

Creating a Scalable and Highly Available Environment for the Café

TASK 1 — Update Network for Multi-AZ High Availability

Task 1.1 — Create NAT Gateway in Public Subnet 2

Steps

1. Open **VPC Console**.
2. Left menu → **NAT Gateways**.
3. Click **Create NAT Gateway**.
4. Configure:
 - **Subnet:** Public Subnet 2
 - **Elastic IP:** Allocate Elastic IP
 - **Name:** NAT-GW-AZ2
5. Click **Create NAT Gateway**.

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.
NAT-GW-AZ2
The name can be up to 256 characters long.

Subnet
Select a subnet in which to create the NAT gateway.
subnet-00cee170af2532bc4 (Public Subnet 2) ▼

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-0f77ad31df161d6d3 ▼ [Allocate Elastic IP](#)

Update Private Subnet 2 Route Table

1. VPC console → **Route Tables**.
2. Find the route table for **Private Subnet 2**
3. Select it → **Routes** tab → **Edit routes**.
4. Add route:
 - **Destination:** 0.0.0.0/0
 - **Target:** NAT Gateway → NAT-GW-AZ2

5. Click **Save changes**.

Edit routes

Destination	Target	Status	Propagated	Route Origin
10.0.0.0/16	local	Active	No	CreateRouteTable
<input type="text" value="0.0.0.0/0"/>	<input type="text" value="local"/>			
<input type="text" value="0.0.0.0/0"/>	NAT Gateway	Active	No	CreateRoute
	<input type="text" value="nat-00018f6648f2b7446"/>			

TASK 2 — Create a Launch Template

1. Open Launch Templates

EC2 Console → left menu → **Launch Templates** → **Create launch template**

2. Template Information

- Name: CafeWebServerTemplate

3. AMI

- Application & OS Images → **My AMIs**
- Select: **Cafe WebServer Image**

4. Instance Type

- Choose: **t2.micro**

5. Key Pair

- vokey

6. Network Settings

- Security Group: **CafeSG**

7. Tags

Add tag:

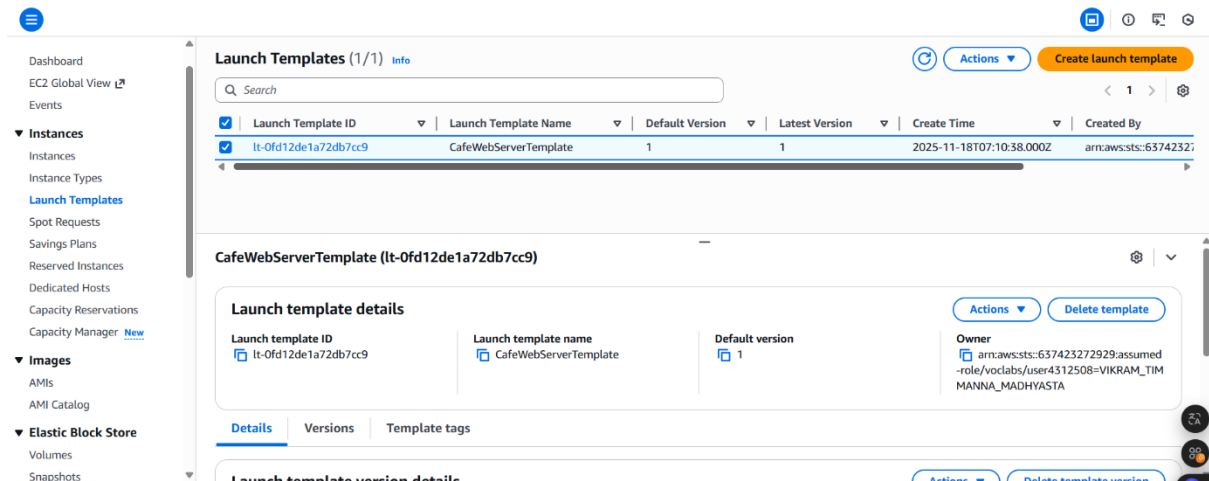
- **Key:** Name
- **Value:** webserver
- **Resource:** Instances

8. IAM Role

- Under Advanced details:
 - **IAM Instance Profile:** CafeRole

9. Create

Click **Create launch template**.



