Auto-scaling and LoadBalancer

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.

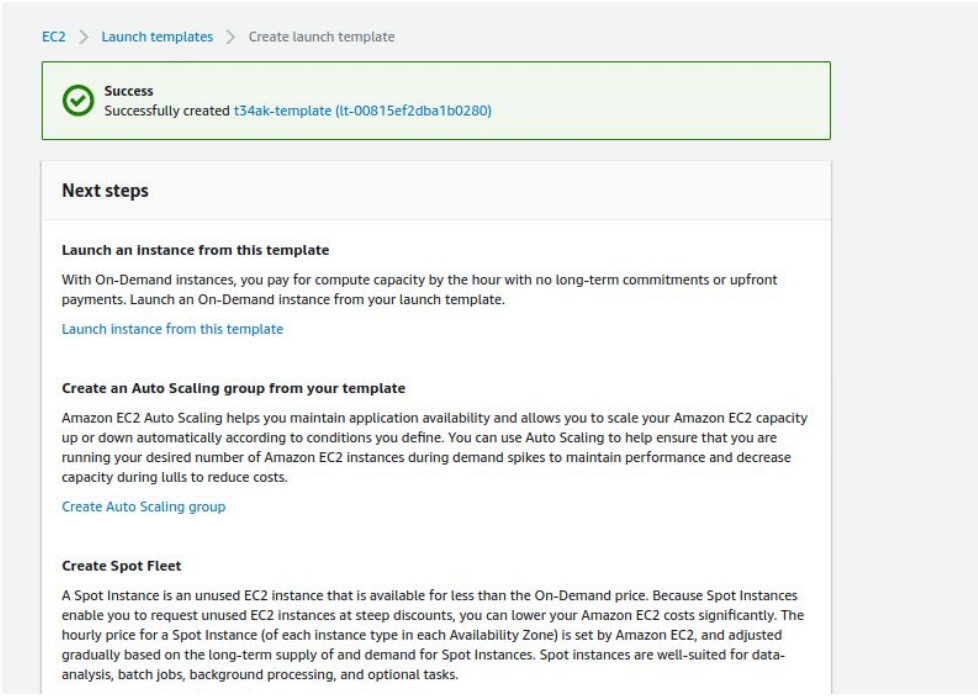
ELB health check--

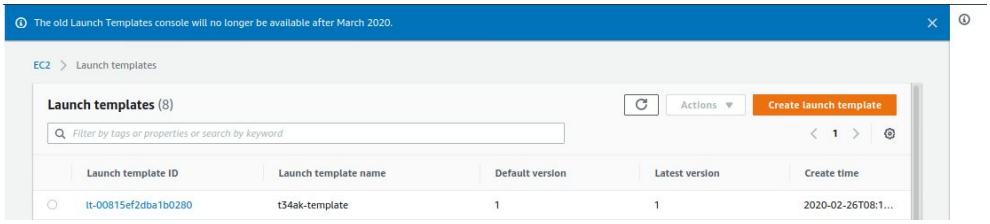
verifies that a specified TCP port on an instance is accepting connections OR a specified web page returns 2xx code. Thus ELB health checks are a little bit smarter and verify that actual app works instead of verifying that just an instance works.

Ques 5:-Create 2 auto-scaling groups with

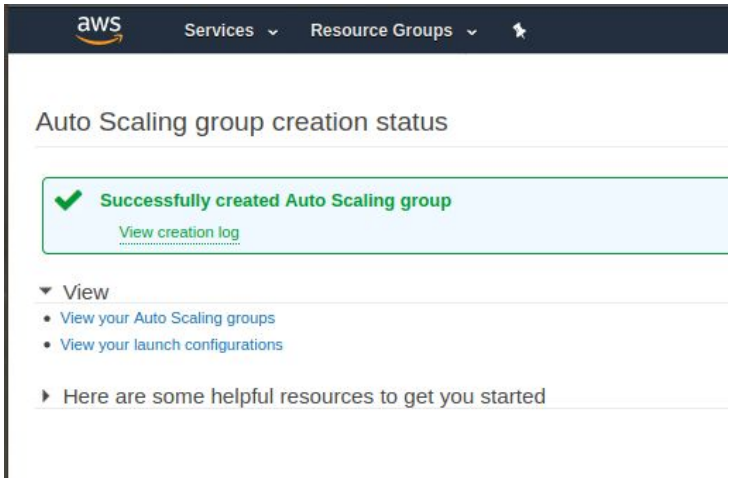
- launch configuration and
- launch template

