

Autoscaling and load balancing:

Q1. Differences between ELB, ALB, and NLB?

ELB:

- An ELB is a software-based load balancer which can be set up and configured in front of a collection of AWS Elastic Compute (EC2) instances. The load balancer serves as a single entry point for consumers of the EC2 instances and distributes incoming traffic across all machines available to receive requests. In addition to providing a single point of entry, the ELB also performs a vital role in improving the fault tolerance of the services which it fronts. The ELB regularly conducts a health check of all instances which have been registered with it, and only routes traffic to those machines which respond as active and healthy to the health check.

NLB:

- This is the distribution of traffic based on network variables, such as IP address and destination ports. It is layer 4 (TCP) and below and is not designed to take into consideration anything at the application layer such as content type, cookie data, custom headers, user location, or the application behavior. It is *context-less*, caring only about the network-layer information contained within the packets it is directing this way and that. This is a TCP Load Balancer only that does some NAT magic at the VPC level. It uses EIPs, so it has a static endpoint unlike ALB and CLBs (by default, contact support if this is a requirement for your CLB or ALB). Each Target can be on different ports.

ALB:

- This is the distribution of requests based on multiple variables, from the network layer to the application layer. It is context-aware and can direct requests based on any single variable as easily as it can a combination of variables. Applications are load balanced based on their peculiar behavior and not solely on server (operating system or virtualization layer) information. This is a feature filled L7 load balancer, HTTP and HTTPS listeners only. Provides the ability to route HTTP and HTTPS traffic based upon rules, host based or path based. Like an NLB, each Target can be on different ports. Even supports HTTP/2. Configurable range of health check status codes (CLB only supports 200 OK for HTTP health checks). The first difference is that the Application Load Balancer (as the name implies) works at the Application Layer (Layer 7 of the OSI model). The network load balancer works at layers 3 & 4 (network and transport layers). The network load balancer just forward requests whereas the application load balancer examines the contents of the HTTP request header to determine where to route the request. So, the application load balancer is performing content based routing.
- The other difference between the two is important because network load balancing cannot assure availability of the *application*. This is because it bases its decisions solely on network and TCP-layer variables and has no awareness of the application at all. Generally a network load balancer will determine "availability" based on the ability of a server to respond to ICMP ping, or to correctly complete the three-way TCP handshake. An application load balancer goes much deeper, and is capable of determining availability based on not only a successful HTTP GET of a particular page but also the verification that the *content* is as was expected based on the input parameters.

Q2. Differences between step scaling and target scaling.

Step Scaling:

We specify MULTIPLE thresholds Along with different responses.

Threshold A - add 1 instance when CPU Utilization is between 40% and 50%

Threshold B - add 2 instances when CPU Utilization is between 50% and 70%

Threshold C - add 3 instances when CPU Utilization is between 70% and 90%

There can be multiple thresholds

Target scaling:

AWS creates the Cloudwatch alarm for you on the basis of the target specified by you. It monitors the alarm and in the event of any alarm breach, tries to keep the metric near the target value by scaling in or scaling out. Additionally, target tracking scaling policy also adjusts to the changes in the metric due to a changing load pattern.

We can go for a target tracking policy if we are sure that our scaling metric increases/decreases in proportion of the number of instances in the auto scaling group. On the other hand, if I'd want more fine grained control of the scaling in or scaling out I'd go for step scaling say for example I've a real time websocket application behind a load balancer (with least connection algorithm) and the average CPU utilization suddenly starts jumping over 80, I'd add maybe 3-4 instances so that all new connections get routed to my new instances and the other instances

Q3. Differences between Launch configuration and launch template.

- Launch template is similar to launch configuration which usually Auto Scaling group uses to launch EC2 instances.
- However, defining a launch template instead of a launch configuration allows you to have multiple versions of a template.
- AWS recommends that we should use launch templates instead of launch configurations to ensure that we can leverage the latest features of Amazon EC2, such as T2 Unlimited instances.

Q4. Differences between EC2 health check and load balancer health check

EC2 health check watches for instance availability from hypervisor and networking point of view. For example, in case of a hardware problem, the check will fail. Also, if an instance was misconfigured and doesn't respond to network requests, it will be marked as faulty.

ELB health check verifies that a specified TCP port on an instance is accepting connections OR a specified web page returns 2xx code. Thus ELB health checks are a little bit smarter and verify that actual app works instead of verifying that just an instance works.

Q5. Create 2 auto-scaling groups with

- launch configuration and
- launch template

Using launch configuration:

The image shows two screenshots from the AWS Management Console. The top screenshot is the 'Create Auto Scaling Group' wizard, step 1. It shows the 'Launch Configuration' option selected. The bottom screenshot is the 'Create Launch Configuration' page, step 5. It shows the 'Assign a security group' section with 'Create a new security group' selected. The security group name is 'AutoScaling-Security-Group-8' and the description is 'AutoScaling-Security-Group-8 (2020-02-26 16:34:46.156+05:30)'. Below this is a table of rules:

Type	Protocol	Port Range	Source
SSH	TCP	22	Anywhere 0.0.0.0/0
HTTP	TCP	80	Anywhere 0.0.0.0/0
Custom TCP Rule	TCP	8080	Anywhere 0.0.0.0/0

Below the table is a warning message: 'Warning: 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.'

Making an autoscaling group:

aws

Services

Resource Groups

vedant.sharma@tothenew.com @ t...

N. Virginia

Support

1. Configure Auto Scaling group details2. Configure scaling policies3. Configure Notifications4. Configure Tags5. Review

Create Auto Scaling Group

Cancel and Exit

No default subnet found

Please choose another subnet in your default VPC, or choose another VPC.

