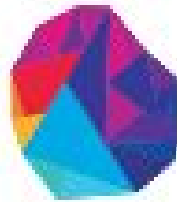


ASSESSMENT ON: AUTO SCALING AND LOAD BALANCER

**TO
THE
NEW.**



1.Differences between ELB, ALB, and NLB. Where will you use which one?

ANS:

	Application Load Balancer	Network Load Balancer	Classic Load Balancer
Quick Info	Application Load Balancer is one of the newer types of load balancers. AWS recommends that you use this instead of the classic load balancer. This ELB operates at OSI Layer 7 (HTTP)	Network Load Balancer is one of the newer types of load balancers. AWS recommends that you use this instead of the classic load balancer. This ELB operates at OSI Layer 4 (TCP)	The classic load balancer is the older type of load balancer in AWS. AWS recommends that you use ALB or NLB instead for your load balancing requirements. Operates at either Layer 4 or Layer 7
Use Case	If you need flexible application management, use an Application Load Balancer.	If extreme performance and static IP is needed for your application, use a Network Load Balancer.	If you have an existing application that was built within the EC2-Classic network, then use a Classic Load Balancer.
Supported Protocols	HTTP and HTTPS	TCP, UDP and TLS	TCP, SSL/TLS, HTTP, HTTPS
Supported Platform	VPC	VPC	VPC and EC2-Classic (deprecated)
Can load balance to multiple ports on the same instance	Yes (Useful for containerized applications)	Yes	No

Unique features	<ul style="list-style-type: none"> • You can set Lambda functions as load balancing targets • Only ALB supports the following content-based routing methods: <ul style="list-style-type: none"> • Path based routing • Host-based routing • HTTP header-based routing • HTTP method-based routing • Query string parameter-based routing • Source IP address CIDR-based routing • Natively supports HTTP/2, IPv6 • Support for multiple SSL certificates on the ALB using • Server Name Indication (SNI) • Allows tag-based IAM permission policies • Can be configured for slow start (linearly increases the number of requests sent to targets) • Supports round-robin load balancing • You can offload the authentication functionality from your apps into ALB • Can redirect an incoming request from one URL to another URL, including HTTP to HTTPS • You can set HTTP or custom responses for incoming requests to the ALB, offloading this task from your application 	<ul style="list-style-type: none"> • High throughput/low latency ELB • Can be assigned a static IP address • Can be assigned an elastic IP address • Preserves source IP address of non-HTTP applications on EC2 instances • Offer multi-protocol listeners, allowing you to run applications such as DNS that rely on both TCP and UDP protocols on the same port behind a Network Load Balancer. • TLS Termination 	<ul style="list-style-type: none"> • You can create custom security policies detailing which ciphers and protocols are supported by the ELB • Supports both IPv4 and IPv6 for EC2-Classical network
-----------------	---	--	---

2.Differences between step scaling and target scaling.

ANS:

With step scaling and simple scaling, you choose scaling metrics and threshold values for the CloudWatch alarms that trigger the scaling process. You also define how your Auto Scaling group should be scaled when a threshold is in breach for a specified number of evaluation periods.

AWS recommends that you use a target tracking scaling policy to scale on a metric like average CPU utilization or the RequestCountPerTarget metric from the Application Load Balancer.

Metrics that decrease when capacity increases and increase when capacity decreases can be used to proportionally scale out or in the number of instances using target tracking. This helps ensure that Amazon EC2 Auto Scaling follows the demand curve for your applications closely. With target tracking scaling policies, you select a scaling metric and set a target value. Amazon EC2 Auto Scaling creates and manages the CloudWatch alarms that trigger the scaling policy and calculates the scaling adjustment based on the metric and the target value. The scaling policy adds or removes capacity as required to keep the metric at, or close to, the specified target value. In addition to keeping the metric close to the target value, a target tracking scaling policy also adjusts to the changes in the metric due to a changing load pattern.

3.Differences between Launch configuration and launch template.

ANS:

A launch configuration is an instance configuration template that an Auto Scaling group uses to launch EC2 instances. When you create a launch configuration, you specify information for the instances. Include the ID of the Amazon Machine Image (AMI), the instance type, a key pair, one or more security groups, and a block device mapping.

A launch template is similar to a launch configuration, in that it specifies instance configuration information. Included are the ID of the Amazon Machine Image (AMI), the instance type, a key pair, security groups, and the other parameters that you use to launch EC2 instances.

