

Auto-scaling and Load Balancer

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

Ans 1:-

Network Load Balancer --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.

Application Load Balancer -- 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.

Elastic Load Balancer basics -An Elastic Load Balancer (ELB) is one of the key architecture components for many applications inside the AWS cloud. In addition to autoscaling, it enables and simplifies one of the most important tasks of our application's architecture: scaling up and down with high availability. Elastic Load Balancing automatically distributes incoming application

traffic across multiple applications, microservices, and containers hosted on Amazon EC2 instances.

Ques 2:-Differences between step scaling and target scaling.

Ans 2:-

Step scaling --

Amazon Web Services has introduced Step Autoscaling Policies for the autoscaling of EC2 instances. ... The Amazon Web Services has modified the Autoscaling Policies and new features have been introduced. It now lets you scale in and scale out your instances in percentage terms of the running instances

Target scaling --

target tracking, you select a load metric for your application, such as "Average CPU Utilization" or the new "Request Count Per Target" metric from Application Load Balancer, set the target value, and Auto Scaling adjusts the number of EC2 instances in your Auto Scaling group as needed to maintain that target. It acts like a home thermostat, automatically adjusting the system to keep the environment at your desired temperature. For example, you can configure target tracking to keep CPU utilization for your fleet of web servers at 50%. From there, Auto Scaling launches or terminates EC2 instances as required to keep the average CPU utilization at 50%.

Ques 3:-Differences between Launch configuration and launch template.

Ans 3:-Launch configuration--

A launch configuration is similar to a launch template, in that it specifies the type of EC2 instance that Amazon EC2 Auto Scaling creates for you. You create the launch configuration by including information such as the

ID of the Amazon Machine Image (AMI) to use, the instance type, the key pair, and security groups.

Launch template--

Launch Templates is a new capability that enables a new way to templatize your launch requests. Launch Templates streamline and simplify the launch process for Auto Scaling, Spot Fleet, Spot, and On-Demand instances.

Ques 4:-Differences between EC2 healthcheck and load balancer health check

Ans 4:-

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.

Ques 5:-Create 2 auto-scaling groups with

- launch configuration and
- launch template

Ans 5:- Creating the launch template

EC2 > Launch templates > Create launch template

 **Success**
Successfully created t34ak-template (lt-00815ef2dba1b0280)

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](#)

Create Spot Fleet

A Spot Instance is an unused EC2 instance that is available for less than the On-Demand price. Because Spot Instances enable you to request unused EC2 instances at steep discounts, you can lower your Amazon EC2 costs significantly. The hourly price for a Spot Instance (of each instance type in each Availability Zone) is set by Amazon EC2, and adjusted gradually based on the long-term supply of and demand for Spot Instances. Spot instances are well-suited for data-analysis, batch jobs, background processing, and optional tasks.

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

EC2 > Launch templates

Launch templates (8)


Filter by tags or properties or search by keyword

Launch template ID	Launch template name	Default version	Latest version	Create time
lt-00815ef2dba1b0280	t34ak-template	1	1	2020-02-26T08:1...

Creating the Auto Scaling Group using Launch Template

aws Services Resource Groups

Auto Scaling group creation status

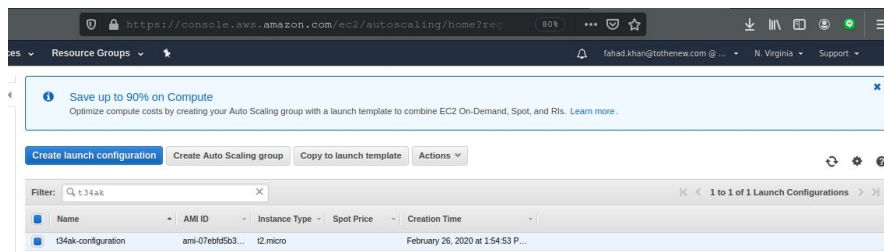
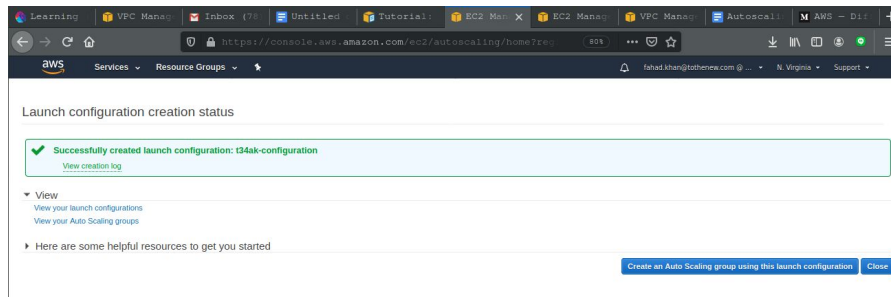
 **Successfully created Auto Scaling group**
[View creation log](#)

View

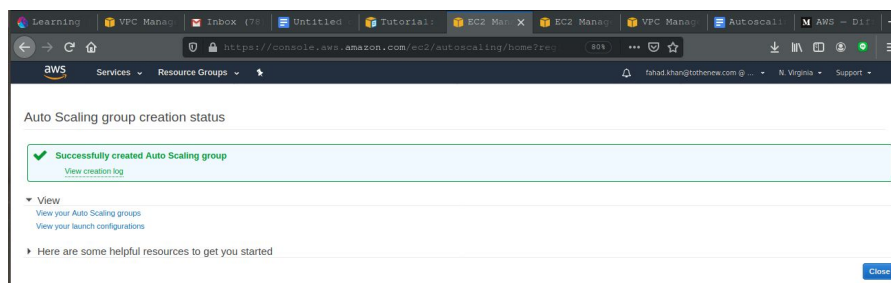
- [View your Auto Scaling groups](#)
- [View your launch configurations](#)

Here are some helpful resources to get you started

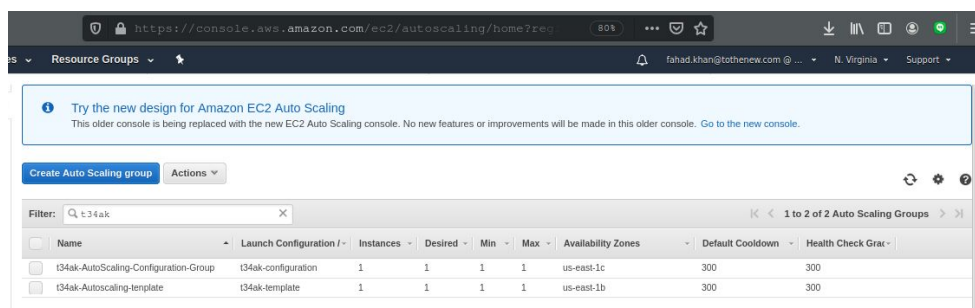
Creating Launch Configuration



Creating Auto Scaling Group using Launch Configuration



Two Auto Scaling Groups with launch template and configuration



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

Ans 6:-

```
fahad@fahad ~/Downloads (master*)  
➤ ssh -i "T34aK.pem" ubuntu@3.210.197.169  
The authenticity of host '3.210.197.169 (3.210.197.169)' can't be established.  
ECDSA key fingerprint is SHA256:xvYXsI/M6yACbtQU2g6bbicS9m1b/nU34P6ySwT39x8.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added '3.210.197.169' (ECDSA) to the list of known hosts.  
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-1057-aws x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
System information disabled due to load higher than 1.0  
  
0 packages can be updated.  
0 updates are security updates.  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.
```

