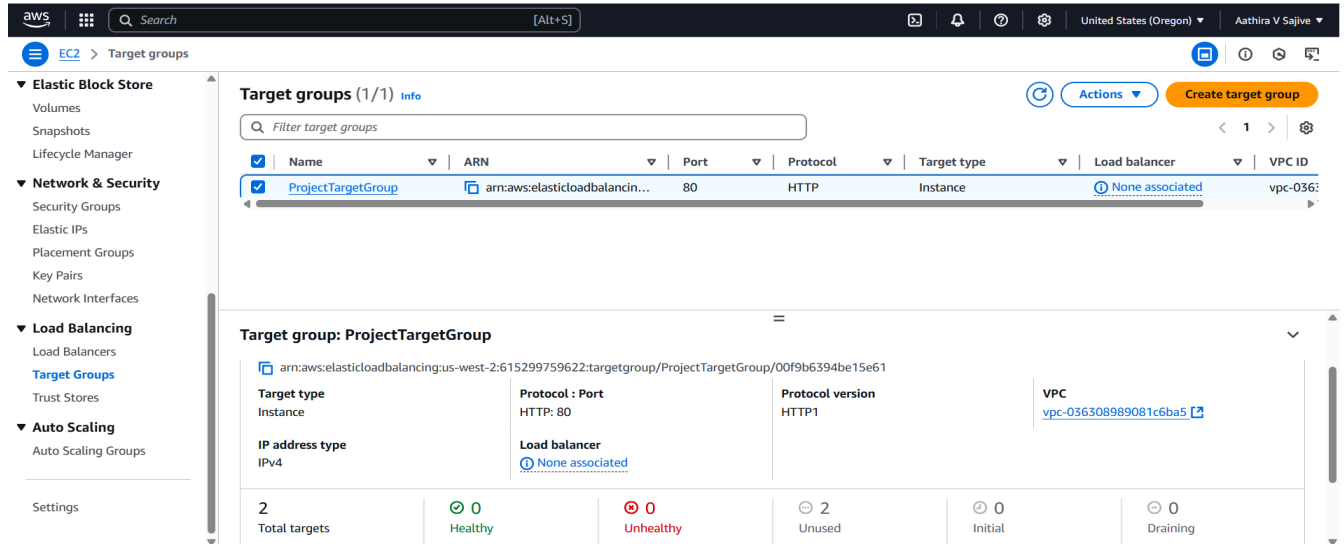
Launch EC2 Instance



- Go to EC2 Dashboard > Instances > Launch Instance
- Name:
 - ProjectServer1
 - ProjectServer2
- AMI: Choose Amazon Linux AMI
- Instance type: t2.micro
- Key pair: Select or create a new key
- Network Settings:
 - VPC: your ProjectVPC
 - Subnet:
 - PublicSubnet1
 - PublicSubnet2
 - Auto-assign public IP: Enabled
- Security Group:
 - Create or select a group that allows:
 - SSH (port 22)
 - HTTP (port 80)
- Add User Data Script in: Advanced > User data field:
- Launch both Instances

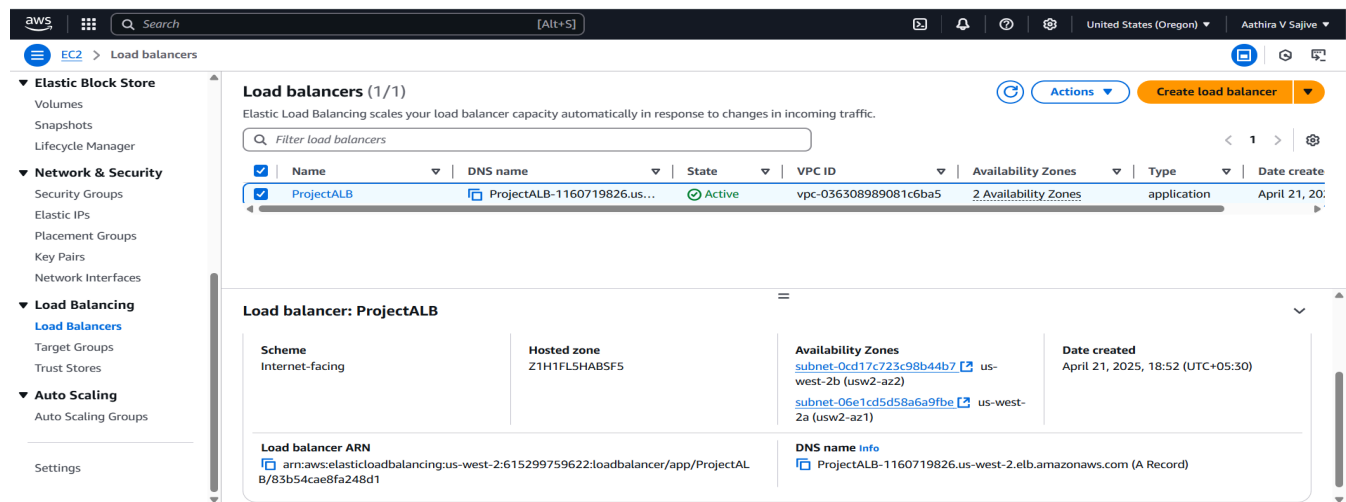
Configure Elastic Load Balancer (ELB)