Ans 5:- Creating the launch template

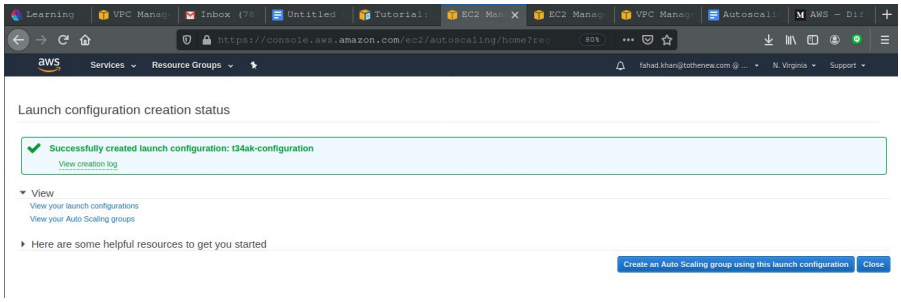


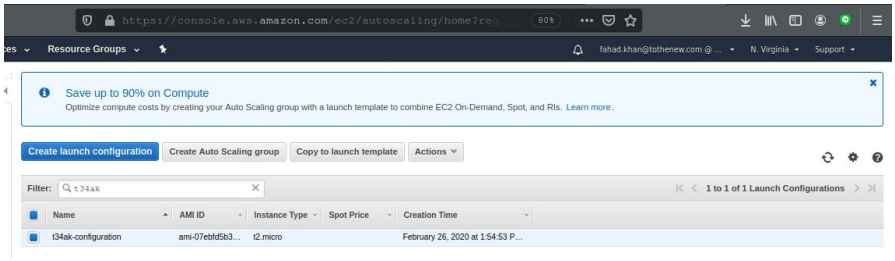


Creating the Auto Scaling Group using Launch Template

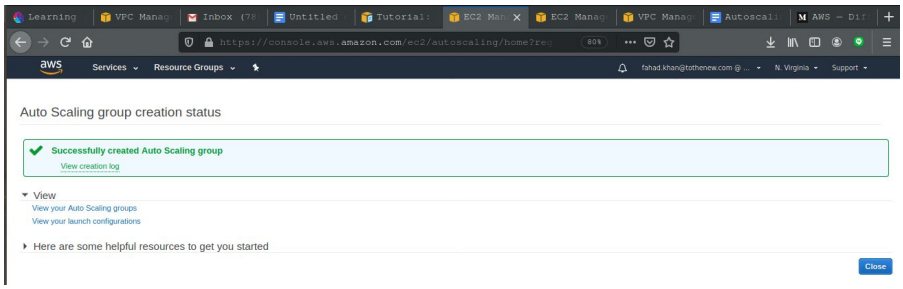


Creating Launch Configuration

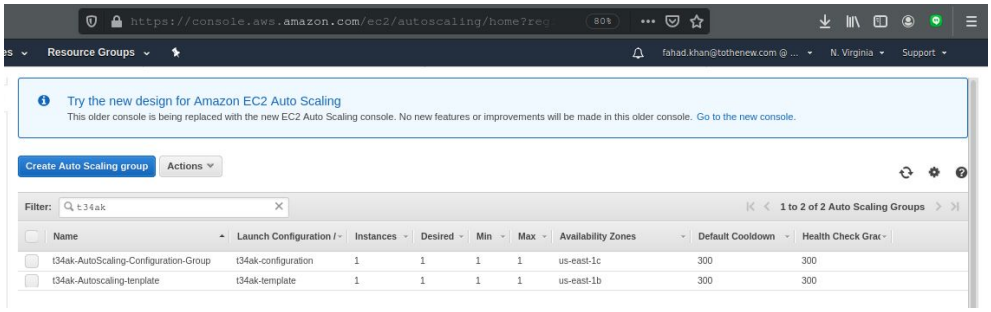




Creating Auto Scaling Group using Launch Configuration

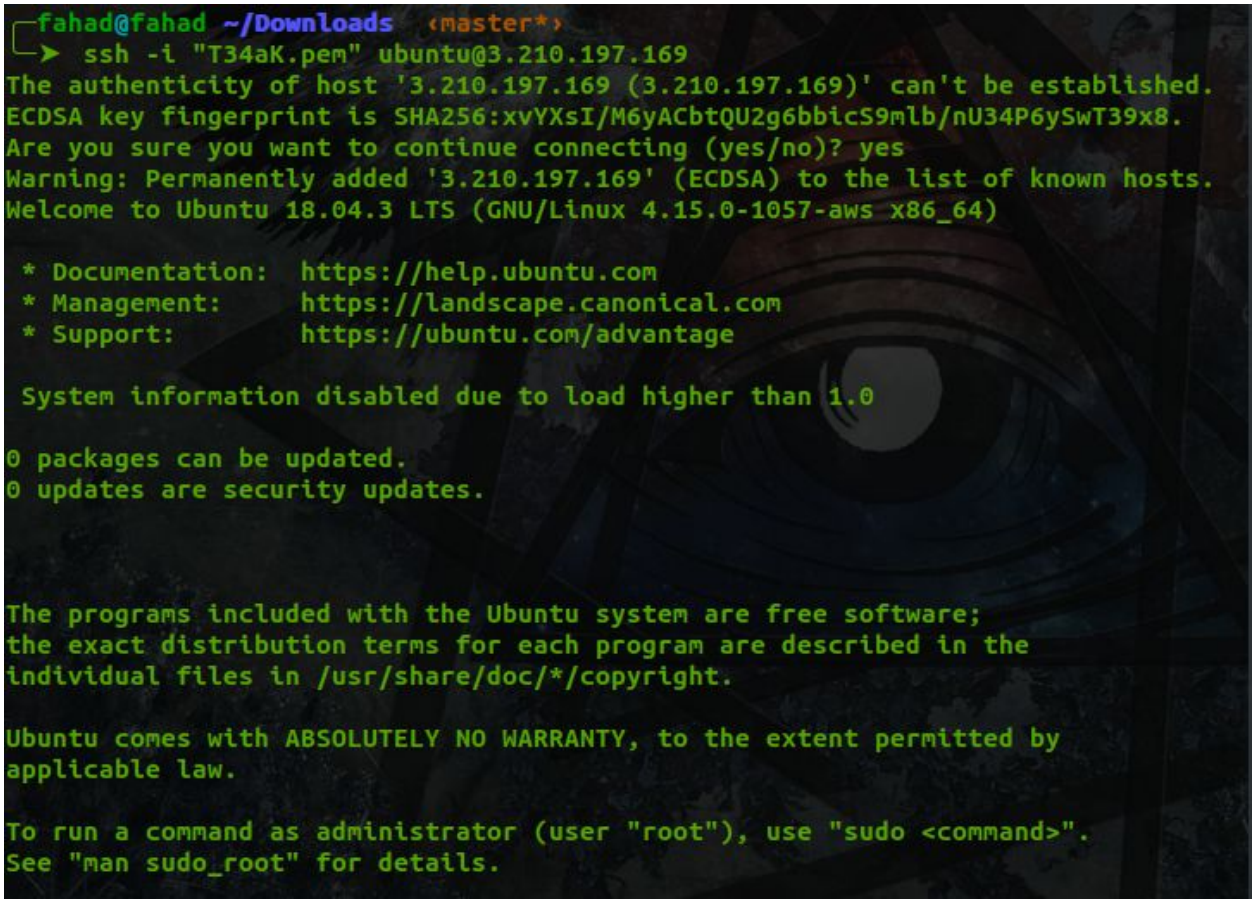


Two Auto Scaling Groups with launch template and configuration ....



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:-



First we have to install and configure the nginx and wordpress

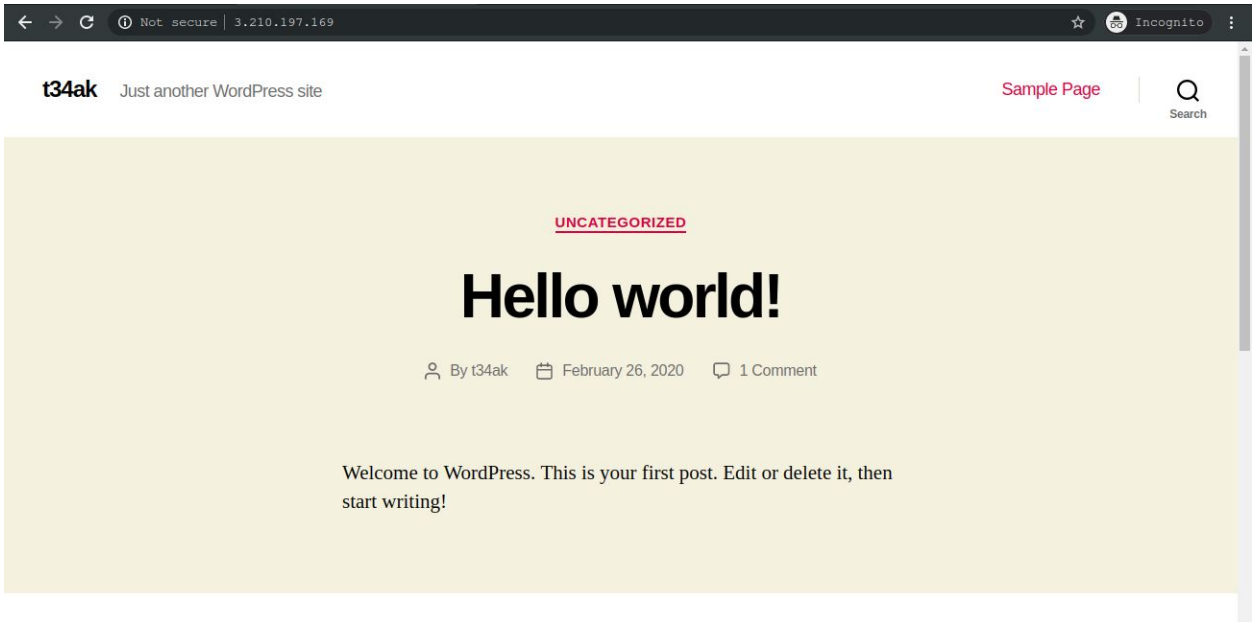
```
ubuntu@ip-10-0-9-145:~$ sudo apt-get install nginx
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  fontconfig-config fonts-dejavu-core libfontconfig1 libgd3 libjbig0
  libjpeg-turbo8 libjpeg8 libnginx-mod-http-geoip
  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
  tinyca
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
```

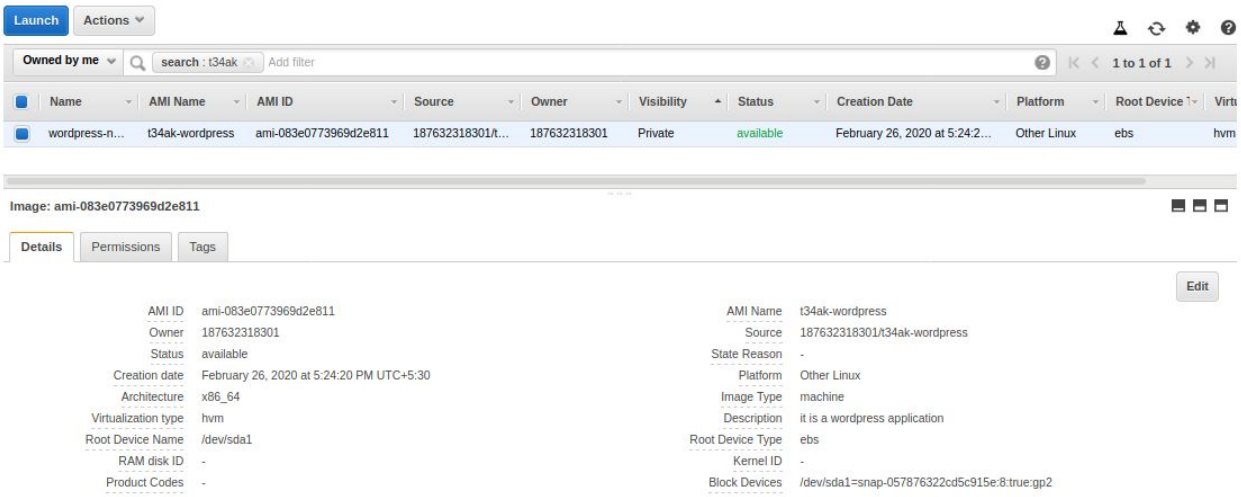
```
ubuntu@ip-10-0-9-145:~$ sudo bash
root@ip-10-0-9-145:~# systemctl enable mariadb.service
root@ip-10-0-9-145:~# mysql_secure_installation

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-comm
n all 1:60ubuntu1 [12.1 kB]
```

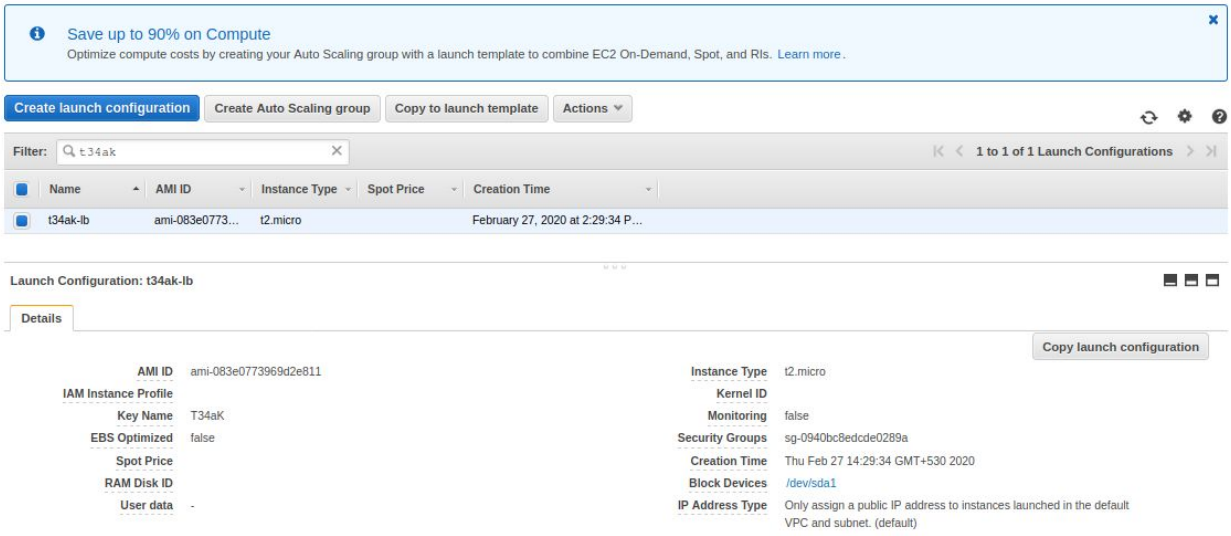
The wordpress start working now we have to put it to the load balancer