*However, defining a launch template instead of a launch configuration allows you to have multiple versions of a template. With versioning, you can create a subset of the full set of parameters and then reuse it to create other templates or template versions. For example, you can create a default template that defines common configuration parameters such as tags or network configurations, and allow the other parameters to be specified as part of another version of the same template.

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

ANS:

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.

EC2 instance health check	Elastic Load Balancer (ELB) health check	Auto Scaling and Custom health checks
<ul style="list-style-type: none"> Amazon EC2 performs automated checks on every running EC2 instance to identify hardware and software issues. Status checks are performed every minute and each returns a pass or a fail status. <ul style="list-style-type: none"> If all checks pass, the overall status of the instance is OK. If one or more checks fail, the overall status is impaired. Status checks are built into EC2, so they cannot be disabled or deleted. You can create or delete alarms that are triggered based on the result of the status checks. There are two types of status checks <ul style="list-style-type: none"> System Status Checks <ul style="list-style-type: none"> These checks detect underlying problems with your instance that require AWS involvement to repair. When a system status check fails, you can choose to wait for AWS to fix the issue, or you can resolve it yourself. Instance Status Checks <ul style="list-style-type: none"> Monitor the software and network configuration of your individual instance. Amazon EC2 checks the health of an instance by sending an address resolution protocol (ARP) request to the ENI. These checks detect problems that require your involvement to repair. 	<ul style="list-style-type: none"> To discover the availability of your registered EC2 instances, a load balancer periodically sends pings, attempts connections, or sends requests to test the EC2 instances. The status of the instances that are healthy at the time of the health check is InService. The status of any instances that are unhealthy at the time of the health check is OutOfService. When configuring a health check, you would need to provide the following: <ul style="list-style-type: none"> a specific port protocol to use <ul style="list-style-type: none"> HTTP/HTTPS health check succeeds if the instance returns a 200 response code within the health check interval. A TCP health check succeeds if the TCP connection succeeds. An SSL health check succeeds if the SSL handshake succeeds. ping path ELB health checks do not support WebSockets. The load balancer routes requests only to the healthy instances. When an instance becomes impaired, the load balancer resumes routing requests to the instance only when it has been restored to a healthy state. The load balancer checks the health of the registered instances using either <ul style="list-style-type: none"> the default health check configuration provided by Elastic Load Balancing or a health check configuration that you configure (auto scaling or custom health checks for example). Network Load Balancers use active and passive health checks to determine whether a target is available to handle requests. <ul style="list-style-type: none"> With active health checks, the load balancer periodically sends a request to each registered target to check its status. After each health check is completed, the load balancer node closes the connection that was established. With passive health checks, the load balancer observes how targets respond to connections, which enables it to detect an unhealthy target before it is reported as unhealthy by active health checks. You cannot disable, configure, or monitor passive health checks. 	<ul style="list-style-type: none"> All instances in your Auto Scaling group start in the healthy state. Instances are assumed to be healthy unless EC2 Auto Scaling receives notification that they are unhealthy. This notification can come from one or more of the following sources: <ul style="list-style-type: none"> Amazon EC2 (default) Elastic Load Balancing A custom health check. After Amazon EC2 Auto Scaling marks an instance as unhealthy, it is scheduled for replacement. If you do not want instances to be replaced, you can suspend the health check process for any individual Auto Scaling group. If an instance is in any state other than running or if the system status is impaired, Amazon EC2 Auto Scaling considers the instance to be unhealthy and launches a replacement instance. If you attached a load balancer or target group to your Auto Scaling group, Amazon EC2 Auto Scaling determines the health status of the instances by checking both the EC2 status checks and the Elastic Load Balancing health checks. Amazon EC2 Auto Scaling waits until the health check grace period ends before checking the health status of the instance. Ensure that the health check grace period covers the expected startup time for your application. Health check grace period does not start until lifecycle hook actions are completed and the instance enters the InService state. With custom health checks, you can send an instance's health information directly from your system to Amazon EC2 Auto Scaling.

5.Create 2 auto-scaling groups with

- launch configuration and

STEP 1: EC2 > Launch Configuration

1. Choose AMI
2. Choose Instance Type
3. Configure details
4. Add Storage
5. Configure Security Group
6. Review

Cancel and Exit

Create Launch Configuration

Free tier eligible

Root device type: ebs

Virtualization type: hvm

SUSE Linux

Free tier eligible

SUSE Linux Enterprise Server 15 SP1 (HVM), SSD Volume Type - ami-0df6cfabf4385b7

SUSE Linux Enterprise Server 15 Service Pack 1 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.