Don't show me this again

Group name

Vedant-test-group

Launch Configuration

Vedant-config

Group size

Start with 1 instances

Network

vpc-d38d68b7 (172.31.0.0/16) | default (default)

Create new VPC

Subnet

subnet-06680a5b651f104dc(172.31.0.0/16) | us-east-1c

Create new subnet

No public IP addresses will be assigned

None of the instances in this Auto Scaling group will be assigned a public IP address because you have not chosen to launch in your default VPC and subnet.

CancelNext: Configure scaling policies

aws

Services

Resource Groups

vedant.sharma@tothenew.com @ t...

N. Virginia

Support

1. Configure Auto Scaling group details2. Configure scaling policies3. Configure Notifications4. Configure Tags5. Review

Create Auto Scaling Group

A tag consists of a case sensitive key-value pair that you can use to identify your group. For example, you could define a tag with Key = Environment and Value = Production. You can optionally choose to apply these tags to instances in the group when they launch. [Learn more.](#)

Key	Value	Tag New Instances
Name	Vedant-test-config-Group	<input checked="" type="checkbox"/>

Add tag49 remaining

aws

Services

Resource Groups

vedant.sharma@tothenew.com @ t...

N. Virginia

Support

Auto Scaling group creation status

Successfully created Auto Scaling group

[View creation log](#)

View

[View your Auto Scaling groups](#)

[View your launch configurations](#)

Here are some helpful resources to get you started

Close

Visible instance status on the autoscaling group activity check:

aws Services Resource Groups

vedant.sharma@tothenew.com @ t... N. Virginia Support

EC2 Dashboard Events Tags Reports Limits

INSTANCES

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

IMAGES

AMIs Bundle Tasks

ELASTIC BLOCK STORE

Volumes

Create Auto Scaling group Actions

Filter: vedant 1 to 2 of 2 Auto Scaling Groups

Name	Launch Configuration /	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grac
Vedant-test-te...	Vedant-test-template-gr...	1	1	1	1	us-east-1c	300	300
Vedant-test-gr...	Vedant-config	1	1	1	1	us-east-1c	300	300

Auto Scaling Group: Vedant-test-group

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

Filter: Any Status Filter scaling history... 1 to 1 of 1 History Items

Status	Description	Start Time	End Time
Successful	Launching a new EC2 instance: i-03ae3ae4f58eb1c25	2020 February 26 16:42:52 UTC+5:30	2020 February 26 16:43:24 UTC+5:30

Using Launch template:

aws Services Resource Groups

vedant.sharma@tothenew.com @ t... N. Virginia Support

New EC2 Experience Tell us what you think

EC2 Dashboard New Events Tags Reports Limits

INSTANCES

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

IMAGES

The old Launch Templates console will no longer be available after March 2020.

EC2 Launch templates

Launch templates (11) Actions Create launch template

Filter by tags or properties or search by keyword

Launch template ID	Launch template name	Default version	Latest version	Create time
lt-06a7560c83c95bd3d	Rishabh-lt	1	1	2020-02-26...
lt-0fbfd9e44571f13af	bootcamp-demo-001	1	2	2020-02-24...
lt-058aba21f142cc89b	Chirag-LT-1	1	1	2020-02-26...
lt-005116261c7bc1b13	vaibhav_wordpress2	1	1	2020-02-24...

Select a launch template above

aws

Services ▾ Resource Groups ▾ ★

vedant.sharma@tothenew.com @ t... ▾ N. Virginia ▾ Support ▾

EC2 > Launch templates > Create launch template

Create launch template

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

Launch template name and description

Launch template name - *required*

Must be unique to this account. Max 128 chars. No spaces or special characters like '&', "'", '@'.

Template version description

Max 255 chars

Auto scaling guidance [Info](#)
Select this if you intend to use this template with auto scaling

☒ Provide guidance to help me set up a template that I can use with auto scaling

Tagging and naming the template with specifying the instance types to be involved and also specifying the keypair to be used for the login.

aws

Services ▾ Resource Groups ▾ ★

vedant.sharma@tothenew.com @ t... ▾ N. Virginia ▾ Support ▾

Create launch template

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

Launch template name and description

Launch template name - *required*

Must be unique to this account. Max 128 chars. No spaces or special characters like '&', "'", '@'.

Template version description

Max 255 chars

Auto scaling guidance [Info](#)
Select this if you intend to use this template with auto scaling

☒ Provide guidance to help me set up a template that I can use with auto scaling

▼ Template tags

Key	Info	Value	Info
<input type="text" value="Name"/>	X	<input type="text" value="Vedant-test-template-group"/>	X
<input type="button" value="Remove tag"/>			

aws Services Resource Groups

vedant.sharma@tothenew.com @ t... N. Virginia Support

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type
ami-07ebf5b5428b6f4d
Catalog: Quick Start architecture: 64-bit (x86) virtualization: hvm

Instance type Info

Instance type

t2.micro
Family: General purpose 1 vCPU 1 GiB Memory
On-Demand Linux pricing: 0.0116 USD per Hour
On-Demand Windows pricing: 0.0162 USD per Hour

Key pair (login) Info

Key pair name

Vedant-Bootcamp

Network settings

aws Services Resource Groups

vedant.sharma@tothenew.com @ t... N. Virginia Support

New EC2 Experience
Tell us what you think

EC2 Dashboard New
Events
Tags
Reports
Limits

▼ INSTANCES
Instances
Instance Types
Launch Templates New
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts New
Scheduled Instances

The old Launch Templates console will no longer be available after March 2020.