First we have to install and configure the nginx and wordpress


```
ubuntu@ip-10-0-9-145:~$ sudo apt-get install nginx
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  fontconfig-config fonts-dejavu-core libfontconfig1 libgd3 libjbig0
  libjpeg-turbo8 libjpeg8 libnginx-mod-http-geoip
  libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter
  libnginx-mod-mail libnginx-mod-stream libtiff5 libwebp6 libxpm4 nginx-common
  nginx-core
Suggested packages:
  libgd-tools fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  fontconfig-config fonts-dejavu-core libfontconfig1 libgd3 libjbig0
  libjpeg-turbo8 libjpeg8 libnginx-mod-http-geoip
  libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter
  libnginx-mod-mail libnginx-mod-stream libtiff5 libwebp6 libxpm4 nginx
  nginx-common nginx-core
0 upgraded, 18 newly installed, 0 to remove and 3 not upgraded.
Need to get 2461 kB of archives.
After this operation, 8210 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 l
ibjpeg-turbo8 amd64 1.5.2-0ubuntu5.18.04.3 [110 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/main amd64 fonts-dej
avu-core all 2.37-1 [1041 kB]
```

```
ubuntu@ip-10-0-9-145:~$ sudo apt-get install mariadb-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  galera-3 libaio1 libcgi-fast-perl libcgi-pm-perl libconfig-inifiles-perl
  libdbd-mysql-perl libdbi-perl libencode-locale-perl libfcgi-perl
  libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl
  libhttp-date-perl libhttp-message-perl libio-html-perl libjemalloc1
  liblwp-mediatypes-perl libmysqlclient20 libterm-readkey-perl
  libtimedate-perl liburi-perl mariadb-client-10.1 mariadb-client-core-10.1
  mariadb-common mariadb-server-10.1 mariadb-server-core-10.1 mysql-common
  socat
Suggested packages:
  libclone-perl libmldbm-perl libnet-daemon-perl libsql-statement-perl
  libdata-dump-perl libipc-sharedcache-perl libwww-perl mailx mariadb-test
  tinyc
The following NEW packages will be installed:
  galera-3 libaio1 libcgi-fast-perl libcgi-pm-perl libconfig-inifiles-perl
  libdbd-mysql-perl libdbi-perl libencode-locale-perl libfcgi-perl
  libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl
  libhttp-date-perl libhttp-message-perl libio-html-perl libjemalloc1
  liblwp-mediatypes-perl libmysqlclient20 libterm-readkey-perl
  libtimedate-perl liburi-perl mariadb-client-10.1 mariadb-client-core-10.1
  mariadb-common mariadb-server mariadb-server-10.1 mariadb-server-core-10.1
```



```
ubuntu@ip-10-0-9-145:~$ sudo bash
root@ip-10-0-9-145:~# systemctl enable mariadb.service
root@ip-10-0-9-145:~# mysql_secure_installation
```

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

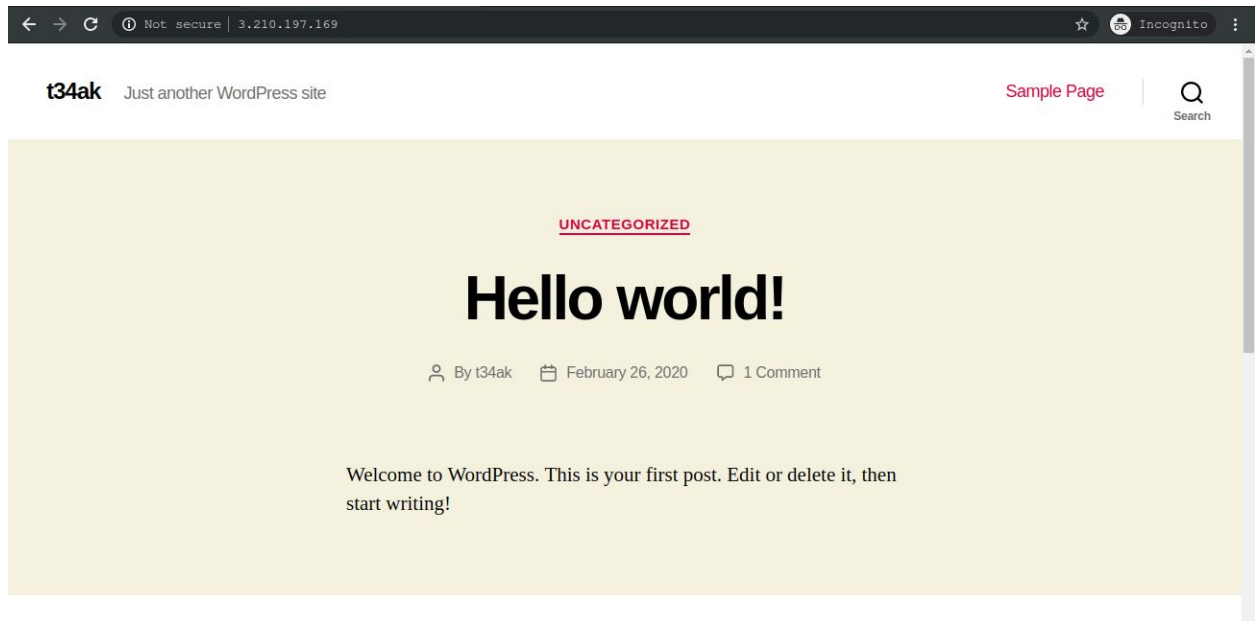
Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

Set root password? [Y/n] y

```
root@ip-10-0-9-145:~# apt-get install php7.2 php7.2-cli php7.2-fpm php7.2-mysql
php7.2-json php7.2-opcache php7.2-mbstring php7.2-xml php7.2-gd php7.2-curl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libsodium23 php-common php7.2-common php7.2-readline
Suggested packages:
  php-pear
The following NEW packages will be installed:
  libsodium23 php-common php7.2 php7.2-cli php7.2-common php7.2-curl
  php7.2-fpm php7.2-gd php7.2-json php7.2-mbstring php7.2-mysql php7.2-opcache
  php7.2-readline php7.2-xml
0 upgraded, 14 newly installed, 0 to remove and 3 not upgraded.
Need to get 4833 kB of archives.
After this operation, 20.6 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/main amd64 libsodium
23 amd64 1.0.16-2 [143 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/main amd64 php-commo
n all 1:60ubuntu1 [12.1 kB]
```

The wordpress start working now we have to put it to the load
balancer



To set up the load balancer we first have to create the AMI of running Instance

A screenshot of the AWS Management Console interface. At the top, there are 'Launch' and 'Actions' buttons. Below them is a search bar with 'search: t34ak' and an 'Add filter' button. A table lists the AMIs, with one entry selected: 'wordpress-n...' with AMI Name 't34ak-wordpress', AMI ID 'ami-083e0773969d2e811', Source '187632318301/t...', Owner '187632318301', Visibility 'Private', Status 'available', Creation Date 'February 26, 2020 at 5:24:2...', Platform 'Other Linux', Root Device 'ebs', and Virtualization type 'hvm'. Below the table, the 'Details' tab is active, showing the following information:

AMI ID	ami-083e0773969d2e811	AMI Name	t34ak-wordpress
Owner	187632318301	Source	187632318301/t34ak-wordpress
Status	available	State Reason	-
Creation date	February 26, 2020 at 5:24:20 PM UTC+5:30	Platform	Other Linux
Architecture	x86_64	Image Type	machine
Virtualization type	hvm	Description	it is a wordpress application
Root Device Name	/dev/sda1	Root Device Type	ebs
RAM disk ID	-	Kernel ID	-
Product Codes	-	Block Devices	/dev/sda1=snap-057876322cd5c915e:8:true:gp2

An 'Edit' button is located at the bottom right of the details section.