64-bit

Select

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-07ebfd5b3428b6f4d

Ubuntu Server 18.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

64-bit

Select

STEP 2: Provide security group

1. Choose AMI
2. Choose Instance Type
3. Configure details
4. Add Storage
5. Configure Security Group
6. Review

Create Launch Configuration

<input type="checkbox"/>	sg-073675c2618ca346c	launch-wizard-maithely	vpc-d38d68b7	launch-wizard-44 created 2020-02-20T13:12:35.227+05:30
<input type="checkbox"/>	sg-01d6c24eb1c7a1dda	pooja	vpc-01d9bca1ea53fdce9	http,https,tcp
<input type="checkbox"/>	sg-0065b5d69c556e3fc	Revant_launch	vpc-d38d68b7	AutoScaling-Security-Group-17 (2020-02-27 11:32:21.350+05:30)
<input type="checkbox"/>	sg-0b0eead678ad6a5ac	Rishabh-sg	vpc-01d9bca1ea53fdce9	Rishabh-sg
<input checked="" type="checkbox"/>	sg-051d671c160aea760	sarthak	vpc-00470a42fc196d84e	ssh https http
<input type="checkbox"/>	sg-04b34e5ffb64d6bdb	sarthak-ttn-sg	vpc-d38d68b7	launch-wizard-71 created 2020-02-20T19:45:04.355+05:30
<input type="checkbox"/>	sg-0ff39f601b256bf73	SG1	vpc-d38d68b7	SSH
<input type="checkbox"/>	sg-0674dc0db6c04c368	sgpooja	vpc-d38d68b7	ssh, https, http
<input type="checkbox"/>	sg-0f4171742dce39a3f	shivansh-sg	vpc-d38d68b7	launch-wizard-36 created 2020-02-20T11:52:44.817+05:30
<input type="checkbox"/>	sg-04a00497500bed727	Srima	vpc-0dc75714b5d93c571	tomcat

Inbound rules for sg-051d671c160aea760 Selected security groups: sg-051d671c160aea760.

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ
HTTP	TCP	80	0.0.0.0/0
SSH	TCP	22	0.0.0.0/0

STEP 3: Launch configuration is being created

Launch configuration creation status

✓ Successfully created launch configuration: diksha(scalinggroup1)LC
[View creation log](#)

STEP 4: Select your VPC and all the subnets

1. Configure Auto Scaling group details
2. Configure scaling policies
3. Configure Notifications
4. Configure Tags
5. Review

Create Auto Scaling Group

Group name ⓘ

Launch Configuration ⓘ

Group size ⓘ Start with instances

Network ⓘ [Create new VPC](#)

Subnet ⓘ

×
 ×
 ×

[Create new subnet](#)

STEP 5: Your Auto scaling group has been created

Auto Scaling group creation status



Successfully created Auto Scaling group

[View creation log](#)

- **launch template**

STEP 1: EC2 > Launch Template



New EC2 Experience
[Learn more](#)

Create launch template

Actions ▾

STEP 2:

[Launch Templates](#) > Create launch template

Create launch template



The current Launch Templates console is being replaced by a new Launch Templates console.

We are replacing this console with a new Launch Templates console, which we will continue to improve based on your feedback. After the launch of the new console, we will support bug fixes and patches in this console, but we won't add any new features. Any features they might be part of the source template, such as Auto Scaling guidance, Outposts, and Dedicated Hosts enhancements.

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. You can also create a new version of an existing template. When you create a new template you are creating a template and the first version of that template.

What would you like to do?

- ☒ Create a new template
- ☐ Create a new template version

Launch template name*

diksha(scalinggroup2)LT

[Show Tags](#)

Template version description

e.g. A prod webserver for MyApp (Max 255 chars)

STEP 3: Provide AMI

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

Source template

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.

AMI ID

Instance type

Key pair name

Search for AMI ×

AMI catalog

AMI

Cancel

Select AMI

STEP 4: Select your security 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.

AMI ID [Search for AMI](#) ⓘ

Instance type ⓘ

Key pair name ↻ ⓘ

Network type ☒ VPC ⓘ
☐ Classic

Security Groups ↻ ⓘ

STEP 5: Your template has been created

Create launch template



Success

Your launch template `diksha(scalinggroup2)LT` (lt-0fc2d1c9b69ff3cd7 Version 1) has been successfully created!

Next steps:

Launch an instance from this template

With On-Demand instances, you pay for compute capacity by the hour with no long-term commitments or upfront payments. Launch an On-Demand instance from your launch template.

