



GLA
UNIVERSITY
MATHURA
Established vide U.P. Act 21 of 2010.

MINOR PROJECT REPORT

TOPIC NAME: Load Balancer

Submitted by :-

Jayveer chauhan	Abhinav Sharma
2115800012(09)	2115800001(30)

Submitted to :-

Mr. Krishna Koppula



Department of computer Engineering and Applications

GLA University, Mathura

17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuha

Mathura – 281406

Declaration

Certified that this project report “**Application Load Balancer**” is the bonafide work of “**Jayveer chauhan , Abhinav Sharma**” who carried out the project work under our supervision.

(Signature of Candidates)

Acknowledgement

It gives us a great sense of pleasure to present the report of the BTech(H) project undertaken during the BTech(Hons.)CS III Year. This project is going to be an acknowledgment of the inspiration, drive, and technical assistance that will be contributed to it by many individuals.

We owe a special debt of gratitude to **Mr. KRISHNA KOPPULA, FULL STACK WEB DEVELOPMENT MENTOR**, for providing us with an encouraging platform to develop this project, which thus helped us in shaping our abilities towards a constructive goal, and for his constant support and guidance to our work. His sincerity, thoroughness, and perseverance have been a constant source of inspiration for us. We believe that he will shower us with all his extensively experienced ideas and insightful comments at different stages of the project & also teach us about the latest industry-oriented technologies.

We also do not like miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind guidance and cooperation.

GROUP MEMBERS NAMES :

JAYVEER CHAUHAN

2115800012(09)

AHINAV SHARMA

2115800001(30)

Table of Contents

1. Abstract.....	1
2. Introduction.....	2

3. Problem Definition.....	3-6
3.1 Technology Stack	
3.2 Task Definition	
3.3 Algorithms	
4. Experimental Evaluation.....	7-20
4.1 Methodology	
4.2 Results	
5. Future Work.....	21
6. Conclusion.....	22
References.....	23

1. Abstract

A load balancer is a crucial component in computer networking that plays a pivotal role in distributing incoming network traffic across multiple servers or resources. The primary objective of a load balancer is to optimize resource utilization, ensure high availability, and enhance the overall performance and reliability of a system. This abstract provides a concise overview of load balancers, highlighting their key functionalities, types, and benefits..

- **Scalability and Performance**

Scalability and performance are two critical aspects of a load balancer, as they directly impact the efficiency and effectiveness of distributed systems. Load balancers play a key role in ensuring that applications can scale horizontally to handle increased traffic while maintaining optimal performance. Below, we delve into how load balancers contribute to scalability and performance.

- **Efficient Data Processing**

Efficient data processing is a critical aspect of modern computing, and load balancers play a crucial role in optimizing data handling across servers. Here are several ways in which load balancers contribute to efficient data processing.

- **Intuitive User Interface:**

An efficient data processing load balancer is a crucial component in a distributed computing environment that specializes in optimizing the handling and processing of data across multiple servers. It plays a pivotal role in improving performance, scalability, and resource utilization. Here are key features and strategies employed by load balancers for efficient data processing.

- **Real-time Communication**

A load balancer designed for real-time communication plays a critical role in ensuring the scalability, reliability, and responsiveness of applications that involve instantaneous data exchange. Real-time communication, which often includes services like video conferencing, instant messaging, and live streaming, has specific requirements that a load balancer must address..

- **User Authentication and Authorization**

A load balancer that incorporates user authentication and authorization features plays a crucial role in securing and managing access to applications or services. This type of load balancer helps ensure that only authenticated and authorized users gain access to protected resources. Here are key considerations and features for a load balancer in the context of user authentication and authorization:.

In conclusion, load balancers are indispensable components in modern computing architectures, playing a pivotal role in optimizing the performance, scalability, and reliability of applications and services. The key takeaways regarding load balancers include:.

2. Introduction

An introduction to load balancers is essential to understanding their critical role in modern computing environments. A load balancer acts as a traffic manager, distributing incoming network traffic across multiple servers or resources to optimize performance, enhance scalability, and ensure high availability of applications and services. Here's a comprehensive introduction to load balancers:.

Key Features

- **Traffic Distribution:**

Load balancers evenly distribute incoming network traffic across multiple servers, preventing any single server from becoming overwhelmed. This ensures efficient resource utilization and optimal performance.

