

Application Load Balancer with Autoscaling for a PHP WebApp

1. Host a simple php web application in the aws instance
 2. Configure an application load balancer for the instance containing web application.
 3. Create an auto scaling group with scaling policies for the application instance.
 4. Attach the load balancer with auto scaling group.
 5. Verify the working of scaling policies by increasing the load manually.

STEPS

1. AWS Log in and create the virtual host virtual most
 2. Install Apache and PHP # yum install -y httpd php
 3. Start and enable Apache to start on boot

```
# systemctl start httpd
```

```
# systemctl enable httpd
```

- #### 4. Create a simple php file

```
# echo "<?php phpinfo(); ?>" > /var/www/html/index.php
```

```
PuTTY (inactive)
Verifying : php-common-5.4.16-46.amzn2.0.5.x86_64 5/13
Verifying : httpd-2.4.59-1.amzn2.x86_64 6/13
Verifying : apr-1.7.2-1.amzn2.x86_64 7/13
Verifying : httpd-filesystem-2.4.59-1.amzn2.noarch 8/13
Verifying : php-cli-5.4.16-46.amzn2.0.5.x86_64 9/13
Verifying : mod_http2-1.19.19-1.amzn2.0.2.x86_64 10/13
Verifying : apr-util-1.6.3-1.amzn2.0.1.x86_64 11/13
Verifying : mailcap-2.1.41-2.amzn2.noarch 12/13
Verifying : generic-logos-httdp-18.0.0-4.amzn2.noarch 13/13

Installed:
httpd.x86_64 0:2.4.59-1.amzn2      php.x86_64 0:5.4.16-46.amzn2.0.5

Dependency Installed:
apr.x86_64 0:1.7.2-1.amzn2
apr-util.x86_64 0:1.6.3-1.amzn2.0.1
apr-util-bdb.x86_64 0:1.6.3-1.amzn2.0.1
generic-logos-httdp.noarch 0:18.0.0-4.amzn2
httpd-filesystem.noarch 0:2.4.59-1.amzn2
httpd-tools.x86_64 0:2.4.59-1.amzn2
libzip010-compat.x86_64 0:0.10.1-9.amzn2.0.5
mailcap.noarch 0:2.1.41-2.amzn2
mod_http2.x86_64 0:1.19.19-1.amzn2.0.2
php-cli.x86_64 0:5.4.16-46.amzn2.0.5
php-common.x86_64 0:5.4.16-46.amzn2.0.5

Complete!
[root@ip-172-31-40-136 ~]#
[root@ip-172-31-40-136 ~]#
[root@ip-172-31-40-136 ~]#
[root@ip-172-31-40-136 ~]#
[root@ip-172-31-40-136 ~]#
[root@ip-172-31-40-136 ~]# systemctl start httpd
[root@ip-172-31-40-136 ~]# systemctl enable httpd
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-40-136 ~]#
[root@ip-172-31-40-136 ~]#
[root@ip-172-31-40-136 ~]#
[root@ip-172-31-40-136 ~]#
[root@ip-172-31-40-136 ~]#
[root@ip-172-31-40-136 ~]# sudo echo "<?php phpinfo(); ?>" > /var/www/html/index.php
[root@ip-172-31-40-136 ~]#
[root@ip-172-31-40-136 ~]#
[root@ip-172-31-40-136 ~]#
```

- ## 5. Create a Load Balancer

- Go to the EC2 dashboard and select "Load Balancers" under "Load Balancing".
 - Click "Create Load Balancer" and choose "Application Load Balancer".
 - Configure the load balancer:
 - Name: myLoadBalancer
 - Scheme: Internet-facing
 - Listeners: HTTP (port 80)

- Availability Zones: Select the VPC and subnets.
- Configure security settings, select the existing security group or create a new one.

The screenshot shows the AWS Services page with the search bar set to "load balancers". The main content area displays three types of load balancers:

- Application Load Balancer Info**: Handles HTTP and HTTPS traffic, routing to multiple targets like EC2 instances or containers.
- Network Load Balancer Info**: Handles TCP and UDP traffic, offloading TLS, and routing to targets via ALB or NLB.
- Gateway Load Balancer Info**: Handles TLS traffic, routing to targets via GWLB.

A sidebar on the right is titled "Amazon Q" and shows a conversation about cross-region access. Buttons for "Don't allow" and "Continue" are present.