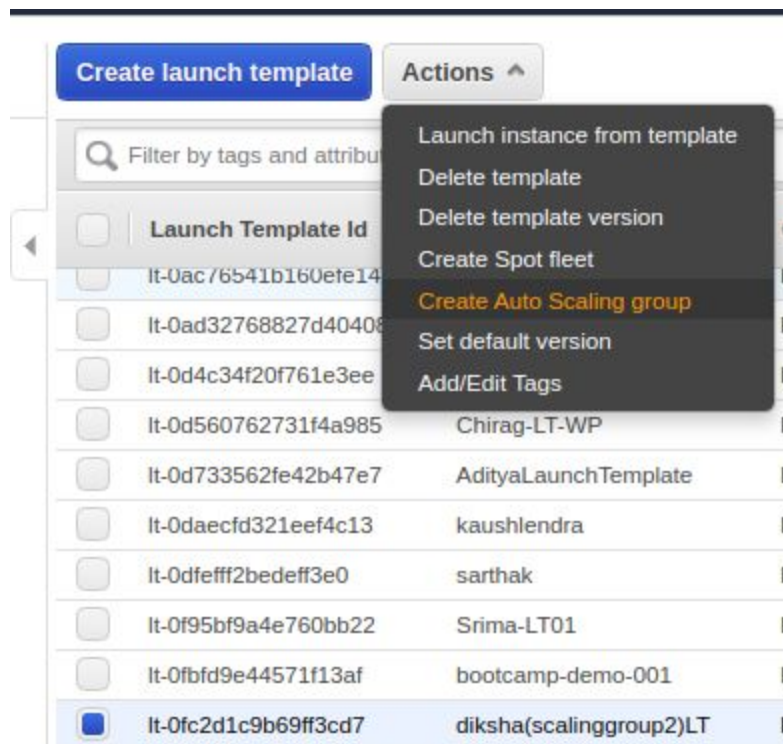
[Launch instance from this template](#)

Create an Auto Scaling group from your template

Amazon EC2 Auto Scaling helps you maintain application availability and allows you to scale your Amazon EC2 capacity up or down automatically according to conditions you define. You can use Auto Scaling to help ensure that you are running your desired number of Amazon EC2 instances during demand spikes to maintain performance and decrease capacity during lulls to reduce costs.

[Create Auto Scaling group](#)

STEP 6: Now create auto scaling group from the template created



1. Configure Auto Scaling group details
2. Configure scaling policies
3. Configure Notifications
4. Configure Tags
5. Review

Create Auto Scaling Group

Group name ⓘ

group2(diksha)

Launch Template ⓘ

lt-0fc2d1c9b69ff3cd7

Launch Template Version ⓘ

Latest

Create new launch template

Launch Template Description ⓘ

version1

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 launched at the lowest price available.

Group size ⓘ

Start with 1 instances

Network ⓘ

vpc-00470a42fc196d84e (10.0.0.0/16) | sarthak


Create new VPC

Subnet ⓘ

subnet-008dcd90bf26a9055(10.0.3.0/24) | sarthak-load-balancer-3 | us-east-1e x
subnet-01d770a77bb69a1f8(10.0.1.0/24) | sarthak-load-balancer-1 | us-east-1b x

STEP 7: Your auto scaling group has been created

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

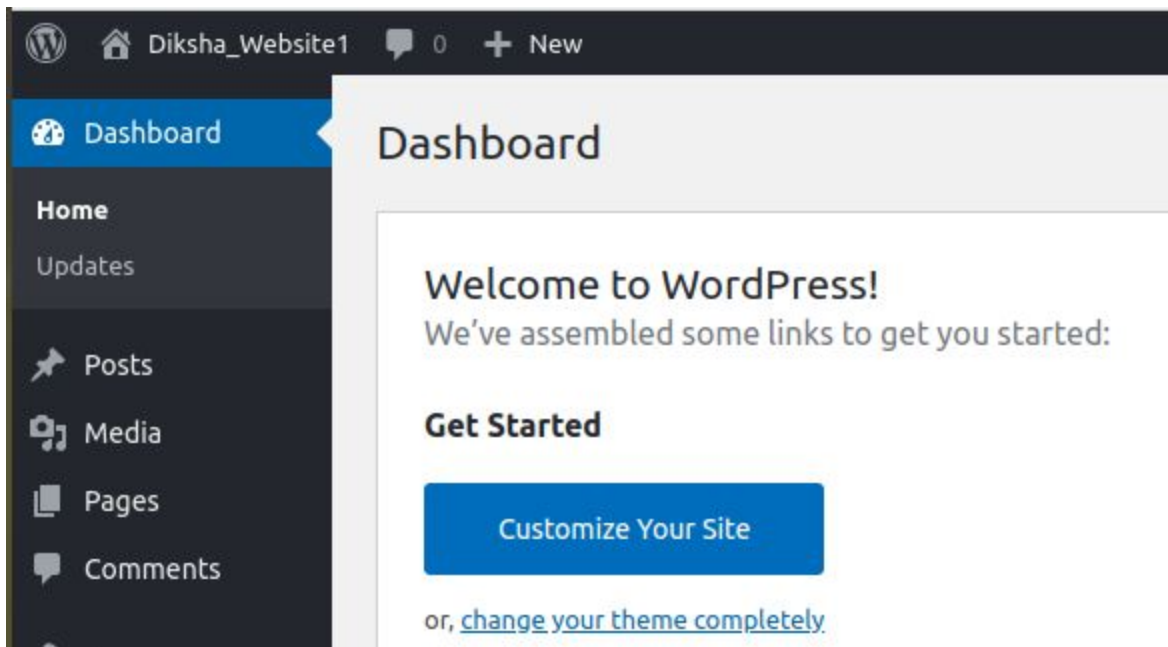
**6.Setup autoscaling Wordpress application with the Application load balancer.
Auto-scaling should be triggered based on CPU usage of EC2 instances.**

