Cycle 08 AWS Homework - ASG + ALB Replication

Part A: Setup and Implementation

1. EC2 Instance Setup

- Launch two Ubuntu EC2 instances in the same region.
- Install Apache web server:

```
bash
CopyEdit
sudo apt update
sudo apt install apache2 -y
```

- Add custom content:
  -
  - On Server 1: update `/var/www/html/index.html`
  - On Server 2: update `/var/www/html/server2.html`
- Test in browser using public IPs.

2. Create AMIs

- Stop both instances.
- Create AMIs for each server from the EC2 console.
- Name them distinctly (e.g., `Server1-AMI`, `Server2-AMI`).

3. Launch Templates

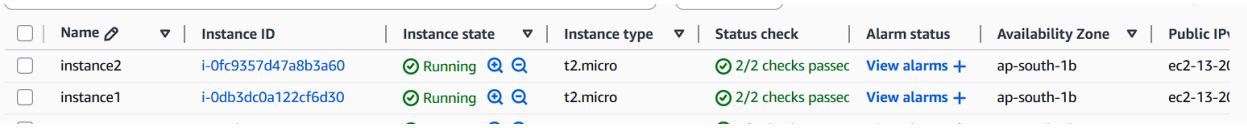
- Create two launch templates using each AMI.
- Select instance type (t2.micro for free tier).
- Ensure networking and security groups allow HTTP access.

4. Auto Scaling Groups (ASG)

- Create two ASGs, one for each launch template.
- Set minimum capacity = 1, maximum = 3.
- Use multiple Availability Zones for redundancy.
- Attach each ASG to its respective target group.

5. Application Load Balancer (ALB)

- Create an ALB (internet-facing).
- Add two target groups: one for each ASG.
- Configure listener rules:
  - Path `/` → Target Group 1
  - Path `/server2.html` → Target Group 2
- Access via ALB DNS to verify routing.

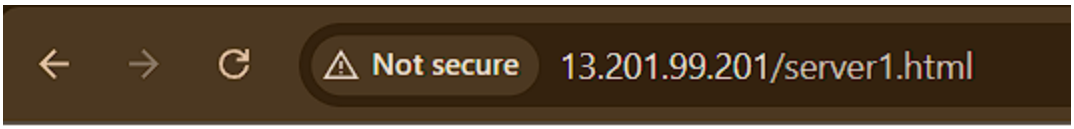


Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
instance2	i-0fc9357d47a8b3a60	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1b	ec2-13-2(
instance1	i-0db3dc0a122cf6d30	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1b	ec2-13-2(

```
root@ip-172-31-2-45:~# cat > /var/www/html/index.html
Welcome to home2
root@ip-172-31-2-45:~# cat > /var/www/html/server2.html
<h1>Welcome to server2<h1>
root@ip-172-31-2-45:~# history
    1  sudo apt-get update
    2  sudo apt-get install apache2
    3  sudo systemctl enable apache2
    4  sudo systemctl restart apache2
    5  sudo systemctl enable apache2
    6  sudo systemctl status apache2
    7  cat > /var/www/html/index.html
    8  cat > /var/www/html/server2.html
    9  history
```

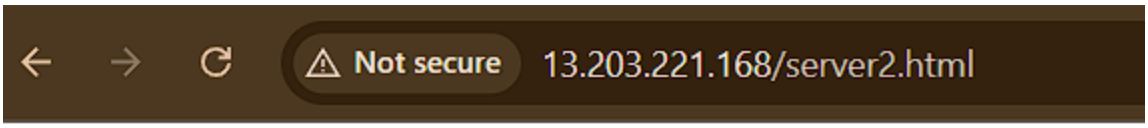
```
root@ip-172-31-11-46:~# cat > /var/www/html/index.html
Welcome to home1
root@ip-172-31-11-46:~# cat > /var/www/html/server1.html
<h1>Welcome to server1<h1>
root@ip-172-31-11-46:~# history
    1  sudo apt-get update
    2  sudo apt-get install apache2
    3  sudo systemctl enable apache2
    4  sudo systemctl restart apache2
    5  sudo systemctl status apache2
    6  cat > /var/www/html/index.html
    7  cat > /var/www/html/server1.html
    8  history
```