- **Load Balancing Algorithms:**

Various algorithms, such as round-robin, least connections, weighted distribution, or content-based routing, are employed to intelligently distribute traffic among servers based on predefined criteria.

- **Health Monitoring:**

Load balancers continuously monitor the health and status of servers through health checks. Unresponsive or underperforming servers are identified and temporarily removed from the pool to maintain system reliability.

- **Scalability:**

Load balancers support horizontal scaling by allowing the dynamic addition or removal of servers based on demand. This ensures that applications can handle varying levels of traffic effectively.

- **SSL Termination:**

Some load balancers can handle SSL/TLS termination, offloading the encryption and decryption process from backend servers. This reduces the computational load on servers and improves overall performance..

3. Problem Definition

The deployment of load balancers in a system is typically aimed at solving specific challenges related to performance, scalability, and reliability. Here are common problem definitions that might prompt the implementation of a load balancer:.

3.1 Technology

Load Balancing Algorithms:

Technology: Load balancers use algorithms to determine how incoming requests should be distributed among servers. Common algorithms include round-robin, least connections, weighted distribution, and content-based routing.

Health Monitoring:

Technology: Load balancers continually monitor the health and status of servers through health checks. This technology involves sending periodic requests to servers and analyzing responses to ensure they are responsive and capable of handling requests.

SSL/TLS Offloading:

Technology: Load balancers can handle SSL/TLS termination, offloading the encryption and decryption process from backend servers. This technology improves overall system performance and allows servers to focus on processing application logic.

Session Persistence/Sticky Sessions:

Technology: For applications requiring session persistence, load balancers support technologies that enable them to direct a user's requests consistently to the same backend server throughout their session. This is crucial for maintaining session state.

WebSocket Support:

Technology: Load balancers designed for real-time communication applications, which use WebSocket protocols, incorporate technologies to efficiently handle persistent connections and bidirectional communication.

Global Server Load Balancing (GSLB):

Technology: GSLB involves technologies that enable load balancers to direct users to the nearest or most available data center. This may include DNS-based load balancing and geographical routing technologies.

Adaptive Load Balancing:

Technology: Some load balancers incorporate adaptive technologies that dynamically adjust their routing decisions based on real-time metrics. This may involve monitoring server load, response times, and other factors to optimize performance.

Content-based Routing:

Technology: Load balancers can employ technologies that inspect the content of incoming requests and route them to servers optimized for handling specific types of content. This is valuable for applications with diverse workloads.

Logging and Monitoring:

Technology: Load balancers provide logging and monitoring capabilities to track performance metrics and generate insights into system behavior. This involves technologies for collecting, analyzing, and presenting data related to traffic distribution and server health.

Integration with Authentication Mechanisms:

Technology: Load balancers can integrate with various authentication mechanisms, such as LDAP or Active Directory, to centralize user authentication and authorization processes.

3.2 Task Definition

A. Problem Statement

- i. Inefficient Job Discovery:** Difficulty for students in discovering relevant job opportunities. Potential candidates may miss out on suitable positions, while recruiters struggle to reach a diverse pool of qualified applicants.
- ii. Limited Analytics for Optimization:** Absence of analytics tools to analyze the effectiveness of recruitment strategies. Recruiters miss opportunities for optimization, resulting in less effective recruitment campaigns.
- iii. Limited Real-time Interaction:** Inefficient channels for real-time communication between recruiters and students. Slower response times and missed opportunities due to delays in communication during the recruitment process.
- iv. Lack of Centralization:** Recruitment activities are dispersed across various channels and platforms. This fragmentation leads to difficulties in discovering and managing opportunities for both recruiters and students.
- v. Manual and Time-Consuming Processes:** The reliance on manual handling of resumes, paperwork, and communication. Time-consuming processes result in delays, creating frustration for recruiters and causing students to miss out on timely opportunities.