ANS:

Prerequisite (VPC >Subnet >IGW >Associate your subnet > Make route Table for your VPC and make entry for your IGW)

STEP 1: Launch an instance with wordpress installed “diksha_wp”

Launch Instance ▾ Connect Actions ▾						
Filter by tags and attributes or search by keyword						
<input type="checkbox"/>	Name	Purpose	Instance ID	Instance Type	Availability Zone	Instance State
<input checked="" type="checkbox"/>	diksha_wp	launching wordpress	i-0c2be1129e02ac0fe	t2.micro	us-east-1a	running
<input type="checkbox"/>			i-015828677e4f4ff85	t2.micro	us-east-1a	running



STEP 2: Create launch Template with the AMI of above instance.

Launch Actions ▾								
Owned by me Filter by tags and attributes or search by keyword								
<input type="checkbox"/>	Name	AMI Name	AMI ID	Source	Owner	Visibility	Status	Creation Date
<input checked="" type="checkbox"/>	diksha_wp_AMI		ami-055a93836aa71dbbf	200332499555/...	200332499555	Private	available	March 4, 2020 at 5:05:39 P...
<input type="checkbox"/>	diksha_wpAMI		ami-0b1420ebc67f60d57	200332499555/...	200332499555	Private	available	March 4, 2020 at 3:54:46 P...

Image: ami-055a93836aa71dbbf

Details		Permissions		Tags	
AMI ID	ami-055a93836aa71dbbf	AMI Name	diksha_wp_AMI		
Owner	200332499555	Source	200332499555/diksha_wp_AMI		
Status	available	State Reason	-		
Creation date	March 4, 2020 at 5:05:39 PM UTC+5:30	Platform	Other Linux		
Architecture	x86_64	Image Type	machine		
Virtualization type	hvm	Description	-		
Root Device Name	/dev/sda1	Root Device Type	ebs		
RAM disk ID	-	Kernel ID	-		
Product Codes	-	Block Devices	/dev/sda1=snap-039205d230d4c3d0b:8.true:gp2		

STEP 3: Create a target group

Create target group

Actions

Filter by tags and attributes or search by keyword

	Name	Port	Protocol	Target type	Load Balancer	VPC ID	Monitoring
<input checked="" type="checkbox"/>	dikshaWpTG	80	HTTP	instance		vpc-09921cb2d9e87f2eb	

Target group: dikshaWpTG

Description

Targets

Health checks

Monitoring

Tags

Basic Configuration

Name

dikshaWpTG

ARN

arn:aws:elasticloadbalancing:us-east-1:200332499555:targetgroup/dikshaWpTG/2f405135ab8b5d99

Protocol

HTTP

Port

80

Target type

instance

VPC

vpc-09921cb2d9e87f2eb

Load balancer

Register and deregister targets

Registered targets

To deregister instances, select one or more registered instances and then click Remove.

Remove

<input type="checkbox"/>	Instance	Name	Port	State	Security groups	Zone
No instances available.						

Instances

To register additional instances, select one or more running instances, specify a port, and then click Add. The default port is the port specified for the target group. If the instance is already registered on the specified port, you must specify a different port.

