Container Design and Deployment

Deploying a Microservice Application over an AWS EC2 Instance and

implementing Auto Scaling Group Policy and load balancing.

Introduction

This research project focuses on the deployment of a Microservice on an Amazon Web Services (AWS) Elastic Compute Cloud (EC2) Instance whilst implementing specific policies and features available within the AWS EC2 Instance, a publicly available cloud-computing platform. The research uses a .jar application called a "Product Service" as a Microservice application to be deployed within the AWS EC2 Instance. As this is an introductory research project, all steps and required software applications are outlined within this research to demonstrate the student'sunderstanding of deploying a Microservice application. Brief terminologies are defined, and key aspects of the demonstration are provided as screenshots within this document. All demonstrations during this exercise are performed using a Free-tier AWS account. The research concludes with an overview of the AWS Auto Scaling group policy, and the challenges and limitations of its usage.

In the dynamic world of cloud computing, managing the scalability and availability of your applications is a critical endeavor. AWS Auto Scaling is a service provided by Amazon Web Services (AWS) that empowers businesses to automatically adjust the capacity of their resources based on specific performance metrics. While this service offers a powerful solution for ensuring optimal resource utilization and application performance, it comes with certain regional limitations.

As of the last knowledge update in September 2021, AWS Auto Scaling is inherently region-specific. This means that Auto Scaling policies are created and managed within a single AWS region and are applied only to resources located within that region. This regional confinement can pose a challenge for organizations with multi-region applications, as it necessitates the setup of Auto Scaling policies separately in each AWS region. This introduction explores the implications of these regional constraints on managing resources in a multi-region context while highlighting potential solutions to streamline the process. However, it's crucial to stay informed about any updates or enhancements that AWS might have introduced to Auto Scaling since that time

Amazon Web Services (AWS): A publicly available cloud computing platform owned by Amazon.

AWS Elastic Compute Cloud (EC2): A web service that you can use to launch and manage Linux/UNIX and Windows Server instances in Amazon data centres.

Amazon EC2 Auto Scaling: A web service that launches or terminates instances automatically based on user-defined policies, schedules, and health checks.

Amazon Machine Image (AMI): An encrypted machine image stored in Amazon Elastic Block Store (Amazon EBS) or Amazon Simple Storage Service. AMIs function like a template of a computer's root drive. They contain the operating system and can also include software and layers of your application, such as database servers, middleware, and web servers.

Microservice: Microservices are an architectural and organizational approach to software development created to speed up deployment cycles, foster innovation, and ownership, improve maintainability and scalability of software applications, and scale organizations delivering software and services by using an agile approach that helps teams work independently.

S3 Bucket is a storage for the internet within the AWS EC2 You can use it to store and retrieve any amount of data at any time, from anywhere on the web

Postman: is a complete API development platform that helps you manage your APIs in every stage of development, from designing and testing, to publishing API documentation and monitoring.

Steps for deploying a Microservice on an AWS EC2 Instance.

The implementation considers that there exists a Free-tier AWS EC2 Instance Account. 1. Logging into your EC2 Instance using an IAM user account. It is important to note that granting users root access to an EC2 Instance is not the best security practice for accessing AWS resources, therefore the root user must set up another user account for multiple access to the resources. This section provides a detailed step for setting up an IAM user account.

Steps for creating a user account from the root user account

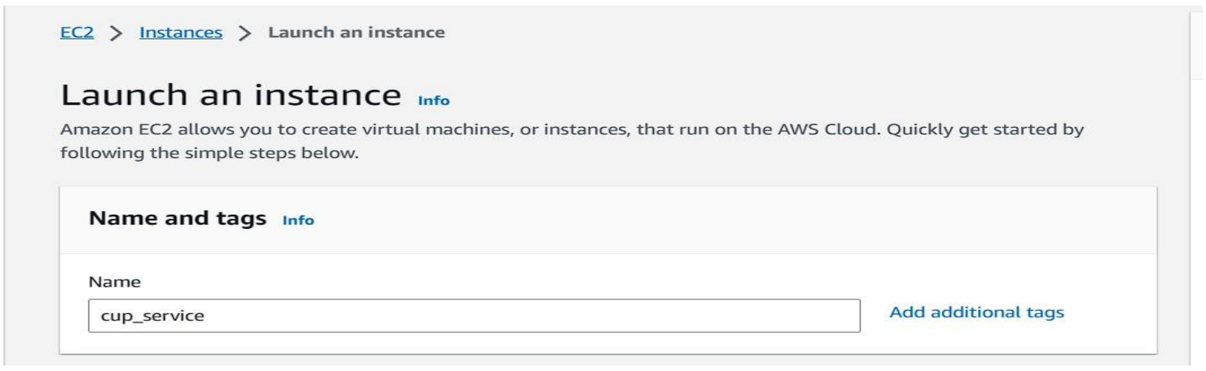
a. Go to the IAM Console (search IAM) select users and click Add user to create a new user

b. Create a username and set a password

c. Set user account policy or assigned user account to a group. For this demo, we set the user policy to Administrator which grants the user all access except billing.

- Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/ write the name of the instance and additional tags if you wish.

Figure 1 shows how to launch an instance using AWS



EC2 > Instances > Launch an instance

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

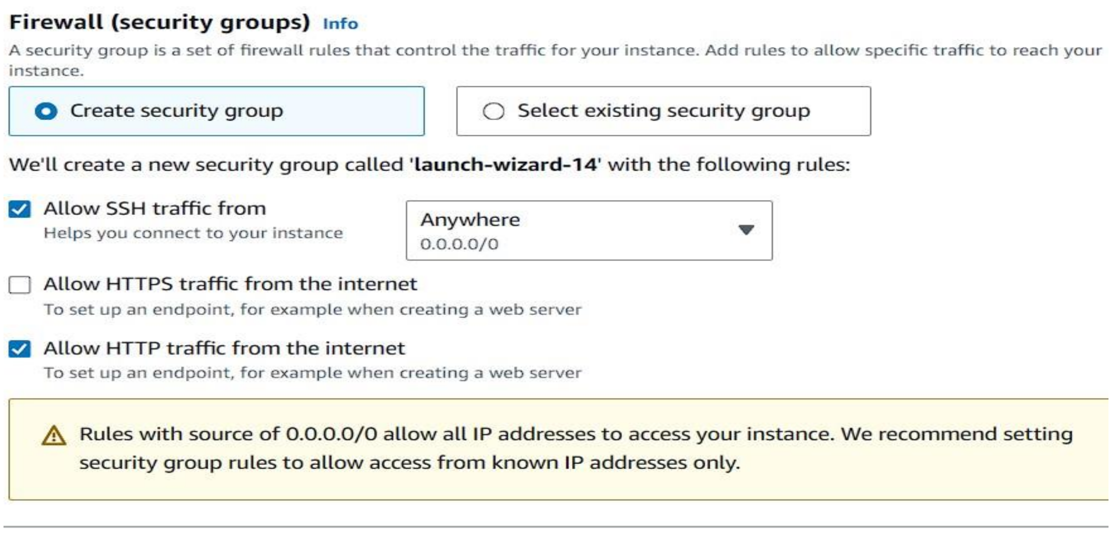
Name: cup_service

Add additional tags

- In the navigation pane, NETWORK & SECURITY, choose Key Pairs.

Note: The navigation pane is on the left side of the Amazon EC2 console. If you do not see the pane, it might be minimized; choose the arrow to expand the pane. You can allow Http traffic to expose port 80

Figure 2 : shows creation of security groups and allow HTTP traffic for port 80



Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

[✓ Create security group] Select existing security group

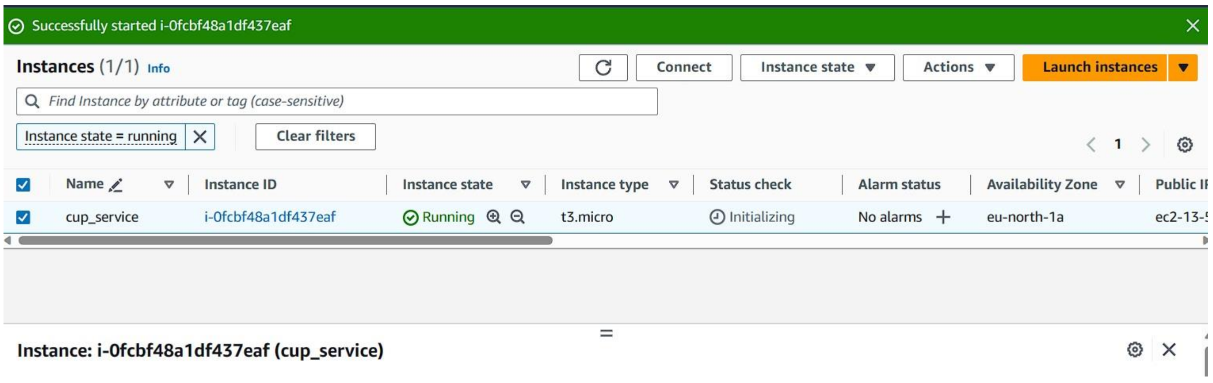
We'll create a new security group called '**launch-wizard-14**' with the following rules:

- [✓] Allow SSH traffic from — Anywhere 0.0.0.0/0
  Helps you connect to your instance
- [ ] Allow HTTPS traffic from the internet
  To set up an endpoint, for example when creating a web server
- [✓] Allow HTTP traffic from the internet
  To set up an endpoint, for example when creating a web server

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

- Choose Create Key Pair.
- Enter a name for the new key pair in the Key pair name field of the Create Key Pair dialog box, and then choose Create.
- The private key file is automatically downloaded by your browser. The base file name is the name you specified as the name of your key pair, and the file name extension is .pem. Save the private key file in a safe place.
  - Important: This is the only chance for you to save the private key file. You'll need to provide the name of your key pair when you launch an instance and the corresponding private key each time you connect to the instance.
- If you will use an SSH client on a Mac or Linux computer to connect to your Linux instance, use the following command to set the permissions of your private key file so that only you can read it.

Figure 3: shows Instance launched form AWS



Successfully started i-0fcbf48a1df437eaf

Instances (1/1) Info

Find Instance by attribute or tag (case-sensitive)

Instance state = running

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
cup_service	i-0fcbf48a1df437eaf	Running	t3.micro	Initializing	No alarms	eu-north-1a	ec2-13-

Instance: i-0fcbf48a1df437eaf (cup_service)

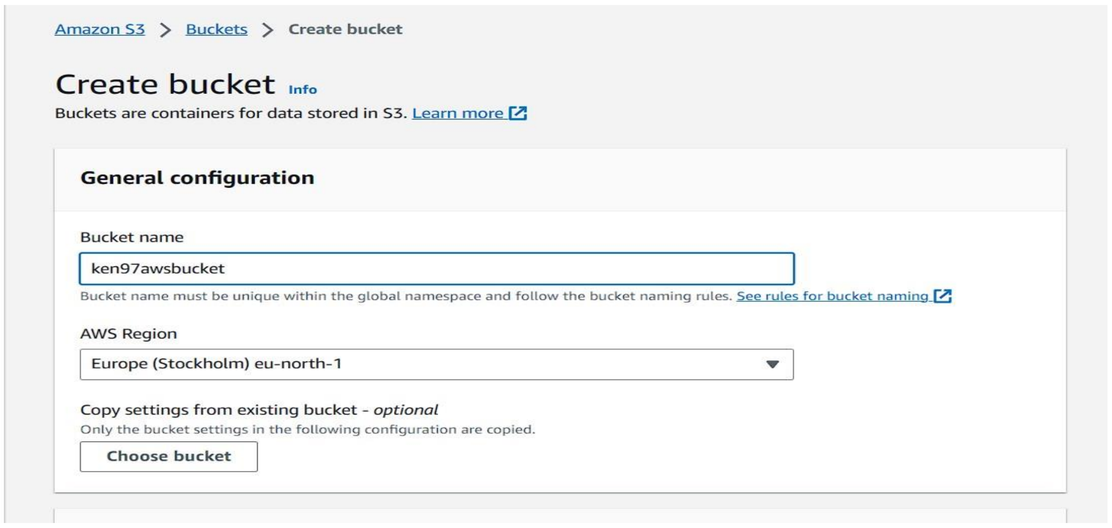
Launch Session Manager and install Java and MariaDb on the EC2 Instance. Create the necessary database and tables for the Product Service Application. Check both installations by checking the application versions. To do so, enter the following command in the terminal.