Then we have to create the Launch Configuration with the help of the AMI

Save up to 90% on Compute
Optimize compute costs by creating your Auto Scaling group with a launch template to combine EC2 On-Demand, Spot, and RIs. [Learn more.](#)

Create launch configurationCreate Auto Scaling groupCopy to launch templateActions

Filter: t34ak1 to 1 of 1 Launch Configurations

Name	AMI ID	Instance Type	Spot Price	Creation Time
t34ak-lb	ami-083e0773...	t2.micro		February 27, 2020 at 2:29:34 P...

Launch Configuration: t34ak-lb

Details

AMI IDami-083e0773969d2e811

IAM Instance Profile

Key NameT34ak

EBS Optimizedfalse

Spot Price

RAM Disk ID

User data-

Instance Type

t2.micro

Kernel ID

Monitoringfalse

Security Groups

sg-0940bc8edcde0289a

Creation Time

Thu Feb 27 14:29:34 GMT+530 2020

Block Devices

/dev/sda1

IP Address Type

Only assign a public IP address to instances launched in the default VPC and subnet. (default)

Copy launch configuration

Then we have to create the Auto Scaling Group with help of Launch configuration

Try the new design for Amazon EC2 Auto Scaling
This older console is being replaced with the new EC2 Auto Scaling console. No new features or improvements will be made in this older console. [Go to the new console.](#)

Create Auto Scaling groupActions

Filter: t34ak1 to 1 of 1 Auto Scaling Groups

Name	Launch Configuration /	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grace
t34ak-ASG	t34ak-lb	4	4	4	4	us-east-1a, us-east-1b, us-e...	300	300

DetailsActivity HistoryScaling PoliciesInstancesMonitoringNotificationsTagsScheduled ActionsLifecycle Hooks

Launch Configuration ⓘt34ak-lb

Desired Capacity ⓘ4

Min ⓘ4

Max ⓘ4

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

Subnet(s) ⓘsubnet-097975492bf68a9df, subnet-01f98a351ed87a94b, subnet-0c169fb103120655c

Classic Load Balancers ⓘ

Target Groups ⓘ

Health Check Type ⓘEC2

Health Check Grace Period ⓘ300

Instance Protection ⓘ

Termination Policies ⓘDefault

Edit

Create Auto Scaling group Actions

Filter: t34ak 1 to 1 of 1 Auto Scaling Groups

Name	Launch Configuration /	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grace
t34ak-ASG	t34ak-lb	4	4	4	4	us-east-1a, us-east-1b, us-e...	300	300

Auto Scaling Group: t34ak-ASG

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

Actions

Filter: Any Health Status Any Lifecycle State Filter instances... 1 to 4 of 4 Instances

Instance ID	Lifecycle	Launch Configuration / Template	Availability Zone	Health Status	Protected from
i-00bde9832bf10c1a6	InService	t34ak-lb	us-east-1a	Healthy	
i-03013d1569b#9791	InService	t34ak-lb	us-east-1b	Healthy	
i-0a18755ca943c08ef	InService	t34ak-lb	us-east-1a	Healthy	
i-0c84f7770b546ef5e	InService	t34ak-lb	us-east-1c	Healthy	

By creating the scaling property we can auto scale the instances.

Here i use step scaling

Create Scaling policy Cancel Create

Name: t34ak-cpu-utilising

Execute policy when: awsec2-t34ak-ASG-CPU-Utilization Create new alarm
breaches the alarm threshold: CPUUtilization >= 70 for 300 seconds
for the metric dimensions AutoScalingGroupName = t34ak-ASG

Take the action: Set to 1 capacity units when 20 <= CPUUtilization < +infinity

Add step

Instances need: 2 seconds to warm up after each step

Create a target tracking scaling policy
Create a simple scaling policy

Now we have to create the Target Group and have to add the instances of the Auto Scaling Group to The target Group

Create target group

Actions

search : t34ak

Add filter

1 to 1 of 1

Name	Port	Protocol	Target type	Load Balan	VPC ID	Monitoring
t34ak-TG	80	HTTP	instance	t34ak-LB	vpc-01d9bca1ea53fdce9	

Target group: t34ak-TG

Description

Targets

Health checks

Monitoring

Tags

The load balancer starts routing requests to a newly registered target as soon as the registration process completes and the target passes the initial health checks. If demand on your targets increases, you can register additional targets. If demand on your targets decreases, you can deregister targets.

Edit

Registered targets

Instance ID	Name	Port	Availability Zone	Status	Description
i-03013d1569b0f9791	t34ak-wp1	80	us-east-1b	healthy	This target is currently passing target group's health checks.
i-0c84f7770b546ef5e	t34ak-wp3	80	us-east-1c	healthy	This target is currently passing target group's health checks.
i-0a18755ca943c08ef	t34ak	80	us-east-1a	healthy	This target is currently passing target group's health checks.
i-00bde9832bf10c1a6	t34ak-wp	80	us-east-1a	healthy	This target is currently passing target group's health checks.

Availability Zones

Availability Zone	Target count	Healthy?
us-east-1b	1	Yes
us-east-1a	2	Yes
us-east-1c	1	Yes

Now we have to create the Load Balancer

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.

Name	<input type="text" value="t34ak-LB"/>
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"/>	