Add to registered

 on port

Search Instances

<input type="checkbox"/>	Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR
<input checked="" type="checkbox"/>	i-0c2be1129e02ac...	diksha_wp	running	launch-wizard-3	us-east-1a	subnet-066f3806db4573748	10.0.1.0/24
<input type="checkbox"/>	i-015828677e4f4f85		running	launch-wizard-3	us-east-1a	subnet-066f3806db4573748	10.0.1.0/24

STEP 4: Create Auto Scaling Group

Create Auto Scaling Group

Launch Template Description ⓘ

-

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 instances

Network ⓘ

vpc-00470a42fc196d84e (10.0.0.0/16) | sarthak



Create new VPC

Subnet ⓘ

subnet-0c3f9366d2e1133a8(10.0.11.0/24) | diksha_lb_wp | us-east-1f x

subnet-008dcd90bf26a9055(10.0.3.0/24) | sarthak-load-balancer-3 | us-east-1e x

Create new subnet

Advanced Details

Launch Template ⓘ

wordpressLT

Launch Template Version ⓘ

1 (Default)

Launch Template Description ⓘ

-

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.

Desired Capacity ⓘ

1

Min ⓘ

1

Max ⓘ

1

Availability Zone(s) ⓘ

us-east-1a x us-east-1b x

Subnet(s) ⓘ

subnet-066f3806db4573748(10.0.1.0/24) | publicsubnet1(diksha) | us-east-1a x

subnet-079390ae765fdcf5e(10.0.12.0/24) | publicsubnet2(diksha) | us-east-1b x

Edit details - ASG_wp_diksha

Desired Capacity

1

Min

1

Max

1

Availability Zone(s)

us-east-1a x us-east-1b x

Subnet(s)

subnet-066f3806db4573748(10.0.1.0/24) | publicsubnet1(diksha) | us-east-1a x

subnet-079390ae765fd5e(10.0.12.0/24) | publicsubnet2(diksha) | us-east-1b x

Classic Load Balancers

Target Groups

dikshaWpTG x

Health Check Type

EC2

Health Check Grace Period

300

Instance Protection

Termination Policies

Default x

Suspended Processes

Cancel

Save

STEP 5: Create an Application Load balancer

Select load balancer type

Elastic Load Balancing supports three types of load balancers: Application Load Balancer, Network Load Balancer, and Classic Load Balancer. We recommend Application Load Balancer for you.

Application Load Balancer

HTTP
HTTPS

Create

Choose an Application Load Balancer when you need a flexible feature set for your web applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.

[Learn more >](#)

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 with a single listener on port 80.

Name	<input type="text" value="dikshaLBwordpress"/>
Scheme	<input checked="" type="radio"/> internet-facing <input type="radio"/> internal
IP address type	<input type="text" value="ipv4"/>

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
<input type="text" value="HTTP"/>	<input type="text" value="80"/>
<input type="button" value="Add listener"/>	

Availability Zones

Specify the Availability Zones to enable for your load balancer. The load balancer routes traffic to the targets in these Availability Zones or Zones to increase the availability of your load balancer.

VPC ⓘ

vpc-09921cb2d9e87f2eb (10.0.0.0/16) | dikshaVPC ▼

Availability Zones

☒ us-east-1a

subnet-066f3806db4573748 (publicsubnet1(diksha)) ▼

IPv4 address ⓘ

Assigned by AWS

☒ us-east-1b

subnet-079390ae765fdc5e (publicsubnet2(diksha)) ▼

IPv4 address ⓘ

Assigned by AWS

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,

- Assign a security group:
- ☐ Create a new security group

☒ Select an existing security group

	Security Group ID	Name	Description
<input type="radio"/>	sg-02728bf96d082a4fa	default	default VPC security group
<input checked="" type="radio"/>	sg-0ecfdca3b5cd4885b	launch-wizard-3	launch-wizard-3 created 2020-03-04T15:48:13.105+05:30

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 balancer.

Target group

Target group ⓘ Existing target group ▼

Name ⓘ dikshaWpTG ▼

Target type

- ☒ Instance
- ☐ IP
- ☐ Lambda function

Protocol ⓘ HTTP ▼

Port ⓘ 80

Health checks

Protocol ⓘ HTTP ▼

Path ⓘ /

► Advanced health check settings

Step 5: Register Targets

Register targets with your target group. If you register a target in an enabled Availability Zone, the load balancer starts routing requests to the target.

Registered targets

The following targets are registered with the target group that you selected. You can only modify this list after you create the load balancer.