EC2 > Launch templates

Launch templates (17)

filter by tags or properties or search by keyword

search: vedant X Clear filters

	Launch template ID	Launch template name	Default version	Latest version
	lt-0a84345f4c66f897a	Vedant-Test-template-group	1	1

Now creating an auto scaling group using the launch template created;

aws

Services ▾ Resource Groups ▾ ★

vedant.sharma@tothenew.com @ t... ▾ N. Virginia ▾ Support ▾

Create Auto Scaling Group

Cancel and Exit

● Launch Configuration

You can continue to use your launch configurations if they support the Amazon EC2 features you need. [Learn more](#)

Create a new launch configuration

● Launch Template New

Launch templates give you the option of launching one type of instance, or a combination of instance types and purchase options. Launch templates include the latest Amazon EC2 features and can be updated and versioned. [Learn more](#)

Create new launch template

Cancel

Next Step

Specifying a source for the template file:

aws

Services ▾ Resource Groups ▾ ★

vedant.sharma@tothenew.com @ t... ▾ N. Virginia ▾ Support ▾

Key Info

Value Info

Q Name X

Q Vedant-Test-Template X

Remove tag

Add tag

49 remaining (Up to 50 tags maximum)

▼ Source template

You can optionally specify a source template if you would like to create a template from another existing template

Launch template name

Name: Vedant-Test-template-group

lt-0a84345f4c66f897a

Source template version

1 (Default)

test template for autoscaling group

Launch template contents

Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

Creating a Auto scaling group by mentioning the template:

aws

Services

Resource Groups

vedant.sharma@tothenew.com @ t...

N. Virginia

Support

1. Configure Auto Scaling group details2. Configure scaling policies3. Configure Notifications4. Configure Tags5. Review

Create Auto Scaling Group

Cancel and Exit

No default subnet found

Please choose another subnet in your default VPC, or choose another VPC.

Don't show me this again

Group name

vedant-test-group

Launch Template

lt-0d4c34f20f761e3ee

Launch Template Version

Default

Create new launch template

Launch Template Description

test template group

Fleet Composition

Adhere to the launch template

The launch template determines the instance type and purchase option (On-Demand or Spot).

Combine purchase options and instances

Choose a mix of On-Demand Instances and Spot Instances and multiple instance types. Spot Instances are automatically launched at the lowest price available.

Group size

Start with 1 instances

Network

vpc-d38d68b7 (172.31.0.0/16) | default (default)

Create new VPC

CancelNext: Configure scaling policies

aws

Services

Resource Groups

vedant.sharma@tothenew.com @ t...

N. Virginia

Support

Auto Scaling group creation status

✓ Successfully created Auto Scaling group

[View creation log](#)

View

View your Auto Scaling groups

View your launch configurations

Here are some helpful resources to get you started

Close

Visible launched instances:

aws Services Resource Groups

vedant.sharma@tothenew.com @ t... N. Virginia Support

EC2 Dashboard New

Events Tags Reports Limits

INSTANCES

Instances

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

IMAGES

Launch Instance Connect Actions

search: vedant Add filter

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
Vedant-test-c...	i-03ae3ae4f58eb1c25	t2.micro	us-east-1c	running	2/2 checks ...	None	-
Vedant-test-t...	i-04fbc258b09657c57	t2.micro	us-east-1c	running	Initializing	None	-
Vedant-Publi...	i-078a1ccb42ab7c6e6	t2.micro	us-east-1c	shutting-do...		None	3
Vedant-Priva...	i-085e3055557277e...	t2.micro	us-east-1c	shutting-do...		None	-
Vedant-Repa...	i-0f2d06d5fd17d886c	t2.micro	us-east-1c	shutting-do...		None	-

Instance: i-04fbc258b09657c57 (Vedant-test-template-group) Private IP: 172.31.234.49

Description Status Checks Monitoring Tags

Instance ID i-04fbc258b09657c57 Public DNS (IPv4) -

Instance state running IPv4 Public IP -

Instance type t2.micro IPv6 IPs -

aws Services Resource Groups

vedant.sharma@tothenew.com @ t... N. Virginia Support

EC2 Dashboard

Events Tags Reports Limits

INSTANCES

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

IMAGES

AMIs Bundle Tasks

ELASTIC BLOCK STORE

Volumes

Create Auto Scaling group Actions

Filter: Q vedant

Name	Launch Configuration /	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grac
Vedant-test-te...	Vedant-test-template-gr...	1	1	1	1	us-east-1c	300	300
Vedant-test-gr...	Vedant-config	1	1	1	1	us-east-1c	300	300

