

ABSTRACT

Cloud computing is latest emerging technology for large scale distributed computing and parallel computing. Cloud computing gives large pool of shared resources, software packages, information, storage and many different applications as per user demands at any instance of time. Cloud computing is emerging quickly; a large number of users are attracted towards cloud services for more satisfaction. Balancing the load has become more interesting research area in this field. Better load balancing algorithm in cloud system increases the performance and resources utilization by dynamically distributing workload among various nodes in the system. **Cloud load balancing** refers to distributing client requests across multiple application servers that are running in a cloud environment. Like other forms of load balancing, cloud load balancing enables you to maximize application performance and reliability; its advantages over traditional load balancing of on-premises resources are the (usually) lower cost and the ease of scaling the application up or down to match demand.

INTRODUCTION

cloud computing is called as internet based computing in which many different services such as server, storage, virtualization and various application are given to the users and organization over the internet. There are several issues in cloud computing paradigm but balancing the load is major issue (challenge) in cloud computing environment. Load balancing is a methodology which provides methods to maximize throughput, utilization of resources and performance of system. As a part of its services, it gives easy and flexible process to keep data or files and make them available for large scale of users. To make the use of resources most efficiently in cloud system, there are several load balancing algorithms.

ADVANTAGES OF A LOAD BALANCER

- Increased Scalability
- Redundancy
- Reduced Downtime, Increased Performance
- Efficiently Manages Failures
- Increased Flexibility

STEPS

Step1: Instances On AWS:

The two instances are created on AWS. The following is the list of instances. Our two instances are with the name 'MyEC2LoadBalancerInstances'.

Launch Instance	Connect	Actions							
Filter by tags and attributes or search by keyword									
	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public	
	AUTO_SCALE	i-0185e6bef607c99a1	t2.micro	us-east-1a	running	2/2 checks ...	None	ec2-54	
	AUTO_SCALE1	i-094de74013653000a	t2.micro	us-east-1c	running	2/2 checks ...	None	ec2-54	
	AUTO_Scale4	i-0836a75c69a669e98	t2.micro	us-east-1c	running	2/2 checks ...	None	ec2-34	
	AUTO_SCALE5	i-0a3691001c2e78772	t2.micro	us-east-1a	running	2/2 checks ...	None	ec2-18	
	AutoScale-WebServers	i-0517f04c756e3e239	t2.micro	us-east-1c	running	2/2 checks ...	None	ec2-54	
	AUTO_Scale3	i-070660265c6822870	t2.micro	us-east-1c	running	2/2 checks ...	None	ec2-52	
	MyEC2LoadBalancedInstances	i-0165e9d627aa45880	t2.micro	us-east-1c	running	2/2 checks ...	None	ec2-35	
	MyEC2LoadBalancedInstances	i-0e404b4415abf3ed0	t2.micro	us-east-1c	running	2/2 checks ...	None	ec2-35	

- Information For Instance 1:

The instance is identified by its unique ID. The following is the information about Instance1.

Description	Status Checks	Monitoring	Tags
Instance ID	i-0165e9d627aa45880	Public DNS (IPv4)	ec2-35-170-187-226.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	35.170.187.226
Instance type	t2.micro	IPv6 IPs	-
Elastic IPs		Private DNS	ip-172-31-87-228.ec2.internal
Availability zone	us-east-1c	Private IPs	172.31.87.228
Security groups	launch-wizard-1. view inbound rules	Secondary private IPs	
Scheduled events	No scheduled events	VPC ID	vpc-d520b8ae
AMI ID	Windows_Server-2012-R2_RTM-English-64Bit-Base-2018.03.24 (ami-eb1ecd96)	Subnet ID	subnet-e98485c6
Platform	windows	Network interfaces	eth0

- Information for Instance 2:

The instance is identified by its unique ID. The following is the information about Instance2.