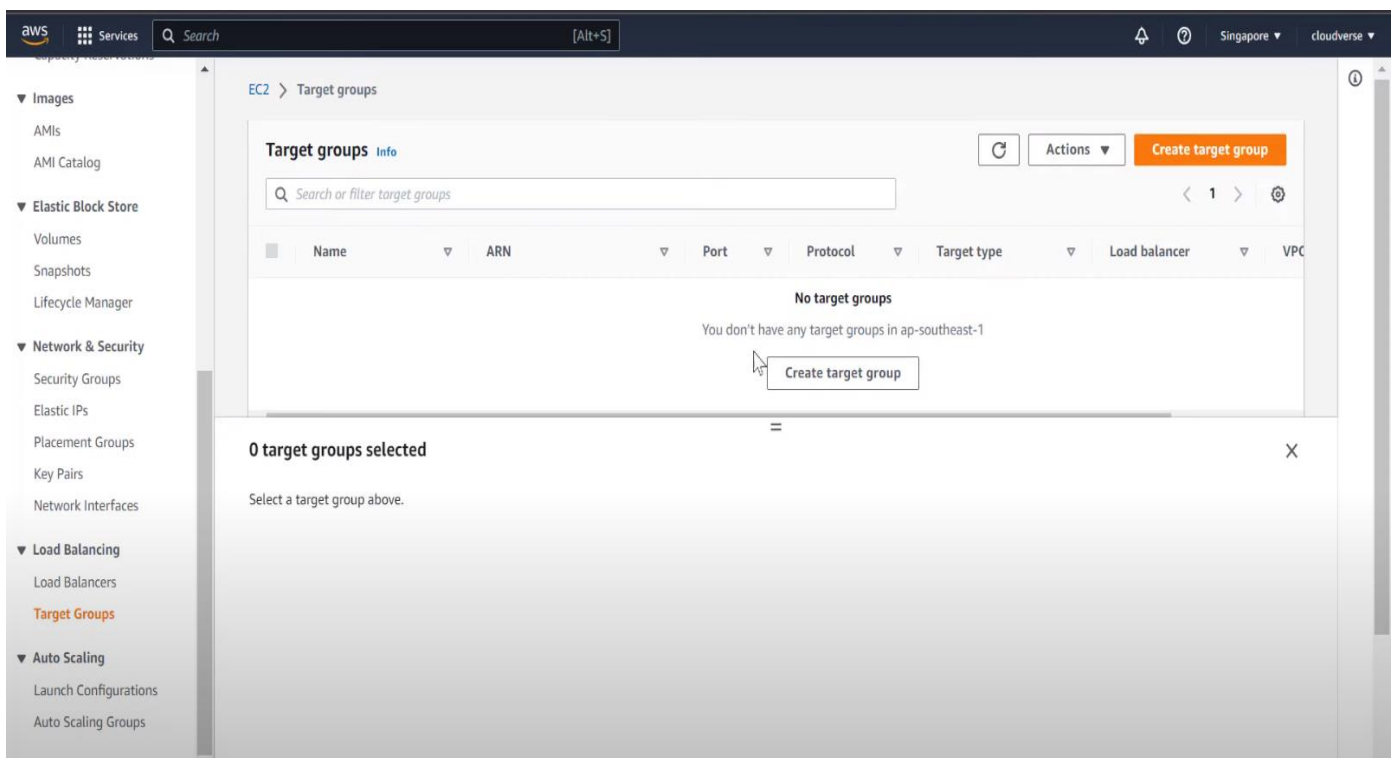
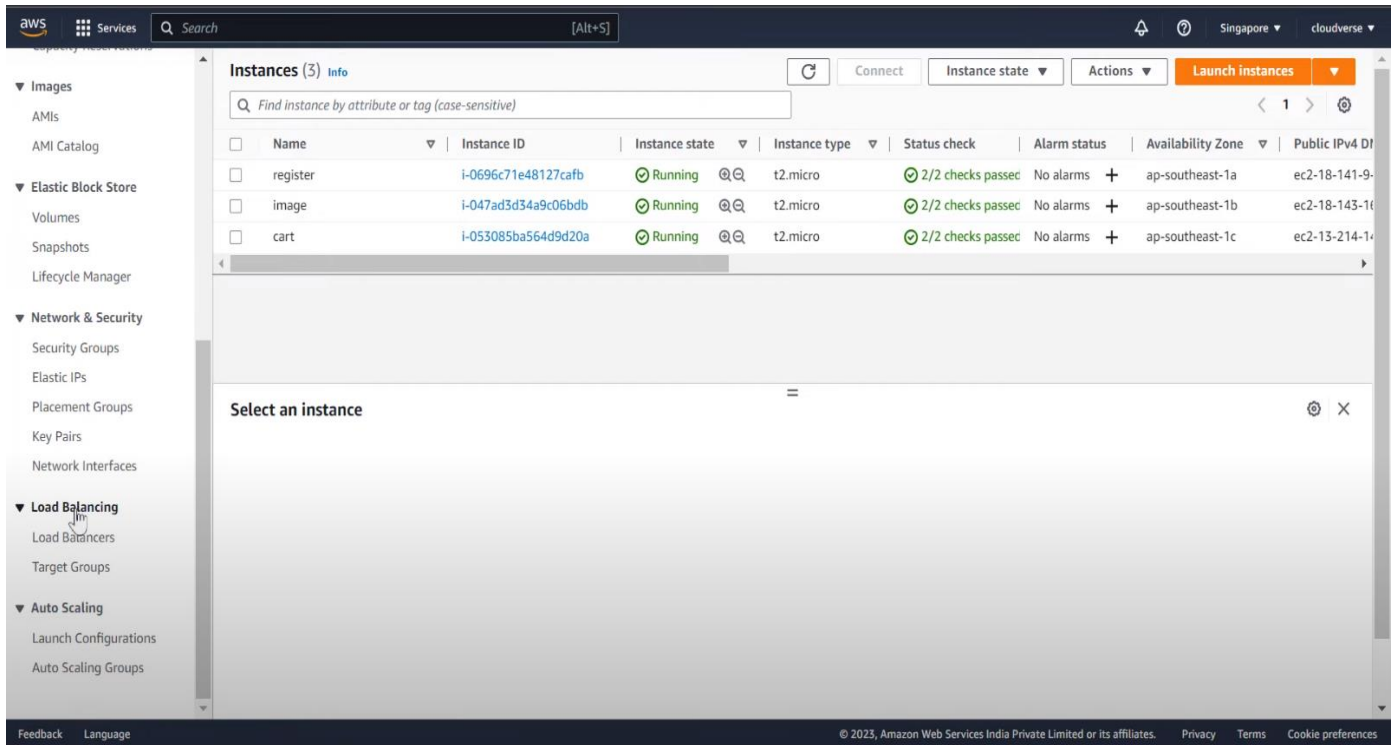
B. Proposed Solution