Welcome to home1

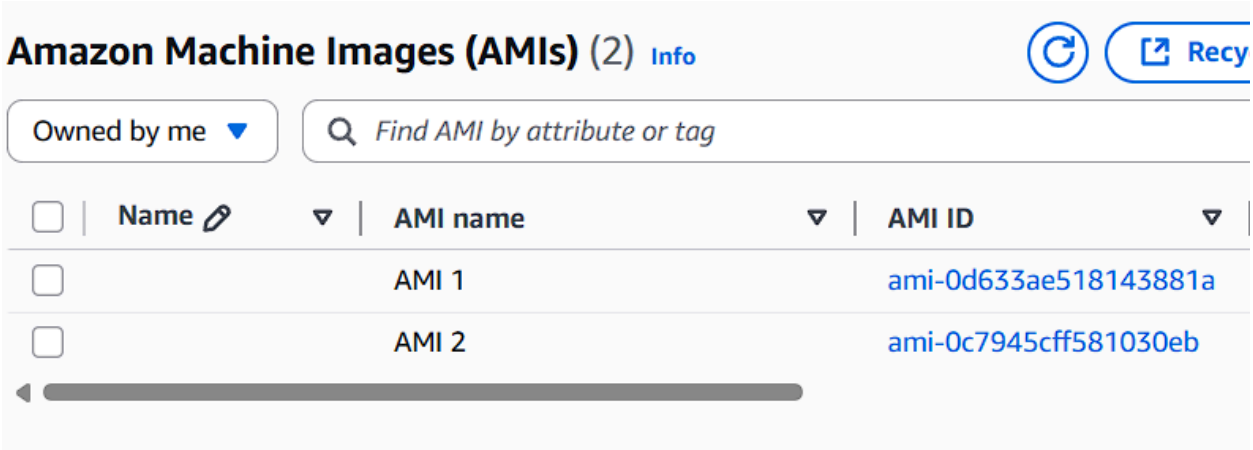


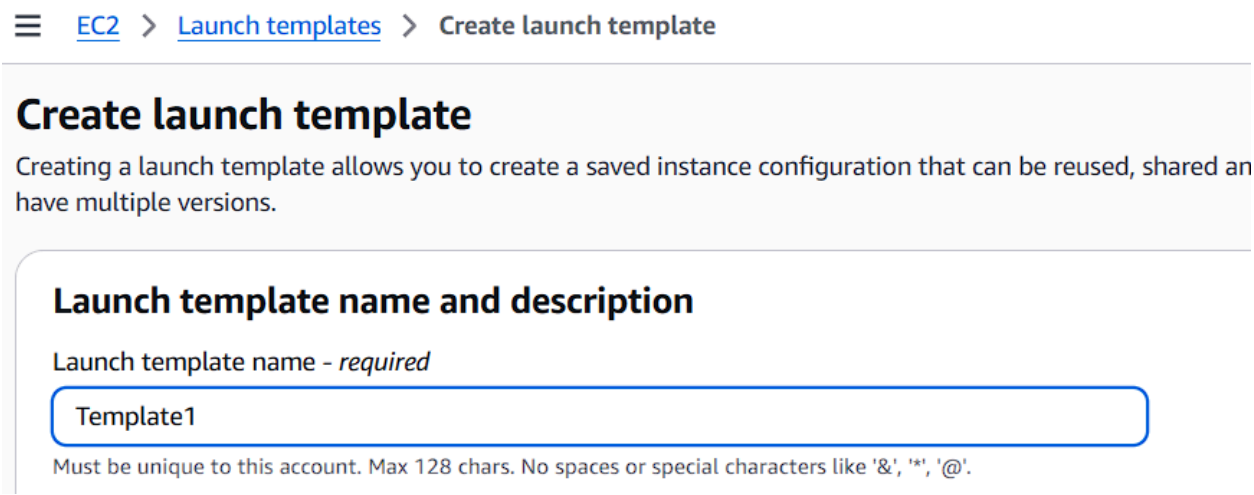
Welcome to server1

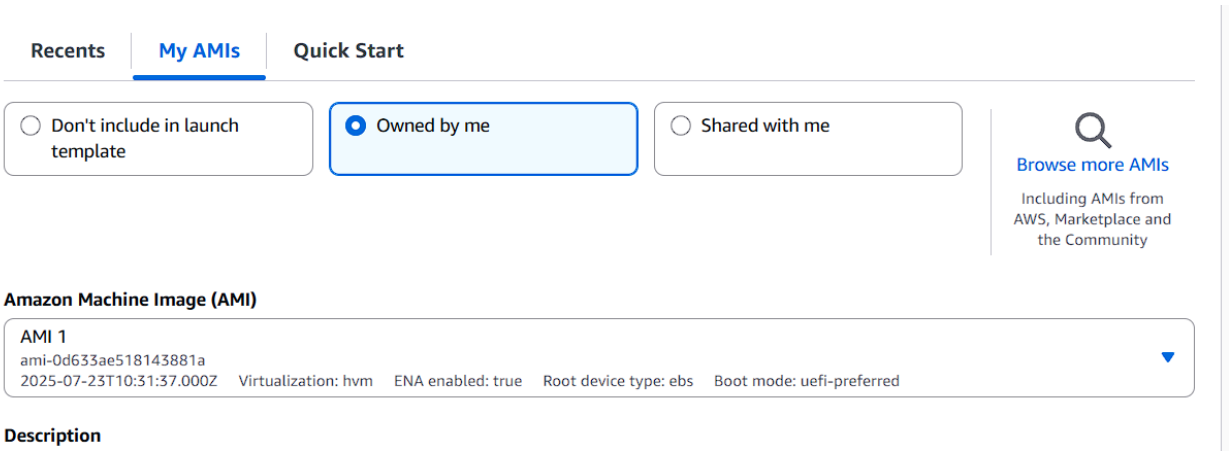
Welcome to home2



Welcome to server2







Create launch template

Creating a launch template allows you to create a saved instance configuration that can be reused, shared a
have multiple versions.