Create a Target Group



- Go to EC2 Dashboard > Target Group > Create a new target group
- Target type: Instance
- Target group name: ProjectTargetGroup
- Protocol: HTTP
- Port: 80
- VPC:Your ProjectVPC
- Create a target group.
- Register your both EC2 instances into the target group

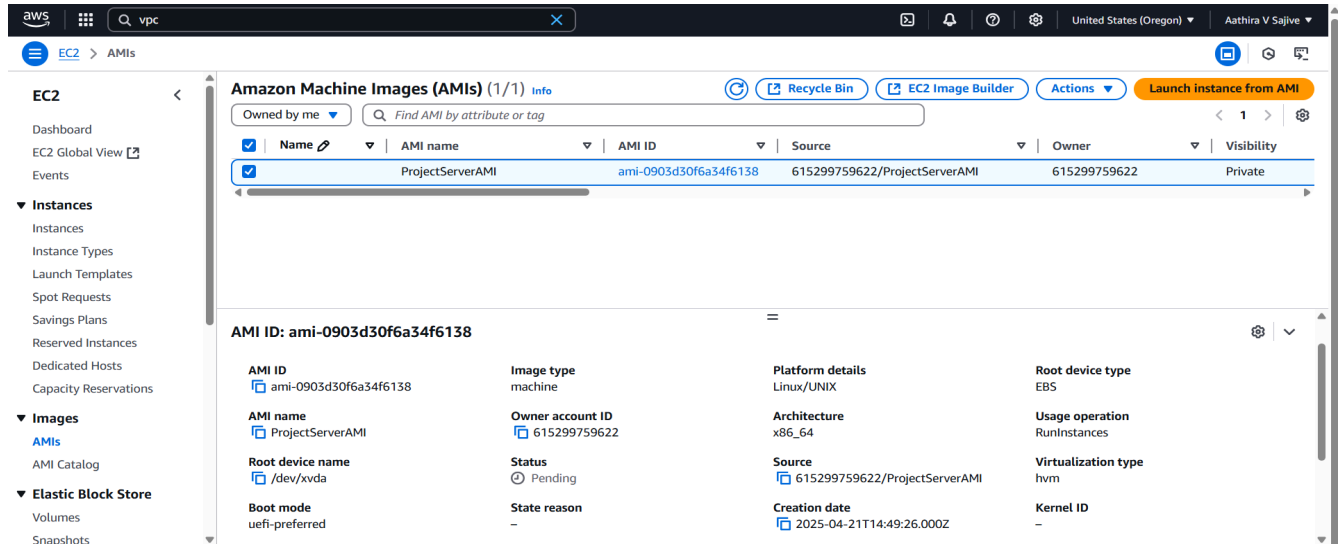
Create an Application Load Balancer



- Go to EC2 Dashboard > Load Balancers
- Click Create Load Balancer > Choose Application Load Balancer
- Set:
 - Name: **ProjectALB**
- Network mapping:
 - VPC: your ProjectVPC
 - Availability Zones:
 - Choose different AZs: **us-west-2a** and **us-west-2b** and attach:
 - **PublicSubnet1**
 - **PublicSubnet2**
- Security Group:
 - Create or choose one that allows: HTTP (port 80)
- Listeners:
 - Listener on port 80 → forward to your ProjectTargetGroup
- Create Load Balancer