MyEC2LoadBalancedInstances		i-0e404b4415abf3ed0	t2.micro	us-east-1c	running	2/2 checks ...	None	ec2
Instance: i-0e404b4415abf3ed0 (MyEC2LoadBalancedInstances)		Public DNS: ec2-35-173-233-41.compute-1.amazonaws.com						
Description	Status Checks	Monitoring	Tags					
Instance ID	i-0e404b4415abf3ed0			Public DNS (IPv4)	ec2-35-173-233-41.compute-1.amazonaws.com			
Instance state	running			IPv4 Public IP	35.173.233.41			
Instance type	t2.micro			IPv6 IPs	-			
Elastic IPs				Private DNS	ip-172-31-83-250.ec2.internal			
Availability zone	us-east-1c			Private IPs	172.31.83.250			
Security groups	launch-wizard-1 . view inbound rules			Secondary private IPs				
Scheduled events	No scheduled events			VPC ID	vpc-d520b8ae			

Step2. Creation of AMI's

Services

Resource Groups

kajal

N. Virginia

Support

Launch

Actions

Owned by me

Filter by tags and attributes or search by keyword

<

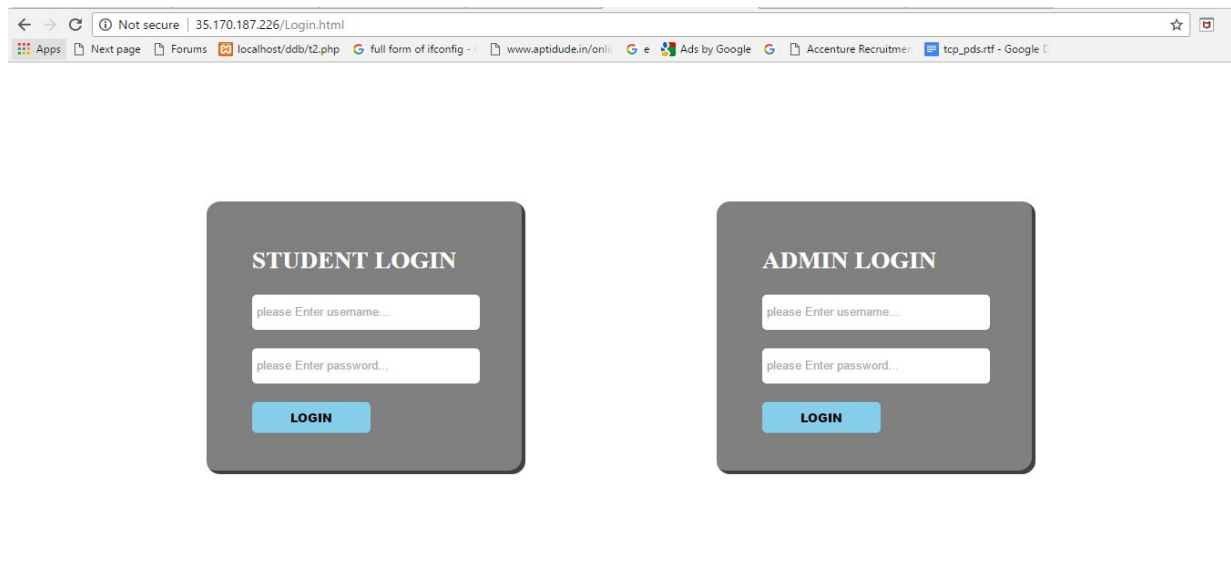
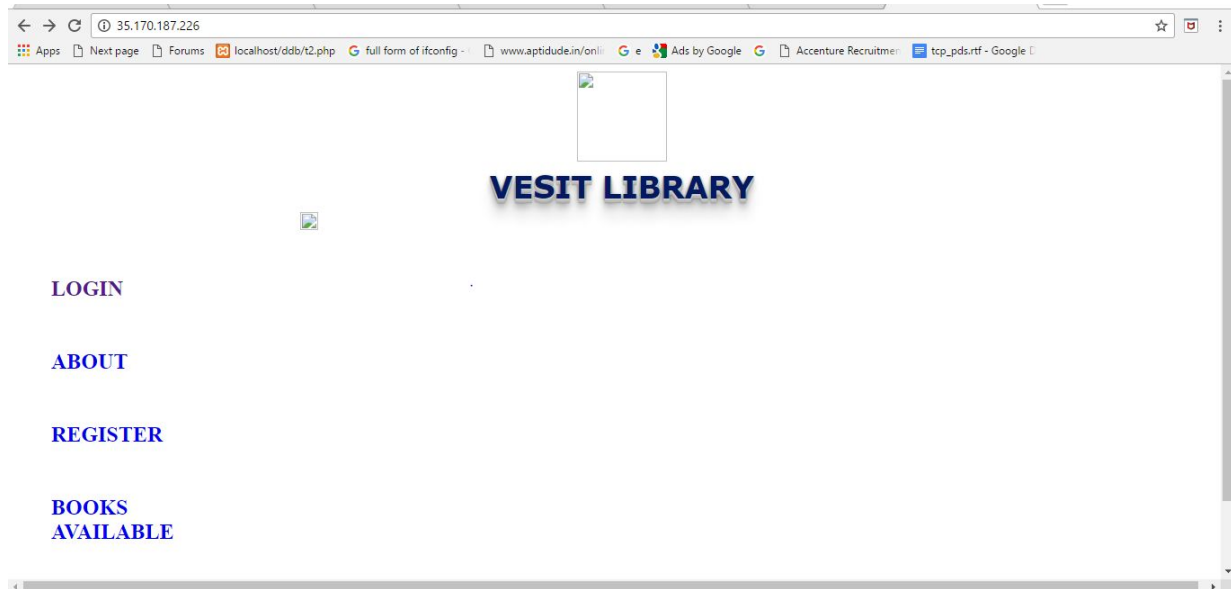
1 to 4 of 4

