

Assignment 4:

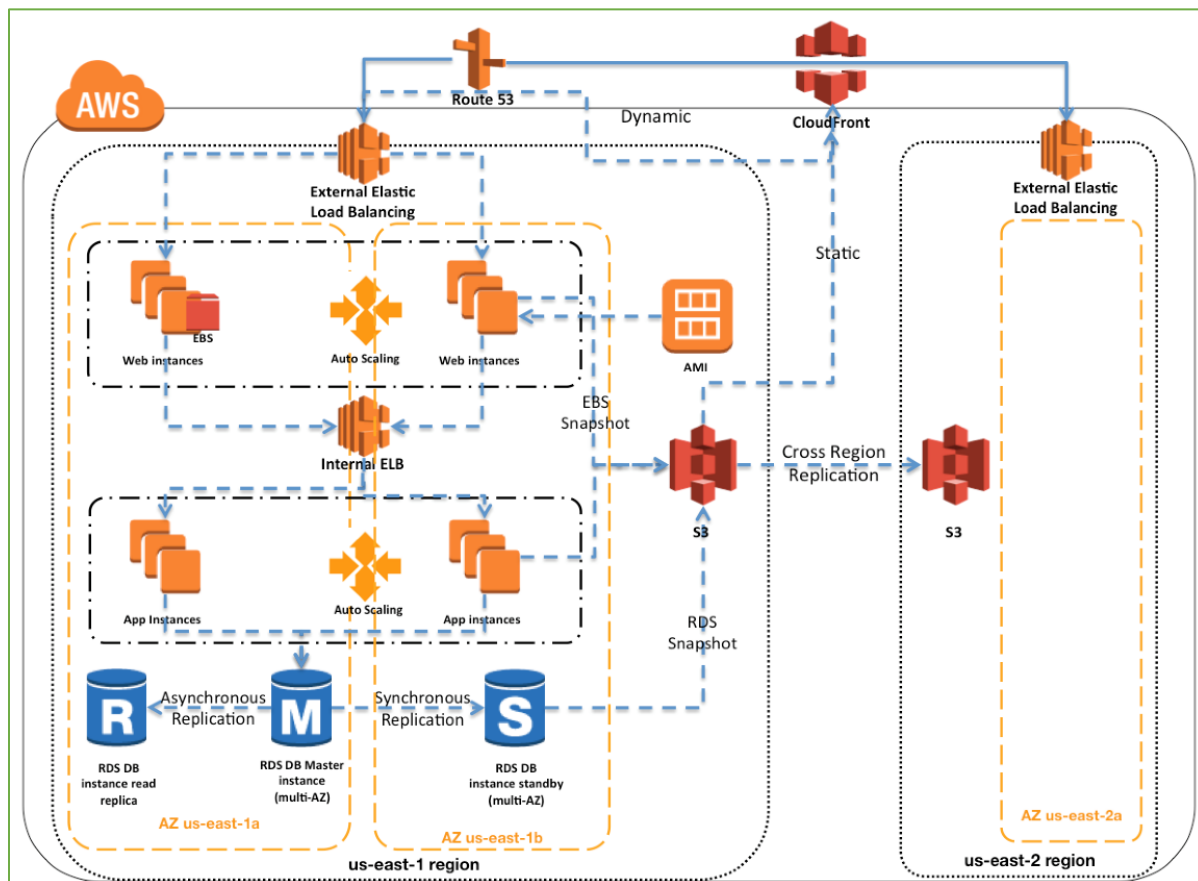
Set-up highly available, fault-tolerant, elastic and scalable architecture using cloud services to deploy web application which reactively scale-in or scale-out based on the demand.

Below is the brief description about fault-tolerant and scalable architecture.