Auto Scaling Group: Vedant-test-template-group

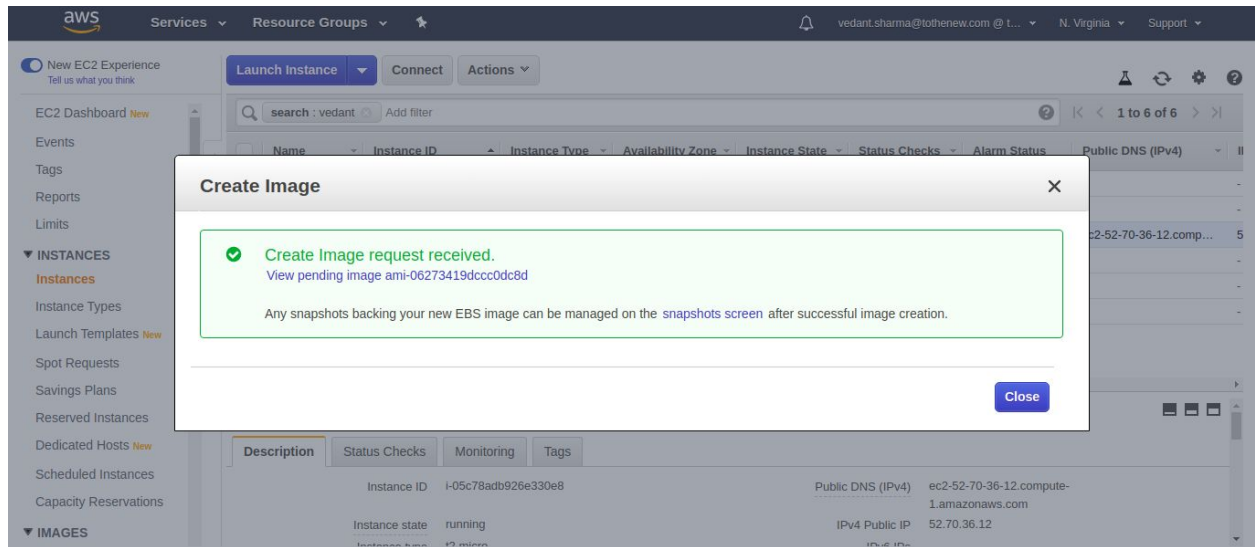
Details Activity History Scaling Policies Instances Monitoring Notifications Tags Scheduled Actions Lifecycle Hooks

Filter: Any Status Filter scaling history...

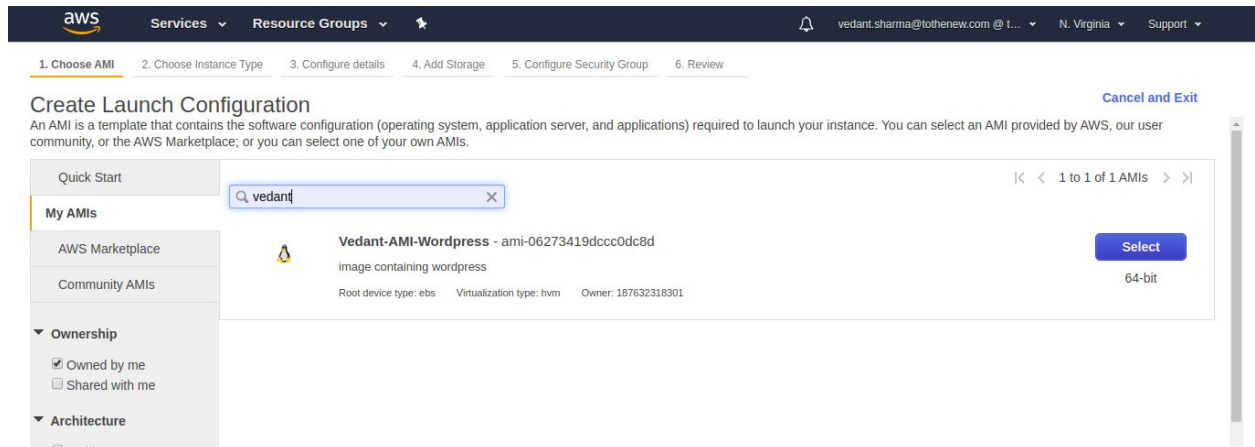
Status	Description	Start Time	End Time
Successful	Launching a new EC2 Instance: i-04fbc258b09657c57	2020 February 26 17:10:14 UTC+5:30	2020 February 26 17:10:46 UTC+5:30

Q6. .Setup autoscaling Wordpress application with the Application load balancer. Auto-scaling should be triggered based on CPU usage of EC2 instances.

Creating an AMI of an instance with wordpress running:



Create an auto scaling group using Ami



aws

Services

Resource Groups

vedant.sharma@tothenew.com @ t...

N. Virginia

Support

1. Configure Auto Scaling group details

2. Configure scaling policies

3. Configure Notifications

4. Configure Tags

5. Review

Create Auto Scaling Group

Keep this group at its initial size

Use scaling policies to adjust the capacity of this group

Scale between 3 and 3 instances. These will be the minimum and maximum size of your group.

Scale Group Size

Name: Scale Group Size

Metric type: Average CPU Utilization

Target value: 70

Instances need: 300 seconds to warm up after scaling

Disable scale-in:

Scale the Auto Scaling group using step or simple scaling policies

Cancel

Previous

Review

Next: Configure Notifications

aws

Services

Resource Groups

vedant.sharma@tothenew.com @ t...

N. Virginia

Support

1. Configure Auto Scaling group details

2. Configure scaling policies

3. Configure Notifications

4. Configure Tags

5. Review

Create Auto Scaling Group

Auto Scaling Group Details

Group name

Group size

Minimum Group Size

Maximum Group Size

Subnet(s)

Health Check Grace Period

Detailed Monitoring

Instance Protection

Service-Linked Role

Vedant-test-group

3

3

5

subnet-01d770a77bb69a1f8,subnet-0b14d740243a0165c

300

No

None

AWSServiceRoleForAutoScaling

Scaling Policies

Scale Group Size

Maintain metric type Average CPU Utilization at target value 40, with 300 seconds for instances to warm up.

Notifications

Cancel

Previous

Create Auto Scaling group

Auto Scaling group creation status

✓ Successfully created Auto Scaling group

[View creation log](#)

View

[View your Auto Scaling groups](#)

[View your launch configurations](#)

Here are some helpful resources to get you started

Close

Creating a target group:

The screenshot shows the 'Create target group' dialog in the AWS Management Console. The dialog is titled 'Create target group' and includes a close button (X) in the top right corner. Below the title, there is a descriptive text: 'Your load balancer routes requests to the targets in a target group using the target group settings that you specify, and performs health checks on the targets using the health check settings that you specify.'