- ➢ sudo su – (to change to root directory of the EC2 Instance from the session manager)
- ➢ mariadb –version (to check if mariadb server is installed) else enter the next command
- ➢ yum install -y mariadb-server (to install mariadb server on your EC2) spercify the version if mariadb is not found using yum search mariadb
- ➢ systemctl enable mariadb (to enable the mariadb server for running)
- ➢ systemctl start mariadb (to start the mariadb server)
- ➢ mysql_secure_installation (to setup mysql root user access. You may choose to set a password that will be used to log on to your mysql. Follow the prompts and allow default settings for all other prompts)
- ➢ mysql -uroot -p (to securely login to mysql and enter your password if you have setup a login password from the previous command and your mysql environment should be set and ready for your sql commands) Enter the following sql commands to create your database and related tables.
- ➢ create database servicedb; (to create a database called serviced)
- ➢ java –version (to check if the Java application is installed) else enter the next command ➢ yum install -y java (to install java on your EC2)

Creating s3 bucket

Create the S3 Bucket: Use the aws s3api create-bucket command to create a new S3 bucket. Replace your-bucket-name with your desired bucket name and your-region with the AWS region where you want to create the bucket. You can also add additional options like versioning, logging, and ACLs by including additional parameters in the command.

Figure 4: shows creation of S3 bucket in AWS



Amazon S3 > Buckets > Create bucket

Create bucket Info

Buckets are containers for data stored in S3. Learn more

General configuration

Bucket name: ken97awsbucket

Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming

AWS Region: Europe (Stockholm) eu-north-1

Copy settings from existing bucket - *optional*

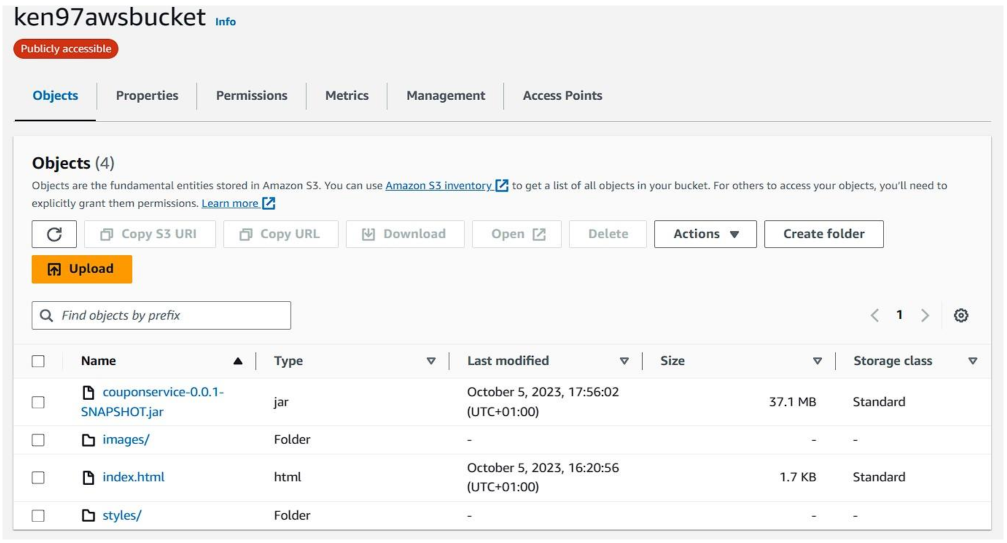
Only the bucket settings in the following configuration are copied.

Choose bucket

Remember that S3 bucket names must be globally unique within AWS, and they have naming rules (e.g., no uppercase letters, no underscores). Also, keep in mind that there may be some limitations on the number of S3 buckets you can create within your AWS account.

Once your S3 bucket is created, you can start uploading and managing objects (files) within it using various methods, such as the AWS Management Console, AWS CLI, or SDKs.

Figure 5: shows properties of S3 AWS bucket



ken97awsbucket Info

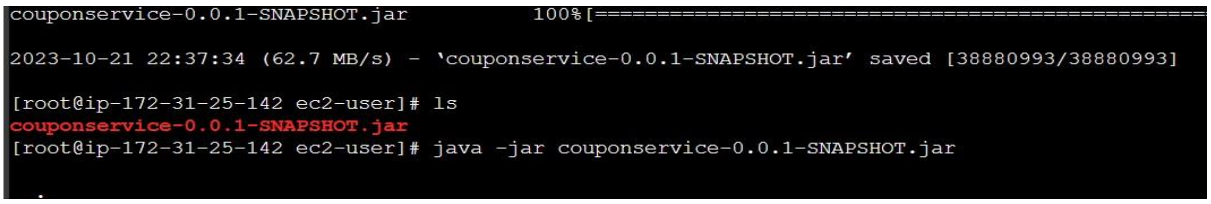
Publicly accessible

Objects (4)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
couponservice-0.0.1-SNAPSHOT.jar	jar	October 5, 2023, 17:56:02 (UTC+01:00)	37.1 MB	Standard
images/	Folder	-	-	-
index.html	html	October 5, 2023, 16:20:56 (UTC+01:00)	1.7 KB	Standard
styles/	Folder	-	-	-

Copy the microservice couponservice.jar to the s3 bucket and make it accessible public. In the EC2 console update the rc.local to start the jar application automatically. The rc.local is a script file that is called to run set commands once all the services in a Linux shell have been started. **Figure 6: Run the java file in AWS console**

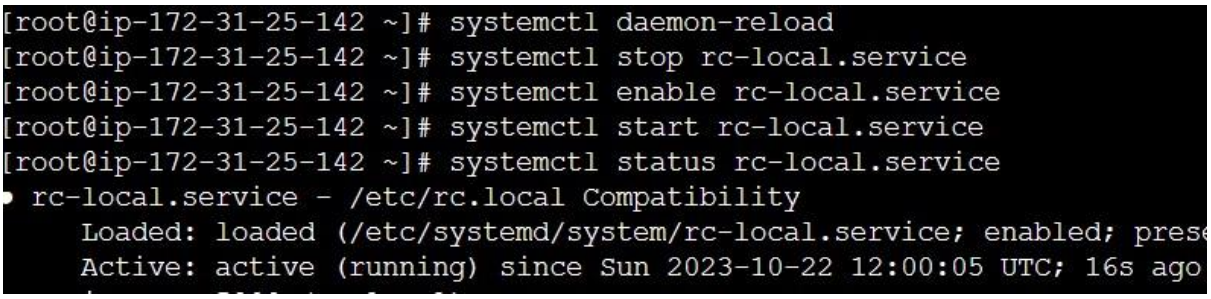


```
couponservice-0.0.1-SNAPSHOT.jar          100%[================================================

2023-10-21 22:37:34 (62.7 MB/s) - ‘couponservice-0.0.1-SNAPSHOT.jar’ saved [38880993/38880993]

[root@ip-172-31-25-142 ec2-user]# ls
couponservice-0.0.1-SNAPSHOT.jar
[root@ip-172-31-25-142 ec2-user]# java -jar couponservice-0.0.1-SNAPSHOT.jar
```

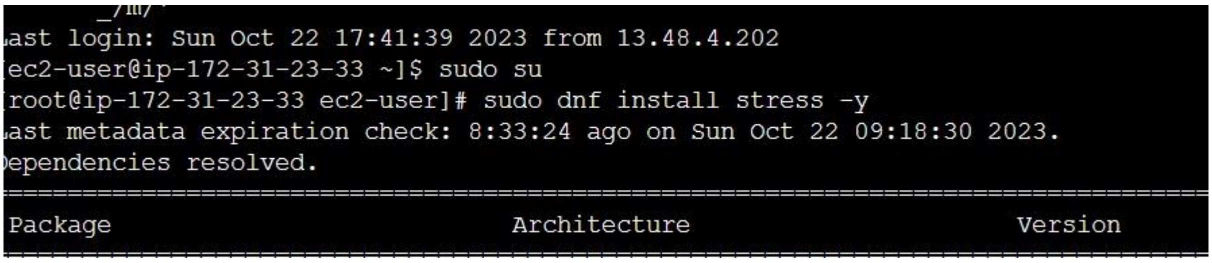
Figure 7: Installation of Rc.local and make it active



```
[root@ip-172-31-25-142 ~]# systemctl daemon-reload
[root@ip-172-31-25-142 ~]# systemctl stop rc-local.service
[root@ip-172-31-25-142 ~]# systemctl enable rc-local.service
[root@ip-172-31-25-142 ~]# systemctl start rc-local.service
[root@ip-172-31-25-142 ~]# systemctl status rc-local.service
• rc-local.service - /etc/rc.local Compatibility
    Loaded: loaded (/etc/systemd/system/rc-local.service; enabled; prese
    Active: active (running) since Sun 2023-10-22 12:00:05 UTC; 16s ago
```

Install amazon-linux extras and install stress package (package to simulate load)

Figure 8: Install amazon-linux extras and install stress package (package to simulate load)



```
ast login: Sun Oct 22 17:41:39 2023 from 13.48.4.202
ec2-user@ip-172-31-23-33 ~]$ sudo su
root@ip-172-31-23-33 ec2-user]# sudo dnf install stress -y
ast metadata expiration check: 8:33:24 ago on Sun Oct 22 09:18:30 2023.
ependencies resolved.
================================================================================
 Package                          Architecture                       Version
================================================================================
```

Creating an Amazon Machine Image (AMI)

is a process that involves capturing a snapshot of an Amazon EC2 instance. An AMI allows you to launch new EC2 instances with the same configuration and data as the original instance. Here are the steps to create an AMI using the AWS Management Console:

Using the AWS Management Console:

Sign into the AWS Management Console:

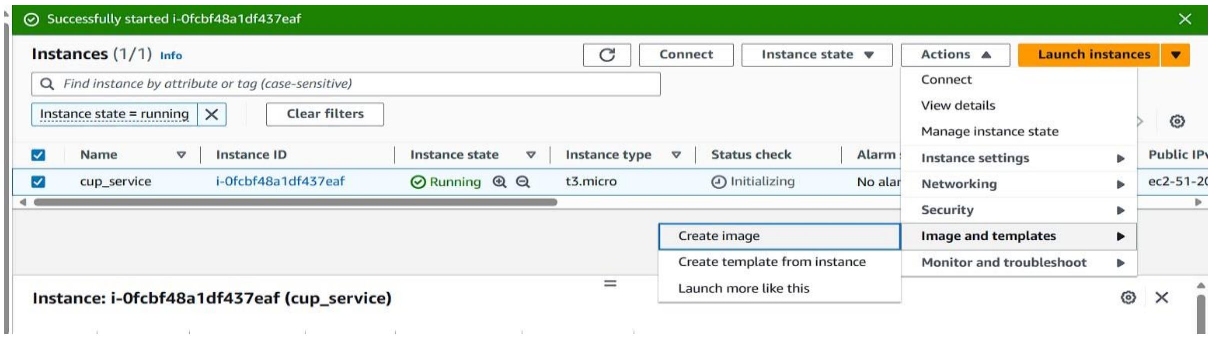
Launch an EC2 Instance: If you don't already have an EC2 instance running, you'll need to launch one.

Select the EC2 Instance: sIn the EC2 Dashboard, select the EC2 instance for which you want to create an AMI.

Create an AMI:

Right-click on the instance or use the "Actions" dropdown menu, and then select "Create Image." Alternatively, you can select the instance and click the "Create Image" button from the top menu.

Figure 9: creation for AMI image from an instance


Successfully started i-0fcbf48a1df437eaf

Instances (1/1) Info

Find instance by attribute or tag (case-sensitive)

Instance state = running

	Name	Instance ID	Instance state	Instance type	Status check	Alarm	Public IPv
✓	cup_service	i-0fcbf48a1df437eaf	Running	t3.micro	Initializing	No alar	ec2-51-2

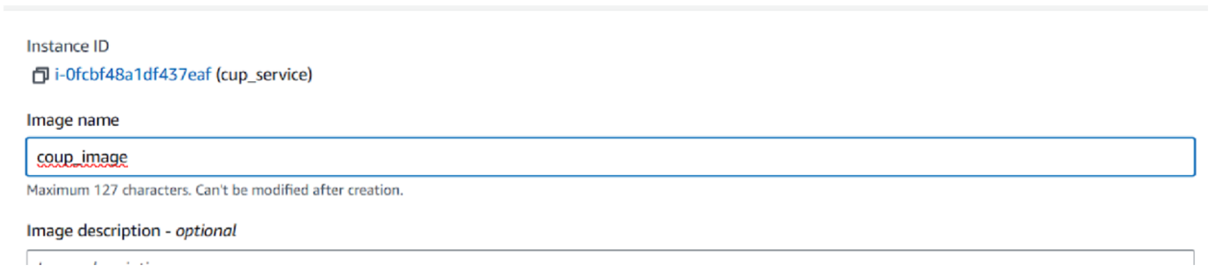
Instance: i-0fcbf48a1df437eaf (cup_service)

Configure the AMI: In the "Create Image" dialog box, you can provide the following information.