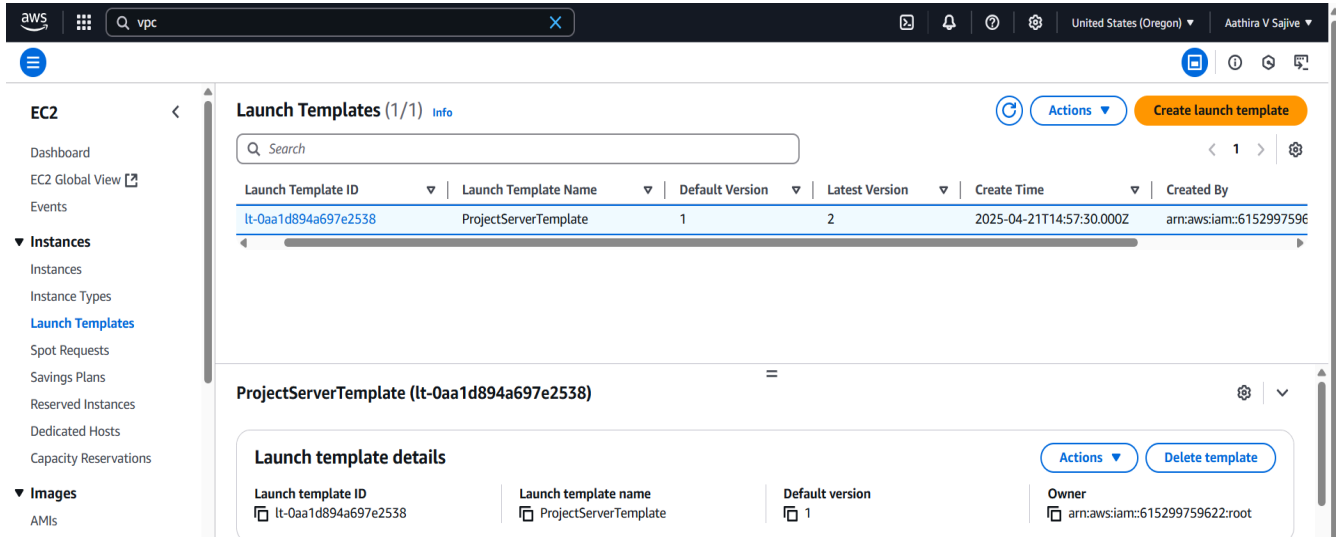
Configure Auto Scaling Group with Elastic Load Balancer

Create an AMI from Instance



- Go to EC2 Dashboard > Instances
- Select your configured instance
- Click Actions > Image > Create image
- Name: **ProjectServerAMI**
- Click Create image

Create Launch Template



Launch Templates (1/1) Info

Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time	Created By
lt-0aa1d894a697e2538	ProjectServerTemplate	1	2	2025-04-21T14:57:30.000Z	arn:aws:iam::615299759622:root

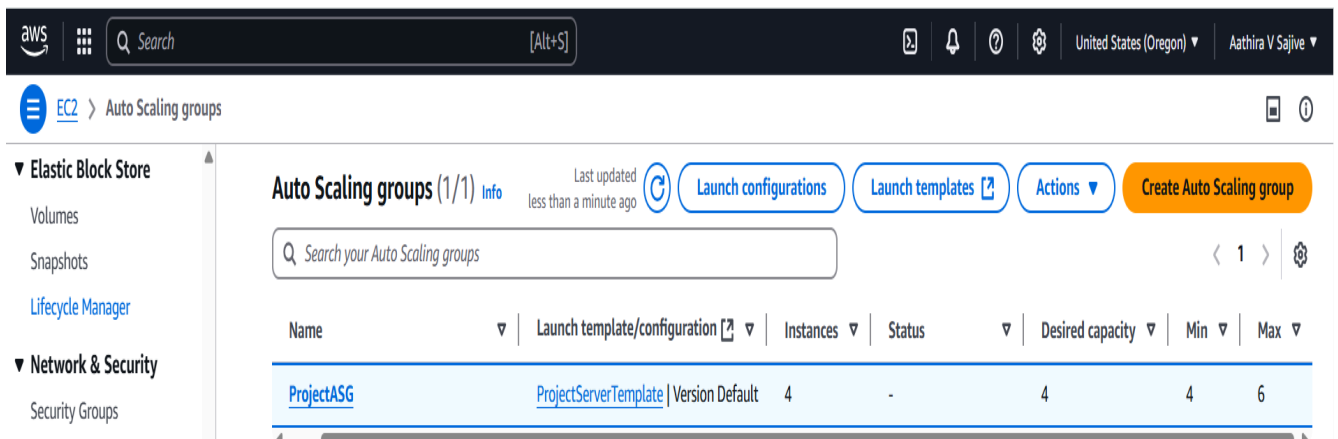
ProjectServerTemplate (lt-0aa1d894a697e2538)

Launch template details

Launch template ID	Launch template name	Default version	Owner
lt-0aa1d894a697e2538	ProjectServerTemplate	1	arn:aws:iam::615299759622:root

- Go to EC2 Dashboard > Launch Templates > Create launch template
- Name: **ProjectServerTemplate**
- AMI: Select your AMI: ProjectServerAMI
- Instance type: t2.micro
- Key pair: Choose or create one
- Click Create launch template