>

	Name	AMI Name	AMI ID	Source	Owner	Visibility	Status	Creation Date	
<input type="checkbox"/>		liveserver2	ami-0168bc06efd201b7a	135596314834/li...	135596314834	Private	available	April 3, 2018 at 11:22:27 AM...	V
<input type="checkbox"/>		liveserver3	ami-07389f07dfa7c3ea8	135596314834/li...	135596314834	Private	available	April 3, 2018 at 12:08:58 PM...	V
<input type="checkbox"/>		liveserver5	ami-086e92337f55cbf98	135596314834/li...	135596314834	Private	available	April 3, 2018 at 12:25:28 PM...	V
<input type="checkbox"/>		liveserver4	ami-0e435ec6511ced92a	135596314834/li...	135596314834	Private	available	April 3, 2018 at 12:09:15 PM...	V

Step3: Content available.

- On Instance 1: Login is available. Registration is not available



- On Instance 2: Registration is available. Login is not available.

← → ↻ Not secure | 35.173.233.41/registration.html#2

Apps Next page Forums localhost/ddb/t2.php full form of ifconfig - www.aptitude.in/onli e Ads by Google Accenture Recruitment tcp_pds.rtf - Google

Registration Form

[HOME](#)

Student or Teacher Librarian

Student

Name:

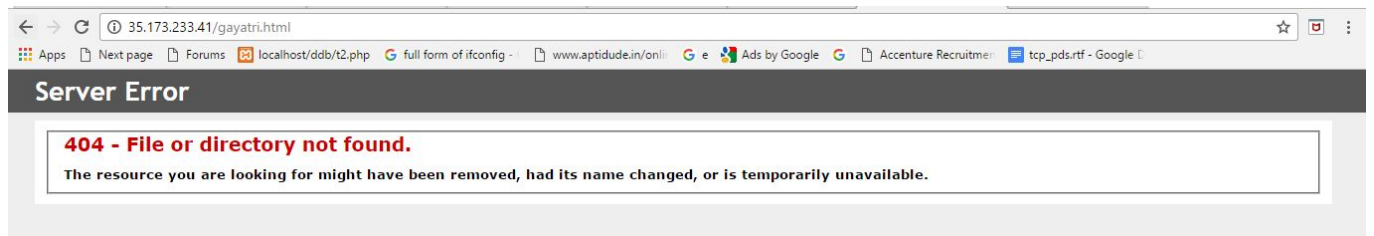
Email Id:

Password:

Confirm Password:

Mobile No.:

Reset Register



Step4: Creation of Auto scaling groups:

- Launch Configuration:

aws Services Resource Groups

EC2 Dashboard
Events
Tags
Reports
Limits

INSTANCES
Instances
Launch Templates
Spot Requests
Reserved Instances
Dedicated Hosts
Scheduled Instances

IMAGES
AMIs
Bundle Tasks

ELASTIC BLOCK STORE
Volumes
Snapshots

Create launch configuration Create Auto Scaling group Actions

Filter: Filter launch configurations...

1 to 6 of 6 Launch Configurations

	Name	AMI ID	Instance Type	Spot Price	Creation Time
<input type="checkbox"/>	liveserver5	ami-086e92337...	t2.micro		April 3, 2018 12:28:56 PM UTC+...
<input type="checkbox"/>	liveserver4	ami-0e435ec65...	t2.micro		April 3, 2018 12:15:23 PM UTC+...
<input type="checkbox"/>	liveserver3	ami-07389f07d...	t2.micro		April 3, 2018 12:13:42 PM UTC+...
<input type="checkbox"/>	liveserver2group	ami-0168bc06e...	t2.micro		April 3, 2018 11:46:56 AM UTC+...
<input type="checkbox"/>	liveserver1	ami-0643e305f...	t2.micro		April 3, 2018 11:40:11 AM UTC+...
<input type="checkbox"/>	MyLaunchConfig	ami-076c1bc91...	t2.micro		April 2, 2018 10:35:58 PM UTC+...

Select a launch configuration above

- Auto scaling groups

Services Resource Groups

Create Auto Scaling group Actions

Filter: Filter Auto Scaling groups...

1 to 6 of 6 Auto Scaling Groups

	Name	Launch Configuration	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grace
<input type="checkbox"/>	liveserver5group	liveserver5	1	1	1	1	us-east-1a, us-east-1c	300	300
<input type="checkbox"/>	liveserver4group	liveserver4	1	1	1	1	us-east-1a, us-east-1c	300	300
<input type="checkbox"/>	liveservergroup3	liveserver3	1	1	1	1	us-east-1a, us-east-1c	300	300
<input type="checkbox"/>	liveserver2group	liveserver2group	1	1	1	1	us-east-1a, us-east-1c	300	300
<input type="checkbox"/>	liveserver1group	liveserver1	1	1	1	1	us-east-1a, us-east-1c	300	300
<input type="checkbox"/>	MyAutoScaleG...	MyLaunchConfig	1	1	1	4	us-east-1a, us-east-1b, us-ea...	300	300

Select an Auto Scaling group above

- Set target for auto scaling groups

Filter:

1 to 6 of 6 Auto Scaling Groups

Name	Launch Configuration /	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grace
liveserver5group	liveserver5	1	1	1	1	us-east-1a, us-east-1c	300	300
liveserver4group	liveserver4	1	1	1	1	us-east-1a, us-east-1c	300	300

Auto Scaling Group: liveserver4group

Details | Activity History | Scaling Policies | Instances | Monitoring | Notifications | Tags | Scheduled Actions | Lifecycle Hooks

Edit

Launch Configuration: liveserver4

Launch Template

Launch Template Version

Service-Linked Role: arn:aws:iam::135596314834:role/aws-service-role/autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling

Load Balancers

Target Groups: liveauto1

Desired: 1

Min: 1

Availability Zone(s): us-east-1a, us-east-1c

Subnet(s): subnet-e98485c6, subnet-2d2d2b70

Filter:

1 to 6 of 6 Auto Scaling Groups

Name	Launch Configuration /	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grace
liveserver5group	liveserver5	1	1	1	1	us-east-1a, us-east-1c	300	300
liveserver4group	liveserver4	1	1	1	1	us-east-1a, us-east-1c	300	300

Auto Scaling Group: liveserver5group

Details | Activity History | Scaling Policies | Instances | Monitoring | Notifications | Tags | Scheduled Actions | Lifecycle Hooks

Edit

Launch Configuration: liveserver5

Launch Template

Launch Template Version

Service-Linked Role: arn:aws:iam::135596314834:role/aws-service-role/autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling

Load Balancers

Target Groups: liveauto2

Desired: 1

Min: 1

Availability Zone(s): us-east-1a, us-east-1c

Subnet(s): subnet-e98485c6, subnet-2d2d2b70

Step5: Creation Of Load Balancer.

Amazon EC2 Auto Scaling integrates with Elastic Load Balancing to enable you to attach one or more load balancers to an existing Auto Scaling group. After you attach the load balancer, it automatically registers the instances in the group and distributes incoming traffic across the instances.

The screenshot displays the AWS Management Console interface for an Elastic Load Balancing resource. At the top, there's a navigation bar with 'services', 'Resource Groups', and user information. Below this, a 'Create Load Balancer' button and an 'Actions' dropdown are visible. A search bar allows filtering by tags and attributes. A table lists the load balancers, with 'LiveLoadBal' selected. The details for 'LiveLoadBal' are shown below the table, including its Name, ARN, DNS name, Scheme, Type, Creation time, Hosted zone, State, VPC, and IP address type.

Name	DNS name	State	VPC ID	Availability Zones	Type
LiveLoadBal	LiveLoadBal-1618415413.us-east-1.elb.amazonaws.com	active	vpc-d520b8ae	us-east-1a, us-east-1c	application
liveloadbalancer	liveloadbalancer-810895777.us-east-1.elb.amazonaws.com	active	vpc-d520b8ae	us-east-1a, us-east-1c	application

Load balancer: LiveLoadBal

Description | Listeners | Monitoring | Tags

Basic Configuration

Name:	LiveLoadBal	Creation time:	April 3, 2018 at 12:39:53 PM UTC+5:30
ARN:	arn:aws:elasticloadbalancing:us-east-1:135596314834:loadbalancer/app/LiveLoadBal/dda618682a7c1894	Hosted zone:	Z35SXDOTRQ7X7K
DNS name:	LiveLoadBal-1618415413.us-east-1.elb.amazonaws.com (A Record)	State:	active
Scheme:	internet-facing	VPC:	vpc-d520b8ae
Type:	application	IP address type:	ipv4
		AWS WAF Web ACL:	

Step6: Set Rules for Load Balancer

The rules that you define for your listener determine how the load balancer routes requests to the targets in one or more target groups.

- **To add a rule using the console**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation pane, under **LOAD BALANCING**, choose **Load Balancers**.
3. Select the load balancer and choose **Listeners**.
4. For the listener to update, choose **View/edit rules**.
5. Choose the **Add rules** icon (the plus sign) in the menu bar, which adds **Insert Rule** icons at the locations where you can insert a rule in the priority order.

Load balancer: **LiveLoadBal**

Description Listeners Monitoring Tags

A listener checks for connection requests using its configured protocol and port, and the load balancer uses the listener or update listeners and listener rules.

Add listener Actions ▾

<input type="checkbox"/>	Listener ID	Security policy	SSL Certificate	Rules
<input type="checkbox"/>	HTTP:80 arn...fe853fae0e4b3563 ▾	N/A	N/A	Default: forwarding to liveauto2 View/edit rules