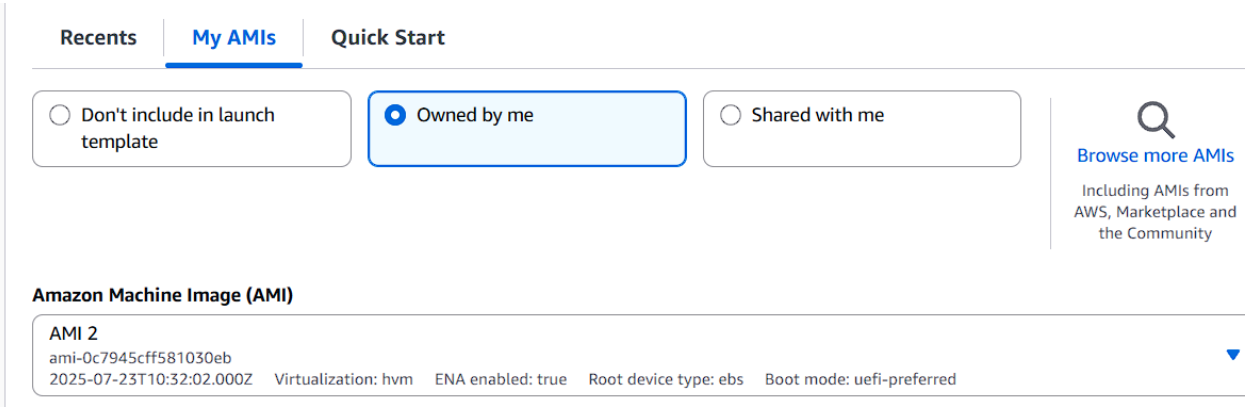
Launch template name and description

Launch template name - *required*

Template2

Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '*', '@'.

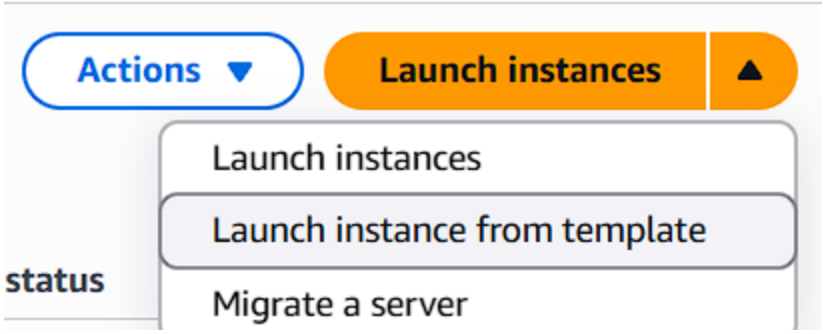
Template version description



Launch Templates (2) Info

Search

Launch Template ID	Launch Template Name
lt-0b2f1913c495aece4	Template1
lt-0c78191ef41ba6795	Template2



Create target group

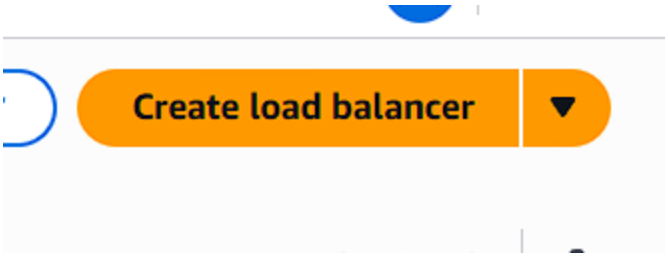
Register targets

This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target g

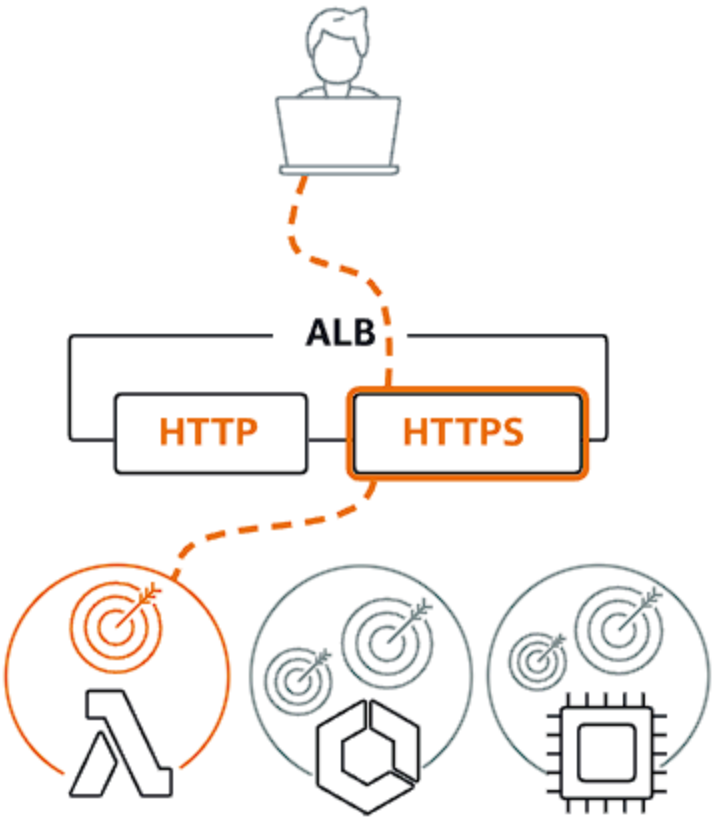
Available instances (1/2)

Filter instances

Instance ID	Name	State
i-0e1bec17f1dc4ce53	Instance2	Running
i-048b2b77172e5c69a	Instance 1	Running


Create load balancer

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

Basic configuration

Load balancer name

Name must be unique within your AWS account and can't be changed after the load balanc