High Availability & Fault Tolerance Architecture

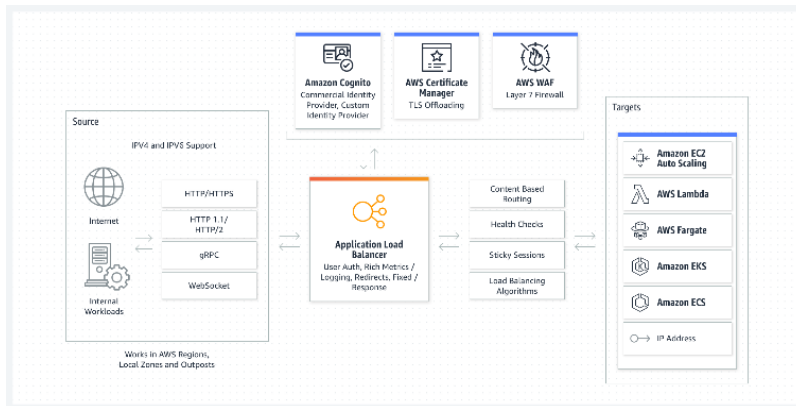
1. Amazon Web Services provides services and infrastructure to build reliable, fault-tolerant, and highly available systems in the cloud.
2. Fault-tolerance defines the ability for a system to remain in operation even if some of the components used to build the system fail.
3. Most of the higher-level services, such as S3, SimpleDB, SQS, and ELB, have been built with fault tolerance and high availability in mind.
4. Services that provide basic infrastructure, such as EC2 and EBS, provide specific features, such as availability zones, elastic IP addresses, and snapshots, that a fault-tolerant and highly available system must take advantage of and use correctly.

Example:



Load balancer can be configured in different ways based on requirements.

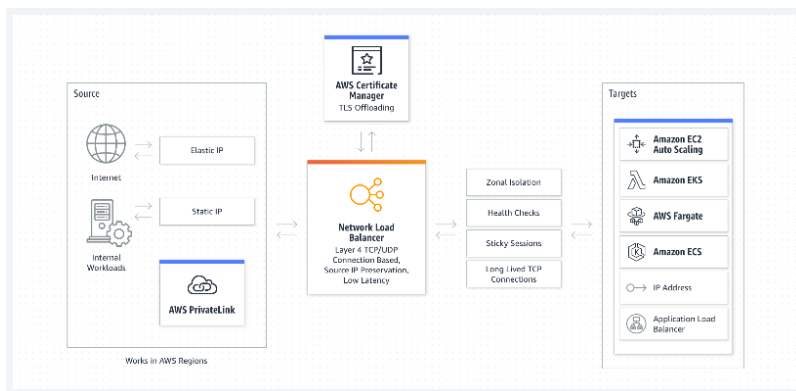
Application load Balancer



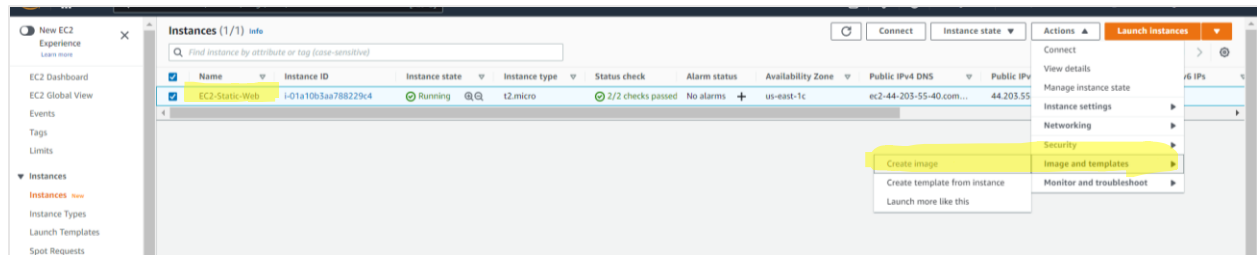
Gateway load balancer



Network load balancer.



1. Create EC2 instance. Install Apache2 and other required software.
2. Check if the installed software services are running fine and default apache2 page displays.
3. Check if you have installed mysql and its running fine and will be able to connect.
4. Run a sample static web app on EC2
5. Create an AMI from existing EC2. Amazon Machine Image (AMI) provides a Template that can be used to define the service instances.