The screenshot shows the "Create Application Load Balancer" wizard at step 1: Basic configuration. The form includes fields for:

- Load balancer name**: MyLoadBalancer
- Scheme**: Internet-facing
- Load balancer IP address type**: Dualstack

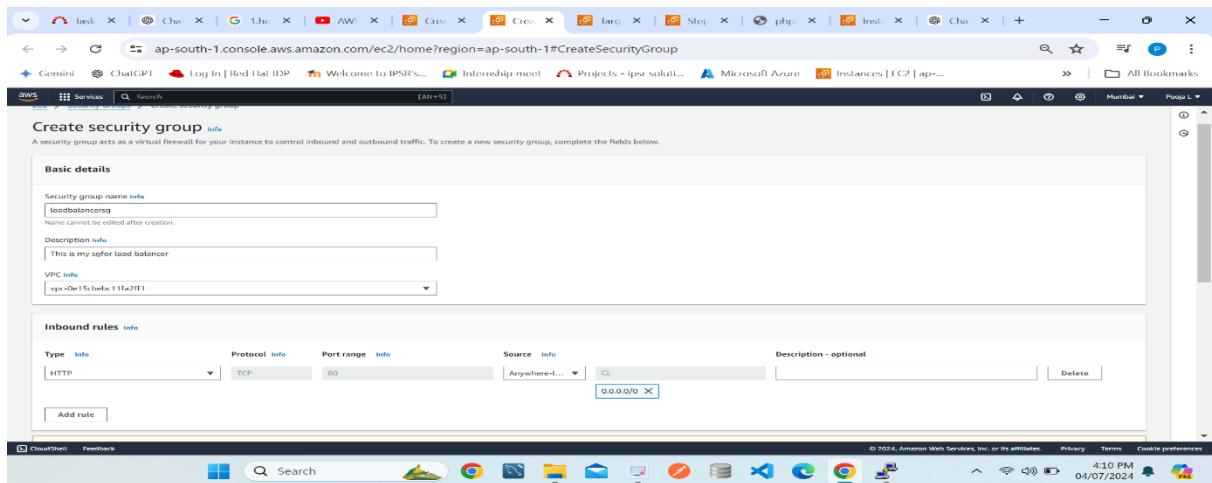
A sidebar on the right shows an Amazon Q AI assistant interface.

The screenshot shows the "Create Application Load Balancer" wizard at step 2: Network mapping. The network mapping section shows:

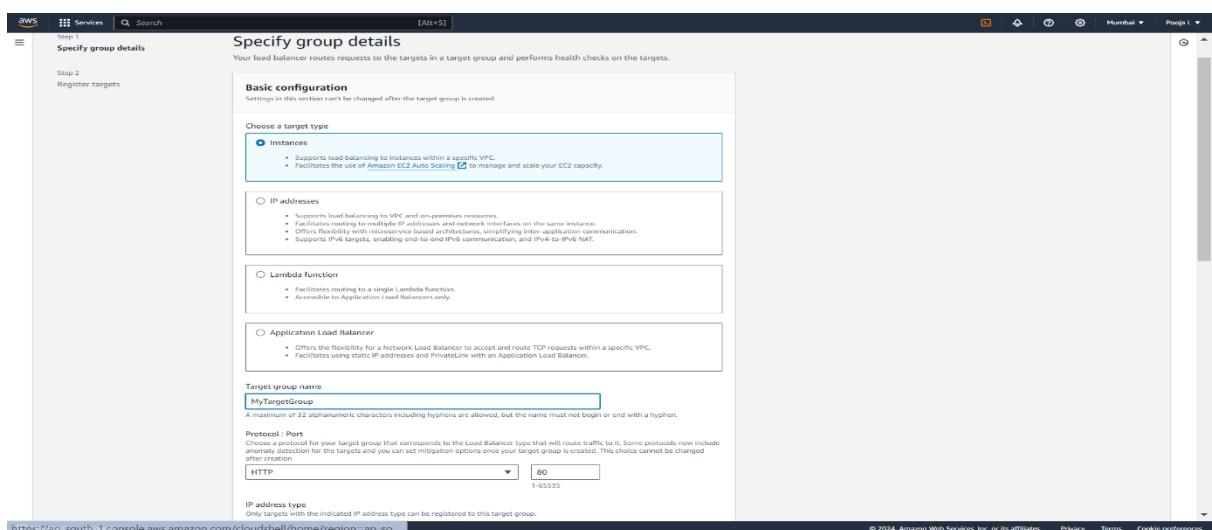
- VPC**: ap-south-1a
- Mappings**: ap-south-1a (ap-s1-a21)
- Subnets**: ap-south-1a-subnet-0, ap-south-1a-subnet-1
- Available Subnets**: ap-south-1b, ap-south-1c

A sidebar on the right shows an Amazon Q AI assistant interface.