LD

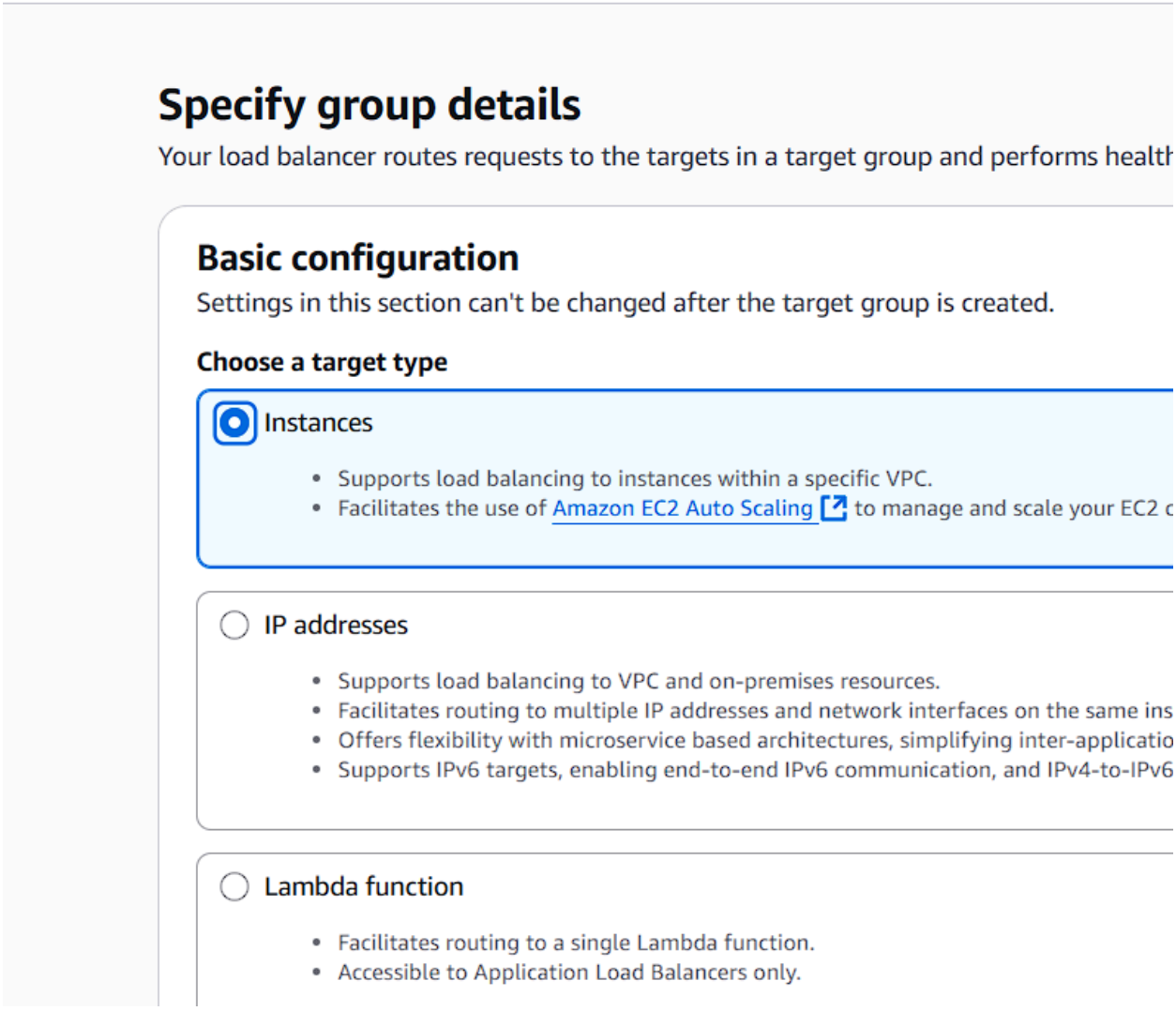
A maximum of 32 alphanumeric characters including hyphens are allowed, but the name m

Scheme | Info

Scheme can't be changed after the load balancer is created.

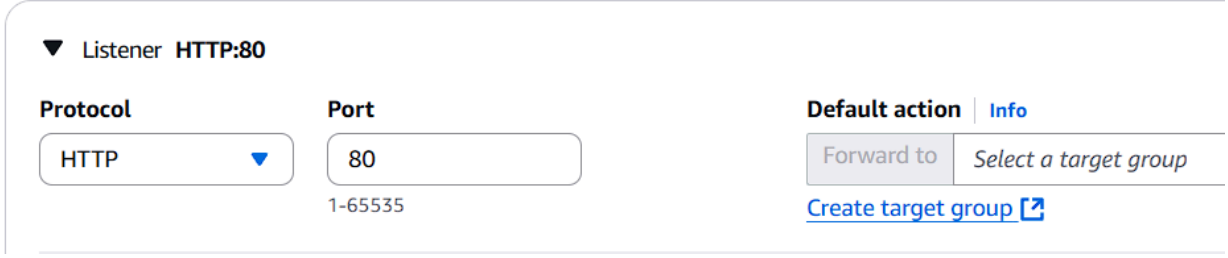
- Internet-facing
  - Serves internet-facing traffic.
  - Has public IP addresses.
  - DNS name resolves to public IPs.
  - Requires a public subnet.