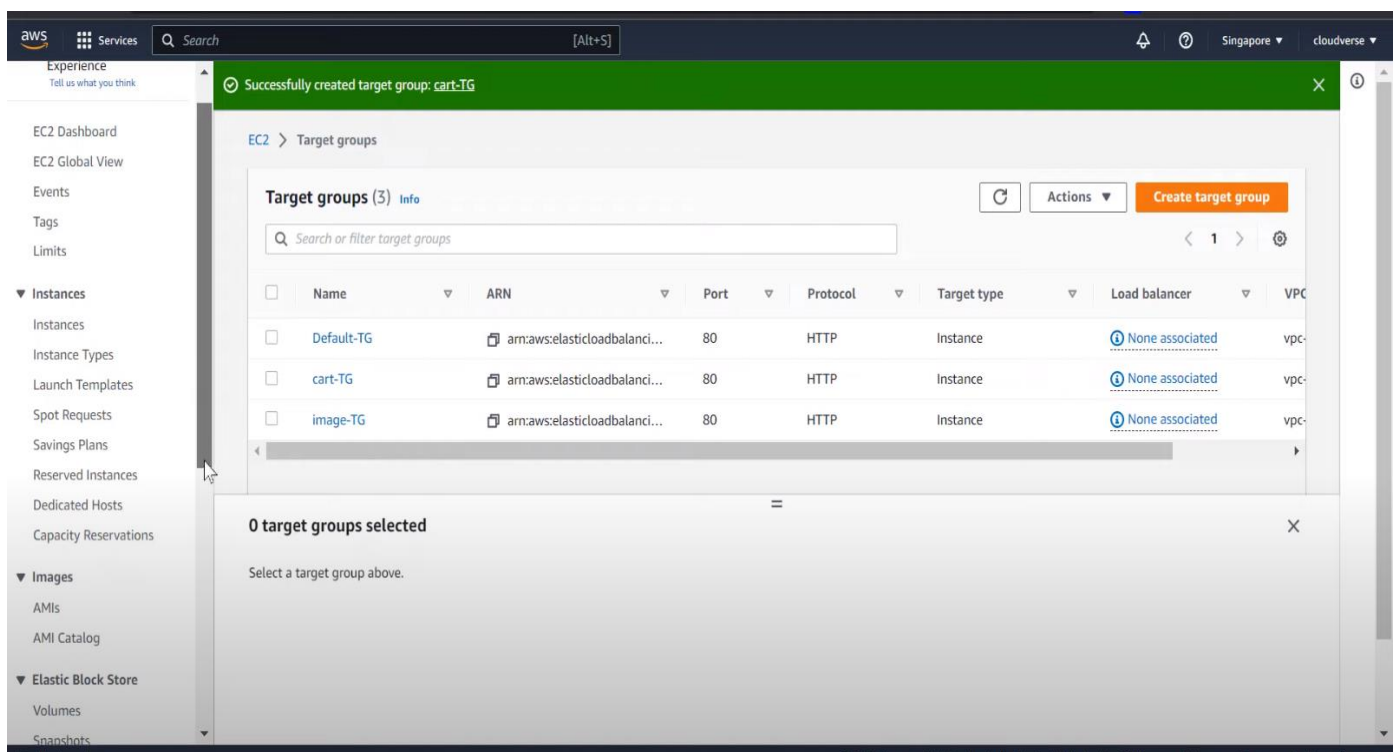
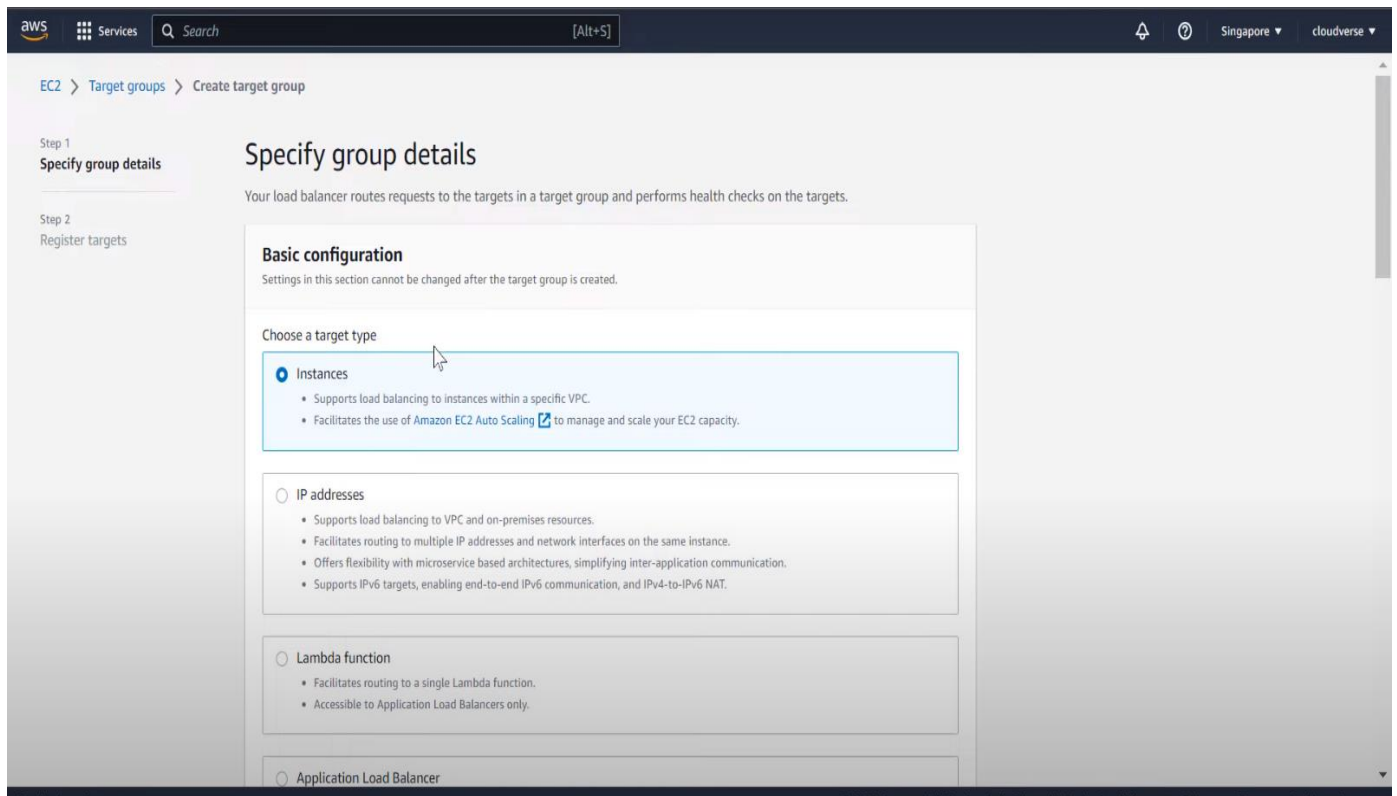
- i. Facilitate Efficient Job Discovery:** Implement features that make it easier for students to discover relevant job opportunities based on their skills, preferences, and career goals.
- ii. Provide Analytics for Optimization:** Integrate analytics tools to offer valuable insights into user interactions, recruitment trends, and platform usage. This empowers recruiters to make data-driven decisions and optimize their recruitment strategies.