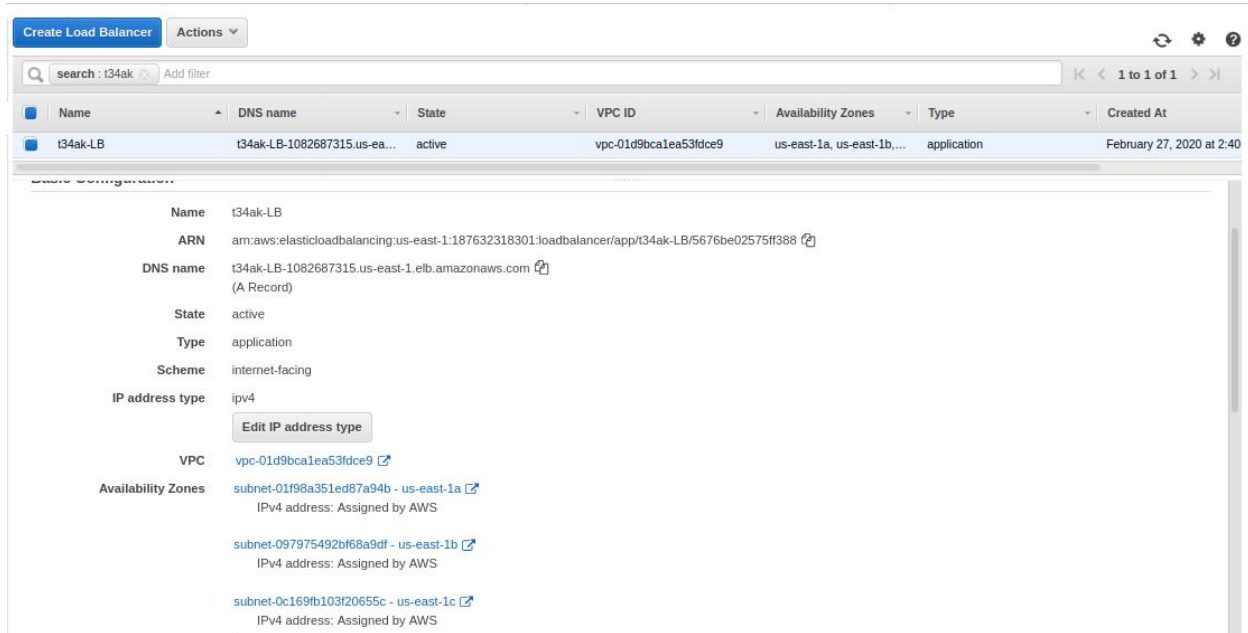
We have to specify the VPC and attach the public subnets to them

Availability Zones

Specify the Availability Zones to enable for your load balancer. The load balancer routes traffic to the targets in these Availability Zones only. You can specify only one subnet per Availability Zone. You must specify subnets from at least two Availability Zones to increase the availability of your load balancer.

VPC	<input type="text" value="vpc-01d9bca1ea53fdce9 (10.0.0.0/16) t34ak"/>
Availability Zones	<input checked="" type="checkbox"/> us-east-1a <input type="text" value="subnet-01f98a351ed87a94b (vaibhav1a)"/>
	IPv4 address Assigned by AWS
	<input checked="" type="checkbox"/> us-east-1b <input type="text" value="subnet-097975492bf68a9df (vaibhav1b)"/>
	IPv4 address Assigned by AWS
	<input checked="" type="checkbox"/> us-east-1c <input type="text" value="subnet-0c169fb103f20655c (vaibhav1c)"/>
	IPv4 address Assigned by AWS
	<input type="checkbox"/> us-east-1e <input type="text" value="subnet-01f9e104e99034de5 (t34ak pub2)"/>
	<input type="checkbox"/> us-east-1f <input type="text" value="subnet-0dba4ee75a08a389d (t34ak-private)"/>

Now the load Balancer is created with all the properties



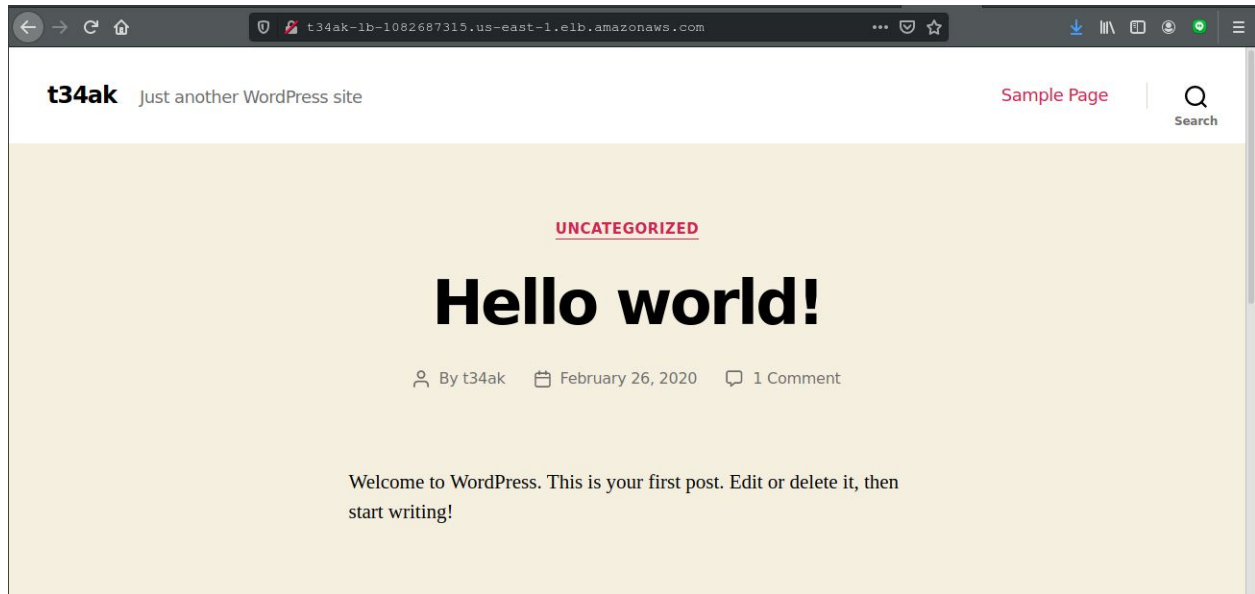
The screenshot displays the AWS Management Console interface for an Application Load Balancer. At the top, there is a 'Create Load Balancer' button and an 'Actions' dropdown menu. Below this is a search bar with the text 'search: t34ak' and an 'Add filter' button. A table lists the load balancer details:

Name	DNS name	State	VPC ID	Availability Zones	Type	Created At
t34ak-LB	t34ak-LB-1082687315.us-east-1.elb.amazonaws.com	active	vpc-01d9bca1ea53fdce9	us-east-1a, us-east-1b, us-east-1c	application	February 27, 2020 at 2:40

Below the table, the 'Details' section provides further information about the load balancer:

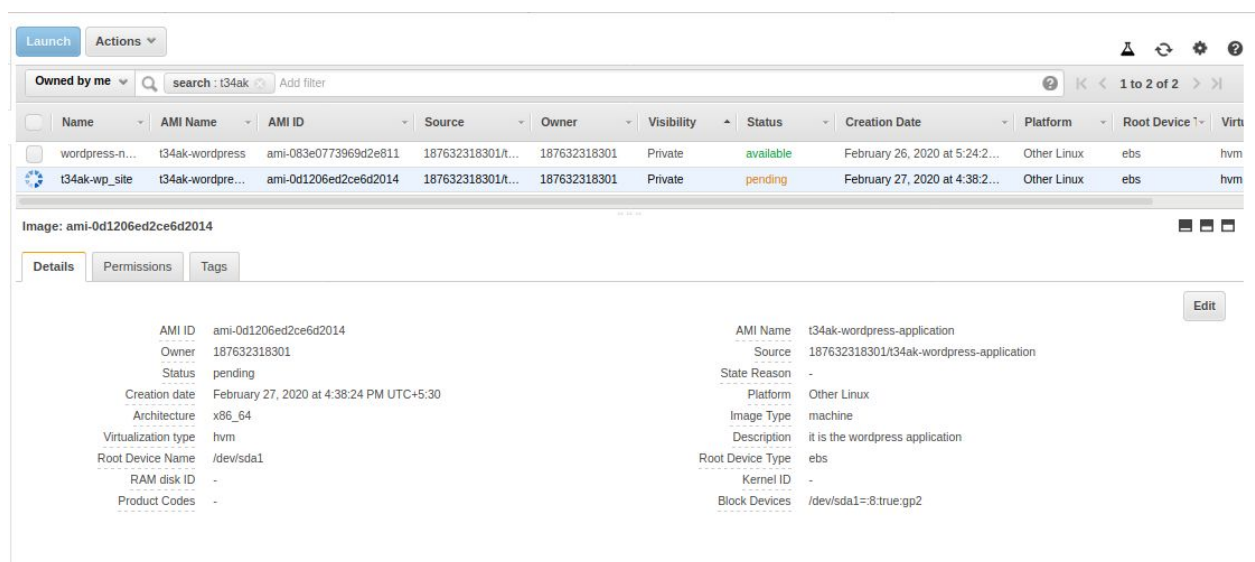
- Name:** t34ak-LB
- ARN:** [arn:aws:elasticloadbalancing:us-east-1:187632318301:loadbalancer/app/t34ak-LB/5676be02575ff388](#)
- DNS name:** [t34ak-LB-1082687315.us-east-1.elb.amazonaws.com](#) (A Record)
- State:** active
- Type:** application
- Scheme:** internet-facing
- IP address type:** ipv4 (with an 'Edit IP address type' button)
- VPC:** [vpc-01d9bca1ea53fdce9](#)
- Availability Zones:**
 - [subnet-01f98a351ed87a94b - us-east-1a](#) (IPv4 address: Assigned by AWS)
 - [subnet-097975492bf68a9df - us-east-1b](#) (IPv4 address: Assigned by AWS)
 - [subnet-0c169fb103f20655c - us-east-1c](#) (IPv4 address: Assigned by AWS)

Now we just have to copy the Dns Name and just have to paste it in the url and then the Load Balancer is loaded



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

Ans 7:-First we have to create the AMI of the new wordpress site




Then we have to create the instance of that AMI

1. Choose AMI2. Choose Instance Type3. Configure Instance4. Add Storage5. Add Tags6. Configure Security Group7. Review

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

▼ AMI Details



t3aak-wordpress-application - ami-0d1206ed2ce6d2014
it is the wordpress application
Root Device Type: ebs Virtualization type: hvm

Edit AMI

▼ Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Edit instance type

▼ Security Groups

Security group name

launch-wizard-184

Description

launch-wizard-184 created 2020-02-27T16:42:08.217+05:30

Type	Protocol	Port Range	Source	Description
This security group has no rules				

Edit instance details

Launch Instance

Connect

Actions

Add filter

1 to 11 of 11

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs
t34ak-wp	i-00bde9832bf10c1a6	t2.micro	us-east-1a	running	2/2 checks ...	None		107.23.150.195	-
t34ak-G3	i-0636adf48021c1f30	t2.micro	us-east-1b	running	2/2 checks ...	None		18.234.76.60	-
t34ak-G3	i-0e5ad75cd44c3810b	t2.micro	us-east-1b	running	2/2 checks ...	None		3.210.197.169	-

Instance: i-0636adf48021c1f30 (t34ak-G3)

Public IP: 18.234.76.60

Description

Status Checks

Monitoring

Tags

Instance ID

Instance state

Instance type

Finding

Private DNS

Private IPs

Secondary private IPs

VPC ID

Subnet ID

Network interfaces

Source/dest. check

T2/T3 Unlimited

EBS-optimized

Root device type

i-0636adf48021c1f30

running

t2.micro

You may not have permission to access AWS Compute Optimizer.

ip-10-0-101-4.ec2.internal

10.0.101.4

vpc-01d9bca1ea53f9ce9 (t34ak)

subnet-097975492bf68a9df (vaihav1b)

eth0

True

Disabled

False

efs

Public DNS (IPv4)

IPv4 Public IP

IPv6 IPs

Elastic IPs

Availability zone

Security groups

Scheduled events

AMI ID

Platform

IAM role

Key pair name

Owner

Launch time

Termination protection

-

18.234.76.60

-

-

us-east-1b

launch-wizard-184, view inbound rules, view outbound rules

No scheduled events

t34ak-wordpress-application (ami-0d1206ed2ce6d2014)

-

-

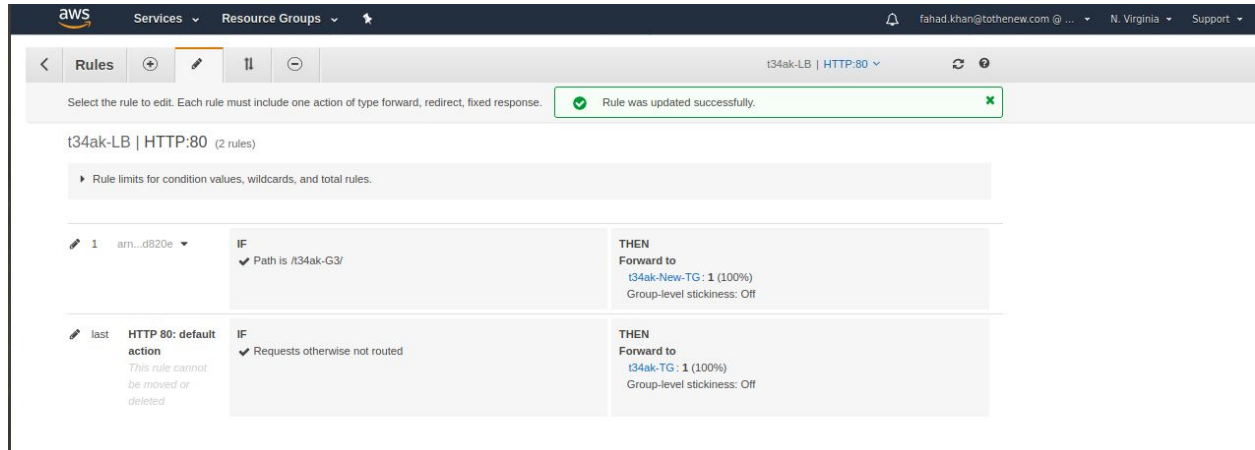
T34ak

187632318301

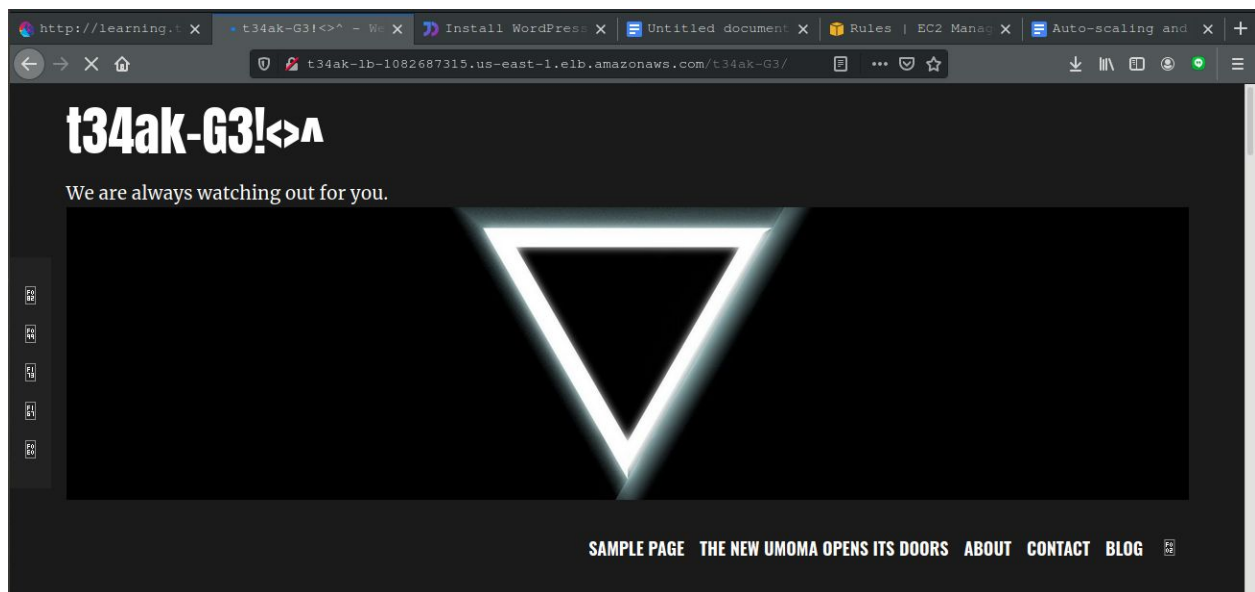
February 27, 2020 at 4:42:35 PM UTC+5:30 (less than one hour)

False

Now we have to add the role in the Load Balancer in Which we have to add the Other Target Group to the Roles of the Load Balancer



Now if we type the url with /t34ak-G3 it will lead us to other landing page.



Ans 8:-

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

Step 5: Review

Please review the load balancer details before continuing

▼ Load balancer

Name

t34ak-NLB

Scheme

internet-facing

Listeners

Port:80 - Protocol:TCP

VPC

vpc-01d9bca1ea53fde9 (t34ak)

Subnets

subnet-01f98a351ed87a94b (vaibhav1a), subnet-097975492b68a9df (vaibhav1b), subnet-0c169fb103f20655c (vaibhav1c)

Tags

Owner:Fahad Khan, Purpose:NLB

▼ Routing

Target group

New target group

Target group name

t34ak-NLB

Port

80

Target type

instance

Protocol

TCP

Health check protocol

HTTP

Path

/

Health check port

traffic port

Healthy threshold

3

Unhealthy threshold

3

Interval

30

Success codes

200-399

▼ Targets

Target name

target-01d9bca1ea53fde9 (t34ak)

Target type

instance

Port

80

Protocol

TCP

Health check protocol

HTTP

Path

/

Health check port

traffic port

Healthy threshold

3

Unhealthy threshold

3

Interval

30

Success codes

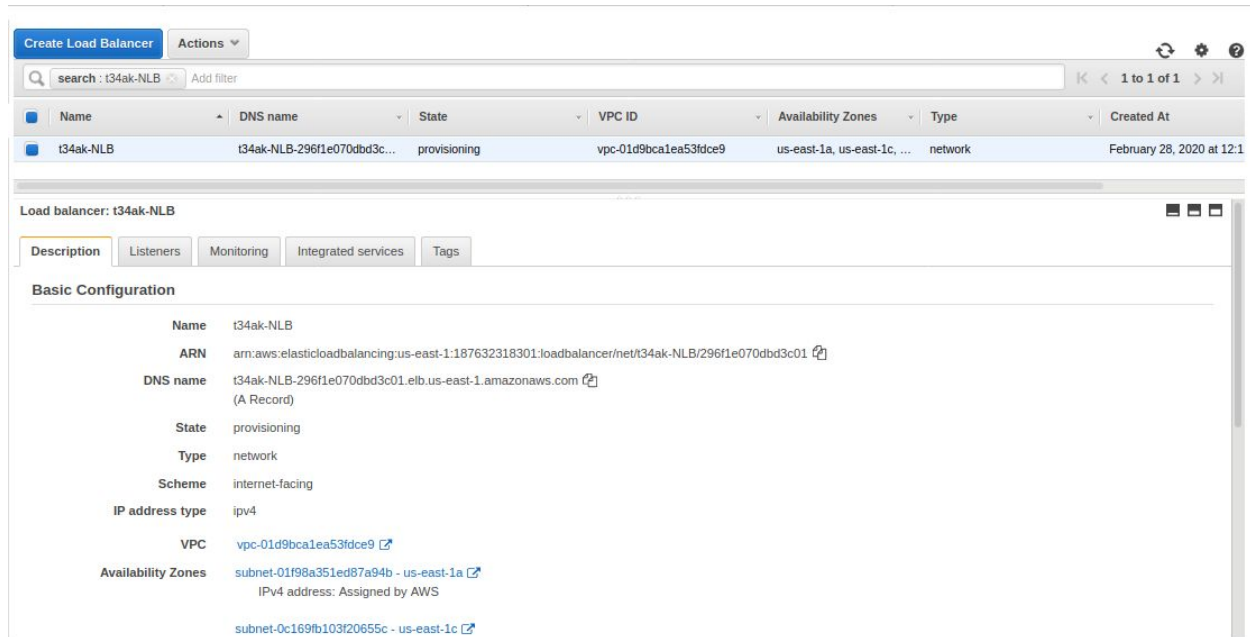
200-399

Cancel

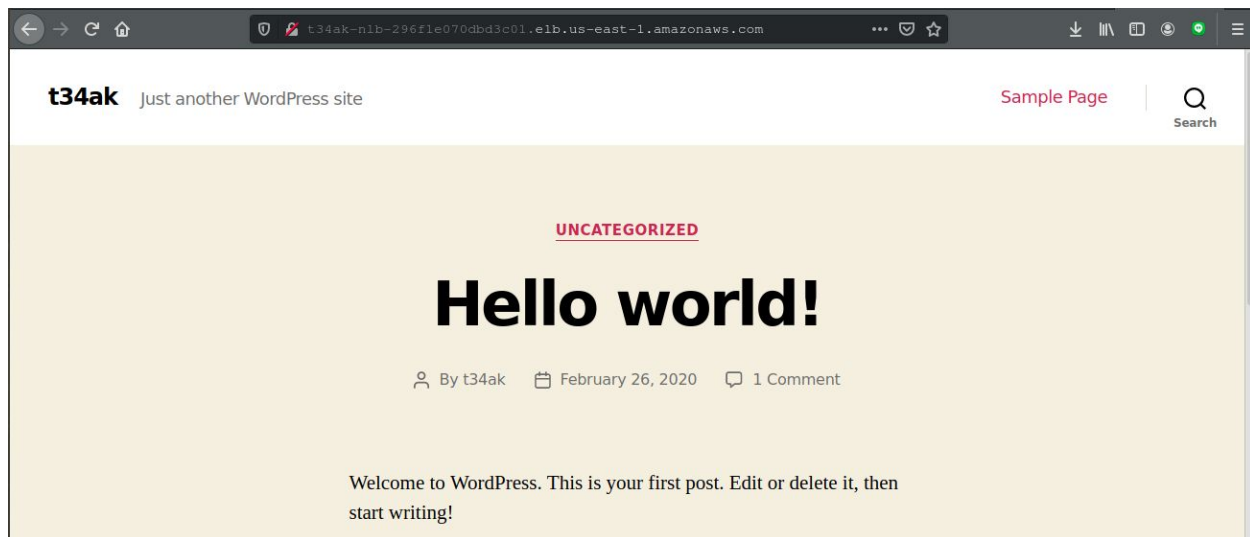
Previous

Create

Now we just have to copy the DNS url and paste it to the webbrowser



NLB starts working

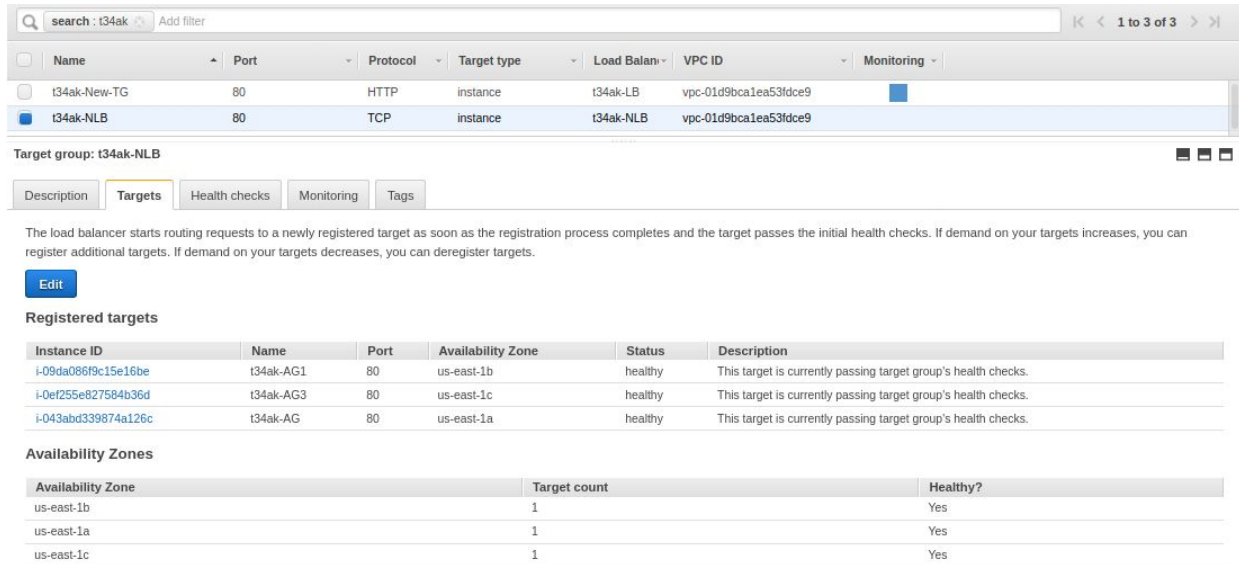


Ques 9. Take an instance out of the ASG.

Ans 9:- For taking the instance out of ASG

First we have to set min value to 1

Previously there were m3 instances running in the NLB and TG

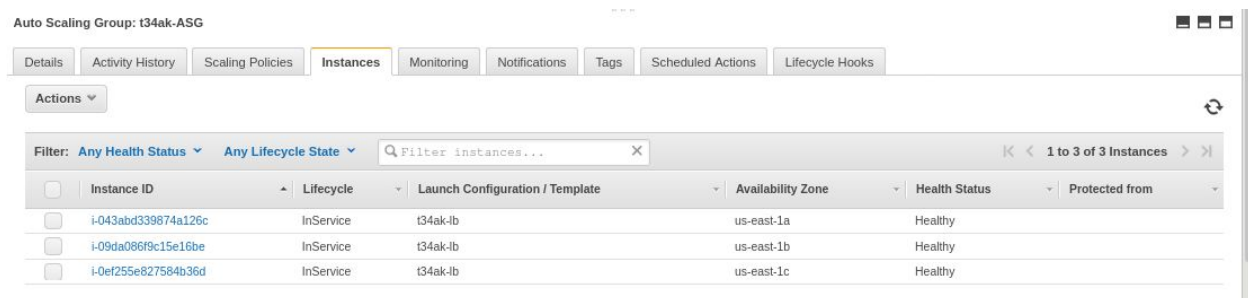


The screenshot shows the AWS Management Console interface for a Target Group named 't34ak-NLB'. The 'Targets' tab is selected, displaying a table of registered targets. Below the table, the 'Availability Zones' section shows the distribution of targets across different zones.

Instance ID	Name	Port	Availability Zone	Status	Description
i-09da086f9c15e16be	t34ak-AG1	80	us-east-1b	healthy	This target is currently passing target group's health checks.
i-0ef255e827584b36d	t34ak-AG3	80	us-east-1c	healthy	This target is currently passing target group's health checks.
i-043abd339874a126c	t34ak-AG	80	us-east-1a	healthy	This target is currently passing target group's health checks.

Availability Zone	Target count	Healthy?
us-east-1b	1	Yes
us-east-1a	1	Yes
us-east-1c	1	Yes

Before changing the value of min and desired



The screenshot shows the AWS Management Console interface for an Auto Scaling Group named 't34ak-ASG'. The 'Instances' tab is selected, displaying a table of instances. The filter is set to 'Any Health Status' and 'Any Lifecycle State'.

Instance ID	Lifecycle	Launch Configuration / Template	Availability Zone	Health Status	Protected from
i-043abd339874a126c	InService	t34ak-lb	us-east-1a	Healthy	
i-09da086f9c15e16be	InService	t34ak-lb	us-east-1b	Healthy	
i-0ef255e827584b36d	InService	t34ak-lb	us-east-1c	Healthy	

After changing the value of min and desired to 1 we can see that the instance count is degraded

Filter: 1 to 1 of 1 Auto Scaling Groups

Name	Launch Configuration /	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grac
t34ak-ASG	t34ak-lb	1	1	1	4	us-east-1a, us-east-1b, us-e...	300	300

Auto Scaling Group: t34ak-ASG

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

Actions

Filter: Any Health Status Any Lifecycle State Filter instances... 1 to 1 of 1 Instances

Instance ID	Lifecycle	Launch Configuration / Template	Availability Zone	Health Status	Protected from
i-043abd339874a126c	InService	t34ak-lb	us-east-1a	Healthy	

Status of the NLB and TG after changing the value of desired and min

search: t34ak Add filter 1 to 3 of 3

Name	Port	Protocol	Target type	Load Balanc	VPC ID	Monitoring
t34ak-New-TG	80	HTTP	instance	t34ak-LB	vpc-01d9bca1ea53fdce9	
t34ak-NLB	80	TCP	instance	t34ak-NLB	vpc-01d9bca1ea53fdce9	

Target group: t34ak-NLB

Description **Targets** Health checks Monitoring Tags

The load balancer starts routing requests to a newly registered target as soon as the registration process completes and the target passes the initial health checks. If demand on your targets increases, you can register additional targets. If demand on your targets decreases, you can deregister targets.

Edit

Registered targets

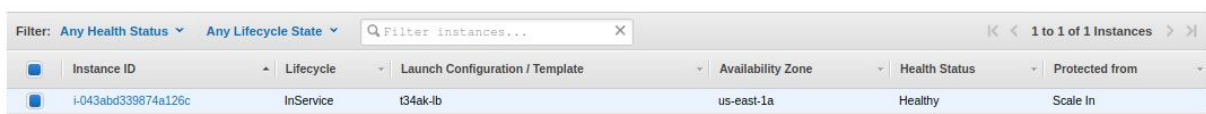
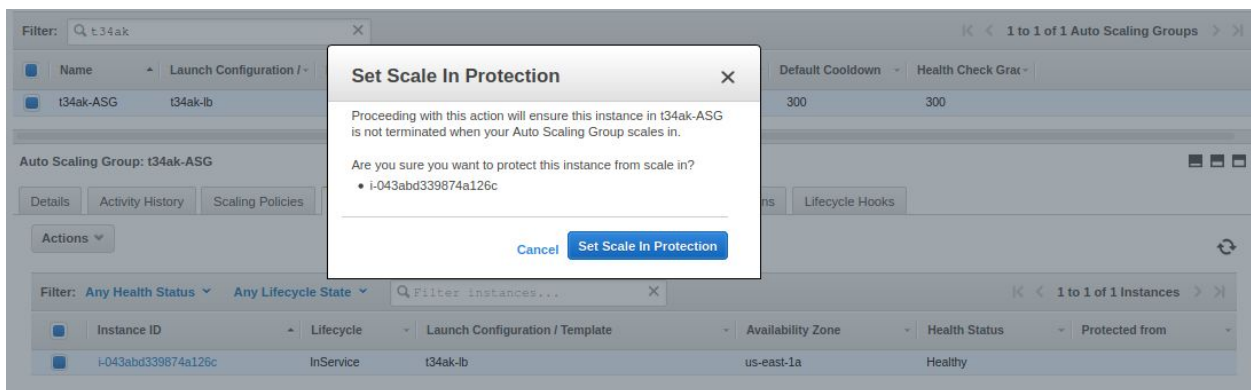
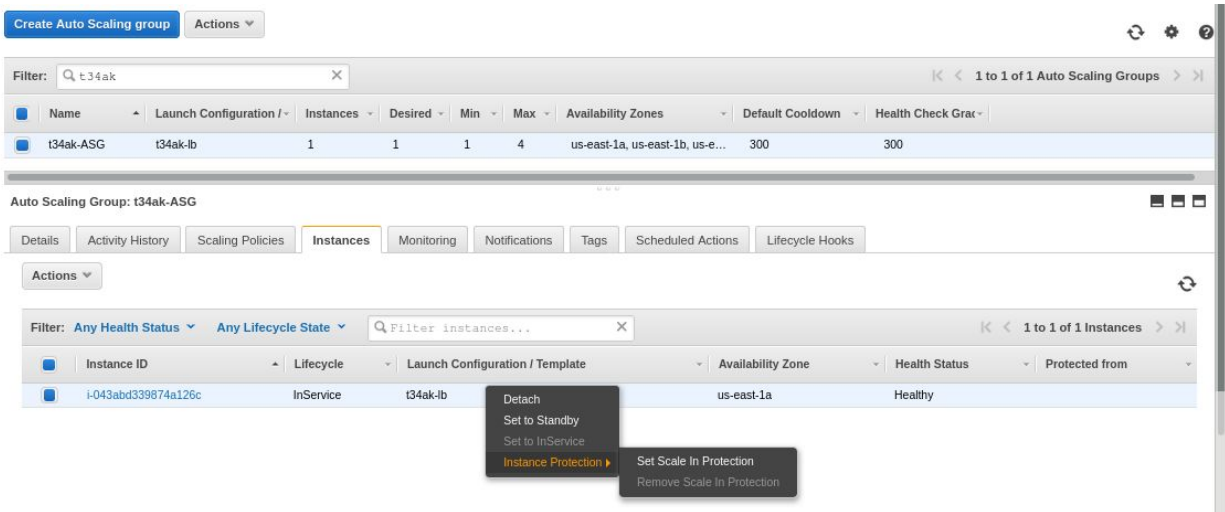
Instance ID	Name	Port	Availability Zone	Status	Description
i-043abd339874a126c	t34ak-AG	80	us-east-1a	healthy	This target is currently passing target group's health checks.

Availability Zones

Availability Zone	Target count	Healthy?
us-east-1a	1	Yes

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

Ans 10:- To set scale-in protection we first have to select the ASG then On Instance tab we have to select the instance on which we have to set the scale-in protection



11. Put Schedules in ASG to:

- Remove all instances of the ASG at 8 PM

To set this we have to go on the schedule actions tab under that click on create Scheduled Action

Create Scheduled Action

Name

remove at 8 pm

Auto Scaling Group

t34ak-ASG

Provide at least one of Min, Max and Desired Capacity

Min

0

Max

0

Desired Capacity

0

Recurrence

Every day

(Cron) 0 20 ***

Start Time

2020-02-29

20 : 00 UTC

Specify the start time in UTC

The first time this scheduled action will run

End Time

2020-03-28

20 : 00 UTC

Specify the end time in UTC

Cancel

Cancel

Create

Filter: t34ak

<< 1 of 1 Auto Scaling Groups >>

Name	Launch Configuration	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grace
t34ak-ASG	t34ak-lb	1	1	1	4	us-east-1a, us-east-1b, us-e...	300	300

Auto Scaling Group: t34ak-ASG

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

Create Scheduled Action Actions

Filter: Filter scheduled actions...

<< 1 of 1 Scheduled Actions >>

Name	Start Time	End Time	Recurrence	Desired Capacity	Min	Max
remove at 8 pm	2020 March 1 01:30:00 UTC+5:30	2020 March 29 01:30:00 UTC+5:30	0 20 ***	0	0	0

- Launch a minimum of 2 instances at 10 AM

To set this we have to go on the schedule actions tab under that click on create Scheduled Action

Create Scheduled Action

Name

t34ak-launch2-instance

Auto Scaling Group

t34ak-ASG

Provide at least one of Min, Max and Desired Capacity

Min

2

Max

4

Desired Capacity

2

Recurrence

Every day

(Cron) 0 22 * * *

Start Time

2020-02-29

22 : 00

UTC

Specify the start time in UTC

The first time this scheduled action will run

End Time

2020-03-28

22 : 00

UTC

Specify the end time in UTC

Cancel

Cancel

Create

Create Auto Scaling group

Actions

Filter: t34ak

1 to 1 of 1 Auto Scaling Groups

Name	Launch Configuration /	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grac
t34ak-ASG	t34ak-lb	1	1	1	4	us-east-1a, us-east-1b, us-e...	300	300

Auto Scaling Group: t34ak-ASG

Details

Activity History

Scaling Policies

Instances

Monitoring

Notifications

Tags

Scheduled Actions

Lifecycle Hooks

Create Scheduled Action

Actions

Filter: Filter scheduled actions...

1 to 2 of 2 Scheduled Actions

Name	Start Time	End Time	Recurrence	Desired Capacity	Min	Max
remove at 8 pm	2020 March 1 01:30:00 UTC+5:30	2020 March 29 01:30:00 UTC+5:30	0 20 * * *	0	0	0
t34ak-launch2-instance	2020 March 1 03:30:00 UTC+5:30	2020 March 29 03:30:00 UTC+5:30	0 22 * * *	2	2	4