Create Auto Scaling Group



Auto Scaling groups (1/1) Info

Last updated less than a minute ago

Launch configurations **Launch templates** **Actions** **Create Auto Scaling group**

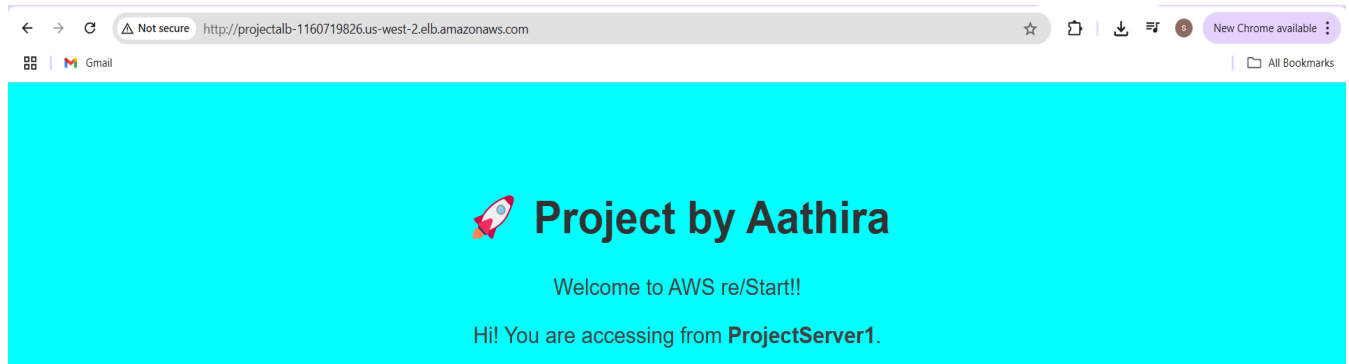
Search your Auto Scaling groups

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max
ProjectASG	ProjectServerTemplate Version Default	4	-	4	4	6

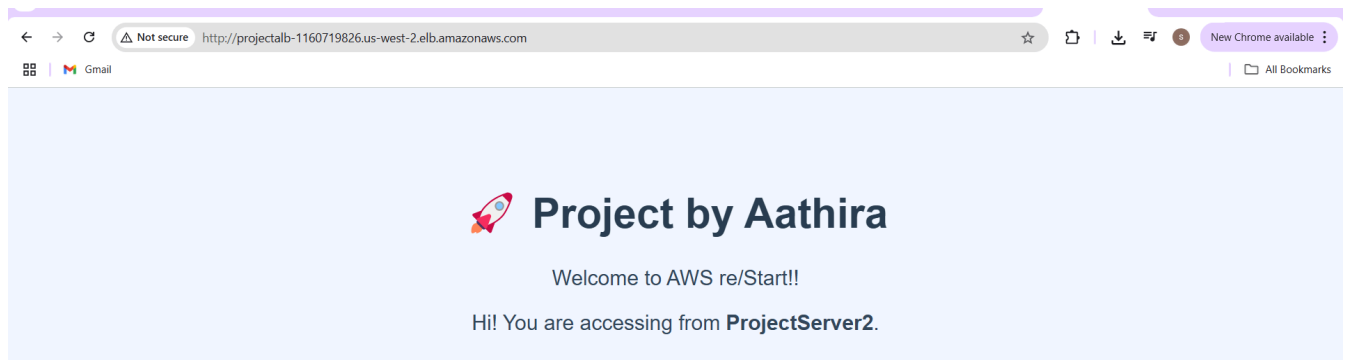
- Go to EC2 Dashboard > Auto Scaling Groups > Create Auto Scaling group
- Name: ProjectASG
- Launch template: select ProjectServerTemplate
- Choose Next
- Choose instance launch options page, in the Network section
- VPC: your ProjectVPC
- Availability Zones and subnets:
 - PublicSubnet1
 - PublicSubnet2
- Choose Next
- Attach to an existing load balancer
 - Choose load balancer target groups: ProjectTargetGroup
- Configure group size and scaling policies
 - Desired capacity: 4
 - Minimum capacity: 4
 - Maximum capacity: 6
 - Configure Health Checks
 - Scaling policies: Target tracking scaling policy
 - Metric type: Average CPU utilization.
 - Target value: 50
 - Choose Next
 - Add tag
 - Key: Name
 - Value: ProjectInstance
 - Choose Next
 - Click Create Auto Scaling Group