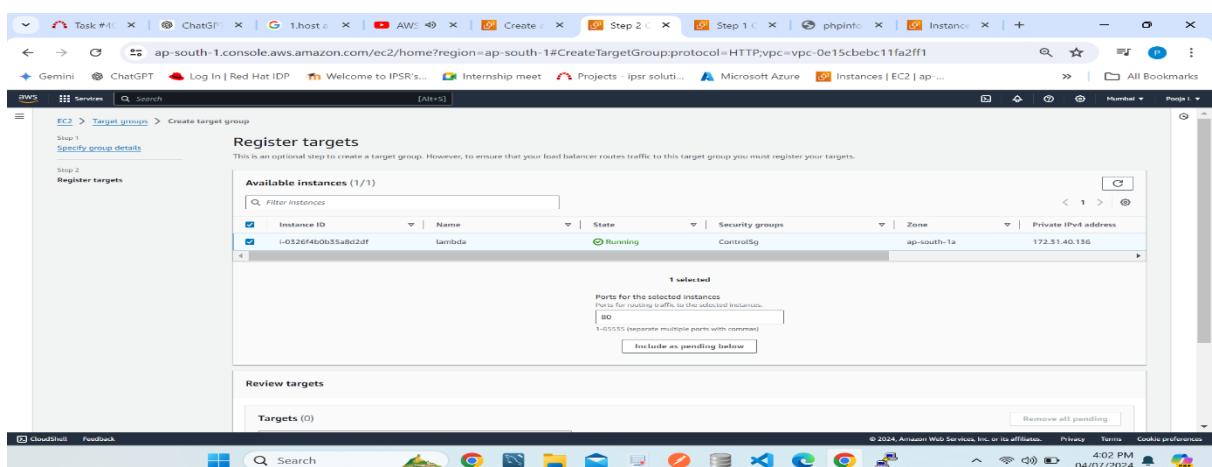
Create security group



- Configure the Target Group
 - Name the target group (e.g., MyTargetGroup), choose Instances as the target type, and HTTP (port 80).
 - Register your EC2 instance with this target group.
 - Review and create the load balancer.



Register the target with instance



Task #4 | ChatGPT | 1.host | AWS | Create | Step 2 | Step 1 | phpinfo | Instances | +

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateTargetGroup:protocol=HTTP;vpc=vpc-0e15cbebc11fa2ff1

Gemini ChatGPT Log In | Red Hat IDP Welcome to IPSR's... Internship meet Projects - ipsr soluti... Microsoft Azure Instances | EC2 | ap-south-1

Services Search [Alt+S]

Instance ID	Name	State	Security groups	Zone	Private IPv4 address
i-0326f4b0b35a8d2df	lambda	Running	ControlSg	ap-south-1a	172.31.40.136

Targets (1)

Instance ID	Name	Port	State	Security groups	Zone	Private IPv4 address	Subnet ID	Launch time
i-0326f4b0b35a8d2df	lambda	80	Running	ControlSg	ap-south-1a	172.31.40.136	subnet-0aed558f3ac5b2d02	July 4, 2024, 15:04 (UTC+04:00)

1 pending

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 4:03 PM 04/07/2024

Target Group Created

Task #4 | ChatGPT | 1.host | AWS | Create | Step 1 | Step 2 | phpinfo | Instances | +

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#TargetGroup:targetGroupArn=arn:aws:elasticloadbalancing:ap-south-1:058264206411:targetgroup/MyTargetGroup:097fba53721b41

Gemini ChatGPT Log In | Red Hat IDP Welcome to IPSR's... Internship meet Projects - ipsr soluti... Microsoft Azure Instances | EC2 | ap-south-1

Services Search [Alt+S]

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

EC2 > Target groups > MyTargetGroup

MyTargetGroup

Details

Target type	Protocol	Protocol version	VPC
Instance	HTTP: 80	HTTP1	vpc-0e15cbebc11fa2ff1
IP address type	Load balancer		
IPv4	None associated		

1 Total targets

Health	Unhealthy	Unused	Initial	Draffing
Healthy	0	1	0	0
Anomalous	0	0	0	0

Distribution of targets by Availability Zone (AZ)

Targets Monitoring Health checks Attributes Tags

Registered targets (1) Info

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 4:04 PM 04/07/2024

Attach the target group in listening and routing section

Task #4 | ChatGPT | 1.host | AWS | Create | Step 2 | Step 1 | phpinfo | Instances | +

Listeners and routing info

A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

Listener HTTP:80

Protocol	Port	Default action	Info
HTTP	80	Forward to MyTargetGroup	HTTP

Listener tags - optional

Add listener tag

Load balancer tags - optional

AWS Web Application Firewall (WAF)

AWS Global Accelerator

Amazon Q

Hello! I'm Amazon Q, your AWS generative AI assistant.