Name: Give a descriptive name to your AMI.

Description: Add an optional description for your AMI.

Figure 10: AMI image name


Instance ID

i-0fcbf48a1df437eaf (cup_service)

Image name

coup_image

Maximum 127 characters. Can't be modified after creation.

Image description - *optional*

create the AMI   After configuring the AMI settings, click the "Create Image" button. This will initiate the AMI creation process.

Monitor the AMI Creation: You can monitor the status of the AMI creation in the "AMI" section under the "Images" category in the EC2 Dashboard.

Use the AMI once the AMI is created successfully, you can use it to launch new EC2 instances with the same configuration as the original instance. While creating an AMI, the instance may be temporarily stopped, and the process might take some time, especially if the instance has a large amount of data. Additionally, the storage costs for the AMI snapshots will be incurred. After creating the AMI, you can launch new instances from it, and they will have the same state as the original instance at the time of the AMI creation.

Figure 11: Properties

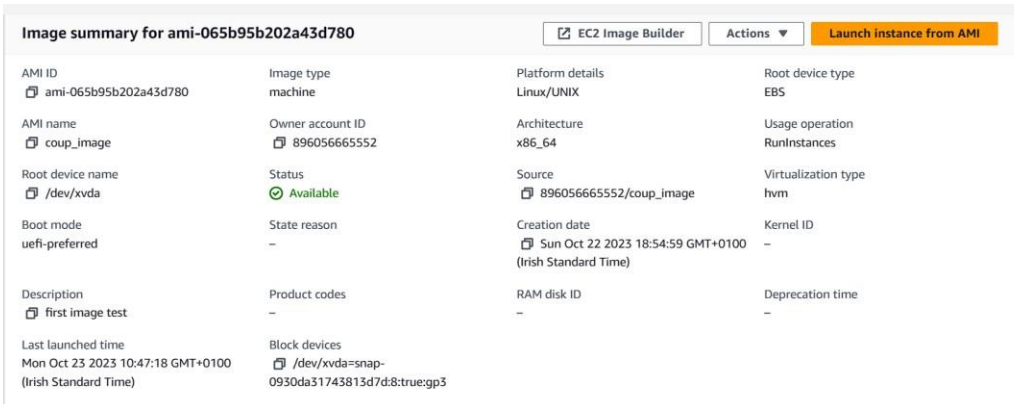
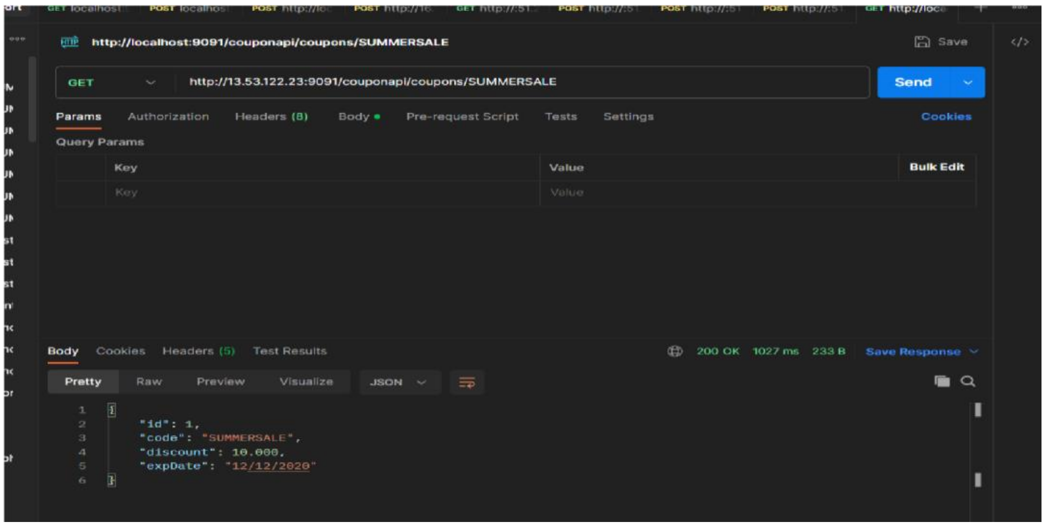


Image summary for ami-065b95b202a43d780

AMI ID ami-065b95b202a43d780	Image type machine	Platform details Linux/UNIX	Root device type EBS
AMI name coup_image	Owner account ID 896056665552	Architecture x86_64	Usage operation RunInstances
Root device name /dev/xvda	Status Available	Source 896056665552/coup_image	Virtualization type hvm
Boot mode uefi-preferred	State reason –	Creation date Sun Oct 22 2023 18:54:59 GMT+0100 (Irish Standard Time)	Kernel ID –
Description first image test	Product codes –	RAM disk ID –	Deprecation time –
Last launched time Mon Oct 23 2023 10:47:18 GMT+0100 (Irish Standard Time)	Block devices /dev/xvda=snap-0930da31743813d7d:8:true:gp3		

Figure 12 testing the Coup Image using postman



http://localhost:9091/couponapi/coupons/SUMMERSALE

GET http://13.53.122.23:9091/couponapi/coupons/SUMMERSALE

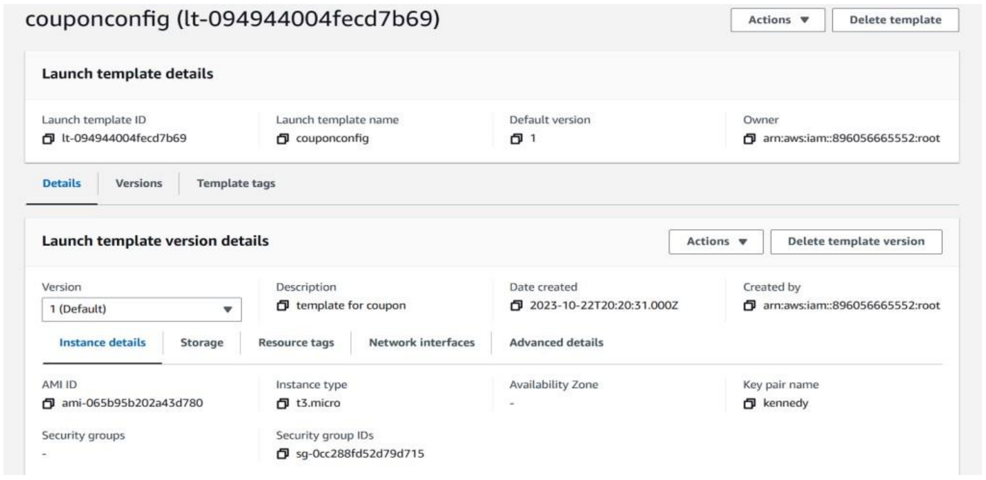
200 OK 1027 ms 233 B

```
{
    "id": 1,
    "code": "SUMMERSALE",
    "discount": 10.000,
    "expDate": "12/12/2020"
}
```

Auto Scalling Group Create launch template

- Under Launch template name and description, enter a name and description for the new launch template.
- Under Key pair (login), for Key pair name, choose the name of the previously created key pair to use when connecting to instances, for example, using SSH.
- Under Network settings, for Security groups, choose one or more previously created security groups.

Figure 13: Launch template details



couponconfig (lt-094944004fecd7b69)

Actions ▾ | Delete template

Launch template details

Launch template ID	Launch template name	Default version	Owner
lt-094944004fecd7b69	couponconfig	1	arn:aws:iam::896056665552:root

Details | Versions | Template tags

Launch template version details

Actions ▾ | Delete template version

Version	Description	Date created	Created by
1 (Default)	template for coupon	2023-10-22T20:20:31.000Z	arn:aws:iam::896056665552:root

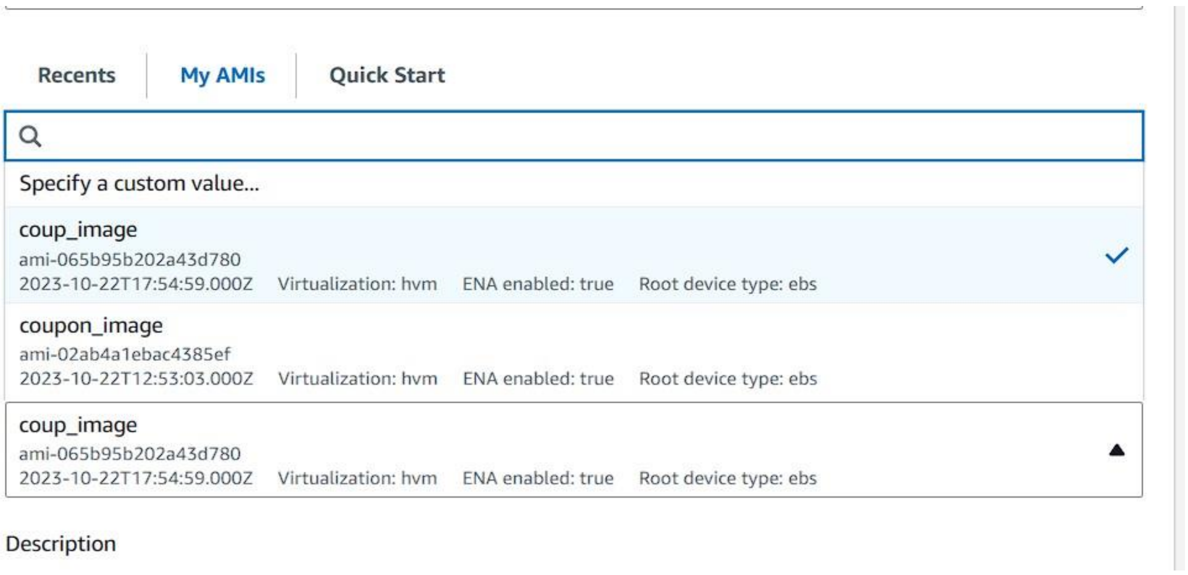
Instance details | Storage | Resource tags | Network interfaces | Advanced details

AMI ID	Instance type	Availability Zone	Key pair name
ami-065b95b202a43d780	t3.micro	-	kennedy

Security groups	Security group IDs
-	sg-0cc288fd52d79d715

- Under Configure storage, update the storage configuration. The default storage configuration is determined by the AMI and the instance type.
- choose Create Auto Scaling group.

Figure 14: image of the launch template



Recents | My AMIs | Quick Start

Specify a custom value...

coup_image
ami-065b95b202a43d780
2023-10-22T17:54:59.000Z Virtualization: hvm ENA enabled: true Root device type: ebs ✓

coupon_image
ami-02ab4a1ebac4385ef
2023-10-22T12:53:03.000Z Virtualization: hvm ENA enabled: true Root device type: ebs

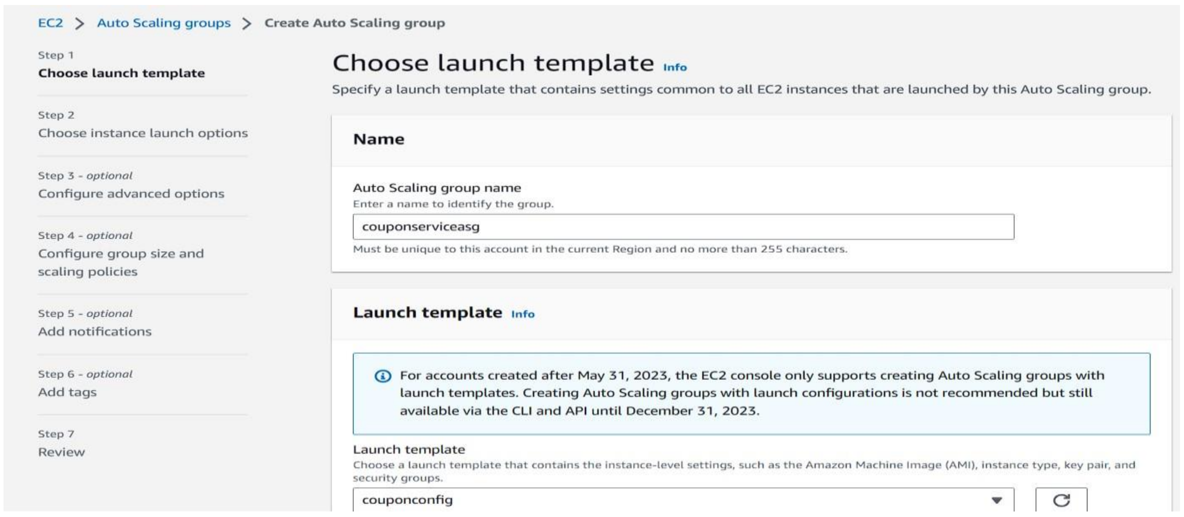
coup_image
ami-065b95b202a43d780
2023-10-22T17:54:59.000Z Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Create an Auto Scaling group

- On the Choose launch template or configuration page, enter a name for the Auto Scaling group. I.e couponserviceasg for group and couponconfig for template

Figure 15: create an auto scaling group


EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1
Choose launch template

Step 2
Choose instance launch options

Step 3 - *optional*
Configure advanced options

Step 4 - *optional*
Configure group size and scaling policies

Step 5 - *optional*
Add notifications

Step 6 - *optional*
Add tags

Step 7
Review

Choose launch template Info

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.