- iii. **Enable Real-time Communication:** Incorporate real-time communication channels using technologies like Socket.io, fostering immediate and direct interaction between recruiters and students.
- iv. **Centralize Recruitment Activities:** Create a centralized platform for recruiters to post job listings, review student profiles, and communicate directly with potential candidates.
- v. **Automate Processes:** Implement features that automate manual tasks, such as resume handling and application tracking, to reduce delays and improve overall efficiency.

- user-centric design. Crafting a seamless and intuitive interface for both recruiters and candidates became a paramount focus. Understanding the diverse needs of users and incorporating feedback led to iterative improvements, emphasizing the significance of agility in development.
- Moreover, the project delved into the complexities of real-time communication and collaboration tools. Implementing features like instant messaging and interview scheduling not only demanded technical proficiency but also required a keen understanding of the dynamic nature of recruitment interactions.
- Collaboration and effective communication with stakeholders played a pivotal role. Engaging with recruiters, HR professionals, and potential candidates provided valuable insights into their expectations and pain points. This iterative feedback loop was instrumental in refining features and ensuring alignment with industry requirements
- The project also underscored the criticality of security in handling sensitive candidate information. Implementing robust measures to safeguard data privacy and comply with regulations became a paramount consideration, reinforcing the importance of ethical software development practices.
- In terms of technical skills, the project provided hands-on experience with cutting-edge technologies, including Node.js, Pug, and JavaScript. The use of AWS for hosting and managing the platform added a cloud-centric dimension to the learning journey, offering insights into scalable and reliable infrastructure.
- Overall, this project has been a transformative experience, bridging the gap between theoretical knowledge and practical application. It has equipped with a versatile skill set, a deeper understanding of user experience considerations, and a heightened awareness of the intricacies involved in developing solutions for real-world challenges.
- As I move forward, these learnings will undoubtedly serve as a solid foundation for future endeavors in software development and contribute to the continuous pursuit of excellence in the field.

Steps to crate a load balancer





aws

Services

Search

[Alt+S]

Singapore

cloudverse

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 Elastic Load balancing works

Basic configuration

Load balancer name

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

ALB-01

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.

Feedback

Language

© 2023, Amazon Web Services India Private Limited or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

aws

Services

Search

[Alt+S]

Singapore

cloudverse

✔ Successfully created load balancer: ALB-01

Note: It might take a few minutes for your load balancer to be fully set up and ready to route traffic. Targets will also take a few minutes to complete the registration process and pass initial health checks.

EC2 > Load balancers > ALB-01 > Create Application Load Balancer

Create Application Load Balancer

ℹ Suggested next steps

- Review, customize, or enable attributes for your load balancer and listeners using the **Description** and **Listeners** tabs within ALB-01.
- Discover other services that you can integrate with your load balancer. Visit the **Integrated services** tab within ALB-01.

View load balancer

Waiting for ap-southeast-1.console.aws.amazon.com...

© 2023, Amazon Web Services India Private Limited or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

aws

Services

Search

[Alt+S]

🔔

🔗

Singapore

cloudverse

New EC2 Experience

EC2 Dashboard

EC2 Global View

Events

Tags

Limits

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Images

AMIs

AMI Catalog

Elastic Block Store

EC2 > Load balancers

Load balancers (1)

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

Filter by property or value

search: ALB-01

Clear filters

	Name	DNS name	State	VPC ID	Availability Zones	Type	Date created
<input type="checkbox"/>	ALB-01	ALB-01-930211095.ap-so...	Provisioning	vpc-08566f645cb9fb383	2 Availability Zones	application	March 3, 2023, 1 (UTC+05:30)

0 load balancers selected

Select a load balancer above.

Feedback

Language

© 2023, Amazon Web Services India Private Limited or its affiliates.