Verify Load Balancing and Auto Scaling

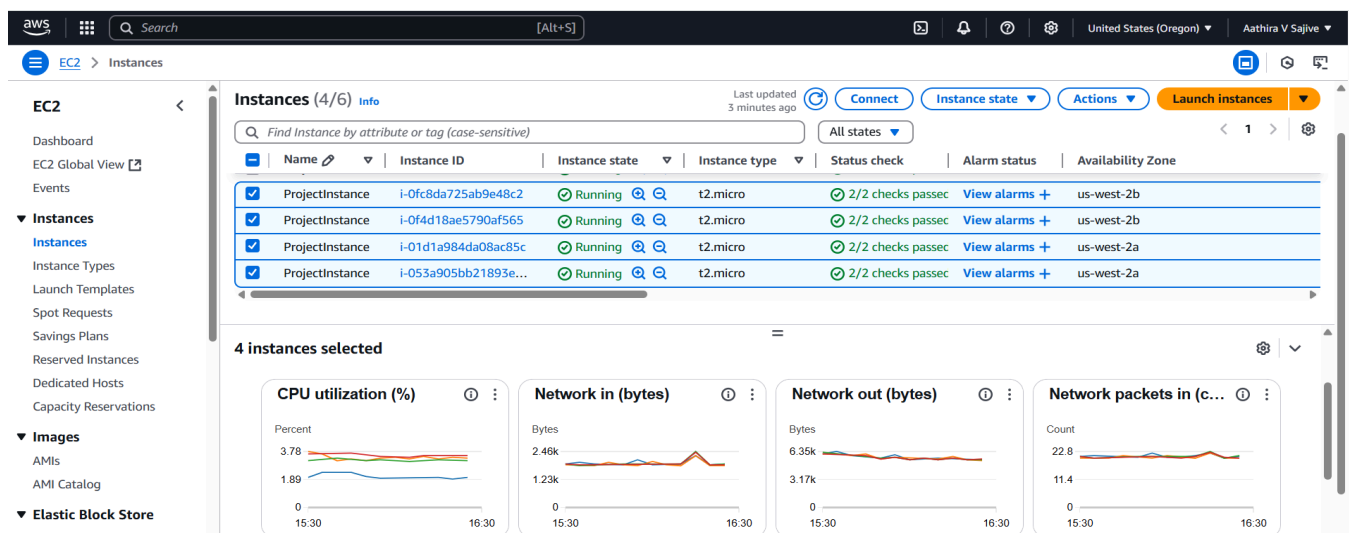
Load Balancer routed traffic to ProjectServer1



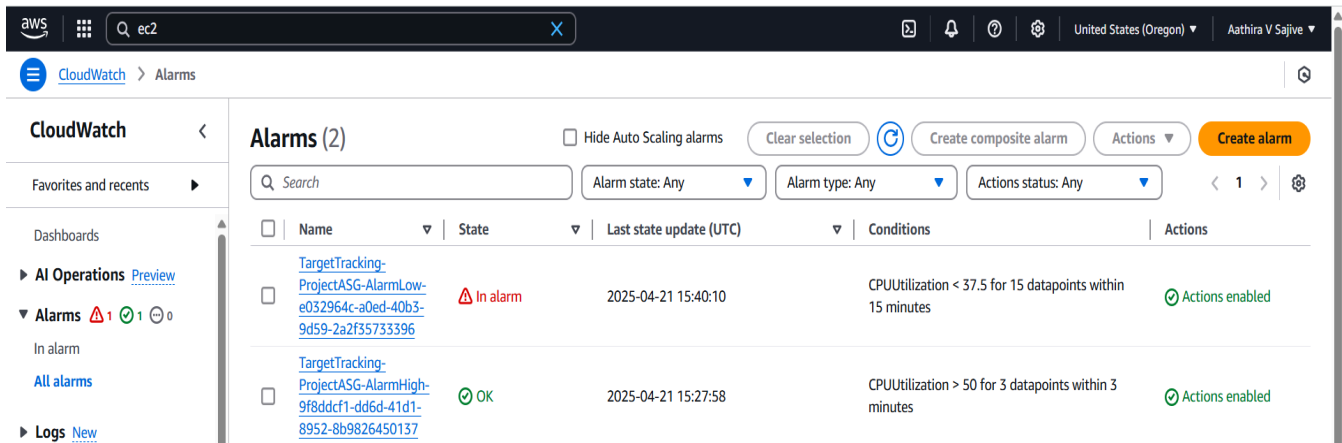
Load Balancer routed traffic to ProjectServer2



Auto Scaling Group Instances and Target Group Registration



CloudWatch Alarms for Auto Scaling Policies



CPU Utilization Graph (CloudWatch)