Name

Auto Scaling group name
Enter a name to identify the group.
couponserviceasg
Must be unique to this account in the current Region and no more than 255 characters.

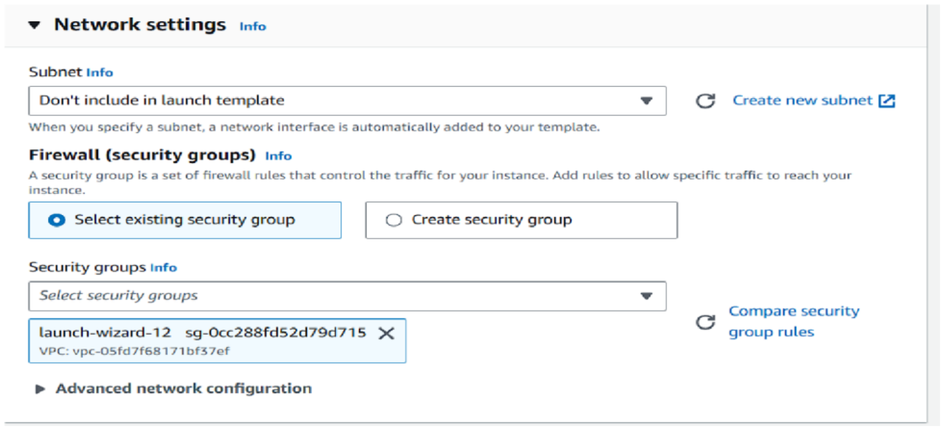
Launch template Info

For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.
couponconfig

- The launch template that you created is already selected for you.
  - For Launch template version, choose whether the Auto Scaling group uses the default, the latest, or a specific version of the launch template when scaling out.
- Choose Next to continue to the next step.
- On the Choose instance launch options page, if you're not using multiple instance types, you can skip the Instance type requirements section to use the EC2 instance type that is specified in the launch template.
- Under Network, for VPC, choose a VPC. The Auto Scaling group must be created in the same VPC as the security group you specified in your launch template.

Figure 16: configuring security groups


Network settings Info

Subnet Info

Don't include in launch template

When you specify a subnet, a network interface is automatically added to your template.

Create new subnet

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

[✓ Select existing security group] Create security group

Security groups Info

Select security groups

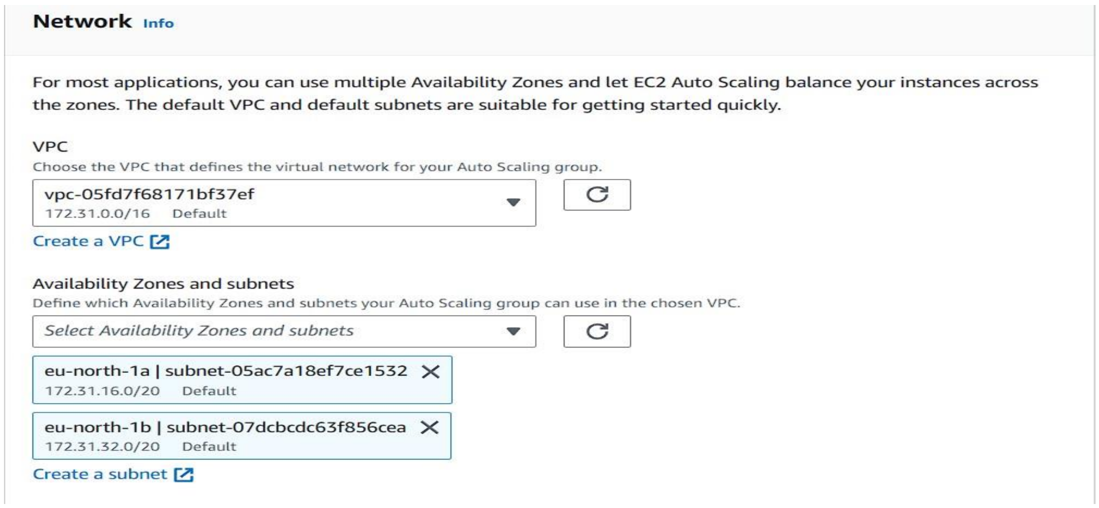
launch-wizard-12 sg-0cc288fd52d79d715
VPC: vpc-05fd7f68171bf37ef

Compare security group rules

Advanced network configuration

- For Availability Zones and subnets, choose one or more subnets in the specified VPC.

Figure 17: configuring subnets


Network Info

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC

Choose the VPC that defines the virtual network for your Auto Scaling group.

vpc-05fd7f68171bf37ef
172.31.0.0/16 Default

Create a VPC

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

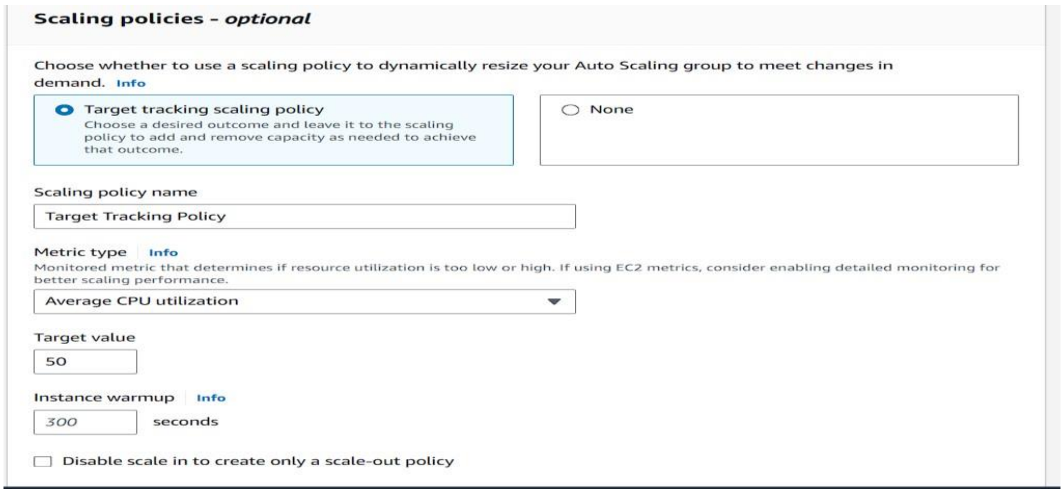
eu-north-1a | subnet-05ac7a18ef7ce1532
172.31.16.0/20 Default

eu-north-1b | subnet-07dcbcdc63f856cea
172.31.32.0/20 Default

Create a subnet

To go to the Configure group size and scaling policies page.

Figure 18: scaling policies configuration


Scaling policies - *optional*

Choose whether to use a scaling policy to dynamically resize your Auto Scaling group to meet changes in demand. Info

[✓ Target tracking scaling policy
Choose a desired outcome and leave it to the scaling policy to add and remove capacity as needed to achieve that outcome.] None

Scaling policy name

Target Tracking Policy

Metric type Info

Monitored metric that determines 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

50

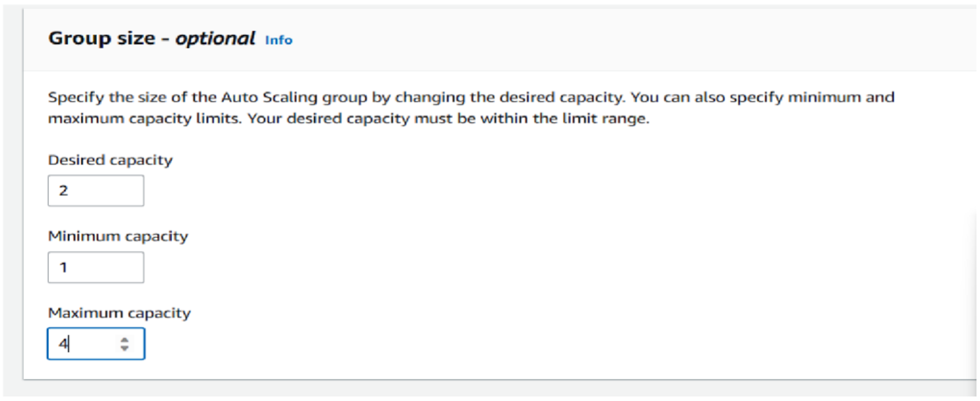
Instance warmup Info

300 seconds

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

- Under Group size, define the Desired capacity initial number of instances to launch immediately after the Auto Scaling group is created, Minimum capacity minimum number of instances, and Maximum capacity maximum number of instances.

Figure 19: Group size configuration



Group size - *optional* Info

Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum capacity limits. Your desired capacity must be within the limit range.

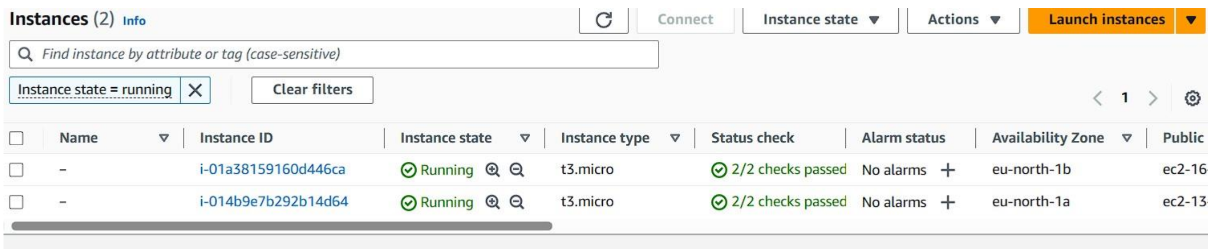
Desired capacity: 2

Minimum capacity: 1

Maximum capacity: 4

- Choose Skip to review.
- On the Review page, choose Create Auto Scaling group.

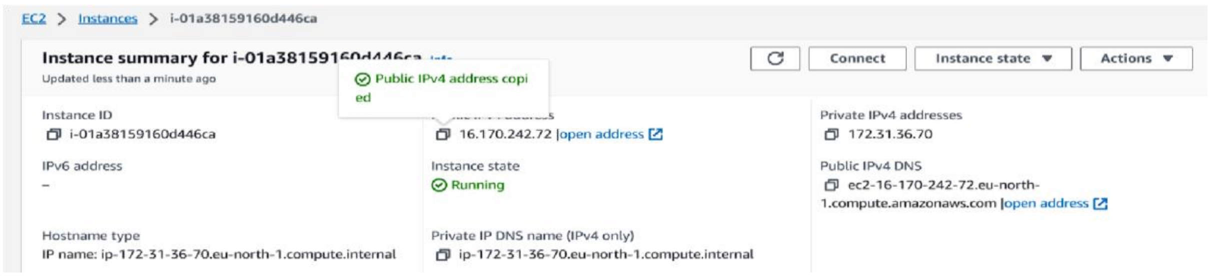
Figure 20: Auto scaling instance



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public
–	i-01a38159160d446ca	Running	t3.micro	2/2 checks passed	No alarms	eu-north-1b	ec2-16
–	i-014b9e7b292b14d64	Running	t3.micro	2/2 checks passed	No alarms	eu-north-1a	ec2-13

Testing your auto scaling group policy: to test the auto scaling, you must remote into an instance of the EC2 that has been instantiated by the auto scaling and running and execute a stress command. Instances created because of an auto scaling policy are unnamed or rather identify using a dash symbol. A demonstration of an auto scaling group policy is shown in the steps below

- click on an instantiated EC2 and copy the IPv4 address to securely remote into the instance
- **Figure 21: IP address of Autoscaling instance**



EC2 > Instances > i-01a38159160d446ca
Connect
Instance state
Actions
Public IPv4 address copied