Ask me anything about AWS services and features or choose a sample question below to start a conversation.

List my S3 buckets

What is the CLI command to list all the t3.micro instances in my account?

Amazon Q can now list and describe your AWS resources. By continuing, you consent to Amazon Q making cross-region calls to access your resources.

You can update cross-region preferences in chat settings. Learn more

Don't allow Continue

Ask me anything about AWS

Max 1000 characters

Use of Amazon Q is subject to the AWS Responsible AI Policy

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Load Balancer Created

The screenshot shows three browser windows demonstrating the creation of a Load Balancer in AWS.

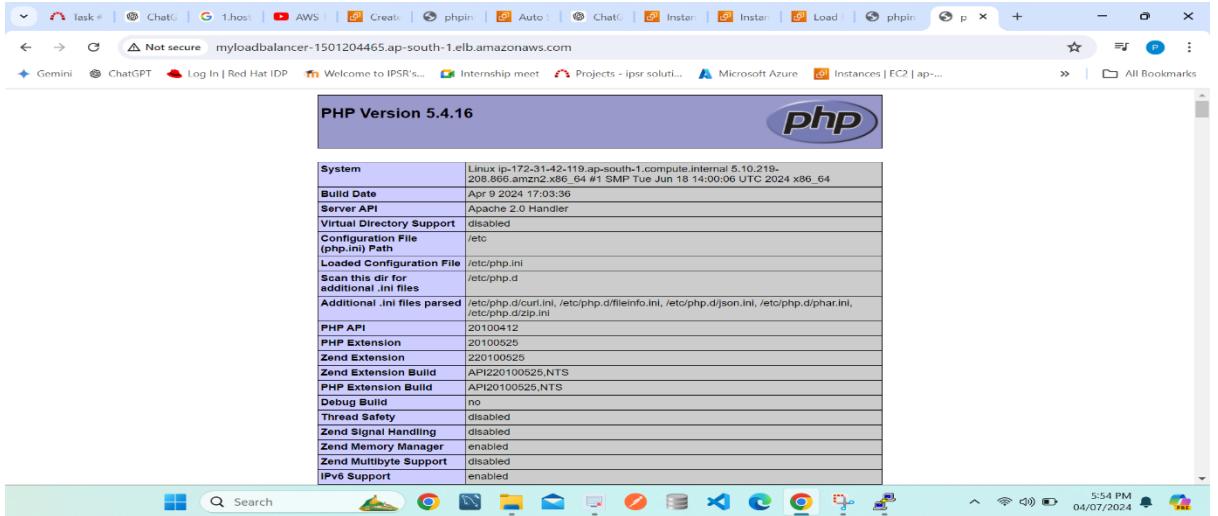
Top Window: AWS Global Accelerator configuration page. Summary section shows a basic configuration for a load balancer named "MY_Load_Balancer". It includes a VPC (vpc-0e15cbebc11fa2#1), security group (ControlSG sg-0d2176215aebc7c8#), and a target group (ap-south-1a). Listener and routing rules are also defined.

Middle Window: EC2 Load Balancers details page for "myloadbalancer". It shows the load balancer's configuration, including its ARN, VPC (vpc-0e15cbebc11fa2#1), and availability zones (ap-south-1a, ap-south-1b). The DNS name is myloadbalancer-1501204465.ap-south-1.elb.amazonaws.com (A Record).

Bottom Window: EC2 Load Balancers details page for the same load balancer. This view shows the listeners and rules configuration, specifically a rule for port 80 forwarding traffic to the target group "MyTargetGroup" (100% weight) and setting target group stickiness to Off.

A sidebar on the right of each window displays the Amazon Q AI assistant, which is asking for permission to access AWS resources. The user has the option to "Don't allow" or "Continue".

we can view the website using the DNS name



6. Create the image of the template

- Select the instance click on action ->Images and template ->create image

7. Create an Auto Scaling Group with Scaling Policies

- Create a Launch Template
 - Go to the EC2 dashboard and select "Launch Templates".
 - Click "Create Launch Template" and configure the details (use the same AMI and instance type as your EC2 instance).