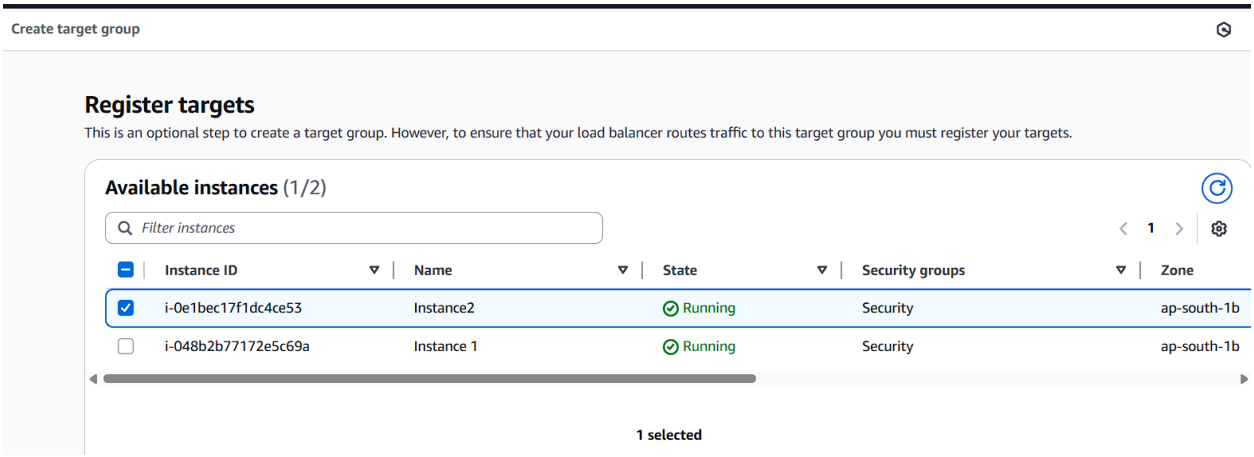
Load balancer IP address type | Info

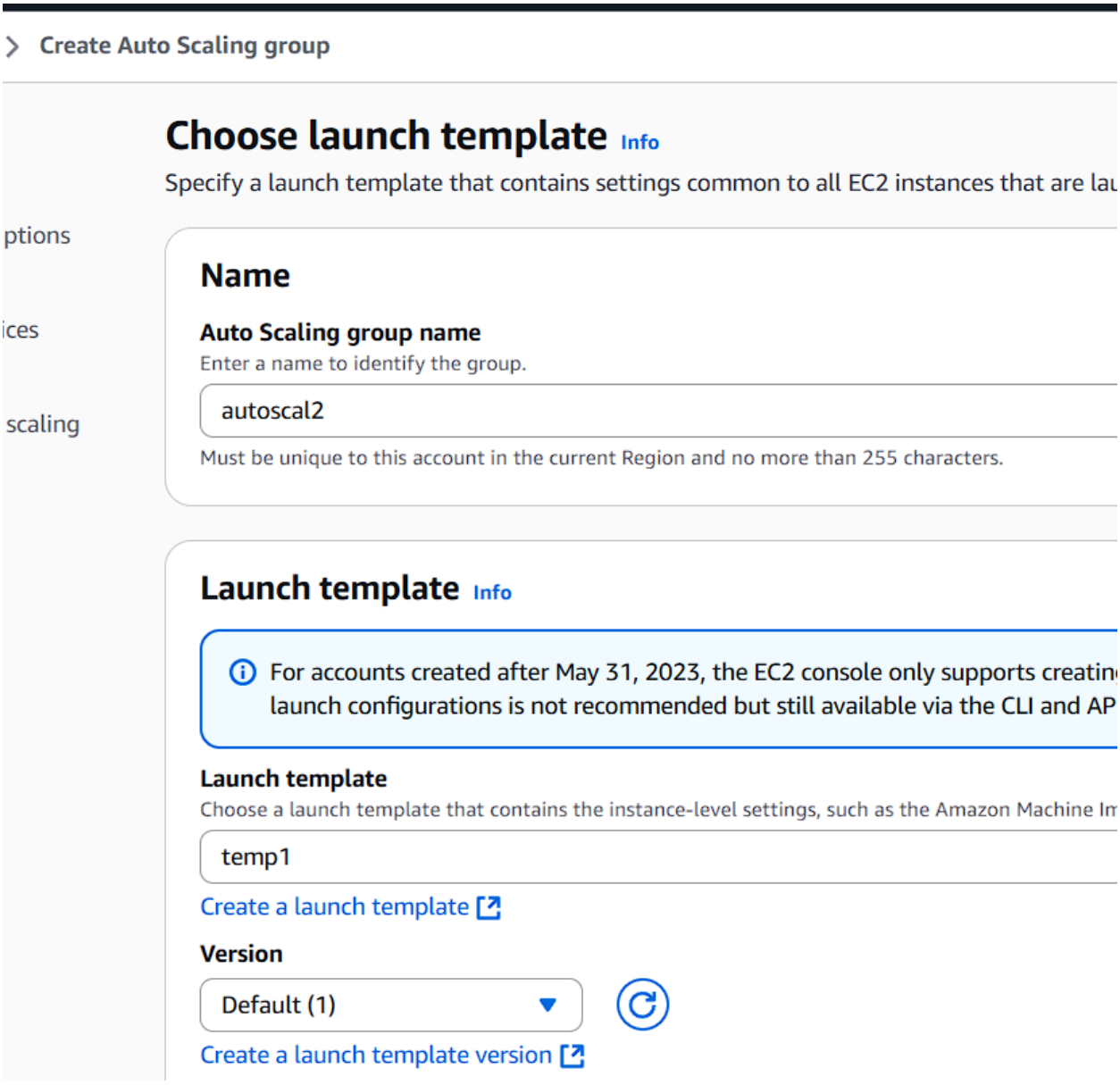


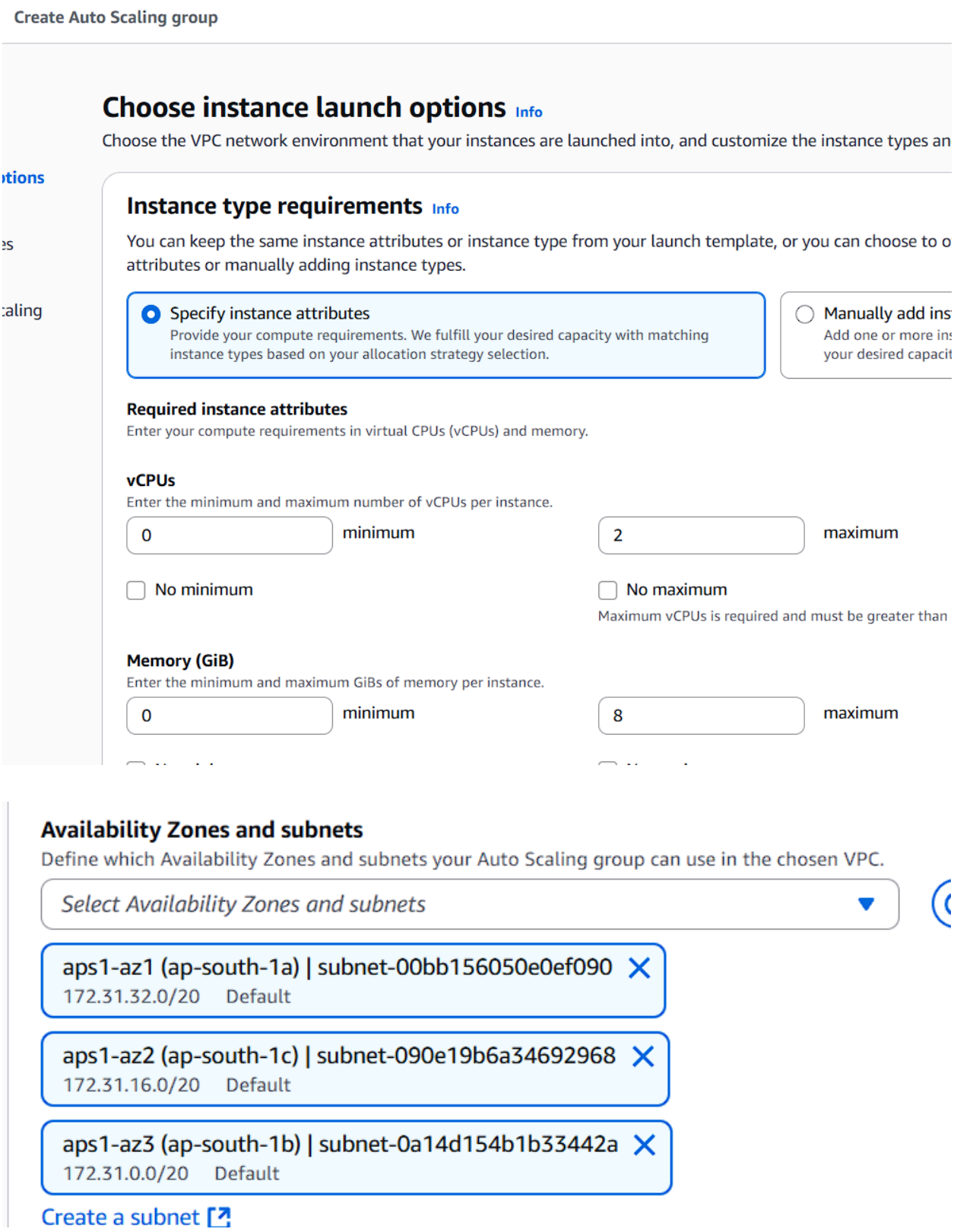
Listeners and routing Info

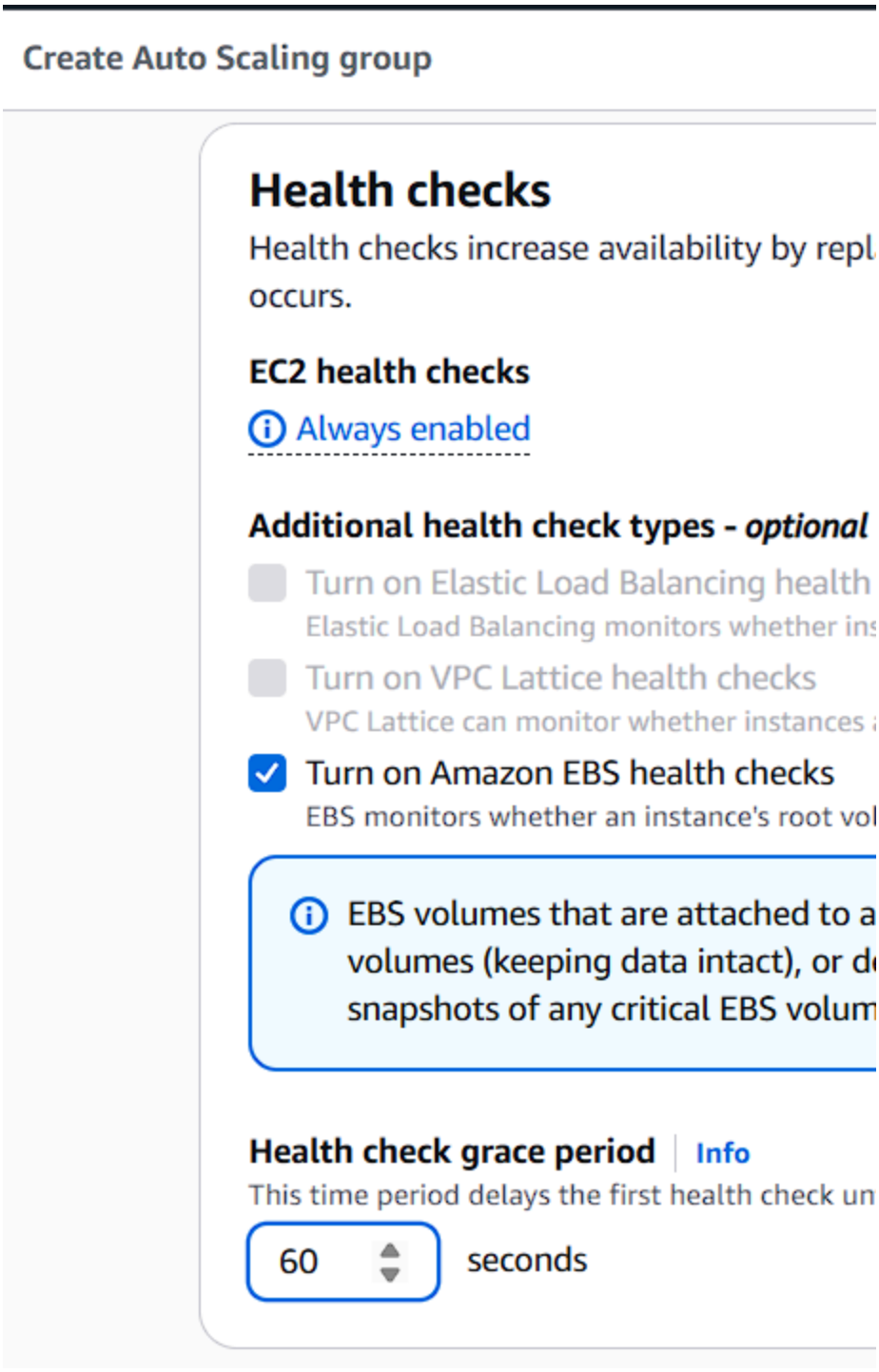
A listener is a process that checks for connection requests using the port and protocol you configure. The rules that y

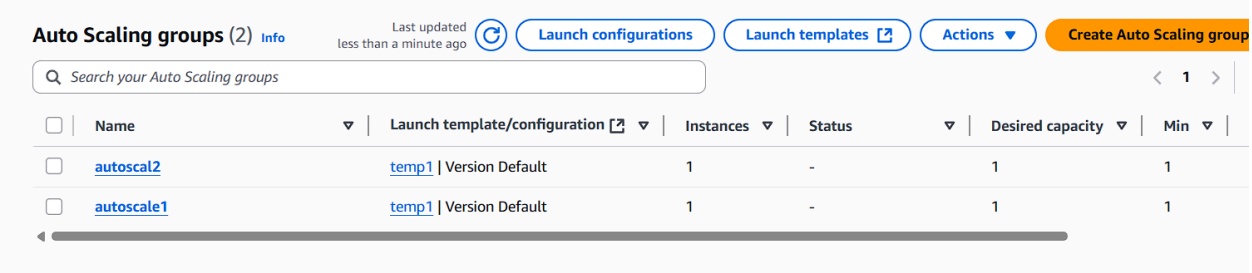


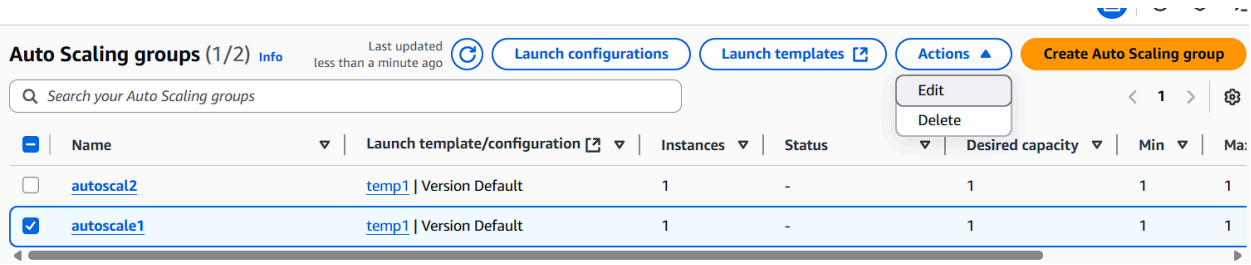

Create target group
Register targets
This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.
Available instances (1/2)
Filter instances
1
Instance ID
Name
State
Security groups
Zone
i-0e1bec17f1dc4ce53
Instance2
Running
Security
ap-south-1b
i-048b2b77172e5c69a
Instance 1
Running
Security
ap-south-1b
1 selected