Instance summary for i-01a38159160d446ca

Updated less than a minute ago

Instance ID: i-01a38159160d446ca

Public IPv4 address: 16.170.242.72 | open address

Private IPv4 addresses: 172.31.36.70

IPv6 address: –

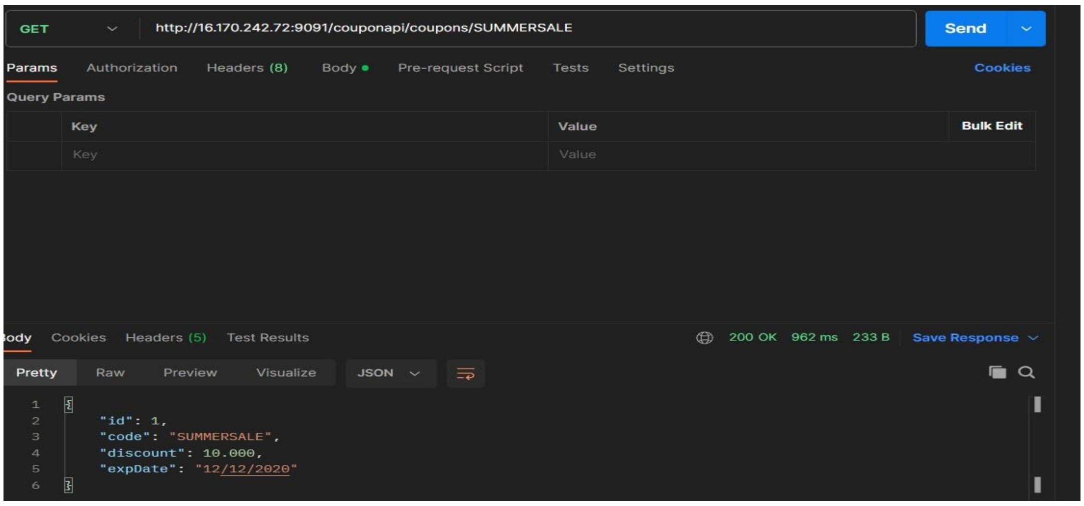
Instance state: Running

Public IPv4 DNS: ec2-16-170-242-72.eu-north-1.compute.amazonaws.com | open address

Hostname type: IP name: ip-172-31-36-70.eu-north-1.compute.internal

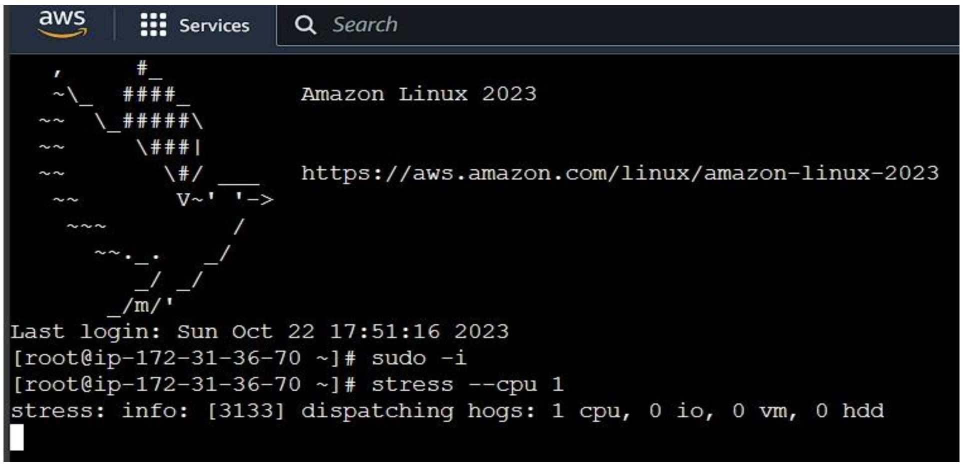
Private IP DNS name (IPv4 only): ip-172-31-36-70.eu-north-1.compute.internal

Figure 22: testing the Autoscaling instance through postman



```
{
    "id": 1,
    "code": "SUMMERSALE",
    "discount": 10.000,
    "expDate": "12/12/2020"
}
```

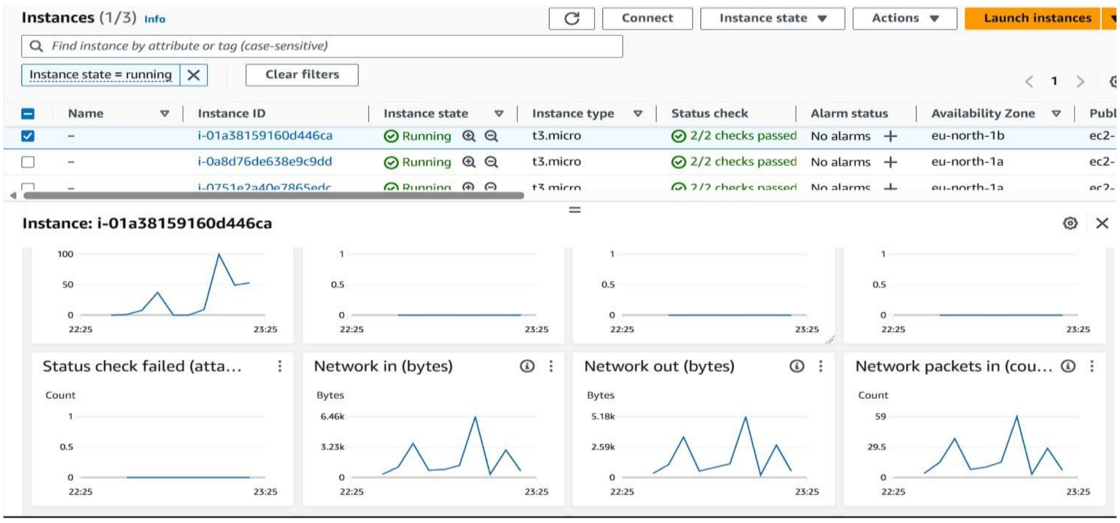
- using a remote client application login into the EC2 instance execute a stress command to stress the system cpu and triggered the instantiation of instances as per the group policy in place. stress - - cpu 1
- **Figure 23 using Linux command to stress instance in the AWS console**



```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Sun Oct 22 17:51:16 2023
[root@ip-172-31-36-70 ~]# sudo -i
[root@ip-172-31-36-70 ~]# stress --cpu 1
stress: info: [3133] dispatching hogs: 1 cpu, 0 io, 0 vm, 0 hdd
```

allow the number of seconds you setup in your configuration, and check to see that the number of instances running have increased due to the stress command executed

Figure 24: Monitoring stress and the new instances scaling


Instances (1/3) Info

Find instance by attribute or tag (case-sensitive)

Instance state = running

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Publ
✓	–	i-01a38159160d446ca	Running	t3.micro	2/2 checks passed	No alarms	eu-north-1b	ec2-
	–	i-0a8d76de638e9c9dd	Running	t3.micro	2/2 checks passed	No alarms	eu-north-1a	ec2-
	–	i-0751e2a40e7865edc	Running	t3.micro	2/2 checks passed	No alarms	eu-north-1a	ec2-

Instance: i-01a38159160d446ca

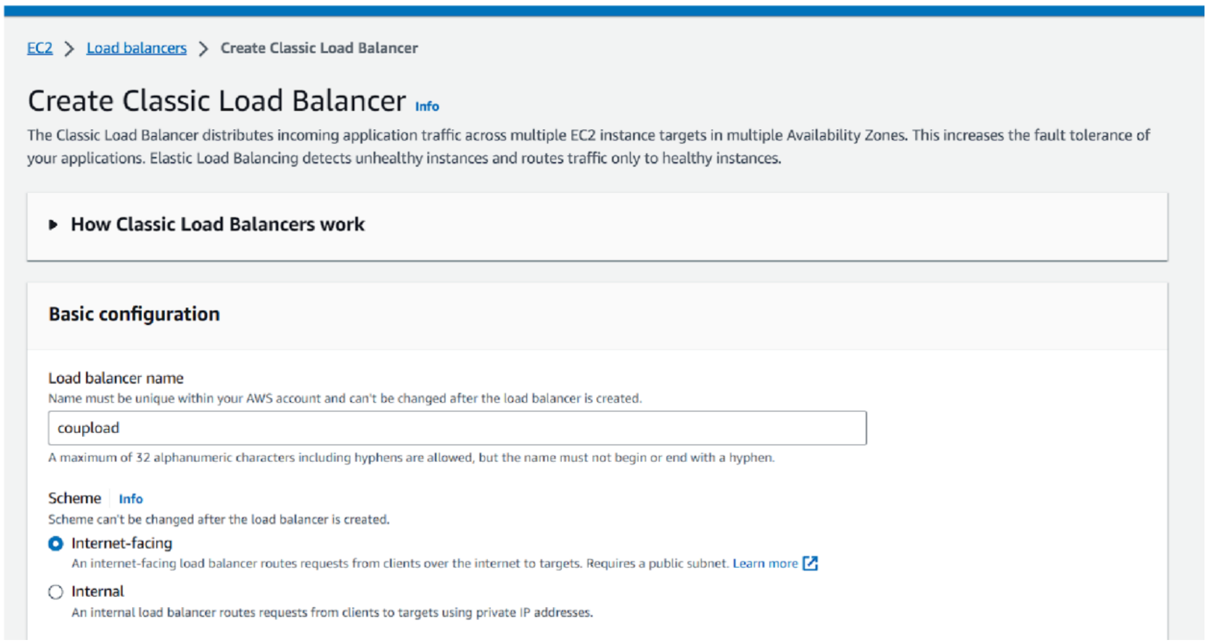
Test that the jar application is also running on the new instance. This is done using “postman”, a software application for testing REST API connections. Use the IPv4 address of the EC2 instance and test the application. After the GET command is send a RESPONSE of 200 is received, it indicates that the application is running perfectly.

Stop the stress command and view your instance. After the stress command has been issued, the instances that were created because of the auto scaling group policy gets terminated. When an instance is terminated, it is deleted from the ESB and it leaves the list of instances after a while. To terminate the stress command, press CTRL + C

CREATING A LOAD BALANCER

- On the navigation pane, under Load Balancing, choose Load Balancers.
- Choose Create Load Balancer.
- Expand the Classic Load Balancer section, then choose Create.
- Basic configuration
    - For Load balancer name, type a name for your load balancer.

      The name of your Classic Load Balancer must be unique within your set of Classic Load Balancers for the Region, can have a maximum of 32 characters, can contain only alphanumeric characters and hyphens, and must not begin or end with a hyphen.
    - For Scheme, select Internet-facing


EC2 > Load balancers > Create Classic Load Balancer
Create Classic Load Balancer Info
How Classic Load Balancers work
Basic configuration

The Classic Load Balancer distributes incoming application traffic across multiple EC2 instance targets in multiple Availability Zones. This increases the fault tolerance of your applications. Elastic Load Balancing detects unhealthy instances and routes traffic only to healthy instances.

Load balancer name
Name must be unique within your AWS account and can't be changed after the load balancer is created.
coupload
A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

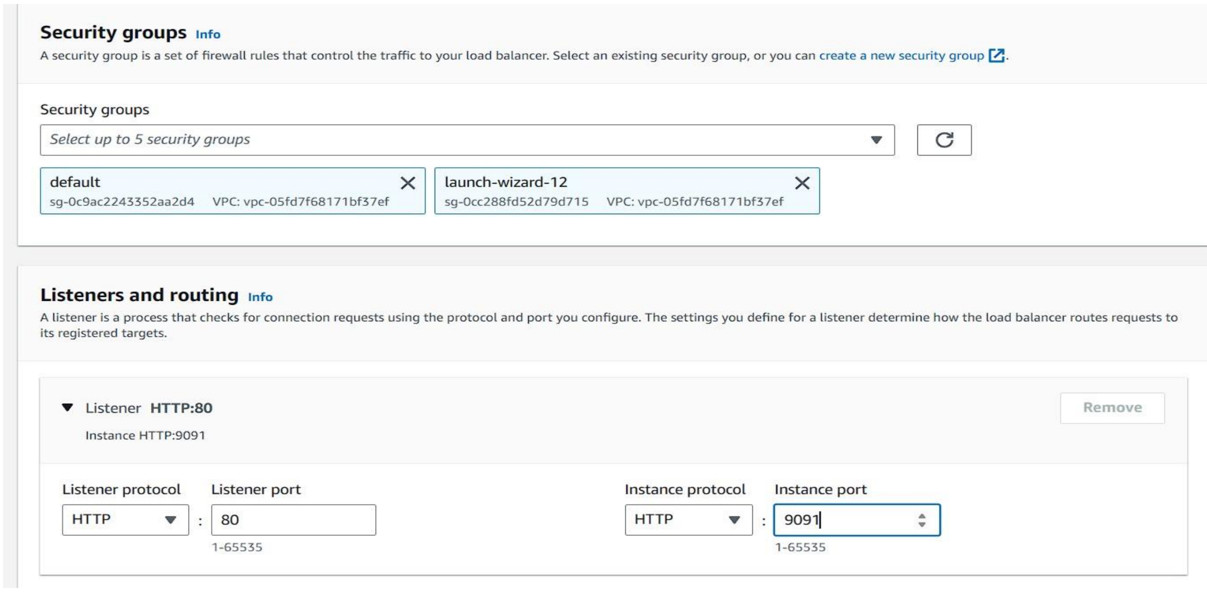
Scheme Info
Scheme can't be changed after the load balancer is created.
[✓ Internet-facing]
An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. Learn more
Internal
An internal load balancer routes requests from clients to targets using private IP addresses.