Instance	Port
i-015828677e4f4ff85	80

Create Load BalancerActions

Filter by tags and attributes or search by keyword

<input checked="" type="checkbox"/>	Name	DNS name	State	VPC ID	Availability Zones	Type
<input checked="" type="checkbox"/>	dikshaLBwordpress	dikshaLBwordpress-175355...	provisioning	vpc-09921cb2d9e87f2eb	us-east-1a, us-east-1b	application

STEP 6: Copy public ip of ALB and check that wordpress page shows.

UNCATEGORIZED

Hello world!

By Diksha Tomar March 4, 2020 1 Comment

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

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

STEP 1: Register the instance with wordpress to a target group.

Create target group

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.

Target group name

TG2dikshawp

Target type

Instance

IP

Lambda function

Protocol

HTTP

Port

80

VPC

vpc-09921cb2d9e87f2eb (10.0.0.0/16) | diks

Health check settings

Protocol

HTTP

Path

/

Advanced health check settings

STEP 2: Make entry in the hosts file of your local against your load balancers ip(dig DNS of your LB)

```
"hosts" [readonly] 10L, 350C 4,27
```

STEP 3: Edit rules and make entry in Your Load balancers rule.

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

New rule was created successfully.

dikshaLBwordpress | HTTP:80 (3 rules)

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

Insert Rule

Rule Number	Condition (IF)	Action (THEN)
1	Host is www.xyz.com	Forward to dikshaWpTG: 1 (100%) Group-level stickiness: Off
2	Host is www.abc.com	Forward to TG2dikshawp: 1 (100%) Group-level stickiness: Off
last	Requests otherwise not routed	Forward to dikshaWpTG: 1 (100%) Group-level stickiness: Off

HTTP 80: default action
This rule cannot be moved or deleted

STEP 4: Hit www.xyz.com in your browser

Not secure | xyz.com

Apps YouTube Maps (3) TO THE NE... (3) Diksha To... Learning | Das... Linux Academy TIMESHEET - T...

Diksha_Website1 Just another WordPress site

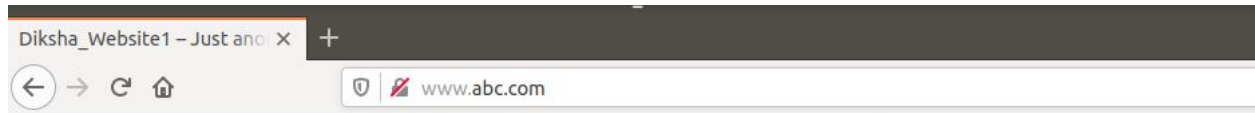
UNCATEGORIZED

Hello world!

By Diksha Tomar March 4, 2020 1 Comment

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

STEP 5: Hit www.abc.com in your browser



Diksha_Website1 — Just another WordPress site

Hello world!

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

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

STEP 1: Launch NLB

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.

Name	<input type="text" value="MyLLB_diksha"/>
Scheme	<input checked="" type="radio"/> internet-facing <input type="radio"/> internal

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
<input type="text" value="TCP"/>	<input type="text" value="80"/>
<input type="button" value="Add listener"/>	

1. Configure Load Balancer2. Configure Security Settings3. Configure Routing4. Register Targets5. Review

Step 4: Register Targets

Register targets with your target group. If you register a target in an enabled Availability Zone, the load balancer starts routing requests to the targets as soon as the registration process completes and the target passes the initial health checks.

Registered targets

To deregister instances, select one or more registered instances and then click Remove.

Remove

<input type="checkbox"/>	Instance	Name	Port	State	Security groups	Zone
<input type="checkbox"/>	i-0c2be1129e02ac0fe	diksha_wp	80	running	launch-wizard-3	us-east-1a

Instances

To register additional instances, select one or more running instances, specify a port, and then click Add. The default port is the port specified for the target group. If the instance is already registered on the specified port, you must specify a different port.

Add to registered

 on port

Search Instances

<input type="checkbox"/>	Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR
<input checked="" type="checkbox"/>	i-0c2be1129e02ac0fe	diksha_wp	running	launch-wizard-3	us-east-1a	subnet-066f3806db4573748	10.0.1.0/24

Load Balancer Creation Status

Successfully created load balancer

Load balancer [MyLLBdiksha](#) 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 [MyLLBdiksha](#).
- Consider using AWS Global Accelerator to further improve the availability and performance of your applications. [AWS Global Accelerator console](#)

← → ↺

Not secure | myllbdiksha-6855976e36c55e0c.elb.us-east-1.amazonaws.com

Apps

YouTube

Maps

(3) TO THE NE...

(3) Diksha To...

Learning | Das...