The form contains the following fields and options:

- Target group name:** A text input field containing 'Wordpress'.
- Target type:** A radio button group with three options: 'Instance' (selected), 'IP', and 'Lambda function'.
- Protocol:** A dropdown menu set to 'HTTP'.
- Port:** A text input field containing '80'.
- VPC:** A dropdown menu showing 'vpc-d38d68b7 (172.31.0.0/16) | default (My VPC)'.
- Health check settings:** A section with a 'Protocol' dropdown menu set to 'HTTP'.

The left sidebar of the console shows the 'EC2 Dashboard' and various navigation links under 'INSTANCES' and 'IMAGES'. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information.

The screenshot shows the 'Health check settings' dialog in the AWS Management Console. The dialog is titled 'Health check settings' and includes a close button (X) in the top right corner. Below the title, there is a 'VPC' dropdown menu showing 'vpc-d38d68b7 (172.31.0.0/16) | default (My VPC)'. The 'Protocol' dropdown menu is set to 'HTTP'.

The 'Path' text input field contains '/test/index.html'.

The 'Advanced health check settings' section is expanded, showing the following fields and options:

- Port:** A radio button group with two options: 'traffic port' (selected) and 'override'.
- Healthy threshold:** A text input field containing '5'.
- Unhealthy threshold:** A text input field containing '2'.
- Timeout:** A text input field containing '5' with a unit of 'seconds'.
- Interval:** A text input field containing '30' with a unit of 'seconds'.
- Success codes:** A text input field containing '200'.

At the bottom right of the dialog, there are two buttons: 'Cancel' and 'Create'.

Adding the instance to the target group

aws Services Resource Groups

vedant.sharma@tothenew.com @ t... N. Virginia Support

New EC2 Experience
Tell us what you think

Create target group Actions

Filter by tags and attributes or search by keyword

Name	Port	Protocol	Target type	Load Balan	VPC ID	Monitoring
AkshayTG	80	HTTP	instance		vpc-0eb8696e500b5c5d	
Wordpress	80	HTTP	instance		vpc-d38d68b7	
kube-api-servers-ig	6443	TCP	instance		vpc-07c3975194af4d40f	
pooja	80	HTTP	instance		vpc-00470a42c196d84e	
test1-LB	80	HTTP	instance		vpc-0c2138f589c308a80	
vedant-target		HTTP	instance		vpc-d38d68b7	
Chirag-TG...		HTTP	instance	Chirag-LB-...	vpc-00470a42c196d84e	
poojacorre...		HTTP	instance	poojacorre...	vpc-00470a42c196d84e	
POOJA		HTTP	instance	POOJALB	vpc-00470a42c196d84e	

Target group: vedant-target-group

Description Targets Health checks Monitoring Tags

Basic Configuration

Creating an application load balancer:

aws Services Resource Groups

vedant.sharma@tothenew.com @ t... N. Virginia Support

1. Configure Load Balancer 2. Configure Security Settings 3. Configure Security Groups 4. Configure Routing 5. Register Targets 6. Review

Step 1: Configure Load Balancer

Basic Configuration

To configure your load balancer, provide a name, select a scheme, specify one or more listeners, and select a network. The default configuration is an Internet-facing load balancer in the selected network with a listener that receives HTTP traffic on port 80.

Name

Scheme ☒ internet-facing ☐ internal

IP address type

Listeners

A listener is a process that checks for connection requests, using the protocol and port that you configured.

Load Balancer Protocol	Load Balancer Port
HTTP	80

Cancel Next: Configure Security Settings

Associating with a security group:

aws Services Resource Groups

vedant.sharma@tothenew.com @ t... N. Virginia Support

1. Configure Load Balancer 2. Configure Security Settings 3. Configure Security Groups 4. Configure Routing 5. Register Targets 6. Review

Step 3: Configure Security Groups

A security group is a set of firewall rules that control the traffic to your load balancer. On this page, you can add rules to allow specific traffic to reach your load balancer. First, decide whether to create a new security group or select an existing one.

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

Filter

Security Group ID	Name	Description	Actions
sg-027522c779c12164c	Chirag-ALB-SG	load-balancer-wizard-1 created on 2020-02-26T19:40:34.731+05:30	Copy to new
sg-0f87044e49cf3814	default	default VPC security group	Copy to new
sg-062b1633ad7e05c21	launch-wizard-177	launch-wizard-177 created 2020-02-26T18:25:24.592+05:30	Copy to new
sg-051d671c160aea760	sarthak	ssh https http	Copy to new

Cancel Previous Next: Configure Routing

Associating target group with ALB

aws

Services

Resource Groups

★

vedant.sharma@tothenew.com @ t...N. VirginiaSupport

1. Configure Load Balancer2. Configure Security Settings3. Configure Security Groups4. Configure Routing5. Register Targets6. Review

Step 4: Configure Routing

Your load balancer routes requests to the targets in this target group using the protocol and port that you specify, and performs health checks on the targets using these health check settings. Note that each target group can be associated with only one load balancer.