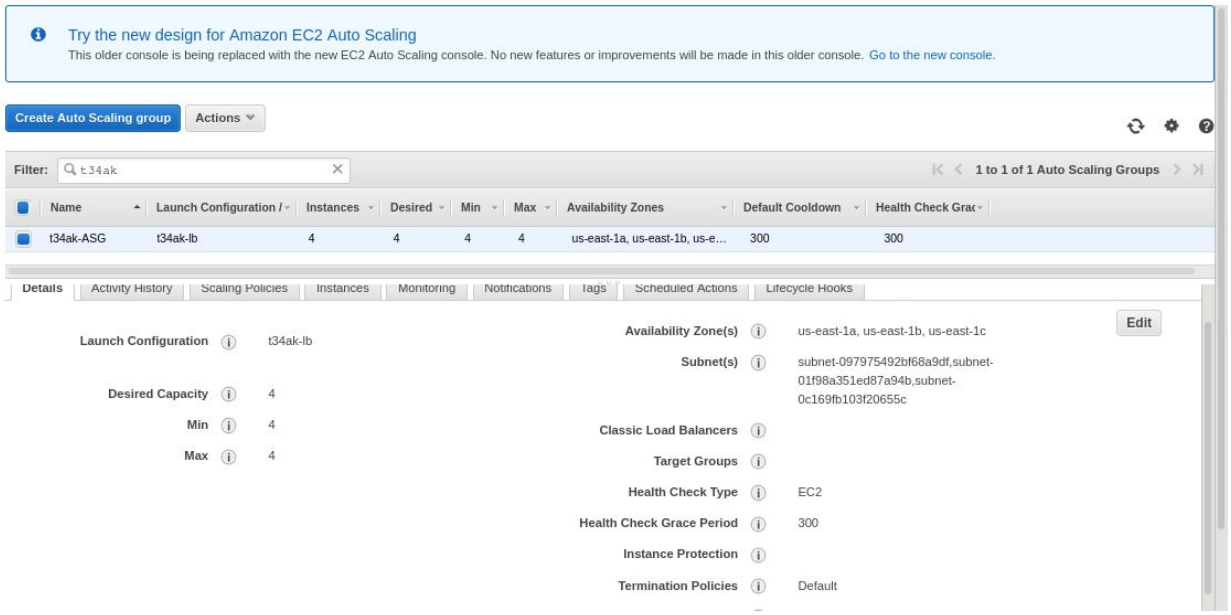
To set up the load balancer we first have to create the AMI of running Instance

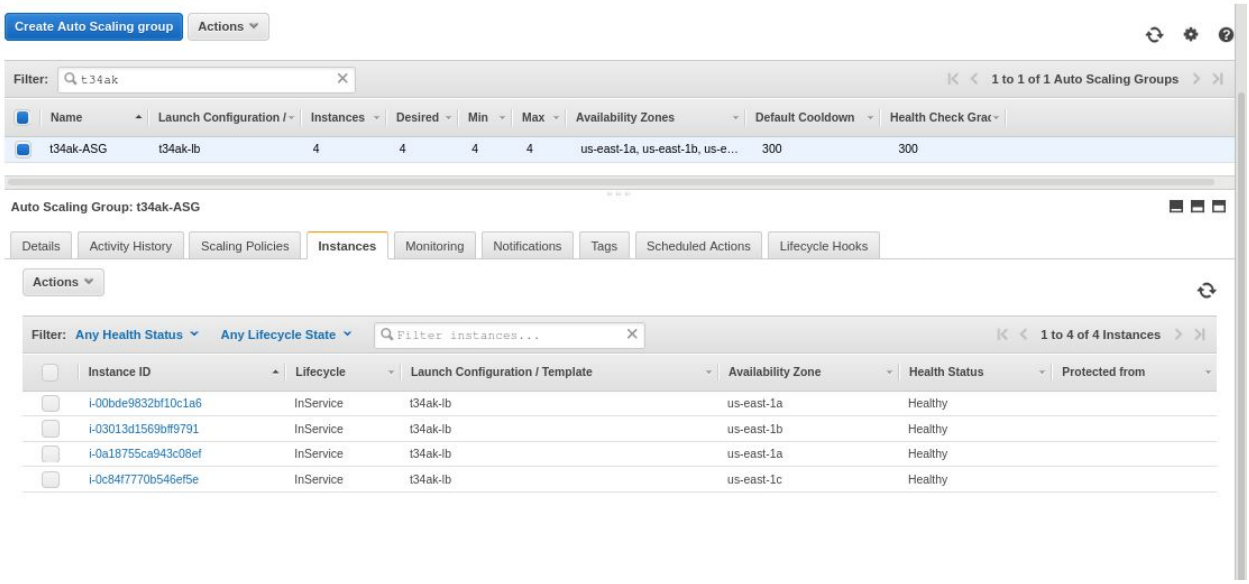


Then we have to create the Launch Configuration with the help of the AMI



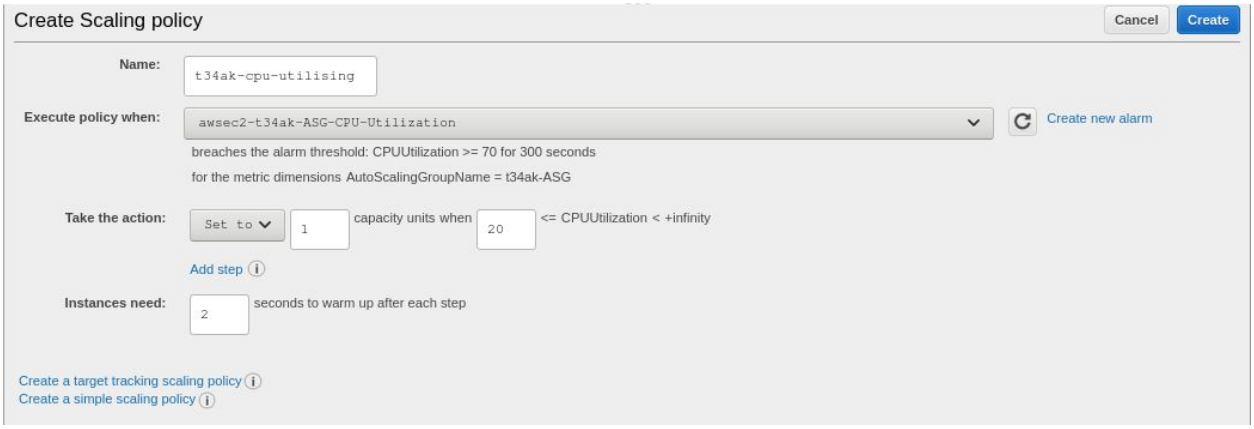
Then we have to create the Auto Scaling Group with help of Launch configuration



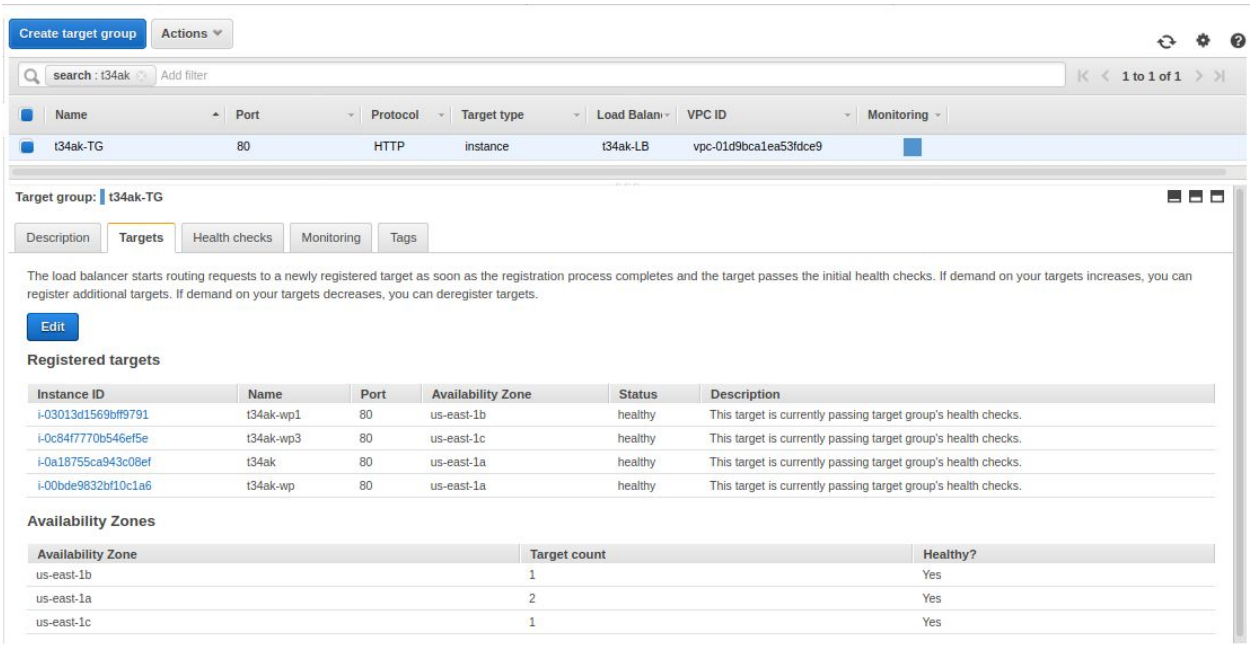


By creating the scaling property we can auto scale the instances.