Figure 25: create a load balancer

- Network mapping
    - For VPC, select the same VPC that you selected for your instances.
    - For Mappings, first select an Availability Zone, then choose a public subnet from its available subnets. You can only select one subnet per Availability Zone. To improve the availability of your load balancer, select more than one Availability Zone and subnet.
- Security groups

  For Security groups, select an existing security group that is configured to allow the required HTTP traffic on port 80.

- Listeners and routing
  - For Listener, ensure the protocol is HTTP and the port is 80.
  - For Instance, ensure the protocol is HTTP and the port is 9091 for the couponservice.



Security groups Info

A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can create a new security group.

Security groups

Select up to 5 security groups

default
sg-0c9ac2243352aa2d4 VPC: vpc-05fd7f68171bf37ef

launch-wizard-12
sg-0cc288fd52d79d715 VPC: vpc-05fd7f68171bf37ef

Listeners and routing Info

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

Listener **HTTP:80**
Instance HTTP:9091

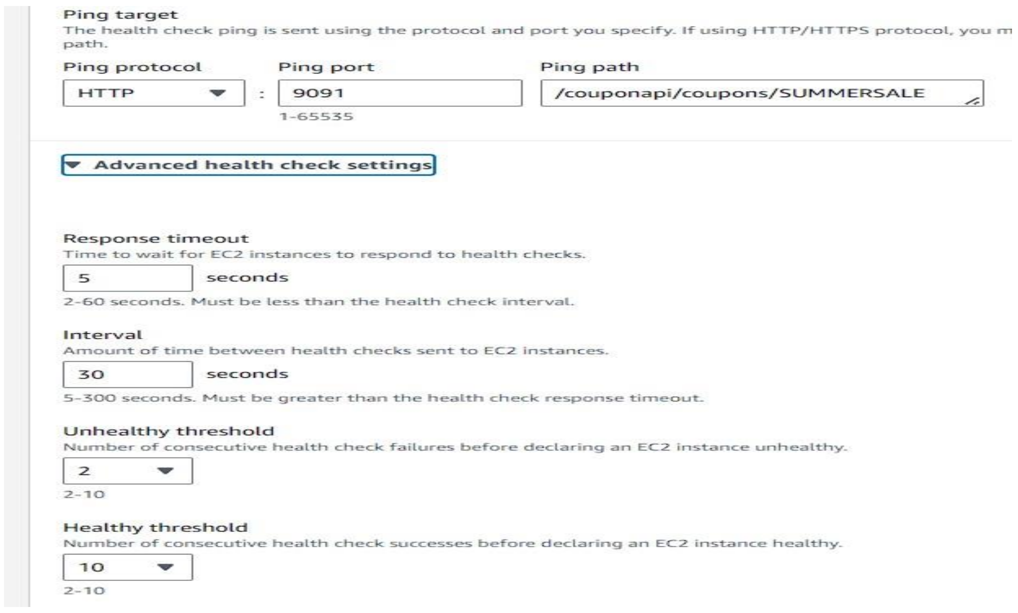
Remove

Listener protocol	Listener port	Instance protocol	Instance port
HTTP	80 (1-65535)	HTTP	9091 (1-65535)

Figure 26: security group configuration

- Health checks
  - For Ping Protocol, ensure the protocol is HTTP.
  - For Ping Port, ensure the port is 80.
  - For Ping Path, ensure the path is /couponapi/coupons/SUMMERSALE
  - For Advanced health check settings, use the values you desire but I recommend default

2. Figure 27 Advance health check settings configuration



Ping target

The health check ping is sent using the protocol and port you specify. If using HTTP/HTTPS protocol, you m path.

Ping protocol	Ping port	Ping path
HTTP	9091 (1-65535)	/couponapi/coupons/SUMMERSALE

Advanced health check settings

Response timeout
Time to wait for EC2 instances to respond to health checks.
5 seconds
2-60 seconds. Must be less than the health check interval.

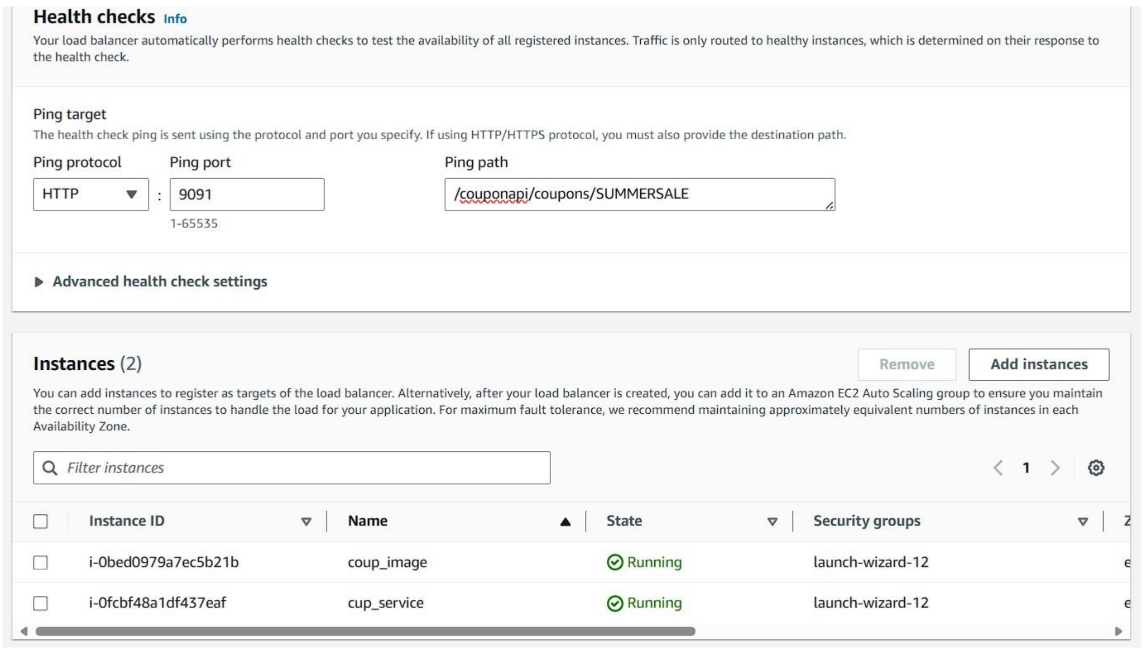
Interval
Amount of time between health checks sent to EC2 instances.
30 seconds
5-300 seconds. Must be greater than the health check response timeout.

Unhealthy threshold
Number of consecutive health check failures before declaring an EC2 instance unhealthy.
2
2-10

Healthy threshold
Number of consecutive health check successes before declaring an EC2 instance healthy.
10
2-10

- Instances
    - Select Add instances, to bring up the instance selection screen.
    - Under Available instances, you can select from the current instances that are available to the load balancer, based on the current network settings. In my case is coup service and cup _image.
    - select Confirm to add the instances to be registered to the load balancer

Figure28: Choosing Instances from the coupon load balancer



Health checks Info

Your load balancer automatically performs health checks to test the availability of all registered instances. Traffic is only routed to healthy instances, which is determined on their response to the health check.

Ping target

The health check ping is sent using the protocol and port you specify. If using HTTP/HTTPS protocol, you must also provide the destination path.

Ping protocol	Ping port	Ping path
HTTP	9091 (1-65535)	/couponapi/coupons/SUMMERSALE

▸ Advanced health check settings

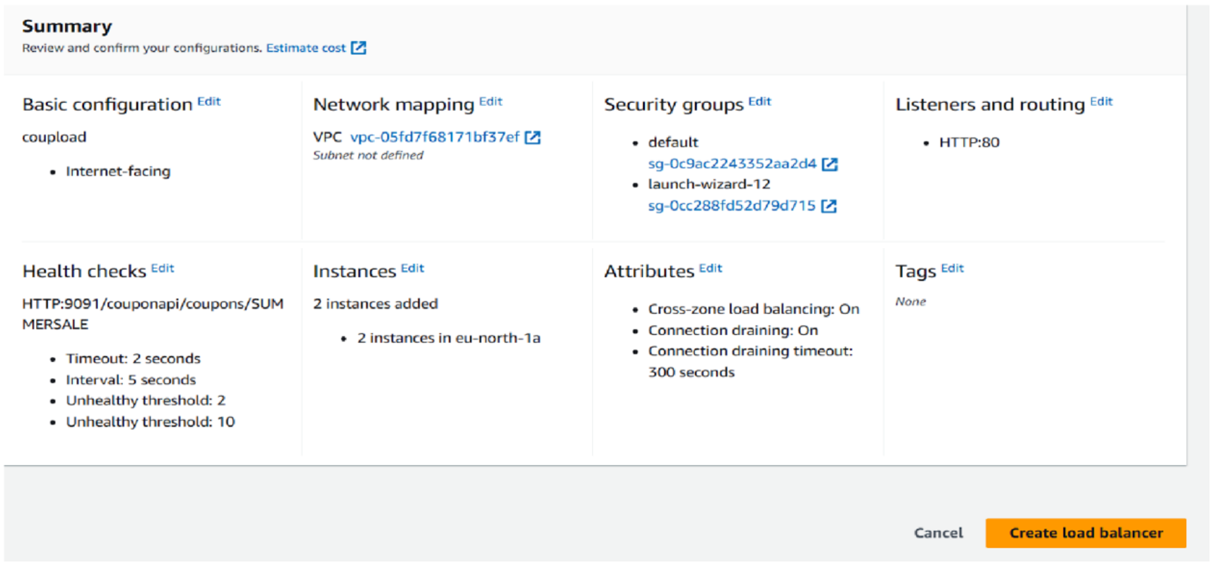
Instances (2)

You can add instances to register as targets of the load balancer. Alternatively, after your load balancer is created, you can add it to an Amazon EC2 Auto Scaling group to ensure you maintain the correct number of instances to handle the load for your application. For maximum fault tolerance, we recommend maintaining approximately equivalent numbers of instances in each Availability Zone.

Instance ID	Name	State	Security groups	Z
i-0bed0979a7ec5b21b	coup_image	Running	launch-wizard-12	e
i-0fcbf48a1df437eaf	cup_service	Running	launch-wizard-12	e

- Attributes
    - For Enable cross-zone load balancing, enable connection draining, and Timeout keep the default values if you want.
- Load balancer tags (optional)
    - To add another tag, select Add new tag then input your values into the Key field, and optionally the Value field.
    - To remove an existing tag, select Remove next to the tag you want to remove.
- Summary and creation
    - If you need to change any settings, select Edit next to the setting needing to be changed.
    - After you're satisfied with all the settings shown in the summary, select Create load balancer to begin creation of your load balancer.
    - On the final creation page, select View load balancer to view your load balancer in the Amazon EC2 console.