Below is the list of AMI created.

Name	AMI ID	AMI name	Source	Owner	Visibility	Status	Creation date
-	ami-02a891ca7cb6c5f9c	ec2-ubuntu-apache-php-high-a...	862781485761/ec2-ubuntu-apache-ph...	862781485761	Private	Pending	2022/10/09 14:50 GMT+5:30

Create an instance from the saved AMI

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
myWebFromAnImage	i-0e5b21d81dfa38283	Running	t2.micro	-	No alarms	us-east-1c	ec2-18-204-204-244.co...	18.204.204.244
EC2-Static-Web	i-01a10b3aa788229c4	Running	t2.micro	2/2 checks passed	No alarms	us-east-1c	ec2-44-203-55-40.com...	44.203.55.40

Access the web application to test if the created instance is running with webserver.

Load Test
RDS

Meta-Data	Value
Instanceid	i-0e5b21d81dfa38283
Availability Zone	us-east-1c

Current CPU Load: 1%

Note: Now we use the AMI and start with creation of target group, ELB, Launch config and finally auto scaling group.

Create a VPC spanning across 2 Availability Zones

This VPC should have 2 public and 2 private subnets

1. Click Subnet and create your Subnet with:
2. Public Subnet 1 and Public Subnet 2 valid Name & VPC.
3. Valid Subnet range which is valid IPv4 CIDR Block.
4. Repeat steps 2 & 3, with Private Subnet too.

Create a Target group for the ELB

EC2 > Target groups > Create target group

Step 1
Specify group details

Step 2
Register targets

Specify group details

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

Basic configuration

Settings in this section cannot be changed after the target group is created.

Choose a target type

☒ **Instances**

- Supports load balancing to instances within a specific VPC.
- Facilitates the use of [Amazon EC2 Auto Scaling](#) to manage and scale your EC2 capacity.

☐ **IP addresses**

- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice based architectures, simplifying inter-application communication.
- Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.

☐ **Lambda function**

- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.

☐ **Application Load Balancer**

- Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
- Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

Target group name

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

ProtocolPort

HTTP: 80

VPC

Select the VPC with the instances that you want to include in the target group.

ecom-mahesh-vpc
vpc-049dbf464ebb1eb42
IPv4: 10.0.0.0/16

Protocol version

☒ HTTP1
Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.

☐ HTTP2
Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.

☐ gRPC
Send requests to targets using gRPC. Supported when the request protocol is gRPC.

Health checks

The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol

HTTP

Health check path

Use the default path of "/" to ping the root, or specify a custom path if preferred.

/

Up to 1024 characters allowed.

► Advanced health check settings

► Tags - optional

Consider adding tags to your target group. Tags enable you to categorize your AWS resources so you can more easily manage them.

Cancel

Next

EC2 > Target groups

Target groups (1) [info](#)

Search or filter target groups

Actions

Create target group

1

<input type="checkbox"/>	Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
<input type="checkbox"/>	myTargetGroup	arn:aws:elasticloadbalancin...	80	HTTP	Instance	None associated	vpc-049dbf464eb1eb42

Create Load Balancer ELB, associate with Target Group.

EC2 > Load balancers > Create Application Load Balancer

Create Application Load Balancer [info](#)

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

► How Application Load Balancers work

Basic configuration

Load balancer name

Name must be unique within your AWS account and cannot be changed after the load balancer is created.

myLoadBalancer

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme [info](#)

Scheme cannot 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.

IP address type [info](#)

Select the type of IP addresses that your subnets use.

☒ IPv4

Recommended for internal load balancers.

☐ Dualstack

Includes IPv4 and IPv6 addresses.