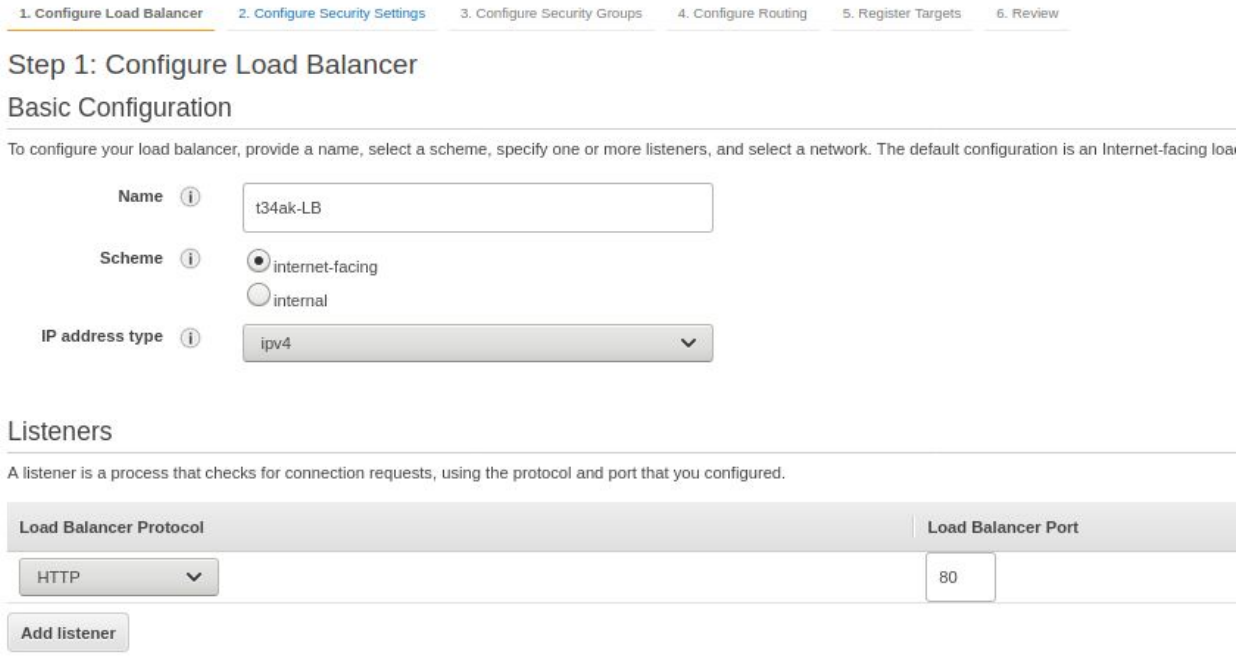
Here i use step scaling



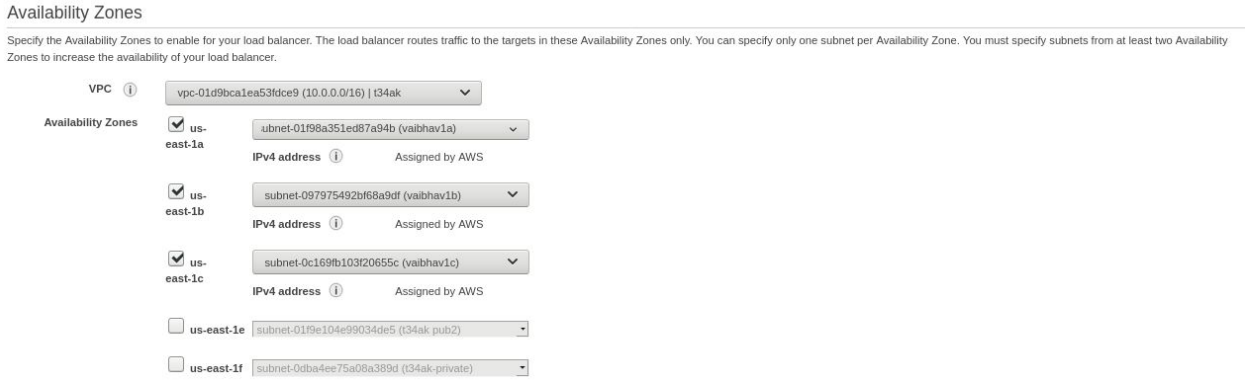
Now we have to create the Target Group and have to add the instances of the Auto Scaling Group to The target Group



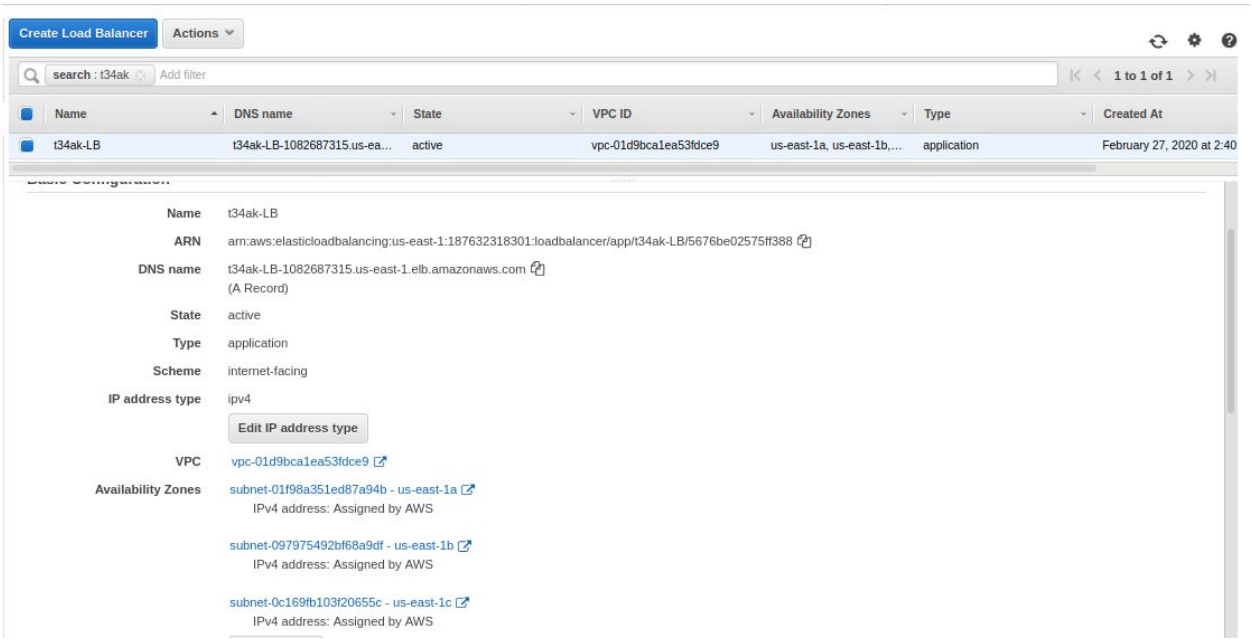
Now we have to create the Load Balancer



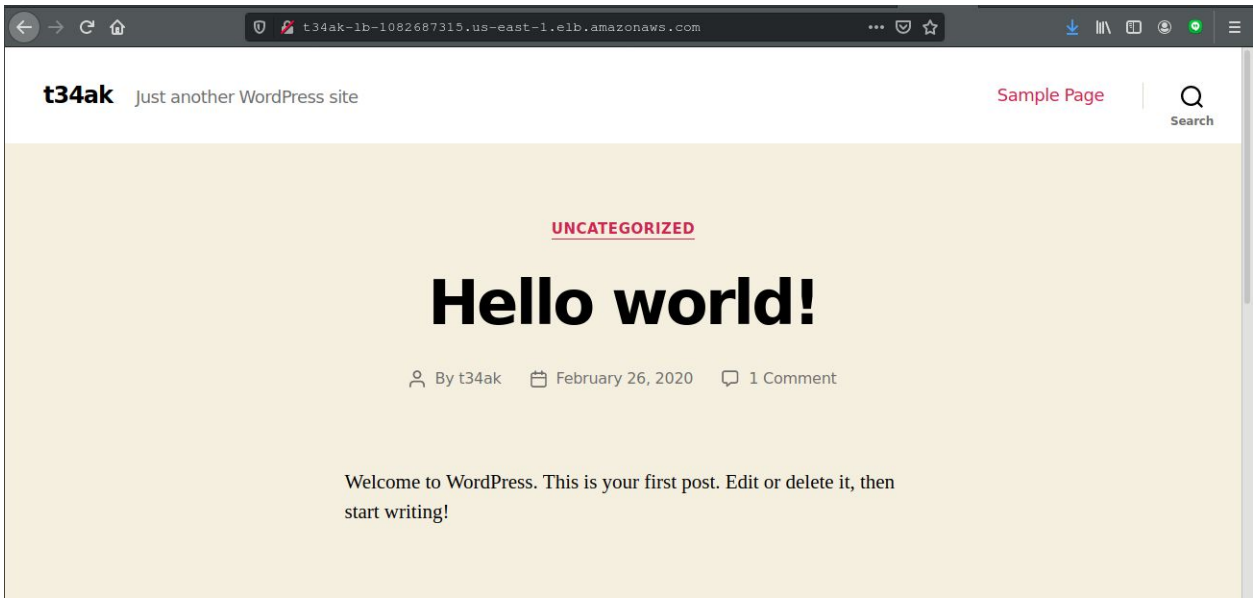
We have to specify thhe VPC and attach the public subnets to them



Now the load Balancer is created with all the properties

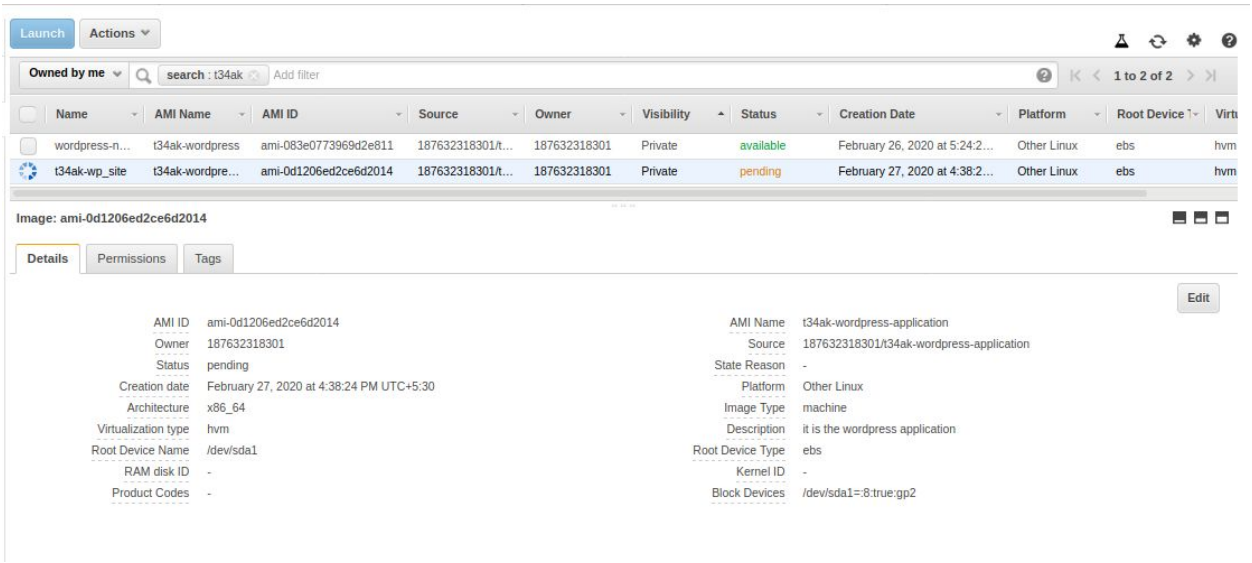


Now we just have to copy the Dns Name and just have to paste it in the url and then the Load Balancer is loaded

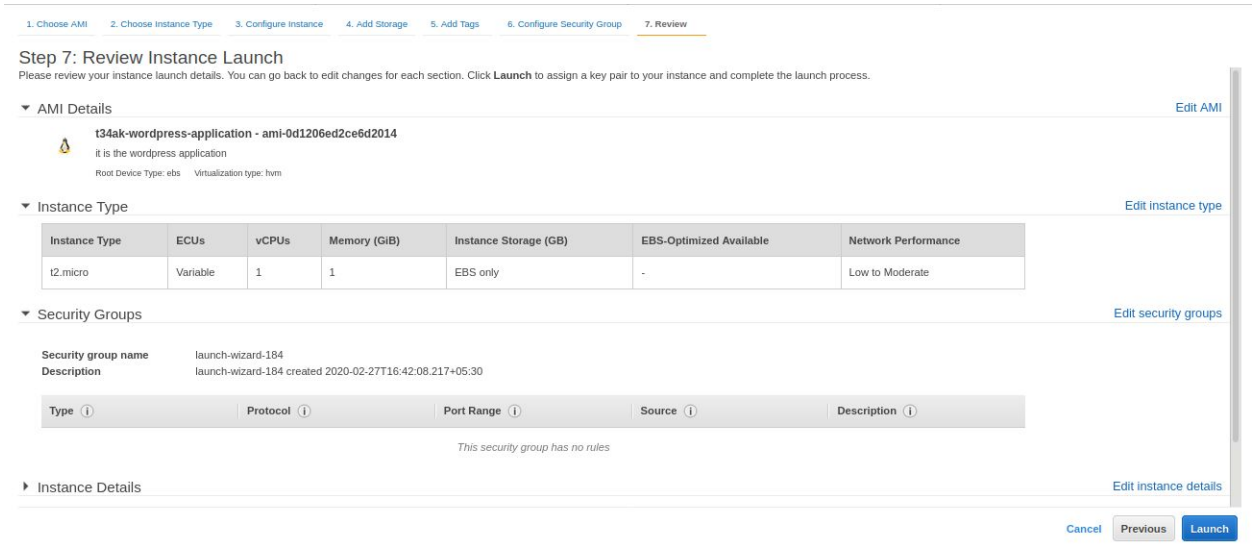


Ques 7:-Create another Wordpress website and use the ALB created above to send traffic to this website based on the hostname

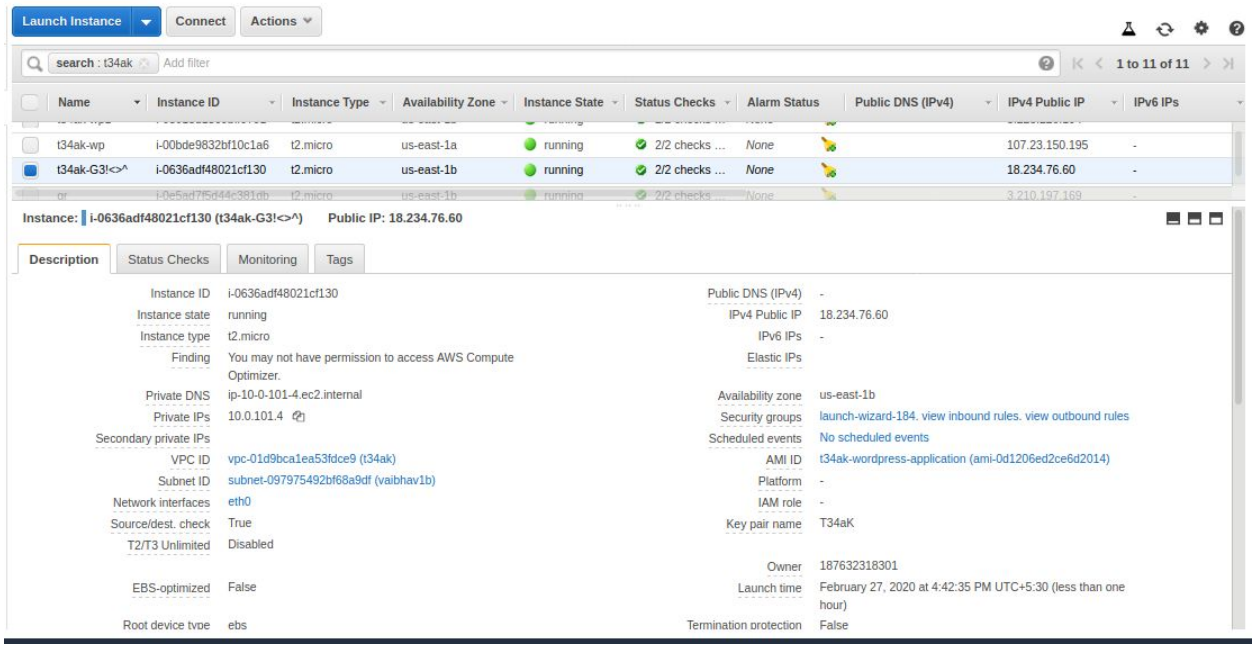
Ans 7:-First we have to create the AMI of the new wordpress site



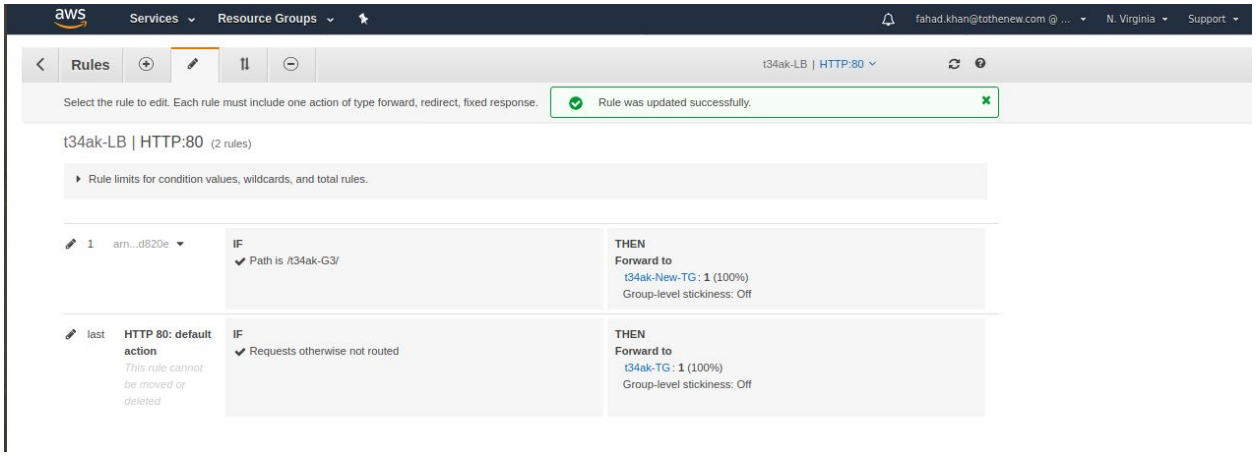
Then we have to create the instance of that AMI



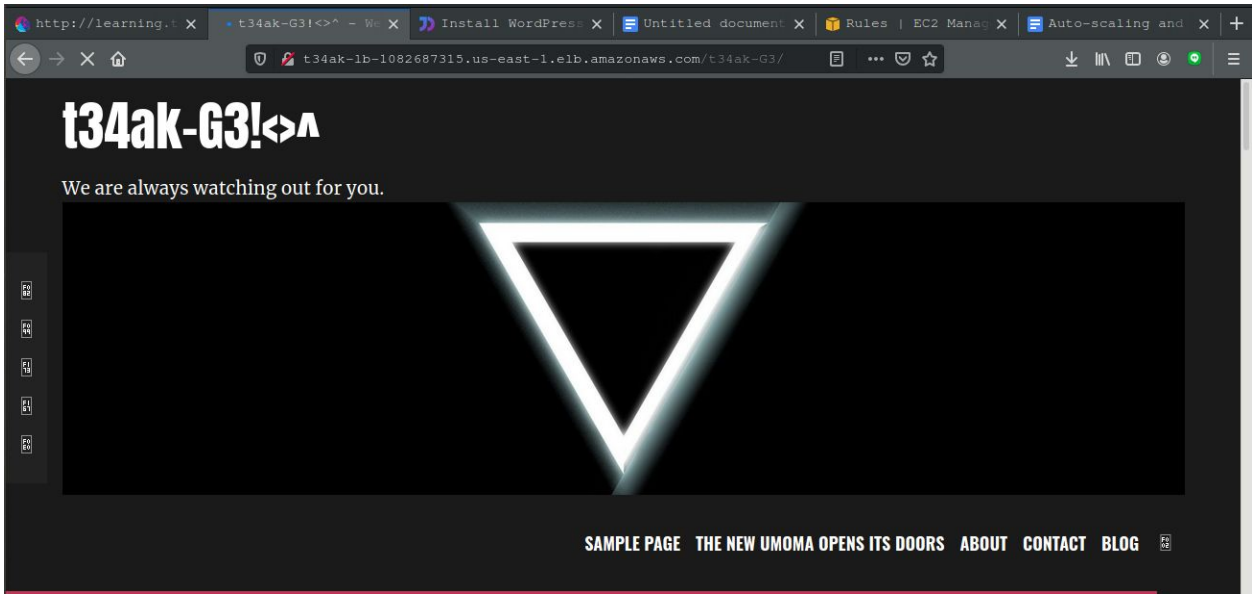
Now we have to add this instance to Other Target Group



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

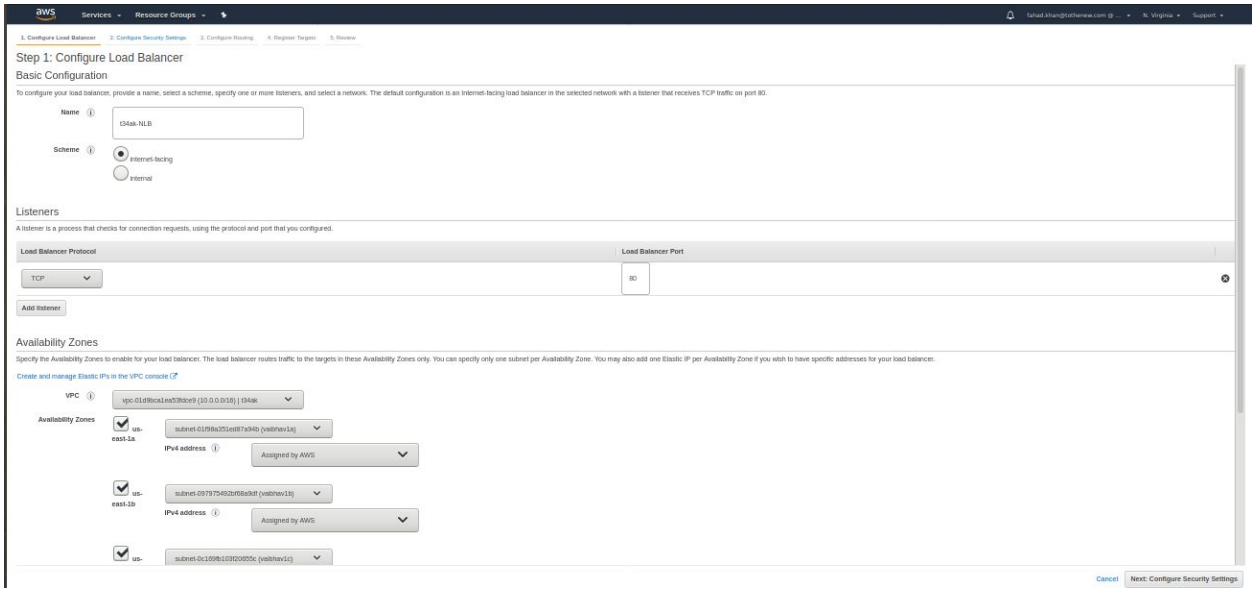


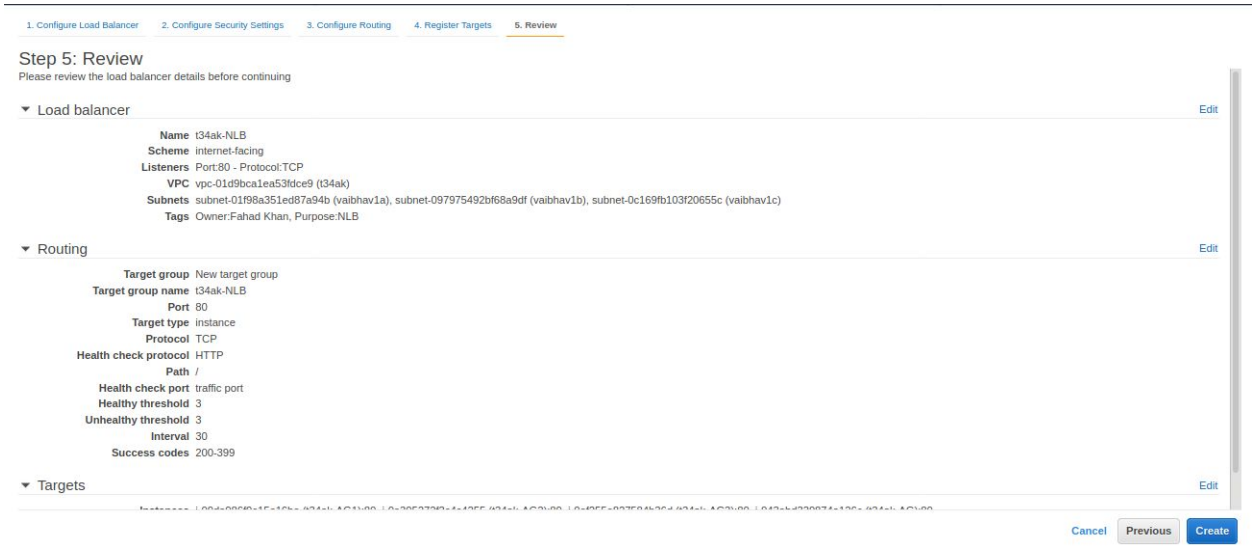
Now if we type the url with /t34ak-G3 it will lead us to other landing page.



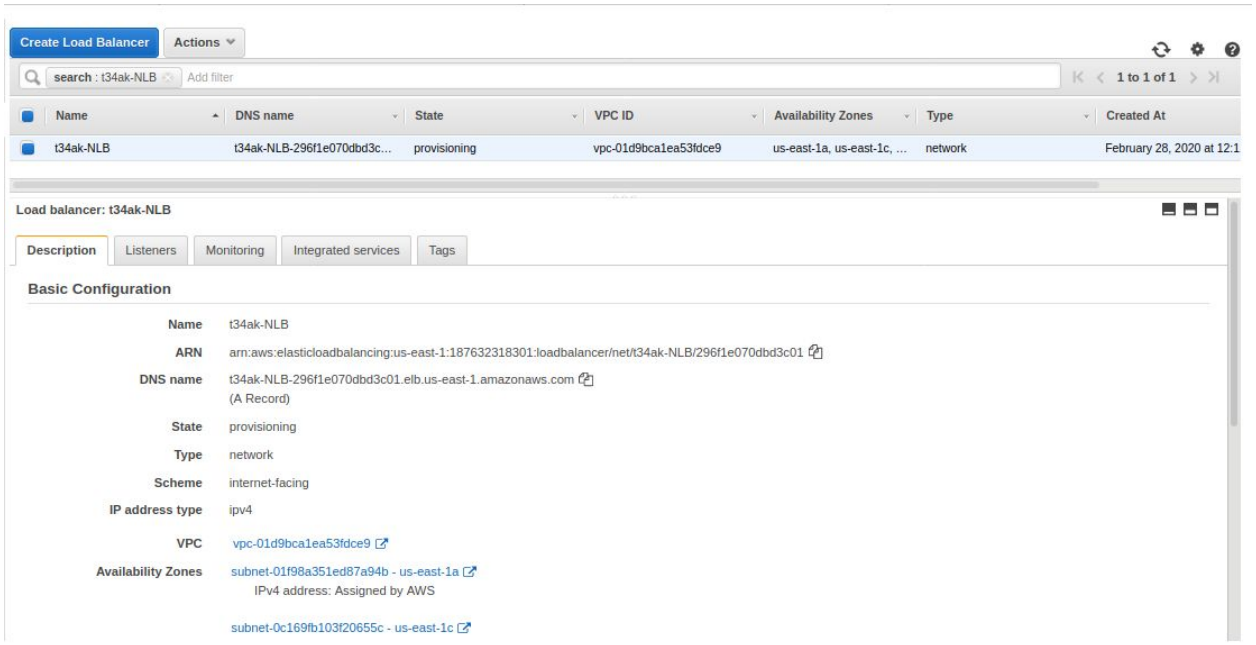
Ques 8:-Use NLB that replaces the ALB in the above setup.

Ans 8:-

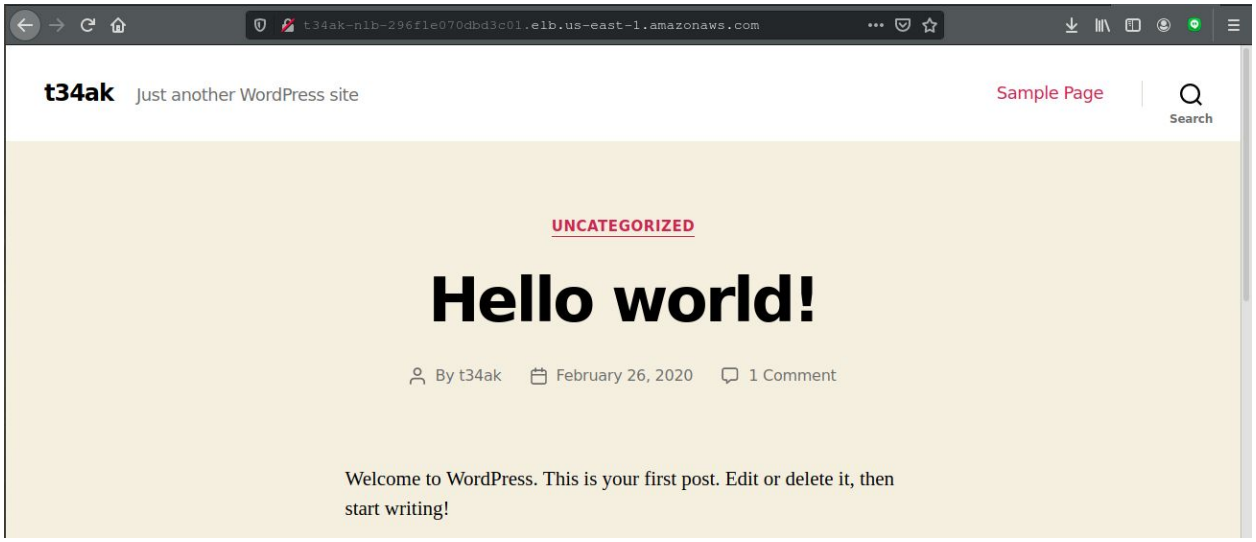




Now we just have to copy the DNS url and paste it to the webbrowser



NLB starts working

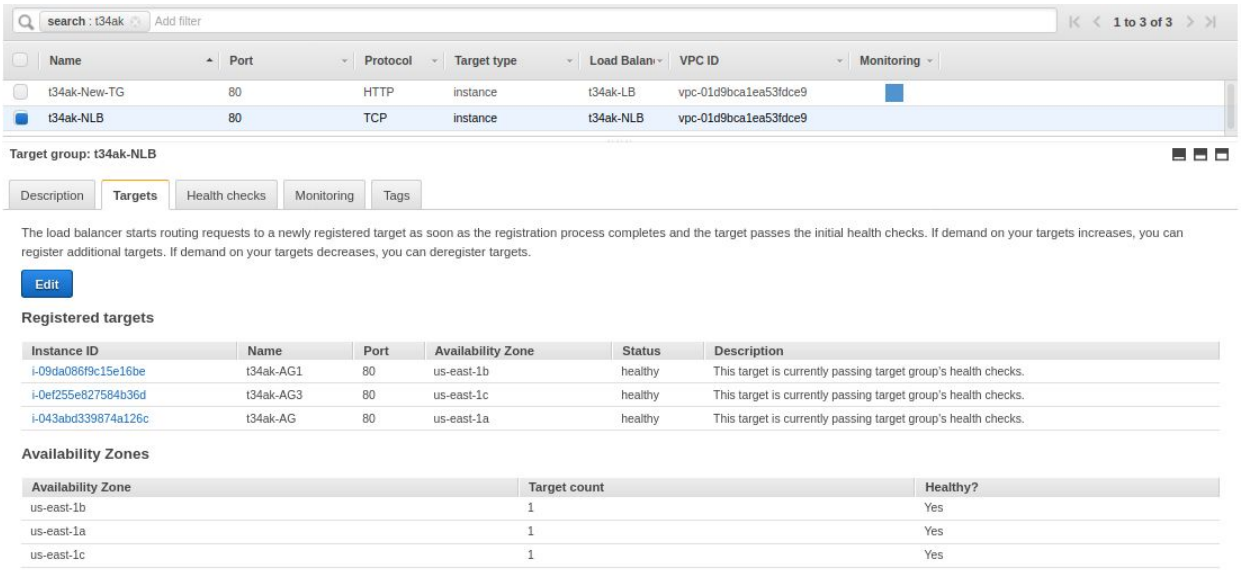


Ques 9.Take an instance out of the ASG.

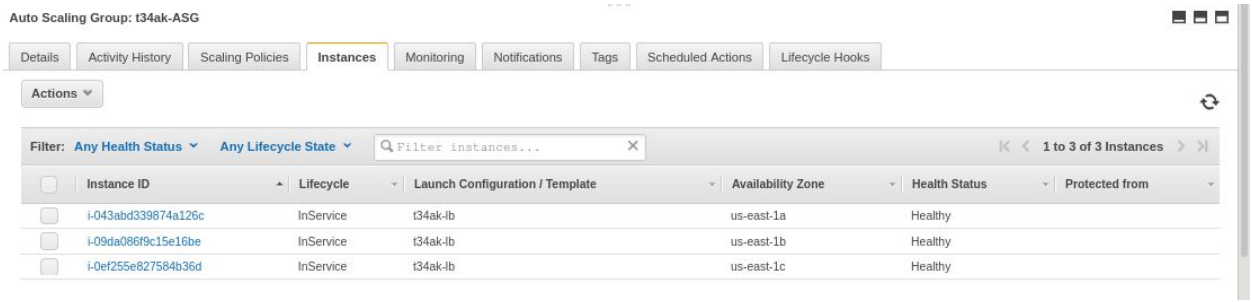
Ans 9:- For taking the instance out of ASG

First we have to set min value to 1

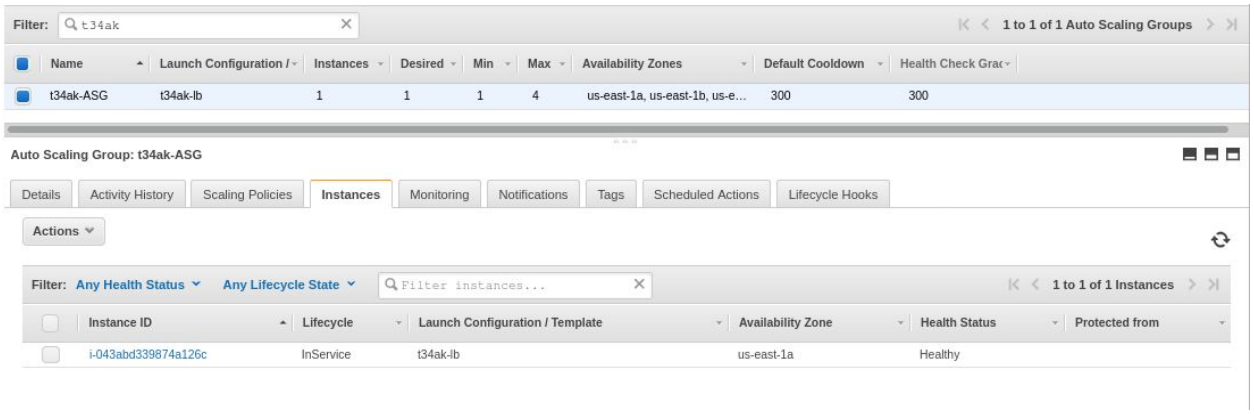
Previously there were m3 instances running in the NLB and TG



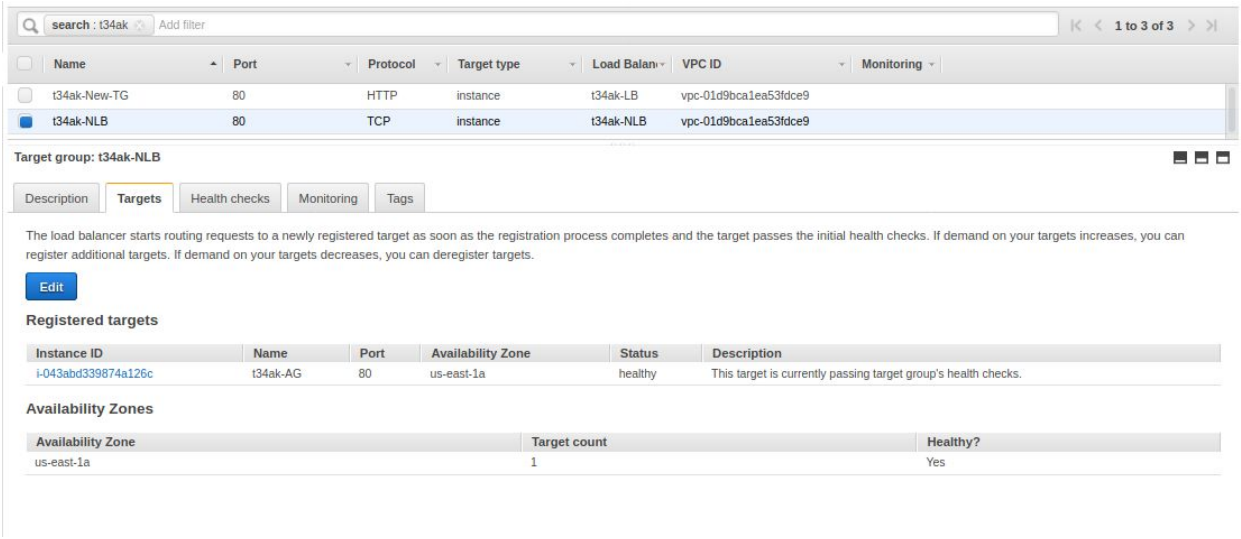
Before changing the value of min and desired



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

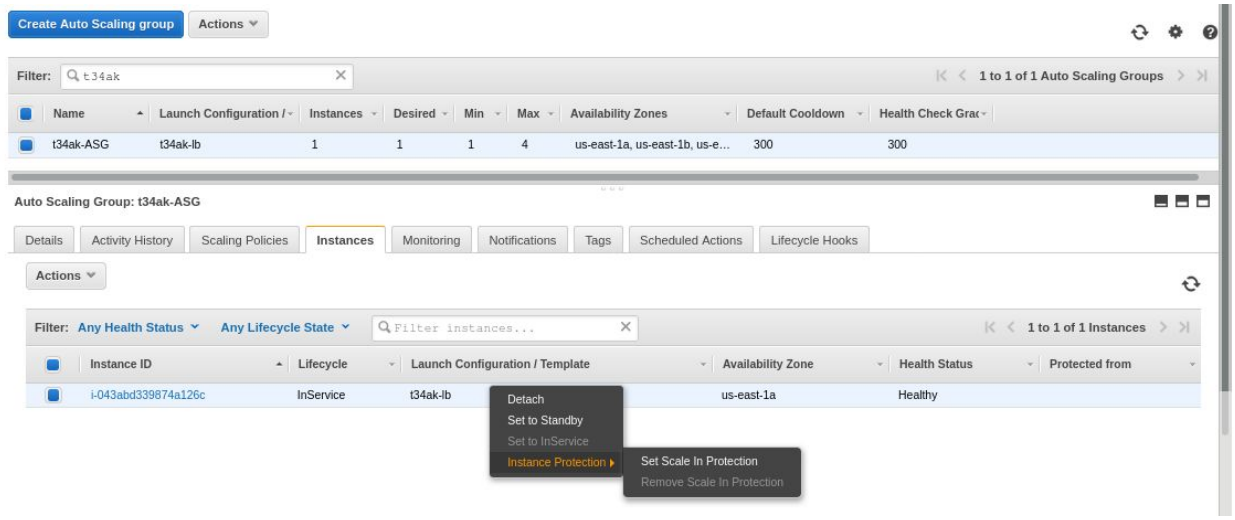


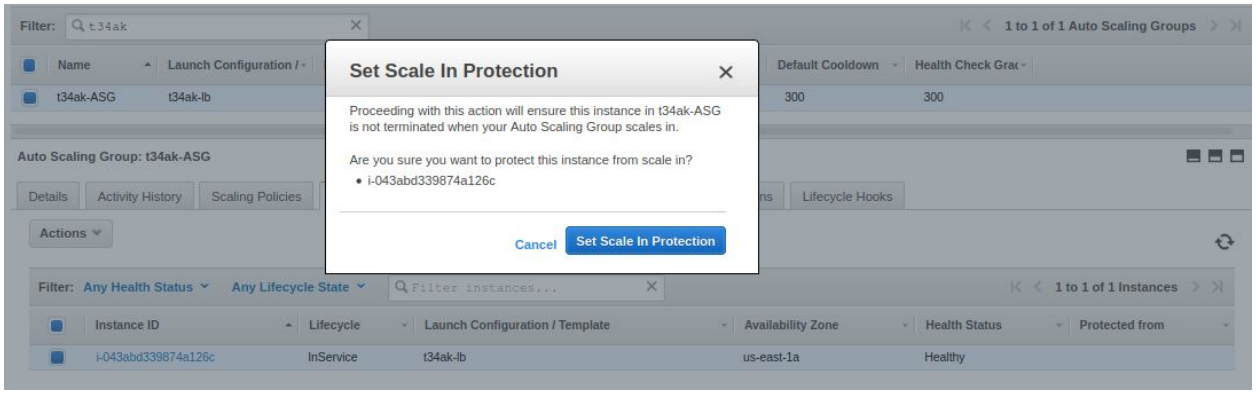
Status of the NLB and TG after changing the value of desired and min

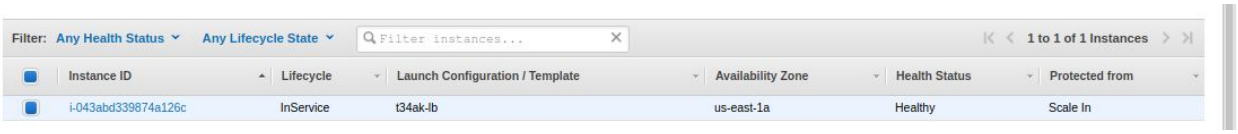


10.Put scale-in protection on an instance in the ASG.

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



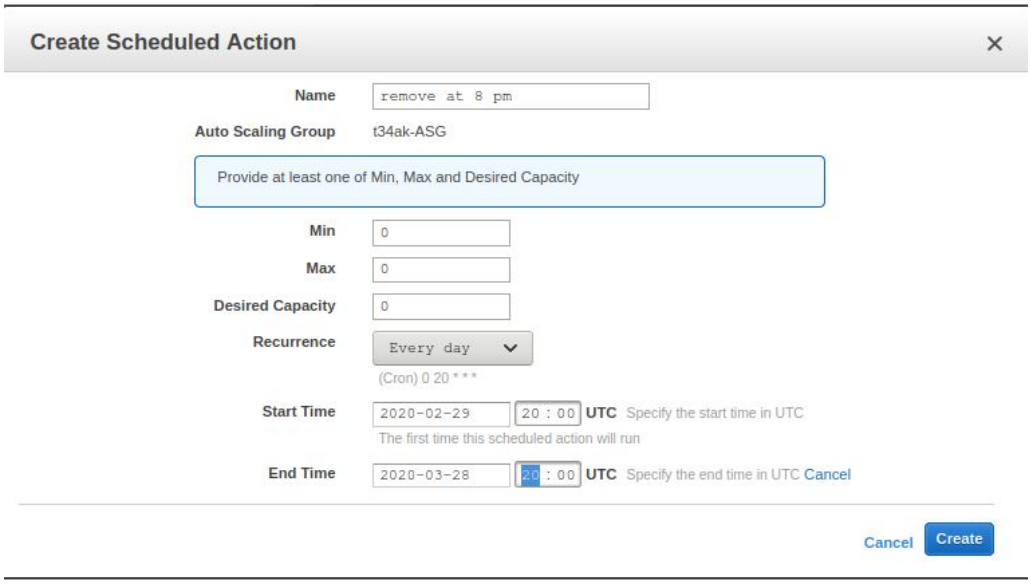


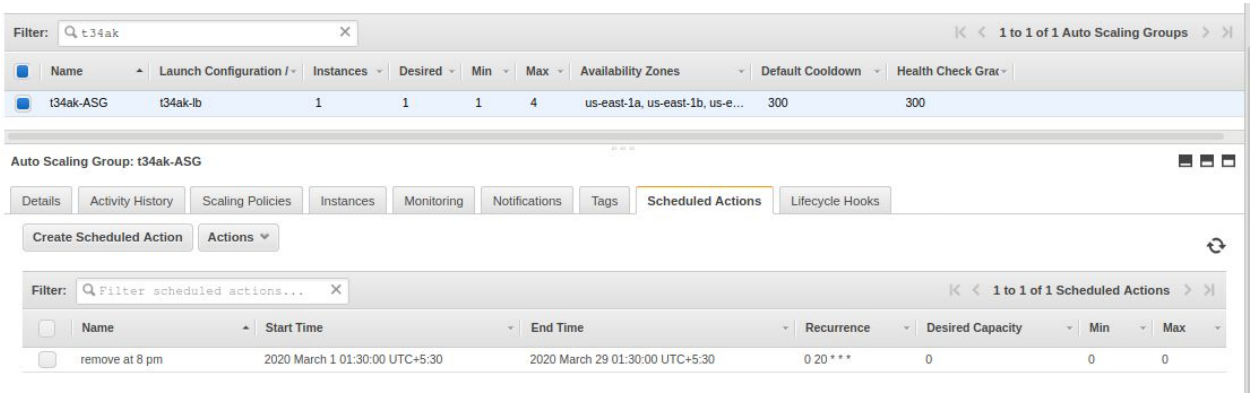


11.Put Schedules in ASG to:

- Remove all instances of the ASG at 8 PM

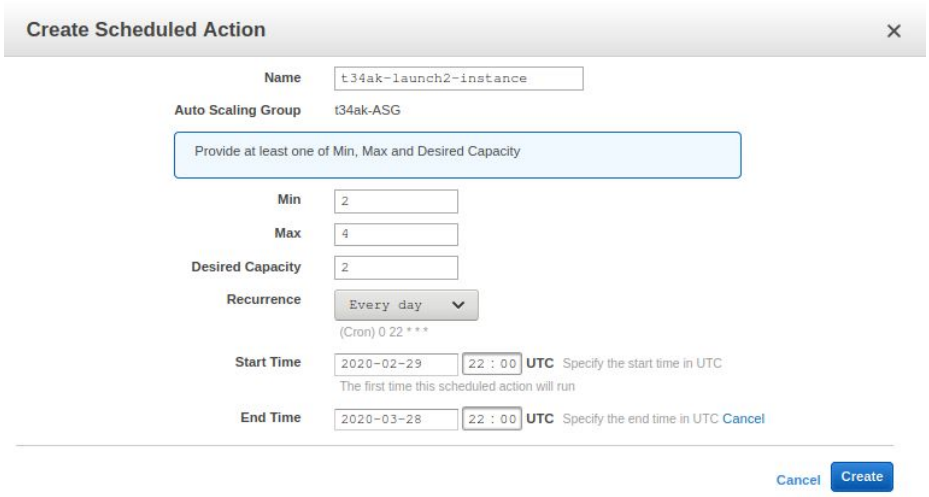
  To set this we have to go on the schedule actions tab under that click on create Scheduled Action

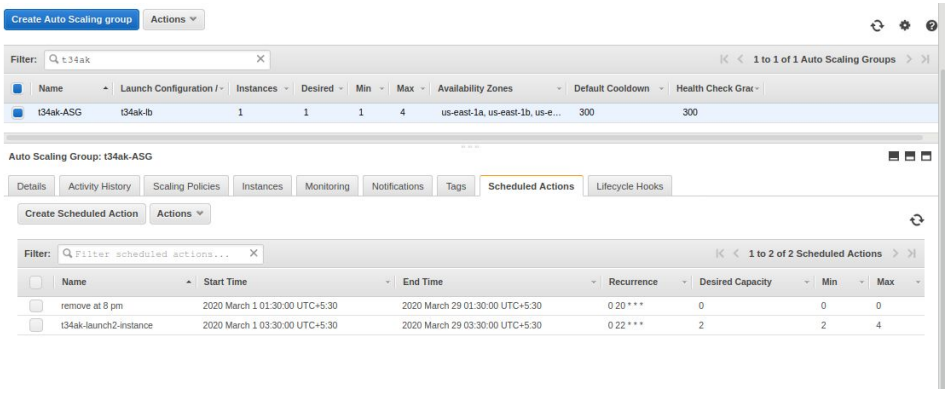




- Launch a minimum of 2 instances at 10 AM

  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