Create and Launch Template

Amazon Machine Image (AMI)

launchimage
ami-0290b6681d573d615 2024-07-04T12:57:26.000Z Virtualization: hvm ENA enabled: true Root device type: ebs

Description: image for auto scaling group

Architecture: x86_64 AMI ID: ami-02b0b6681d573d615

Instance type

t2.micro Family: t2 1 vCPU 1 GiB Memory Current generation: true Free tier eligible

On-Demand Linux base pricing: 0.0124 USD per Hour On-Demand Windows base pricing: 0.017 USD per Hour On-Demand Amazon RDS base pricing: 0.0124 USD per Hour On-Demand SUSE base pricing: 0.0124 USD per Hour

Additional costs apply for AMIs with pre-installed software

Key pair (login)

Control Create new key pair

Network settings

Subnet: Don't include in launch template Create new subnet

Firewall (security groups): Select existing security group Create security group

Security groups Info: loadbalancerssg sg-061bfbdccdd5c4817 X

Storage (volumes)

EBS Volumes Hide details

Summary

Software Image (AMI): iamge for auto scaling group ami-0290b6681d573d615

Virtual server type (instance type): t2.micro

Firewall (security group): loadbalancerssg

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Create launch template

Key pair (login)

Control Create new key pair

Network settings

Subnet: Don't include in launch template Create new subnet

Firewall (security groups): Select existing security group Create security group

Security groups Info: loadbalancerssg sg-061bfbdccdd5c4817 X

Storage (volumes)

EBS Volumes Hide details

Summary

Software Image (AMI): iamge for auto scaling group ami-0290b6681d573d615

Virtual server type (instance type): t2.micro

Firewall (security group): loadbalancerssg

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Create launch template

Success

Successfully created MyTemplate (lt-00e20ba30c2967a15).

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

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.

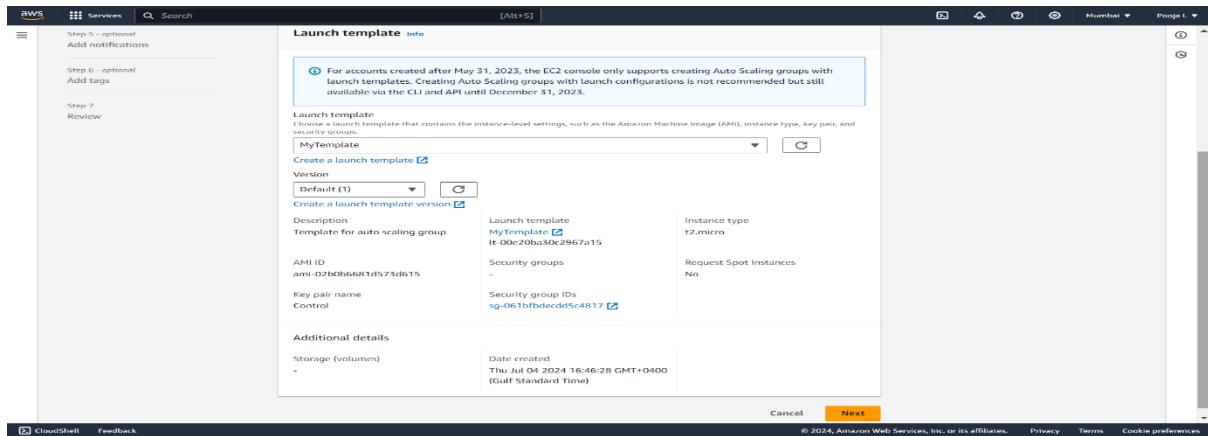
Create Auto Scaling group

Create Spot Fleet

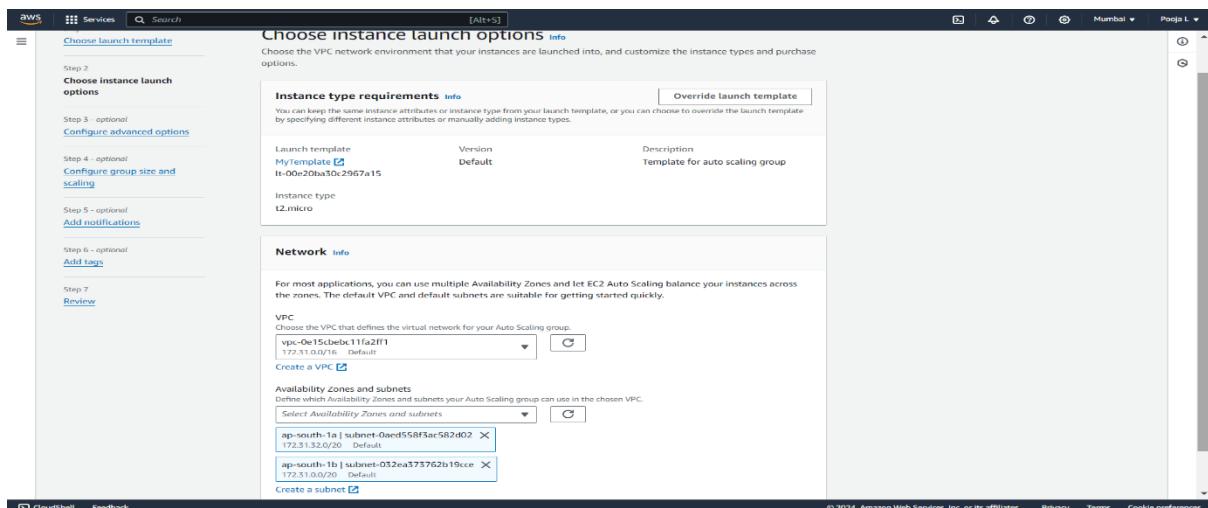
A Spot Instance is an unused EC2 instance that is available for less than the On-Demand price. Because Spot Instances enable you to request unused EC2 instances at steep discounts, you can lower your Amazon EC2 costs significantly. The hourly price for a Spot Instance (of each instance type in each Availability 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