Target group

Target group

Existing target group

Name

Vedant-test-targetgroup

Target type

Instance

IP

Lambda function

Protocol

HTTP

Port

80

Health checks

Protocol

HTTP

Cancel

Previous

Next: Register Targets

aws

Services

Resource Groups

★

vedant.sharma@tothenew.com @ t...N. VirginiaSupport

1. Configure Load Balancer2. Configure Security Settings3. Configure Security Groups4. Configure Routing5. Register Targets6. Review

Step 6: Review

Please review the load balancer details before continuing

▼ Load balancer

Name

Vedant-ALB

Scheme

internet-facing

Listeners

Port:80 - Protocol:HTTP

IP address type

ipv4

VPC

vpc-00470a42fc196d84e (sarthak)

Subnets

subnet-01d770a77bb69a1f8 (sarthak-load-balancer-1), subnet-0b14d740243a0165c (sarthak-loadbalancer-2)

Tags

Name:Vedant-Test-ALB

▼ Security groups

Security groups

sg-010a51e206309b0c9

▼ Routing

Target group

Existing target group

Target group name

Vedant-test-targetgroup

Port

80

Cancel

Previous

Create

Creating an ALB

aws Services Resource Groups

vedant.sharma@tothenew.com @ t... N. Virginia Support

New EC2 Experience Tell us what you think

STORE

- Volumes
- Snapshots
- Lifecycle Manager

NETWORK & SECURITY

- Security Groups
- Elastic IPs *New*
- Placement Groups *New*
- Key Pairs *New*
- Network Interfaces

LOAD BALANCING

- Load Balancers**
- Target Groups

AUTO SCALING

- Launch Configurations
- Auto Scaling Groups

Create Load Balancer Actions

search : Vedant Add filter

Name	DNS name	State	VPC ID	Availability Zones	Type
Vedant-test-ALB	Vedant-test-ALB-161999225...	active	vpc-00470a42fc196d84e	us-east-1e, us-east-1b...	application

Load balancer: Vedant-test-ALB

Description Listeners **Monitoring** Integrated services Tags

CloudWatch alarms: ✔ No alarms configured [Create Alarm](#)

CloudWatch metrics: Showing data for: Last Hour [View](#)

Below are your CloudWatch metrics for the selected resources (a maximum of 10). Click on a graph to see an expanded view. All times shown are in UTC.

Load Balancer Creation Status



Successfully created load balancer

Load balancer **Vedant-ALB** was successfully created.

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

Suggested next steps

- Discover other services that you can integrate with your load balancer. Visit the **Integrated services** tab within **Vedant-ALB**
- Consider using AWS Global Accelerator to further improve the availability and performance of your applications. [AWS Global Accelerator console](#)

Close

Copying the available DNS name and putting it on the browser in order to access the page

AWS Management Console interface showing the 'Load Balancers' page. The left sidebar contains navigation options: STORE (Volumes, Snapshots, Lifecycle Manager), NETWORK & SECURITY (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), LOAD BALANCING (Load Balancers, Target Groups), and AUTO SCALING (Launch Configurations, Auto Scaling Groups). The main content area displays a table of load balancers with columns: Name, DNS name, State, VPC ID, Availability Zones, and Type. The table lists several load balancers, with 'Vedant-Test-ALB' selected. Below the table, the 'Basic Configuration' section shows details for 'Vedant-Test-ALB':

Property	Value
Name	Vedant-Test-ALB
ARN	arn:aws:elasticloadbalancing:us-east-1:187632318301:loadbalancer/app/Vedant-Test-ALB/357de9c18f299ec2
DNS name	Vedant-Test-ALB-737151566.us-east-1.elb.amazonaws.com (A Record)
State	active

Browser screenshot showing the website 'vedant' at the URL 'vedant-test-alb-737151566.us-east-1.elb.amazonaws.com'. The page features a navigation bar with links: Home, About, Blog, Contact, and a search icon. The main content area displays the title 'The New UMoMA Opens its Doors' in large, bold, black text. Below the title is a large, solid red rectangular block.

Q7. Create another Wordpress website and use the ALB created above to send traffic to this website based on the hostname

Adding rules to the load balancer for path based forwarding

The screenshot shows the AWS Management Console interface for creating a new load balancer. The 'Listeners' tab is active, displaying a table with one listener: 'Vedant-Test-ALB' on port 80, forwarding to 'Vedant-Target-Group'. The 'Rules' column shows a default rule: 'Default: forwarding to Vedant-Target-Group'. The 'Add listener' button is visible, along with 'Edit' and 'Delete' buttons for the existing listener.

Setting /wp-admin to be forwarded to the instances running in the different target group

The screenshot shows the 'Rules' page for the 'Vedant-Test-ALB | HTTP:80' listener. It displays two rules:

- Rule 1:** IF Path is /wp-admin, THEN Forward to Vedant-test-TG2: 1 (100%), Group-level stickiness: Off.
- Rule 2 (Default):** IF Requests otherwise not routed, THEN Forward to Vedant-Target-Group: 1 (100%), Group-level stickiness: Off.

The 'last' rule is marked as 'HTTP 80: default action' and 'This rule cannot be moved or deleted'.

aws

Services

Resource Groups

vedant.sharma@tothenew.com @ t...

N. Virginia

Support

<

Rules

+

↑

↓

Vedant-Test-ALB | HTTP:80

↺

?

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

Cancel

Save

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

Insert Rule

RULE ID	IF (all match)	THEN
1 A rule ID (ARN) is generated when you save your rule.	<div>Path...</div> <div>is /wp-admin</div> <div>or Value</div> <div>✓</div> <div>+ Add condition</div>	<div>1. Forward to...</div> <div>Target group : Weight (0-999)</div> <div>Vedant-test-TG2</div> <div>1</div> <div>Traffic distribution 100%</div> <div>Select a target group</div> <div>0</div> <div>Group-level stickiness</div> <div>✓</div> <div>+ Add action</div>

←

→

↺

Not secure

testwordpress.com/wp-login.php?redirect_to=http%3A%2F%2Fvedant-test-alb-737151566.us-east-1.elb.amazonaws.com%2Fwp-admin%2F

☆

🔒

📱

🔍

🔧

🔧

🔧

Apps

Gmail


YouTube

Maps

Dashboard

Devhints

Bash-Scripting



Username or Email Address

Password

☐ Remember Me

Log In

Lost your password?