< Rules + ✎ ⏏ ⏪ LiveLoadBal | HTTP:80 ▾ ↺ ⓘ

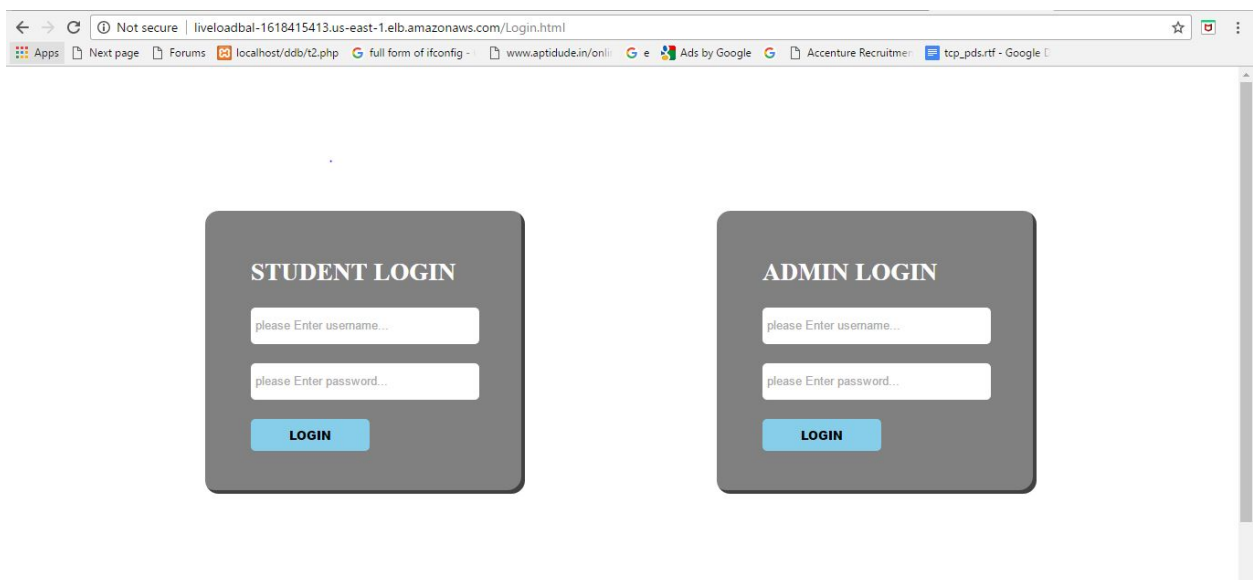
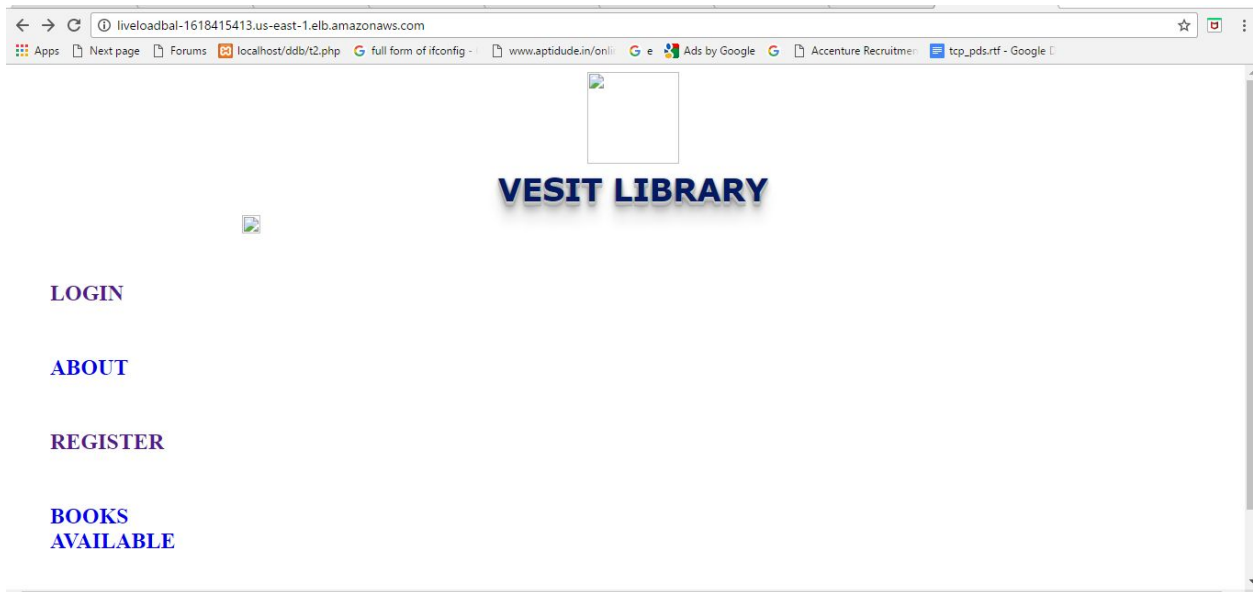
To edit, select a mode above.

LiveLoadBal | HTTP:80 (3 rules)

1	arn...aaa61 ▾	IF ✓ Path is /Login.html/*	THEN Forward to liveauto2
2	arn...a7042 ▾	IF ✓ Path is /registration.html/*	THEN Forward to liveauto1
last	HTTP 80: default action <i>This rule cannot be moved or deleted</i>	IF ✓ Requests otherwise not routed	THEN Forward to liveauto2

Step7: Accessing Load balancer using DNS name.

Here Both Registration and Login are available.



Registration Form

HOME

Student or Teacher Librarian

Student

Name: Full Name

Email Id: Enter Active Email Id

Password: enter password of min 8 and max 15 digits

Confirm Password: Confirm your password

Mobile No.: Mobile no.

Reset Register

Conclusion:

By introducing load balancing, the traffic can be spread across multiple servers and the increase in the traffic can be handled in a much easier manner. When you use load balancing for maintaining a website on more than one web server, the impact of hardware failure on a site's overall uptime can be limited significantly. By implementing load balancing you can achieve redundancy. This means that when the website traffic is sent to two or more web servers and one server fails, then the load balancer will automatically transfer the traffic to the other working servers. Load balancing helps in detecting failures early on and manages them efficiently, making sure that failure of any kind doesn't affect the servers or the workload.