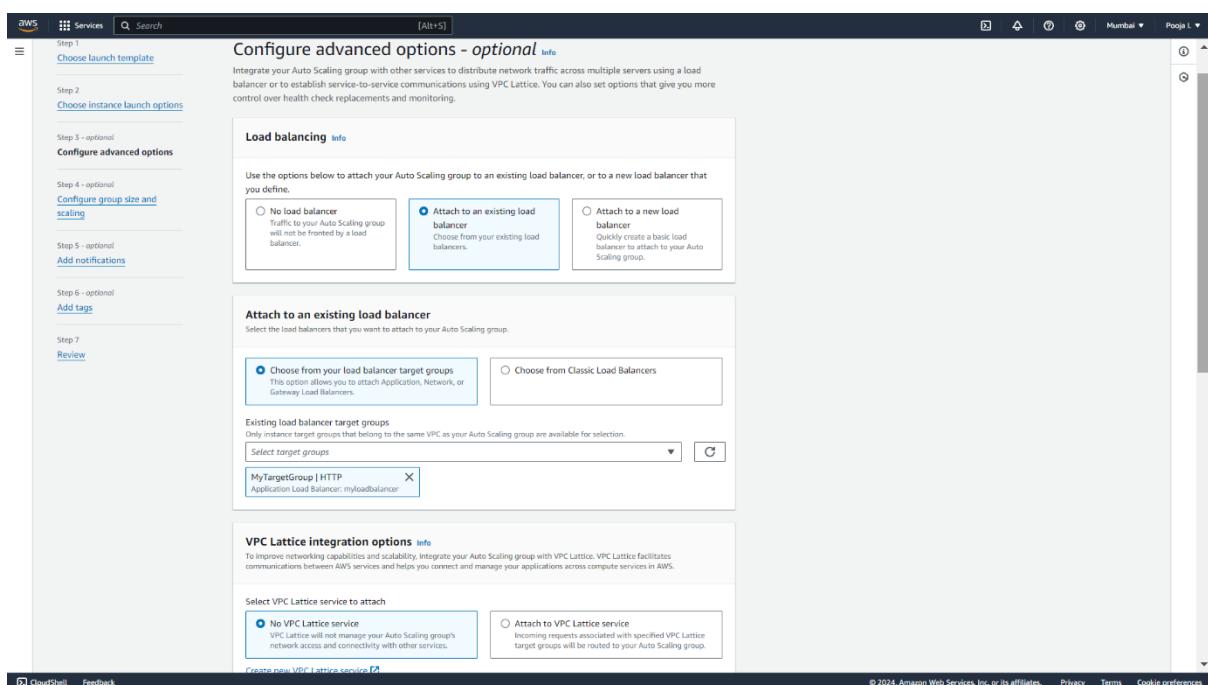
Here we can choose the launch template and click next



Choose the instances we want to launch. Here we select the VPC and availability zone



Click next and attach an existing load balancer



Health checks

Health checks increase availability by replacing unhealthy instances. When you use multiple health checks, all are evaluated, and if at least one fails, instance replacement occurs.

EC2 health checks

Always enabled

Additional health check types - optional

Turn on Elastic Load Balancing health checks (Recommended)

EC2 Auto Scaling will start to detect and act on health checks performed by Elastic Load Balancing. To avoid unexpected terminations, first verify the settings of these health checks in the [Load Balancer console](#).