→ Back to vedant

Q8. Use NLB that replaces the ALB in the above setup.

aws

Services

Resource Groups

vedant.sharma@tothenew.com @ t...N. VirginiaSupport

1. Configure Load Balancer

2. Configure Security Settings

3. Configure Routing

4. Register Targets

5. Review

Step 1: Configure Load Balancer

Basic Configuration

To configure your load balancer, provide a name, select a scheme, specify one or more listeners, and select a network. The default configuration is an Internet-facing load balancer in the selected network with a listener that receives TCP traffic on port 80.

Name

Vedant-Test-NLB

Scheme

internet-facing

internal

Listeners

A listener is a process that checks for connection requests, using the protocol and port that you configured.

Load Balancer Protocol

TCP

Load Balancer Port

80

Add listener

Cancel

Next: Configure Security Settings

1. Configure Load Balancer

2. Configure Security Settings

3. Configure Routing

4. Register Targets

5. Review

Step 3: Configure Routing

Your load balancer routes requests to the targets in this target group using the protocol and port that you specify, and performs health checks on the targets using these health check settings. Note that each target group can be associated with only one load balancer.

Target group

Target group

New target group

Name

Vedant-NLB-Test

Target type

Instance

IP

Protocol

TCP

Port

80

Health checks

Protocol

HTTP

Cancel

Previous

Next: Register Targets

Load Balancer Creation Status

Successfully created load balancer

Load balancer **Vedant-Test-NLB** was successfully created.

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

Suggested next steps

- Discover other services that you can integrate with your load balancer. Visit the **Integrated services** tab within **Vedant-Test-NLB**.
- Consider using AWS Global Accelerator to further improve the availability and performance of your applications. [AWS Global Accelerator console](#)

Close

NLB using a different target group:

aws Services Resource Groups vedant.sharma@tothenew.com @ t... N. Virginia Support

New EC2 Experience
Tell us what you think

EC2 Dashboard **New**

Events

Tags

Reports

Limits

▼ INSTANCES

Instances

Instance Types

Launch Templates **New**

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts **New**

Scheduled Instances

Capacity Reservations

▼ IMAGES

Create Load Balancer Actions

search : vedant Add filter

	Name	DNS name	State	VPC ID	Availability Zones	Type
<input type="checkbox"/>	Vedant-Test-ALB	Vedant-Test-ALB-737151566...	active	vpc-00470a42fc196d84e	us-east-1e, us-east-1b...	application
<input checked="" type="checkbox"/>	Vedant-Test-NLB	Vedant-Test-NLB-b318c30a6...	provisioning	vpc-00470a42fc196d84e	us-east-1c, us-east-1e...	network

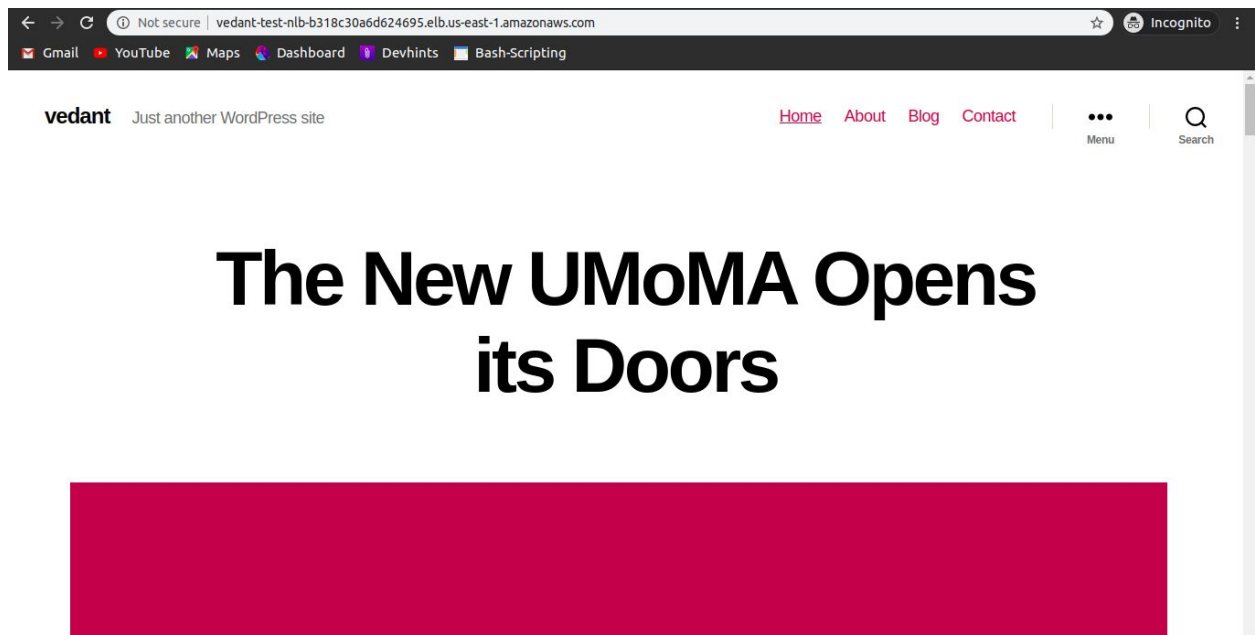
A listener checks for connection requests using its configured protocol and port, and the load balancer uses the listener rules to route requests to targets. You can add, remove, or update listeners and listener rules.