Network mapping [info](#)

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC [info](#)

Select the virtual private cloud (VPC) for your targets. Only VPCs with an internet gateway are enabled for selection. The selected VPC cannot be changed after the load balancer is created. To confirm the VPC for your targets, view your target groups

ecom-mahesh-vpc

vpc-049dbf464eb1eb42

IPv4: 10.0.0.0/16

Mappings [info](#)

Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

☒ us-east-1a

Subnet

subnet-036b3e730b743d0a0

ecom-mahesh-subnet-public1-us-east-1a

IPv4 settings

Assigned by AWS

☒ us-east-1b

Subnet

subnet-001e8d3ff75906038

ecom-mahesh-subnet-public2-us-east-1b

IPv4 settings

Assigned by AWS

Security groups Info

A security group is a set of firewall rules that control the traffic to your load balancer.

Security groups

Select up to 5 security groups

Create new security group [Create new security group](#)

default sg-029c08b3fad820998

VPC: vpc-049dbf464ebbb1eb42

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

Remove

Protocol

Port

HTTP

:

80

1-65535

Default action Info

Forward to myTargetGroup

Target type: Instance, IPv4

HTTP

Create target group [Create target group](#)

Listener tags - *optional*

Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

Add listener tag

You can add up to 50 more tags.

Create Load Balancer

Actions ▼

search: myLoadBalancer

Add filter

Name	DNS name	State	VPC ID	Availability Zones	Type	Created At	Monitoring
myLoadBalancer	myLoadBalancer-751546970...	Provisioning	vpc-049dbf464ebbb1eb42	us-east-1b, us-east-1a	application	October 10, 2022 at 3:22:35 ...	

Create Launch config

EC2 > Launch configurations > Create launch configuration

Create launch configuration Info

⚠

Instead of using launch configurations to create your EC2 Auto Scaling groups, we recommend that you use launch templates and make use of the Auto Scaling guidance option. For more information on migrating launch configurations and using launch templates, see the [documentation](#).

Create launch template

Launch configuration name

Name

myLaunchConfig

Amazon machine image (AMI) Info

AMI

ec2-ubuntu-apache-php-high-avail-app

Instance type Info

Instance type

t2.small (1 vCPUs, 2 GiB, EBS Only)

Choose instance type

Internal

Assign a security group

☐ Create a new security group
☒ Select an existing security group

Security groups Copy to new View rules

< 1 >

<input type="checkbox"/>	Security group ID	Name	VPC ID	Description
<input type="checkbox"/>	sg-0a76081d257df7ccd	AutoScaling-Security-Group-mh	vpc-08d745f8388d541cc	AutoScaling-Security-Group-2 (2022-10-09T19:06:11.318Z)
<input checked="" type="checkbox"/>	sg-029c08b3fad820998	default	vpc-049dbf464ebb1eb42	default VPC security group
<input type="checkbox"/>	sg-0b76c754c5672ac61	default	vpc-08d745f8388d541cc	default VPC security group

⚠ 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.

Key pair (login) [Info](#)

Key pair options

Existing key pair

☒ I acknowledge that I have access to the selected private key file (ec2static.pem), and that without this file, I won't be able to log into my instance.

Cancel Create launch configuration

EC2 > Launch configurations

Launch configurations (1) [Info](#) Actions Copy to launch template Create launch configuration

< 1 >

<input type="checkbox"/>	Name	AMI ID	Instance type	Spot price	Creation time
<input type="checkbox"/>	myLaunchConfig	ami-02a891ca7cb6c5f9c	t2.small	-	Mon Oct 10 2022 03:27:23 GMT+0530 (India Standard Time)

Create Autoscaling Group (ASG) using Launch config. Selected the create ASG from drop down.

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1: Choose launch template or configuration

Choose launch template or configuration [Info](#)

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group. If you currently use launch configurations, you might consider migrating to launch templates.

Name

Auto Scaling group name

Enter a name to identify the group.

Must be unique to this account in the current Region and no more than 255 characters.

Launch configuration [Info](#) Switch to launch template

⚠ Instead of using launch configurations to create your EC2 Auto Scaling groups, we recommend that you use launch templates and make use of the Auto Scaling guidance option. For more information on migrating launch configurations and using launch templates, [see the documentation](#).

Launch configuration

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

[Create a launch configuration](#)

Launch configuration	AMI ID	Date created
myLaunchConfig	ami-02a891ca7cb6c5f9c	Mon Oct 10 2022 03:27:23 GMT+0530 (India Standard Time)
Security groups	Instance type	Key pair name
sg-029c08b3fad820998	t2.small	ec2static

Cancel Next

From the subnet drop down select the private subnets

Choose instance launch options [Info](#)

Choose the VPC network environment that your instances are launched into, and customize the instance types and purchase options.

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-049dbf464ebb1eb42 (ecom-mahesh-vpc)
10.0.0.0/16

[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

us-east-1a | subnet-03c84febd090aa615 (ecom-mahesh-subnet-private1-us-east-1a)
10.0.1.0/24

us-east-1b | subnet-0a69d48680128d042 (ecom-mahesh-subnet-private2-us-east-1b)
10.0.3.0/24

[Create a subnet](#)

Cancel Previous Skip to review Next

Select the existing load balancer and select the target group.

Configure advanced options [Info](#)

Choose a load balancer to distribute incoming traffic for your application across instances to make it more reliable and easily scalable. You can also set options that give you more control over health check replacements and monitoring.

Load balancing - optional [Info](#)

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

☐ No load balancer
Traffic to your Auto Scaling group will not be fronted by a load balancer.

☒ Attach to an existing load balancer
Choose from your existing load balancers.

☐ Attach to a new load balancer
Quickly create a basic load balancer to attach to your Auto Scaling group.

Attach to an existing load balancer
Select the load balancers that you want to attach to your Auto Scaling group.

☒ Choose from your load balancer target groups
This option allows you to attach Application, Network, or Gateway Load Balancers.

☐ Choose from Classic Load Balancers

Existing load balancer target groups
Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

Select target groups

myTargetGroup | HTTP
Application Load Balancer: myLoadBalancer

Health check gives the heart rate of the application. ELB checks the webserver is up and running every 300 seconds.

Health checks - optional

Health check type [Info](#)
EC2 Auto Scaling automatically replaces instances that fail health checks. If you enabled load balancing, you can enable ELB health checks in addition to the EC2 health checks that are always enabled.

☒ EC2 ☒ ELB

Health check grace period
The amount of time until EC2 Auto Scaling performs the first health check on new instances after they are put into service.

300 seconds

Additional settings - optional

Monitoring [Info](#)
☒ Enable group metrics collection within CloudWatch

Default instance warmup [Info](#)
The amount of time that 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

Cancel Previous Skip to review Next

Below is the important step which define the resources group size for auto scaling based on the scaling policy.

Configure group size and scaling policies [Info](#)

Set the desired, minimum, and maximum capacity of your Auto Scaling group. You can optionally add a scaling policy to dynamically scale the number of instances in the group.

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

Minimum capacity

Maximum capacity

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

Metric type

Average CPU utilization ▼

Target value

Instances need

 seconds warm up before including in metric

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

Instance scale-in protection - optional

Instance scale-in protection

If protect from scale in is enabled, newly launched instances will be protected from scale in by default.

☐ Enable instance scale-in protection

Cancel

Previous

Skip to review

Next

Click next and review and finally create ASG

Auto Scaling groups (1) [Info](#)

↻

Edit

Delete

Create an Auto Scaling group

<

1

>

⚙

<input type="checkbox"/>	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
<input type="checkbox"/>	myASG	myLaunchConfig	0	⌂ Updating capacity	2	2	4	us-east-1a, us-east-1b

Next open Load Balancer (ELB) description and copy the DNS name.