Figure 29: load balancer summary



Summary

Review and confirm your configurations. Estimate cost

Basic configuration Edit

coupload

- Internet-facing

Network mapping Edit

VPC vpc-05fd7f68171bf37ef

Subnet not defined

Security groups Edit

- default
  sg-0c9ac2243352aa2d4
- launch-wizard-12
  sg-0cc288fd52d79d715

Listeners and routing Edit

- HTTP:80

Health checks Edit

HTTP:9091/couponapi/coupons/SUMMERSALE

- Timeout: 2 seconds
- Interval: 5 seconds
- Unhealthy threshold: 2
- Unhealthy threshold: 10

Instances Edit

2 instances added

- 2 instances in eu-north-1a

Attributes Edit

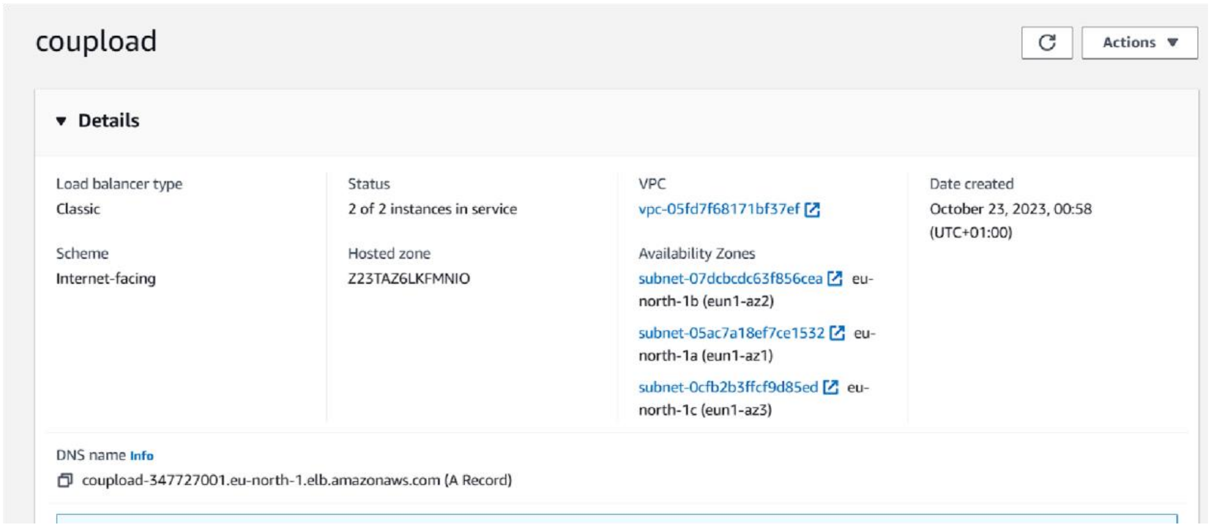
- Cross-zone load balancing: On
- Connection draining: On
- Connection draining timeout: 300 seconds

Tags Edit

None

- Verify
  - Select your new load balancer.
  - On the Target instances tab, check the health status column. After at least one of your EC2 instances is In-service, you can test your load balancer.
  - In the Details section, copy the load balancers DNS name, which would look like coupload-347727001.eu-north-1.elb.amazonaws.com

Figure 30 : load balancer details



coupload

Details

Load balancer type
Classic

Scheme
Internet-facing

Status
2 of 2 instances in service

Hosted zone
Z23TAZ6LKFMNIO

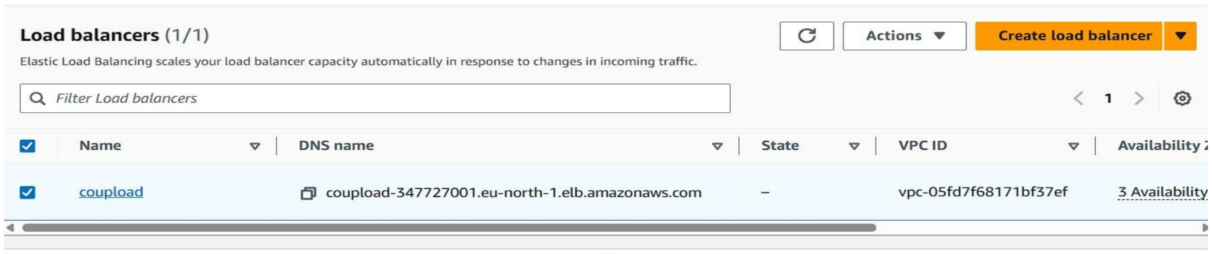
VPC
vpc-05fd7f68171bf37ef

Availability Zones
subnet-07dcbcdc63f856cea eu-north-1b (eun1-az2)
subnet-05ac7a18ef7ce1532 eu-north-1a (eun1-az1)
subnet-0cfb2b3ffcf9d85ed eu-north-1c (eun1-az3)

Date created
October 23, 2023, 00:58 (UTC+01:00)

DNS name Info
coupload-347727001.eu-north-1.elb.amazonaws.com (A Record)

Figure 31:The load balancer



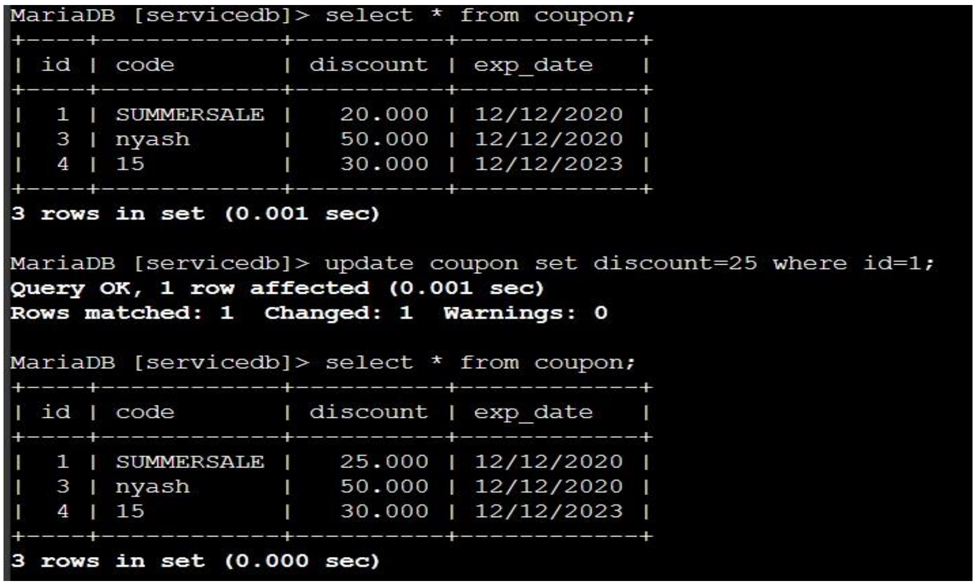
Load balancers (1/1)

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

Name	DNS name	State	VPC ID	Availability
coupload	coupload-347727001.eu-north-1.elb.amazonaws.com	–	vpc-05fd7f68171bf37ef	3 Availability

To see the difference in the load balancer we need to change the database(mariadb) from one instance in the AWS.

Figure 31: updating database of one instance



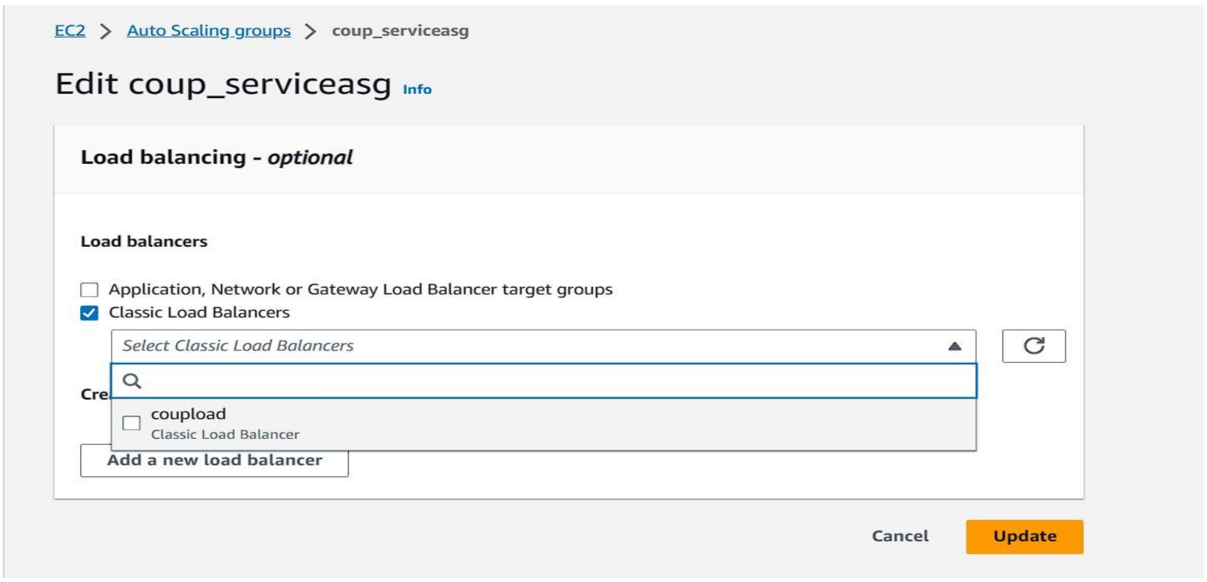
```
MariaDB [servicedb]> select * from coupon;
+----+------------+----------+------------+
| id | code       | discount | exp_date   |
+----+------------+----------+------------+
|  1 | SUMMERSALE |   20.000 | 12/12/2020 |
|  3 | nyash      |   50.000 | 12/12/2020 |
|  4 | 15         |   30.000 | 12/12/2023 |
+----+------------+----------+------------+
3 rows in set (0.001 sec)

MariaDB [servicedb]> update coupon set discount=25 where id=1;
Query OK, 1 row affected (0.001 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [servicedb]> select * from coupon;
+----+------------+----------+------------+
| id | code       | discount | exp_date   |
+----+------------+----------+------------+
|  1 | SUMMERSALE |   25.000 | 12/12/2020 |
|  3 | nyash      |   50.000 | 12/12/2020 |
|  4 | 15         |   30.000 | 12/12/2023 |
+----+------------+----------+------------+
3 rows in set (0.000 sec)
```

Paste your load balancers DNS name into the address field of a public internet connected web browser. If your load balancer is functioning correctly, you will see the default page of your server.
Or copy the DNS name to postman to get the 200ms ok.

Attach the load balancer to the Auto scaling group his helps you keep up with the demand from your users while keeping costs down. Amazon's ELB ties in well with auto scaling groups automatically load balancing servers that have been brought up as the result of an auto scaling action. **Figure 32: attaching load balancer to the autoscaling group**


EC2 > Auto Scaling groups > coup_serviceasg

Edit coup_serviceasg Info

Load balancing - *optional*

Load balancers

☐ Application, Network or Gateway Load Balancer target groups
☑ Classic Load Balancers

Select Classic Load Balancers

☐ coupload
Classic Load Balancer

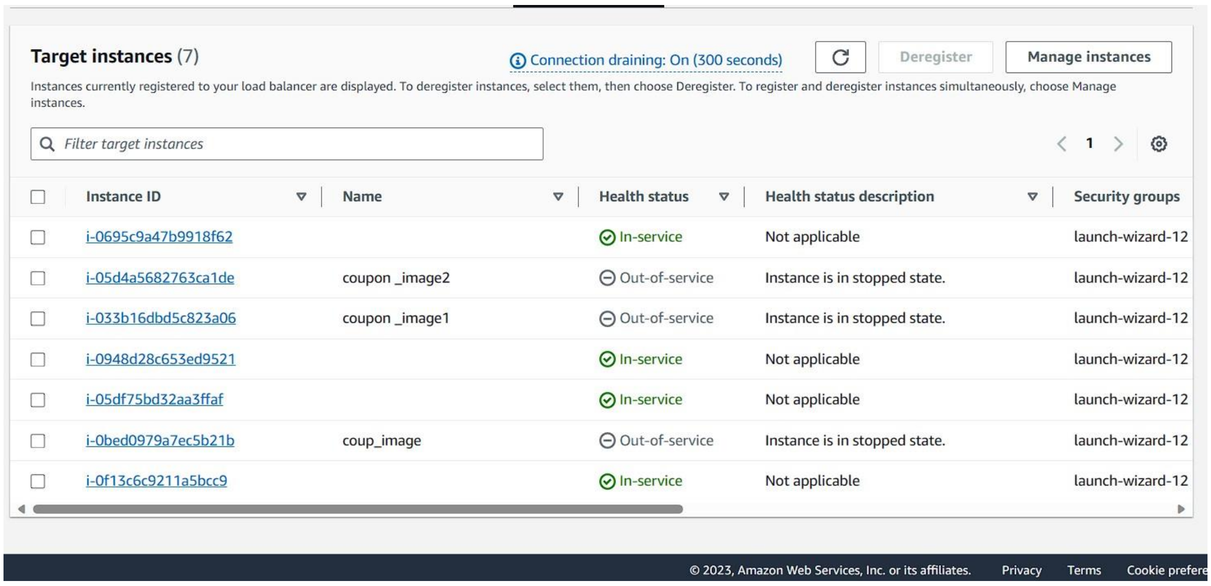
Cre

Add a new load balancer

Cancel Update

by attaching the load balancer to the auto scaling group next time when the EC2 are stress the auto scaling instance will launch from the load balancer instance see on the figure bellow

Figure 33: shows auto scaling instance launching in the load balancer


Target instances (7)

Connection draining: On (300 seconds)