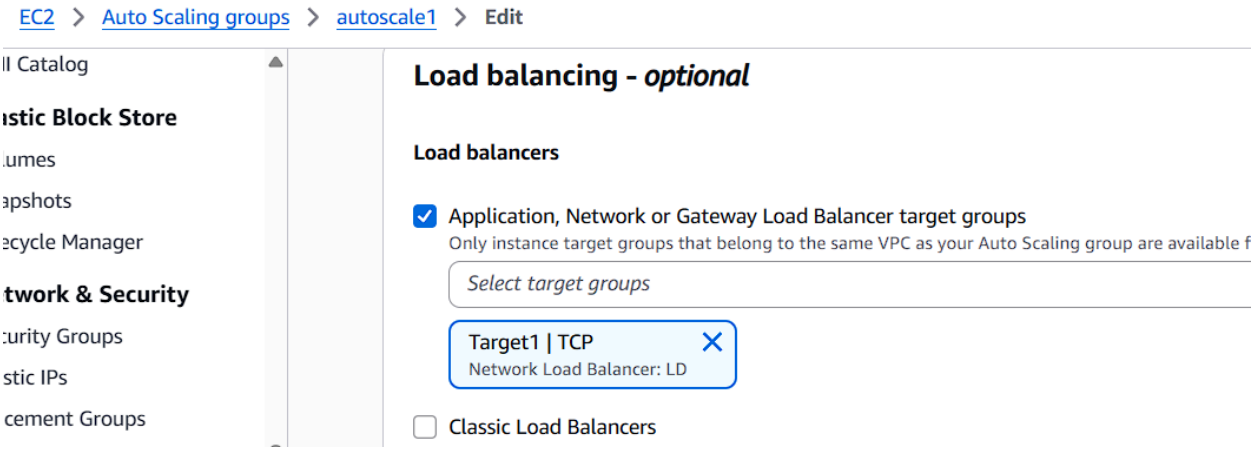

Create Auto Scaling group
Choose launch template Info
Specify a launch template that contains settings common to all EC2 instances that are lau
ptions
ices
scaling
Name
Auto Scaling group name
Enter a name to identify the group.
autoscal2
Must be unique to this account in the current Region and no more than 255 characters.
Launch template Info
For accounts created after May 31, 2023, the EC2 console only supports creatin
launch configurations is not recommended but still available via the CLI and AP
Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Im
temp1
Create a launch template
Version
Default (1)
Create a launch template version


Create Auto Scaling group
Choose instance launch options Info
Choose the VPC network environment that your instances are launched into, and customize the instance types an
Instance type requirements Info
You can keep the same instance attributes or instance type from your launch template, or you can choose to o
attributes or manually adding instance types.
Specify instance attributes
Provide your compute requirements. We fulfill your desired capacity with matching instance types based on your allocation strategy selection.
Manually add ins
Add one or more ins
your desired capacit
Required instance attributes
Enter your compute requirements in virtual CPUs (vCPUs) and memory.
vCPUs
Enter the minimum and maximum number of vCPUs per instance.
0
minimum
2
maximum
No minimum
No maximum
Maximum vCPUs is required and must be greater than
Memory (GiB)
Enter the minimum and maximum GiBs of memory per instance.
0
minimum
8
maximum
Availability Zones and subnets
Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.
Select Availability Zones and subnets
aps1-az1 (ap-south-1a) | subnet-00bb156050e0ef090
172.31.32.0/20 Default
aps1-az2 (ap-south-1c) | subnet-090e19b6a34692968
172.31.16.0/20 Default
aps1-az3 (ap-south-1b) | subnet-0a14d154b1b33442a
172.31.0.0/20 Default
Create a subnet


Create Auto Scaling group
Health checks
Health checks increase availability by repl
occurs.
EC2 health checks
Always enabled
Additional health check types - optional
Turn on Elastic Load Balancing health
Elastic Load Balancing monitors whether ins
Turn on VPC Lattice health checks
VPC Lattice can monitor whether instances
Turn on Amazon EBS health checks
EBS monitors whether an instance's root vol
EBS volumes that are attached to a
volumes (keeping data intact), or d
snapshots of any critical EBS volum
Health check grace period
Info
This time period delays the first health check un
60
seconds







Part B: Additional Topics

1. Advanced Auto Scaling Policies

- Add target tracking or step scaling:
  - CPU utilization > 70%
  - Request count > 100 per target
- Set cooldown period to avoid excessive scaling.

2. Cost Optimization

- Compare pricing of Linux vs Windows EC2 instances using AWS Pricing Calculator.
- Estimate costs for 1000 requests/hour with:
  - ALB
  - 2 EC2 instances
  - ASG usage
- Analyze cost differences and suggest optimization strategies.

3. High Availability Testing

- Terminate an EC2 instance in an ASG and verify auto-replacement.
- Confirm that instances are launched across two or more AZs.
- Validate traffic continuity through ALB during failover.

4. Security Enhancements

- Create a custom security group:
  - Allow HTTP (80), HTTPS (443)
  - Allow SSH only from your IP
- Use AWS Certificate Manager (ACM):
  - Request a public certificate
  - Attach it to the ALB to enable HTTPS

5. Infrastructure as Code (IaC)

- Use Terraform or AWS CDK to automate the setup:
  - Define VPC, ALB, ASGs, launch templates, and listener rules
- Store code in a structured folder (e.g., `IaC/`)