Turn on VPC Lattice health checks

VPC Lattice can monitor whether instances are available to handle requests. If it considers a target as failed a health check, EC2 Auto Scaling replaces it after its next periodic check.

Health check period [Info](#)

This time period delays the first health check until your instances finish initializing. It doesn't prevent an instance from terminating when placed into a non-running state.

300 seconds

Additional settings

Monitoring [Info](#)

Enable group metrics collection within CloudWatch

Default instance warmup [Info](#)

The instance warmup CloudWatch metrics for new instances do not contribute to the group's aggregated instance metrics, as their usage data is not reliable yet.

Enable default instance warmup

Configure the group size, subnets, and attach to the existing load balancer.

Configure group size and scaling - optional [Info](#)

Define your group's desired capacity and scaling limits. You can optionally add automatic scaling to adjust the size of your group.

Group size [Info](#)

Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by using automatic scaling.

Desired capacity type

Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.

Units (number of instances)

Desired capacity

Specify your group size.

1

Scaling [Info](#)

You can resize your Auto Scaling group manually or automatically to meet changes in demand.

Scaling limits

Set limits on how much your desired capacity can be increased or decreased.

Min desired capacity	1	Max desired capacity	1
Equal or less than desired capacity		Equal or greater than desired capacity	

Automatic scaling - optional

Choose whether to use a target tracking policy [Info](#)

You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

No scaling policies

Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.

Target tracking scaling policy

Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.

Scaling policy name

Target Tracking Policy

Metric type [Info](#)

Resource utilization that determines if resource utilization is too low or high. If using EC2 metrics, consider enabling detailed monitoring for better scaling performance.

Define the scaling policy (e.g., keep average CPU utilization at 80%).

Metric type [Info](#)

CloudWatch metrics determine if resource utilization is too low or high. If using EC2 metrics, consider enabling detailed monitoring for better scaling performance.

Average CPU utilization

Target value

80

Instance warmup [Info](#)

300 seconds

Disable scale-in to create only a scale-out policy

Instance maintenance policy [Info](#)

Control your Auto Scaling group's availability during instance replacement events. This includes health checks, instance refreshes, maximum instance lifetime, features and events that happen automatically to keep your group balanced, called rebalancing events.

Choose a replacement behavior depending on your availability requirements

Mixed behavior	Priority availability	Content costs	Flexible
<input checked="" type="radio"/> No policy	<input type="radio"/> Launch before terminating	<input type="radio"/> Terminate and launch	<input type="radio"/> Custom behavior
For rebalancing events, new instances will be launched before terminating others. For instance events, instances terminate and launch at the same time.	Launch new instances before terminating others. This allows you to go above your desired capacity by a given percentage and may temporarily increase costs.	Terminate and launch instances at the same time. This allows you to go below your desired capacity by a given percentage and may temporarily reduce availability.	Set custom values for the minimum and maximum amount of available capacity. This provides greater flexibility in setting how far below and above your desired capacity EC2 Auto Scaling goes when replacing instances.

Instance scale-in protection

Scale-in protection prevents newly launched instances from being terminated by scaling activities. Make sure to remove scale-in protection for the group or individual instances when instances are ready to be terminated.

Enable instance scale-in protection

Next

AWS Auto Scaling Step 7: Review

Step 1: Choose launch template

Group details

- Auto Scaling group name: myautoscalinggroup

Launch template

Launch template	Version	Description
MyTemplate	Default	Template for auto scaling group

Step 2: Choose instance launch options

Network

VPC: vpc-0e15beb11fa2ff1

Availability Zone	Subnet	Range
ap-south-1a	subnet-0aed558f3ac582e02	172.31.32.0/20
ap-south-1b	subnet-052ea575762b19cc	172.31.0.0/20

Instance type requirements

This Auto Scaling group will adhere to the launch template.

Step 3: Configure advanced options

Load balancing

EC2 > Auto Scaling groups

Auto Scaling groups (1) Info

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
myautoscalinggroup	MyTemplate Version Default	1	-	1	1	1	ap-south-1b, ap-sout...