Deregister | Manage instances

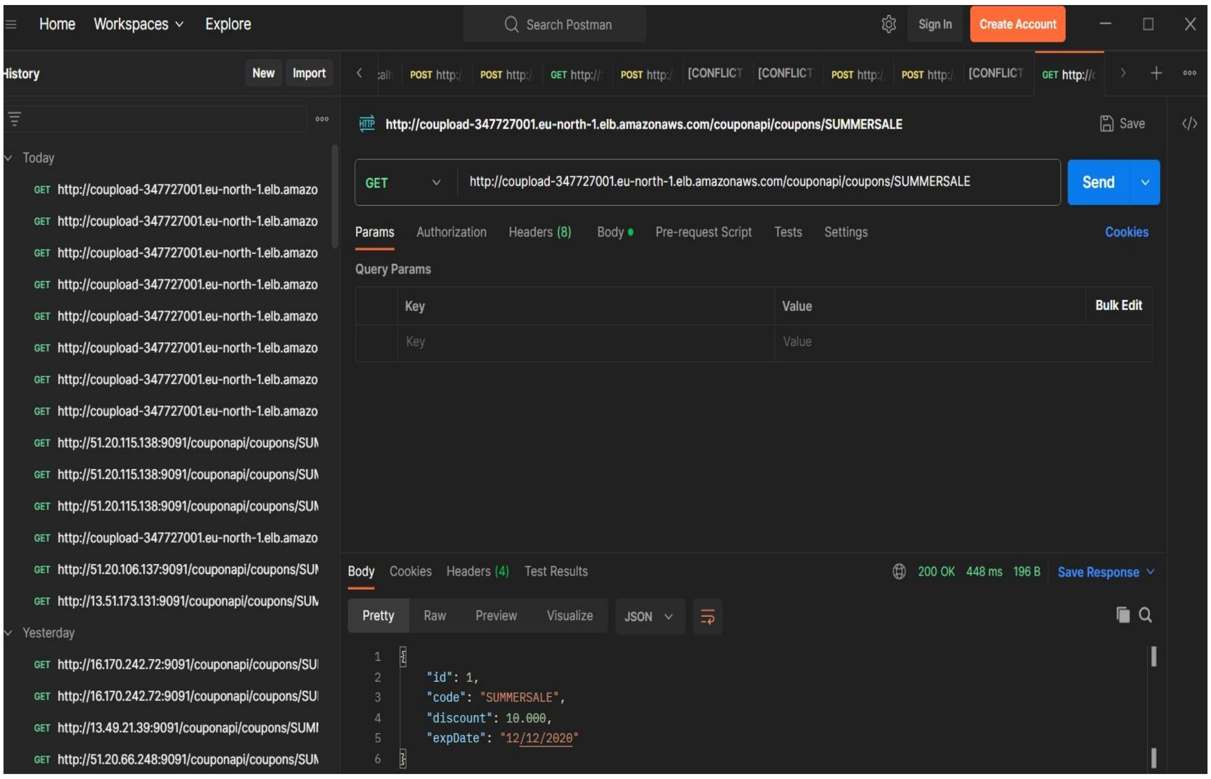
Instances currently registered to your load balancer are displayed. To deregister instances, select them, then choose Deregister. To register and deregister instances simultaneously, choose Manage instances.

Filter target instances

Instance ID	Name	Health status	Health status description	Security groups
i-0695c9a47b9918f62		In-service	Not applicable	launch-wizard-12
i-05d4a5682763ca1de	coupon _image2	Out-of-service	Instance is in stopped state.	launch-wizard-12
i-033b16dbd5c823a06	coupon _image1	Out-of-service	Instance is in stopped state.	launch-wizard-12
i-0948d28c653ed9521		In-service	Not applicable	launch-wizard-12
i-05df75bd32aa3ffaf		In-service	Not applicable	launch-wizard-12
i-0bed0979a7ec5b21b	coup_image	Out-of-service	Instance is in stopped state.	launch-wizard-12
i-0f13c6c9211a5bcc9		In-service	Not applicable	launch-wizard-12

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie prefere

Figure 32: testing the load balancer from one instance



Today

GET http://coupload-347727001.eu-north-1.elb.amazo
GET http://coupload-347727001.eu-north-1.elb.amazo
GET http://coupload-347727001.eu-north-1.elb.amazo
GET http://coupload-347727001.eu-north-1.elb.amazo
GET http://coupload-347727001.eu-north-1.elb.amazo
GET http://coupload-347727001.eu-north-1.elb.amazo
GET http://coupload-347727001.eu-north-1.elb.amazo
GET http://coupload-347727001.eu-north-1.elb.amazo
GET http://51.20.115.138:9091/couponapi/coupons/SUN
GET http://51.20.115.138:9091/couponapi/coupons/SUN
GET http://51.20.115.138:9091/couponapi/coupons/SUN
GET http://coupload-347727001.eu-north-1.elb.amazo
GET http://51.20.106.137:9091/couponapi/coupons/SUN
GET http://13.51.173.131:9091/couponapi/coupons/SUN

Yesterday

GET http://16.170.242.72:9091/couponapi/coupons/SU
GET http://16.170.242.72:9091/couponapi/coupons/SU
GET http://13.49.21.39:9091/couponapi/coupons/SUMI
GET http://51.20.66.248:9091/couponapi/coupons/SUN

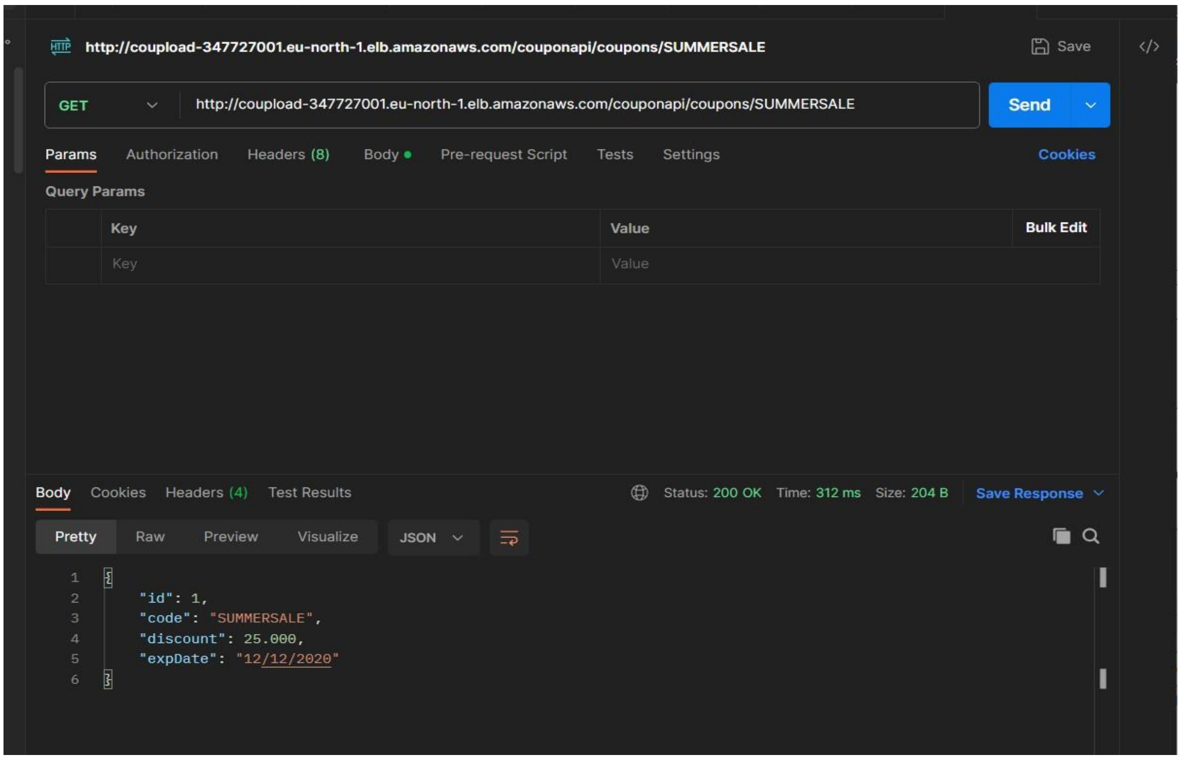
http://coupload-347727001.eu-north-1.elb.amazonaws.com/couponapi/coupons/SUMMERSALE

GET http://coupload-347727001.eu-north-1.elb.amazonaws.com/couponapi/coupons/SUMMERSALE

Body Cookies Headers (4) Test Results — 200 OK 448 ms 196 B

```
{
    "id": 1,
    "code": "SUMMERSALE",
    "discount": 10.000,
    "expDate": "12/12/2020"
}
```

Figure 3: testing the load balancer from another instance



http://coupload-347727001.eu-north-1.elb.amazonaws.com/couponapi/coupons/SUMMERSALE

GET http://coupload-347727001.eu-north-1.elb.amazonaws.com/couponapi/coupons/SUMMERSALE

Body Cookies Headers (4) Test Results — Status: 200 OK Time: 312 ms Size: 204 B

```
{
    "id": 1,
    "code": "SUMMERSALE",
    "discount": 25.000,
    "expDate": "12/12/2020"
}
```

Evaluation

After I created an instance, connected to it. Installed mariadb, gave it the password created the databases from mariadb gave the tables (coupon and product). We later installed java in that instance.

Created a s3 bucket where we uploaded the java file made it accessible publicly and went back to the instance wget the java file then run the java file run it. Copied the EC2 public address and the couponapi/cooupon posted the summersales to postman app for test.

Installed rc.local to enable the jar file automatically when the instance is active.

Create an AMI from the instance and get the image . run the images. Open auto scaling group of the two instance. Apply a stress package from one the instance load The EC2 scaled horizontally and duplicate to the desired cpu parameter to help up with the load.

Create an load balancing template add the template to the auto scaling on the network group use the existing network group from the instance. After finishing up creating load ballancing group,go to one off the EC2 and change the mariadb

Copy the load balancer URL to the postman and change on the EC2 observe the tables to see the deference.

AWS Auto Scaling service can be considered as regionally limited, such that, it is only effective in one region that are selected during the setup of group policies and it's not possible to use auto scaling policies across resources in different regions. It may become a challenge setting up auto scaling policies for multi-regions applications, therefore AWS auto scaling policies will need to be set up separately for each region

The auto-scaling worked efficiently adjusted the number of instances to handle variable workloads, ensuring optimal performance and Auto scaling effectively maintained the desired performance levels by adding or removing instances as needed hence completing all the task required.

key limitations

Defininining Scaling Policies: Configuring effective scaling policies can be challenging. You need to set thresholds that trigger scaling actions, and choosing the right thresholds can require careful monitoring and fine-tuning. I.e cpu percentage or bytes.

Testing Auto Scaling:To demonstrate auto scaling in operation, you can simulate increased traffic to your application. This can be done by using load testing tools, sending additional requests manually, or creating a custom script to generate traffic likes linux stress test or other script languages like python script.

Observe Auto Scaling in Action:As the load on your application increases and breaches the alarm thresholds, you should see the Auto Scaling Group automatically launch new instances based on your scaling policies time i.e 300 sec you will have to wait for that time to see the outcome and another instance can be added or existing one can be stopped.

Scaling Speed**:** AWS Auto Scaling may not always respond instantly to rapid changes in traffic. There can be a delay in launching new instances, which can lead to temporary performance issues during sudden traffic spikes.

cost Management: Without careful monitoring and adjustment of your scaling policies, AWS Auto Scaling can lead to increased costs, while using free tier especially if it scales up resources unnecessarily.

References

1. Amazon Glossary of Terms

Available at: https://docs.aws.amazon.com/general/latest/gr/glos-chap.html [cited 25 October 2023]

2. Developing Cloud Native Applications : Module 1 Glossary: Introduction to Cloud Native Available at: https://learning.edx.org/course/course-v1:IBM+CC0101EN+2T2021/home [cited 25October 2023] 3. Amazon Auto Scaling

Available at: https://docs.aws.amazon.com/autoscaling/ec2/userguide/what-is-amazonec2-autoscaling.html [cited 25 October 2023}

4.Intergrating Load balancer with auto scaling groups

Available at: https://www.loadbalancer.org/blog/integrating-your-load-balancer-with-auto-scalinggroups-in-ec2[cited 25 October 2023}

5 creating an EC2 Instance with Lambda in AWS

Available at: https://www.pluralsight.com/cloud-guru/labs/aws/creating-an-ec2-instance-withlambda-in-AWS[cited 25 October 2023}

6 Amazon Load balancing

Available at: https://aws.amazon.com/elasticloadbalancing/[cited 25 October 2023}

7 Postman

Available at: https://linuxize.com/post/how-to-install-postman-on-ubuntu-20-04/[cited 25 October 2023}