Privacy

Terms

Cookie preferences

aws

Services

Search

[Alt+S]

🔔

🔗

Singapore

cloudverse

New EC2 Experience

EC2 Dashboard

EC2 Global View

Events

Tags

Limits

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Images

AMIs

AMI Catalog

Elastic Block Store

EC2 > Load balancers

Load balancers (1)

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

Filter by property or value

search: ALB-01

Clear filters

	Name	DNS name	State	VPC ID	Availability Zones	Type	Date created
<input type="checkbox"/>	ALB-01	ALB-01-930211095.ap-so...	Active	vpc-08566f645cb9fb383	3 Availability Zones	application	March 3, 2023, 1 (UTC+05:30)

0 load balancers selected

Select a load balancer above.

Feedback

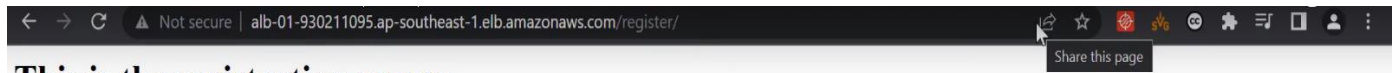
Language

© 2023, Amazon Web Services India Private Limited or its affiliates.

Privacy

Terms

Cookie preferences



This is the registration server



Not Found

The requested URL was not found on this server.



aws Services Search [Alt+S] Singapore cloudverse

New EC2 Experience Tell us what you think

EC2 Dashboard
EC2 Global View
Events
Tags
Limits

▼ Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations

▼ Images
AMIs
AMI Catalog

▼ Elastic Block Store

EC2 > Load balancers

Load balancers (1/1)
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Filter by property or value

<input checked="" type="checkbox"/>	Name	DNS name	State	VPC ID	Availability Zones	Type	Date created
<input checked="" type="checkbox"/>	ALB-01	ALB-01-930211095.ap-so...	Active	vpc-08566f645cb9fb383	3 Availability Zones	application	March 3, 2021 (UTC+05:30)

Load balancer: ALB-01

Listeners (1/1)
A listener checks for connection requests on its port and protocol. Traffic received by the listener is routed according to its rules.

Search

<input checked="" type="checkbox"/>	Protocol:Port	ARN	Security policy	Default SSL cert	Default routing
<input checked="" type="checkbox"/>	HTTP:80	ARN	Not applicable	Not applicable	1. Forward to • Default- • Group-le

Actions

- View listener details
- Edit listener
- Add SSL certificates for SNI
- Manage rules
- Manage tags
- Delete listener

Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

aws Services Search [Alt+S] Singapore cloudverse

Rules ALB-01 | HTTP:80

Click a location for your new rule. Each rule must include one action of type forward, redirect, fixed response. Cancel Save

ALB-01 | HTTP:80 (3 rules)

Rule limits for condition values, wildcards, and total rules.

Insert Rule

RULE ID	IF (all match)	THEN
1	<p>A rule ID (ARN) is generated when you save your rule.</p> <p>+ Add condition</p>	<p>1. Forward to...</p> <p>Target group : Weight (0-999)</p> <p>cart-TG 1</p> <p>Traffic distribution 100%</p> <p>Select a target group 0</p> <p>Group-level stickiness</p> <p>+ Add action</p>
2	arn...e4ce9	IF THEN

aws Services Search [Alt+S] Singapore cloudverse

Rules ALB-01 | HTTP:80

Click a location for your new rule. Each rule must include one action of type forward, redirect, fixed response.

✓ New rule was created successfully.

▶ Rule limits for condition values, wildcards, and total rules.

Insert Rule

Order	ARN	IF	THEN
1	arn...d1a69	✓ Path is /cart*	Forward to cart-TG: 1 (100%) Group-level stickiness: Off
2	arn...e4ce9	✓ Path is /image*	Forward to image-TG: 1 (100%) Group-level stickiness: Off
last	HTTP 80: default action	✓ Requests otherwise not routed	Forward to Default-TG: 1 (100%) Group-level stickiness: Off

This rule cannot be moved or deleted

alb-01-930211095.ap-southeast-1.elb.amazonaws.com/cart

This is the Image Catalogue server

References

A. Google Scholar:

Google Scholar is a freely accessible search engine that indexes scholarly articles. It can be a useful tool to discover research papers on load balancing topics.

B. Research Conferences:

Explore proceedings from reputable conferences in the field, such as: ACM SIGCOMM (Special Interest Group on Data Communication) USENIX (Advanced Computing Systems Association) conferences IEEE INFOCOM (Conference on Computer Communications)