Linux Academy

TIMESHE

Diksha_Website1 — Just another WordPress site

Hello world!

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

Diksha Tomar March 4, 2020 Uncategorized 1 Comment

9. Take an instance out of the ASG.

STEP 1: Edit your ASG and edit desired and max as 2

Edit details - ASG_wp_diksha

Launch Instances Using ⓘ
☒ Launch Template
☐ Launch Configuration

Launch Template ⓘ
wordpressLT

Launch Template Version ⓘ
3

Launch Template Description ⓘ
-

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.

Desired Capacity ⓘ
2

Min ⓘ
1

Max ⓘ
2

Availability Zone(s) ⓘ
us-east-1a x us-east-1b x

Subnet(s) ⓘ
subnet-066f3806db4573748(10.0.1.0/24) | publicsubnet1(diksha) | us-east-1a x
subnet-079390ae765fd5e(10.0.12.0/24) | publicsubnet2(diksha) | us-east-1b x

Cancel Save

Auto Scaling Group: ASG_wp_diksha

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

Actions ▾

Filter: Any Health Status ▾ Any Lifecycle State ▾ <input type="text" value="Filter instances..."/>						
1 to 2 of 2 instances						
<input type="checkbox"/>	Instance ID	Lifecycle	Launch Configuration / Template	Availability Zone	Health Status	Protected from
<input type="checkbox"/>	i-046cd9507cedb5ecb	InService	wordpressLT	us-east-1b	Healthy	
<input checked="" type="checkbox"/>	i-0a16b887712795fd7	Pending	wordpressLT	us-east-1a	Healthy	

STEP 2: Select the instance you want to detach

Auto Scaling Group: ASG_wp_diksha

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

Actions ^

- Detach
- Set to Standby
- Set to InService
- Instance Protection ▶

Filter instances...

	Instance ID	Lifecycle	Launch Configuration / Template	Availability Zone	Health Status
<input checked="" type="checkbox"/>	i-046cd9507cedb5ecb	InService	wordpressLT	us-east-1b	Healthy
<input type="checkbox"/>	i-0a16b8877f2795fd7	InService	wordpressLT	us-east-1a	Healthy

STEP 3: Your instance has been detached

Auto Scaling Group: ASG_wp_diksha

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

Actions v

Filter: Any Health Status v Any Lifecycle State v Filter instances...

	Instance ID	Lifecycle	Launch Configuration / Template	Availability Zone
<input checked="" type="checkbox"/>	i-046cd9507cedb5ecb	Detaching	wordpressLT	us-east-1b
<input type="checkbox"/>	i-0a16b8877f2795fd7	InService	wordpressLT	us-east-1a

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

STEP 1: Click on you ASG and then edit it. In instance protection select “Protect from scale in ”

Edit details - ASG_wp_diksha

Max

1

Availability Zone(s)

us-east-1a x us-east-1b x

Subnet(s)

subnet-066f3806db4573748(10.0.1.0/24) | publicsubnet1(diksha) | us-east-1a x
subnet-079390ae765fd5e(10.0.12.0/24) | publicsubnet2(diksha) | us-east-1b x

Classic Load Balancers

Target Groups

dikshaWpTG x

Health Check Type

EC2

Health Check Grace Period

30

Instance Protection

Protect From Scale In x |

Termination Policies

OldestLaunchConfiguration x Default x

Suspended Processes

Max Instance Lifetime

Placement Groups

Default Cooldown

30

Cancel

Save

11.Put Schedules in ASG to:

- Remove all instances of the ASG at 8 PM

Create Auto Scaling group Actions

Filter: Filter Auto Scaling groups

Name ASG_wp

Auto Scaling G

Details AS

Create Sch

Filter: Filter

Create Scheduled Action

Name

Auto Scaling Group ASG_wp_diksha

Provide at least one of Min, Max and Desired Capacity

Min

Max

Desired Capacity

Recurrence
(Cron) 0 20 * * *

Start Time UTC Specify the start time in UTC
The first time this scheduled action will run

End Time [Set End Time](#)

[Cancel](#) [Create](#)

- Launch a minimum of 2 instances at 10 AM

Create Auto Scaling group Actions

Filter: Filter Auto Scaling groups

Name ASG_wp

Auto Scaling G

Details AS

Create Sch

Filter: Filter

Create Scheduled Action

Name

Auto Scaling Group ASG_wp_diksha

Provide at least one of Min, Max and Desired Capacity

Min

Max

Desired Capacity

Recurrence

Start Time UTC Specify the start time in UTC
The first time this scheduled action will run

[Cancel](#) [Create](#)