The screenshot shows the AWS Management Console interface for a Load Balancer. At the top, there's a 'Create Load Balancer' button and an 'Actions' dropdown. Below is a search bar and a table with columns: Name, DNS name, State, VPC ID, Availability Zones, and Type. The table lists 'myLoadBalancer' with its DNS name 'myLoadBalancer-751546970.us-east-1.elb.amazonaws.com'. Below the table, there's a section for 'Load balancer: myLoadBalancer' with tabs for Description, Listeners, Monitoring, Integrated services, and Tags. The 'Description' tab is active, showing 'Basic Configuration' with details like Name, ARN, DNS name, State, Type, Scheme, IP address type, VPC, and Availability Zones.

Name	DNS name	State	VPC ID	Availability Zones	Type
myLoadBalancer	myLoadBalancer-751546970.us-east-1.elb.amazonaws.com	Active	vpc-049dbf464eb1eb42	us-east-1b, us-east-1a	application

Load balancer: myLoadBalancer

Description | Listeners | Monitoring | Integrated services | Tags

Basic Configuration

Name: myLoadBalancer
ARN: arn:aws:elasticloadbalancing:us-east-1:862781485761:loadbalancer/app/myLoadBalancer/5ee39b0c8d048d07
DNS name: myLoadBalancer-751546970.us-east-1.elb.amazonaws.com (A Record)
State: Active
Type: application
Scheme: Internet-facing
IP address type: ipv4
VPC: vpc-049dbf464eb1eb42
Availability Zones: subnet-001e8d3f75906038 - us-east-1b, subnet-036b3a730b743d8a0 - us-east-1a

Open the same DNS in browser to verify if the webservice is running

The screenshot shows a web browser window with the URL 'myloadbalancer-751546970.us-east-1.elb.amazonaws.com'. The page displays the AWS logo and the text 'Load Test RDS'. Below this, there's a table with 'Meta-Data' and 'Value' columns. The table shows 'InstanceId' as 'i-0ce474affc7fc57df' and 'Availability Zone' as 'us-east-1b'. At the bottom, it says 'Current CPU Load: 0%'.

Meta-Data	Value
InstanceId	i-0ce474affc7fc57df
Availability Zone	us-east-1b

Current CPU Load: 0%

To perform the load testing on the load balancer run the application in the browser.

Keep refreshing the page and observe if we are getting different instance ID of EC2. Elastic Load Balancing (ELB) automatically distributes incoming application traffic across multiple targets and virtual appliances in one or more Availability Zones.

The screenshot shows the AWS Management Console 'Instances' page. It has a search bar and a filter for 'Instance state = running'. Below is a table with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IP. The table lists two instances: 'i-0ce474affc7fc57df' and 'i-02403acd3ce864327', both in 'Running' state, with 't2.small' instance type and 'Initializing' status check.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
-	i-0ce474affc7fc57df	Running	t2.small	Initializing	No alarms	us-east-1b	-
-	i-02403acd3ce864327	Running	t2.small	Initializing	No alarms	us-east-1a	-

Once the server CPU goes above the mentioned threshold, it creates instances for the load balancing.

Not secure | myloadbalancer-751546970.us-east-1.elb.amazonaws.com

ng Home Things OS - Kernel,...

aws Load Test RDS

Meta-Data	Value
InstanceId	i-02403acd3ce864327
Availability Zone	us-east-1a

Current CPU Load: 100%

Once we start performing the load testing, if the CPU usage crosses the average value defined it creates new instance to serve the load of the application.

Instances (4) Info Refresh Connect Instance st

Find instance by attribute or tag (case-sensitive)

Instance state = running X Clear filters

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
<input type="checkbox"/>	-	i-03517ef8de5874a97	Running	t2.small	2/2 checks pas	No alarms	us-east-1b	-
<input type="checkbox"/>	-	i-0ce474affc7fc57df	Running	t2.small	2/2 checks pas	No alarms	us-east-1b	-
<input type="checkbox"/>	-	i-0928f2c9e68afa82f	Running	t2.small	2/2 checks pas	No alarms	us-east-1a	-
<input type="checkbox"/>	-	i-02403acd3ce864327	Running	t2.small	2/2 checks pas	No alarms	us-east-1a	-