TASK 3 — Create Application Load Balancer

1. Open Load Balancer Console

EC2 → Load Balancers → **Create Load Balancer** → Application Load Balancer

2. ALB Configuration

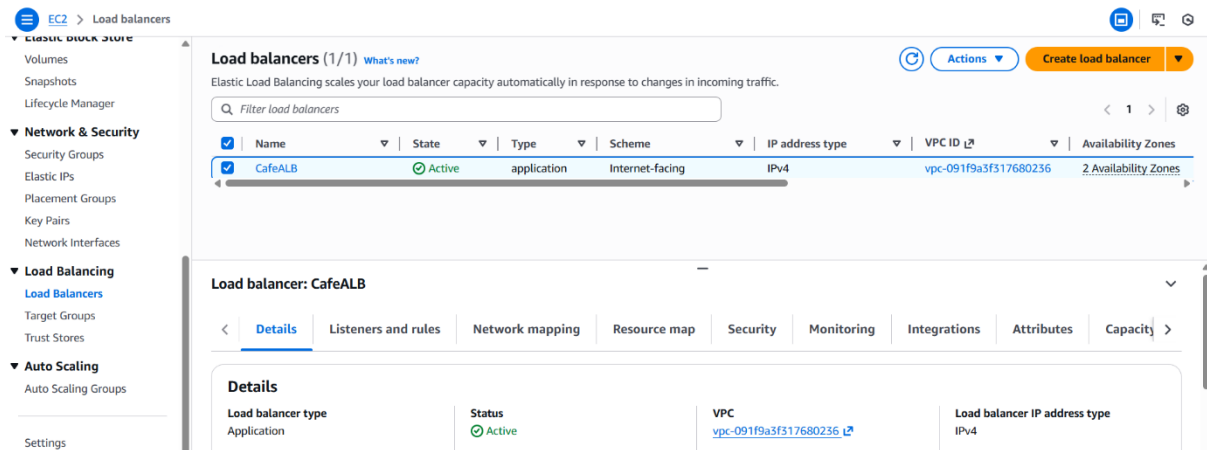
- Name: CafeALB
- Scheme: **Internet-facing**
- IP type: IPv4
- VPC: *lab VPC*
- Subnets:
 - Public Subnet 1
 - Public Subnet 2
- Select security group

3. Create Target Group

- Target type: **Instances**
- Name: CafeTargetGroup
- Protocol: **HTTP**
- Health check path: /cafe
- Click **Create target group**

4. Create

Create Load Balancer



TASK 4 — Create Auto Scaling Group

1. Open ASG Console

EC2 → Auto Scaling Groups → Create Auto Scaling group

2. Basic Details

- ASG name: CafeWebServerASG
- Launch template: CafeWebServerTemplate

3. Network

- VPC: *your lab VPC*
- Subnets (select both):
Private Subnet 1
Private Subnet 2

4. Group Size

- **Desired:** 2
- **Min:** 2
- **Max:** 6

5. Scaling Policy

- Select **Target tracking scaling policy**
 - Metric: **Average CPU Utilization**
 - Target: **25%**
 - Instance warmup: **60 sec**

6. Create ASG

Click **Create Auto Scaling group**.

EC2 > Auto Scaling groups

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

<input checked="" type="checkbox"/>	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones	
<input checked="" type="checkbox"/>	CafeWebServerASG	CafeWebServerTemplate Version Default	2	-	2	2	6	2 Availability Zones	Tu

Auto Scaling group: CafeWebServerASG

[Details](#) [Integrations](#) [Automatic scaling](#) [Instance management](#) [Instance refresh](#) [Activity](#) [Monitoring](#) [Tags - moved](#)

CafeWebServerASG Capacity overview [Edit](#)

arn:aws:autoscaling:us-east-1:637423272929:autoScalingGroup:cc5fc03b-b996-40b9-8630-139cd5d30f21:autoScalingGroupName/CafeWebServerASG

Desired capacity	Scaling limits (Min - Max)	Desired capacity type	Status
2	2 - 6	Units (number of instances)	-

Task 5 — Testing and Validation

5.1 Application Reachability Test

The ALB DNS name was used to verify that the café application was accessible:

<http://cafealb-641114072.us-east-1.elb.amazonaws.com/cafe/>

This confirmed:


- Instances were healthy
- Target group was functioning
- ALB routing was operational

Browser tabs: AWS Academy Cloud, Challenge (Café) lab, Systems Manager, Instances | EC2 | us-east-1, Auto Scaling group, Café, RouteTables | VPC, C...

Not secure cafealb-641114072.us-east-1.elb.amazonaws.com/cafe/


Café

Home About Us Contact Us Menu Order History




Our café offers an assortment of delicious and delectable pastries and coffees that will put a smile on your face. From cookies to croissants, tarts and cakes, each treat is especially prepared to excite your tastebuds and brighten your day!


Frank bakes a rich variety of cookies. Try them all!



Tea, Coffee, Lattes.



Our tarts are always a customer favorite!



Windows taskbar: Search, ENG IN, 13:09, 18-11-2025

5.2 Auto Scaling Stress Test

To validate autoscaling behavior, a stress test was performed using **AWS Systems Manager Session Manager**.

Commands to Run:

```
sudo amazon-linux-extras install epel
```

```
sudo yum install stress -y
```

```
stress --cpu 1 --timeout 600
```

This generated artificial CPU load.

The ASG detected CPU >25% and initiated **scale-out**, launching additional EC2 instances.

Monitoring the ASG dashboard confirmed:

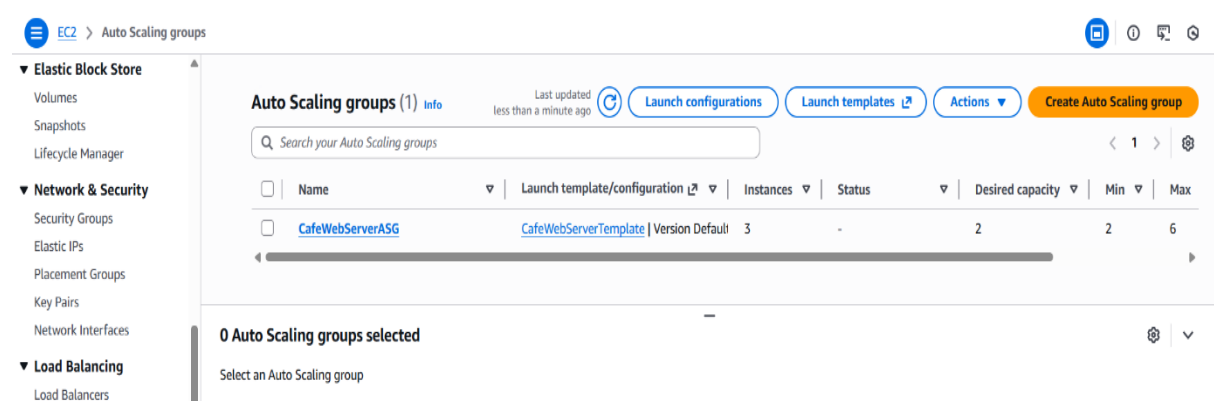
- Scaling policies were applied correctly
- New instances launched in both private subnets
- Load gradually balanced across instances

```
Installing:
stress                                x86_64                                1.0.4-16.el7                                epel                                39 k
Transaction Summary
-----
Install 1 Package

Total download size: 39 k
Installed size: 94 k
Downloading packages:
warning: /var/cache/yum/x86_64/2/epel/packages/stress-1.0.4-16.el7.x86_64.rpm: Header V3 RSA/SHA256 Signature, key ID 352c64e5: NOKEY
Public key for stress-1.0.4-16.el7.x86_64.rpm is not installed
stress-1.0.4-16.el7.x86_64.rpm                                | 39 kB  00:00:00
Retrieving key from file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7
Importing GPG key 0x352c64e5:
 Userid   : "Fedora EPEL (7) <epel@fedoraproject.org>"
Fingerprint: 91e9 7d7c 4a5e 96f1 7f3e 888f 6a2f aea2 352c 64e5
 Package  : epel-release-7-11.noarch (@amzn2extra-epel)
 From     : /etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : stress-1.0.4-16.el7.x86_64                                1/1
  Verifying  : stress-1.0.4-16.el7.x86_64                                1/1

Installed:
  stress.x86_64 0:1.0.4-16.el7

Complete!
sh-4.2$ stress --cpu 1 --timeout 600
stress: info: [3367] dispatching hogs: 1 cpu, 0 io, 0 vm, 0 hdd
```



Dashboard
EC2 Global View
Events
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations

Instances (4) Info

Last updated less than a minute ago

Connect Instance state Actions Launch instances

Find Instance by attribute or tag (case-sensitive)

All states

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
<input type="checkbox"/>	webserver	i-043869f27f8393563	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	-
<input type="checkbox"/>	webserver	i-0cdc809a4a99aa173	Shutting-d...	t2.micro	-	View alarms +	us-east-1a	-
<input type="checkbox"/>	webserver	i-0f94678aeba34169a	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	-
<input type="checkbox"/>	CafeWebAppS...	i-002df20ccf01c176d	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-54-196

EC2
Auto Scaling groups

Auto Scaling groups (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
<input type="checkbox"/>	CafeWebServerASG	CafeWebServerTemplate Version Default	6	Updating capacity...	3	2	6

EC2
Instances

Instances (9) Info

Last updated less than a minute ago

Connect Instance state Actions Launch instances

Find Instance by attribute or tag (case-sensitive)

All states

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
<input type="checkbox"/>	webserver	i-043869f27f8393563	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	-
<input type="checkbox"/>	webserver	i-0cdc809a4a99aa173	Terminated	t2.micro	-	View alarms +	us-east-1a	-
<input type="checkbox"/>	webserver	i-0f94678aeba34169a	Terminated	t2.micro	-	View alarms +	us-east-1b	-
<input type="checkbox"/>	webserver	i-0763a15c42aedd3f4	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	-
<input type="checkbox"/>	webserver	i-0b36ce2b70e23b2d2	Running	t2.micro	Initializing	View alarms +	us-east-1b	-
<input type="checkbox"/>	webserver	i-0ac8d48636c050975	Running	t2.micro	Initializing	View alarms +	us-east-1b	-
<input type="checkbox"/>	CafeWebAppS...	i-002df20ccf01c176d	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-54-196-1t
<input type="checkbox"/>	webserver	i-0fe6bb9080feb56fa	Running	t2.micro	Initializing	View alarms +	us-east-1a	-
<input type="checkbox"/>	webserver	i-078dd64e1b4e6ae0	Running	t2.micro	Initializing	View alarms +	us-east-1a	-

Conclusion

This lab successfully demonstrated the deployment of a highly available, fault tolerant, and scalable web application architecture using core AWS services. A multi-AZ design combined with NAT Gateways, private subnets, an ALB, and an Auto Scaling Group ensures robust performance and resiliency.