Add listener Edit Delete

<input type="checkbox"/>	Listener ID	Security policy	SSL Certificate	Default action
<input type="checkbox"/>	TCP : 80	N/A	N/A	Forward to Vedant-NLB-Test

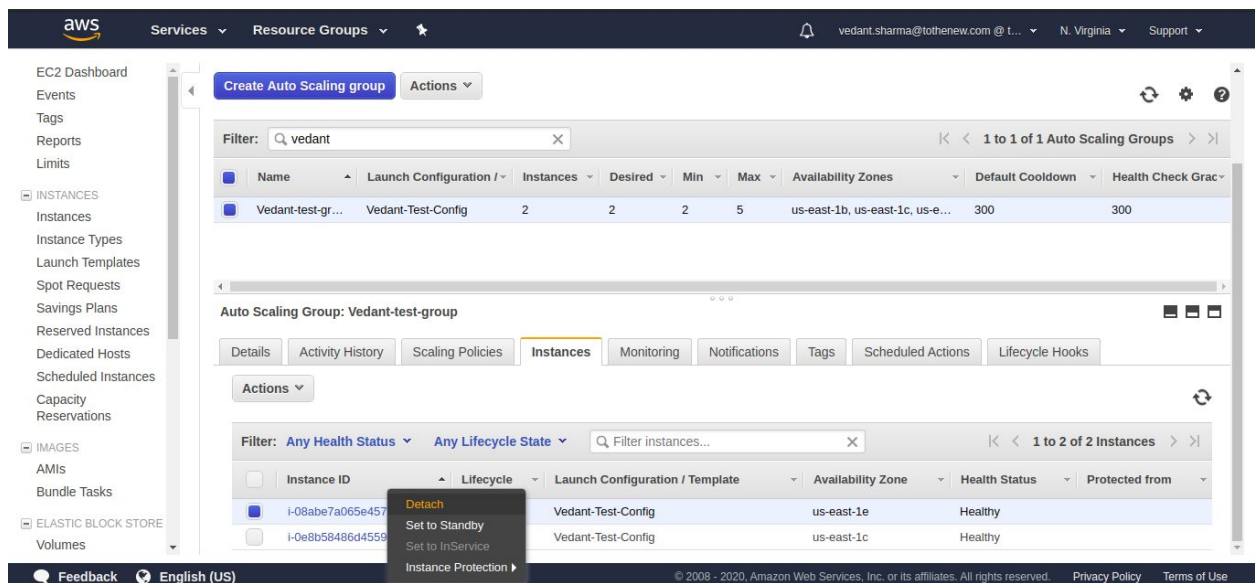
arn...87b1701663bfd566

Wordpress website up using NLB



Q9. Take an instance out of the ASG

Editing the min number of instances required before detaching the instance:



Detaching

The screenshot shows the AWS Management Console interface for an Auto Scaling Group (ASG) named 'Vedant-test-group'. The 'Instances' tab is selected, displaying a table of instances. The first instance, 'i-08abe7a065e4578c8', is in the 'Detaching' lifecycle state, while the second instance, 'i-0e8b58486d4559ee2', is in the 'InService' state. The table columns include Name, Launch Configuration / Template, Instances, Desired, Min, Max, Availability Zones, Default Cooldown, and Health Check Grace Period.

Name	Launch Configuration / Template	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grace Period
Vedant-test-gr...	Vedant-Test-Config	2	2	2	5	us-east-1b, us-east-1c, us-e...	300	300

Q10. Put scale-in protection on an instance in the ASG.

Selecting the instance under the autoscaling group and under actions tag selecting instance protection and then set scale in protection

The screenshot shows the same AWS Management Console interface, but now the instance 'i-05e800eac9dc' is selected. The 'Actions' dropdown menu is open, and the 'Set Scale In Protection' option is highlighted. The table shows three instances: 'i-05e800eac9dc' (Pending), 'i-0c14babf73361c5c' (Healthy), and 'i-0e8b58486d4559ee2' (InService).

Instance ID	Lifecycle	Launch Configuration / Template	Availability Zone	Health Status	Protected from
i-05e800eac9dc	Detaching	Vedant-Test-Config	us-east-1e	Healthy	
i-0c14babf73361c5c	Pending	Vedant-Test-Config	us-east-1b	Healthy	
i-0e8b58486d4559ee2	InService	Vedant-Test-Config	us-east-1c	Healthy	

Protected from scale in:

Q11. Put Schedules in ASG to:

- Remove all instances of the ASG at 8 PM
- Launch a minimum of 2 instances at 10 AM

Creating a Schedule Action

Create Scheduled Action

Name

Removing all instances at 8 pm

Auto Scaling Group

Vedant-test-group

Provide at least one of Min, Max and Desired Capacity

Min

0

Max

0

Desired Capacity

0

Recurrence

Once

Start Time

2020-02-28

20 : 00

UTC

Specify the start time in UTC

The first time this scheduled action will run

Cancel

Create

Create Scheduled Action



Name

Launch a minimum of 2 instances at 10 AM

Auto Scaling Group

Vedant-test-group

Provide at least one of Min, Max and Desired Capacity

Min

2

Max

4

Desired Capacity

2

Recurrence

Once

Start Time

2020-02-28

10 : 00

UTC Specify the start time in UTC

The first time this scheduled action will run

Cancel

Create

Removing all instances at 8 pm

2020 February 29 01:30:00 UTC+5:30

0