0 Auto Scaling groups selected

Auto Scaling Group

Auto Scaling groups (1/1) Info

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max
myautoscalinggroup	MyTemplate Version Default	1	-	1	1	1

Auto Scaling group: myautoscalinggroup

Activity history (1)

Status	Description	Cause	Start time	End time
Successful	Launching a new EC2 instance: i-0a366056fd66b27c2	At 2024-07-04T12:55:13Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 1. At 2024-07-04T12:55:28Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 1.	2024-07-04 04:55:30 PM +04:00	2024-07-04 04:56:00 PM +04:00

Two Instances Created

The screenshot shows the AWS EC2 Instances page. There are two instances listed:

- lambda**: Instance ID i-0326f4b0b35a8d2df, Status: Running, Instance type: t2.micro, Status check: 2/2 checks passed, Alarm status: View alarms, Availability Zone: ap-south-1a, Public IPv4 DNS: ec2-65-0-55-235.ap-south-1.amazonaws.com
- i-0a366056fd66b27c2**: Instance ID i-0a366056fd66b27c2, Status: Running, Instance type: t2.micro, Status check: 2/2 checks passed, Alarm status: View alarms, Availability Zone: ap-south-1a, Public IPv4 DNS: ec2-13-201-61-1.ap-south-1.amazonaws.com

A modal window for the instance **lambda** is open, displaying its details:

- Public IPv4 address:** 65.0.55.239 (open address)
- Private IPv4 addresses:** 172.31.40.136
- Public IPv4 DNS:** ec2-65-0-55-235.ap-south-1.amazonaws.com

My Target Group and Its registered instances

The screenshot shows the AWS Lambda Target Groups page. A target group named **MyTargetGroup** is selected.

Details:

- Protocol: Port 80
- VPC: vpc-0e15checb11fa2#1

Targets:

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

Registered targets (2):

Instance ID	Name	Port	Zone	Health status	Health status details	Launch...	Anomaly detection result
i-0a366056fd66b27c2		80	ap-south-1a	Unused	Target group is not co...	July 4, 20...	Normal
i-0326f4b0b35a8d2df	lambda	80	ap-south-1a	Unused	Target group is not co...	July 4, 20...	Normal

Roles Created

The screenshot shows the AWS IAM Roles page. The left sidebar includes 'Identity and Access Management (IAM)', 'Dashboard', 'Access management' (User groups, Users, Roles), 'Access reports' (Access Analyzer, External access), and 'CloudShell' and 'Feedback' buttons. The main content area displays 'Roles (4) Info' with a table:

Role name	Trusted entities	Last activity
AWSServiceRoleForAutoScaling	AWS Service: autoscaling (Service-Link)	25 minutes ago
AWSServiceRoleForElasticLoadBalancing	AWS Service: elasticloadbalancing (Service-Link)	1 hour ago
AWSServiceRoleForSupport	AWS Service: support (Service-Link)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Link)	-

The screenshot shows the AWS AMI Catalog page. The left sidebar includes 'Amazon Machine Images (AMIs)' (1/1), 'Instances' (Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), 'Images' (AMIs, AMI Catalog), and 'Elastic Block Store' (Volumes, Snapshots, Lifecycle Manager). The main content area displays 'AMI ID: ami-02b0b6681d573d615' with detailed information:

AMI ID	Image type	Platform details	Root device type
ami-02b0b6681d573d615	machine	Linux/UNIX	EBS
AMI name	Owner account ID	Architecture	Usage operation
launchimage	058264296431	x86_64	RunInstances
Root device name	Status	Source	Virtualization type
/dev/xvda	Available	058264296431/launchimage	hvm
Boot mode	State reason	Creation date	Kernel ID
		2024-04-07T12:46:28.000Z	

Launch Template

The screenshot shows the AWS Launch Templates page. The left sidebar includes '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), and 'Elastic Block Store' (Volumes, Snapshots, Lifecycle Manager). The main content area displays 'Launch Templates (1/1) Info' with a table:

Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time	Created By
lt-00e20ba30c2967a15	MyTemplate	1	1	2024-04-07T12:46:28.000Z	arn:aws:iam::058264296431:root

The 'MyTemplate (lt-00e20ba30c2967a15)' details section shows:

Launch template ID	Launch template name	Default version	Owner
lt-00e20ba30c2967a15	MyTemplate	1	arn:aws:iam::058264296431:root

PHP Application Launched

The screenshot shows a web browser window with the URL `65.0.55.239`. The title bar indicates "Not secure". The page content is titled "PHP Version 5.4.16" and displays detailed PHP configuration information in a table.

PHP Version 5.4.16	
System	Linux ip-172-31-40-136.ap-south-1.compute.internal 5.10.219-208.866.amzn2.x86_64 #1 SMP Tue Jun 18 14:00:06 UTC 2024 x86_64
Build Date	Apr 9 2024 17:03:36
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/curl.ini, /etc/php.d/fileinfo.ini, /etc/php.d/json.ini, /etc/php.d/phar.ini, /etc/php.d/zip.ini
PHP API	20100412
PHP Extension	20100525
Zend Extension	220100525
Zend Extension Build	API220100525.NTS
PHP Extension Build	API20100525.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled