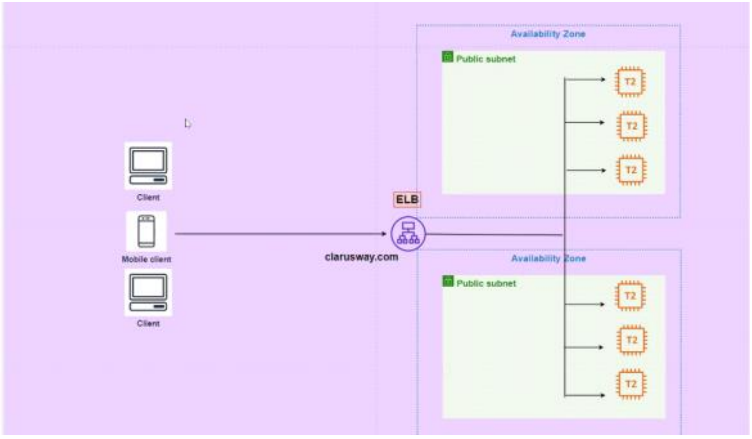
Elastic Load Balancing (ELB)

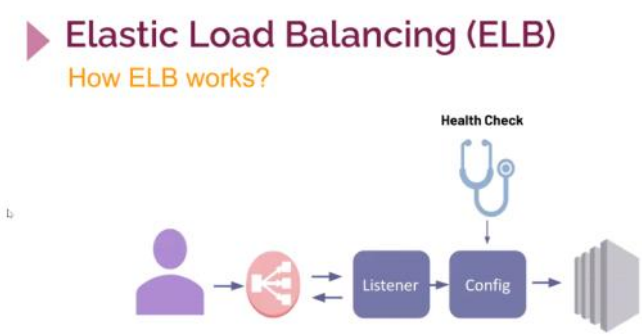
Wednesday, 7 July 2021 19.52

Genellikle production ortaminda yayin yapan web sitemizini kesintiye ugramadan yayin yapmasini isteriz. Ayrica web sitemize gelen trafigin arkada calisan EC2 lara dengeli bir sekide dagitilmasida arzu edilen bir durumdur. Loadbalancer lar bu amac icin uygulanacak network cozumleridir.

Loadbalancerlar gelen trafigin dengeli bir sekilde arkada calisan pclerinize dengeli bir sekilde dagitilmasini saglar. Makinalardan birinde bir sorun olmasi durumunda gelen trafigi kalan makinalara dengeli bir sekilde dagitir. Boylelikle hem gecikme sureleri azaltilmis olur.



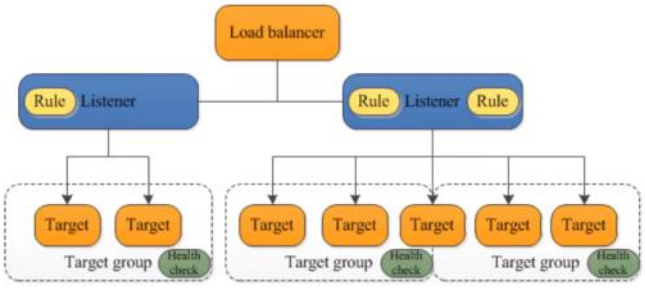
Biz bugun Application Load Balancing uzerinde duracagiz



Belirli componentler mevcut. Bir tane listener tanimliyoruz ve listener a bir veya birden fazla rol veriyoruz. Listener servera gelen talepleri hangi porttan yayin yapiliyorsa listener o portu dinlemeye basliyor. Asagidaki birden fazla target grubuna ec2 tanimliyabiliyoruz. Road balancer belli araliklarla targetlara ping gonderiyor. Eger geri donusu alirsa healty olarak tanimliyor ve bu EC2 ya gelen requestleri gondermeye devam ediyor. Eger unhealthy ise request gondermiyor.

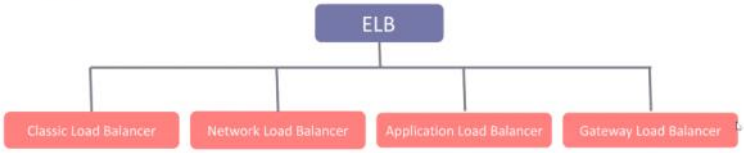
Elastic Load Balancing (ELB)

How ELB works?



Elastic Load Balancing (ELB)

Types of ELB



Elastic Load Balancing (ELB)

	Application Load Balancer	Network Load Balancer	Classic Load Balancer	Gateway Load Balancer
Protocol	HTTP, HTTPS	TCP, TLS, UPD	HTTP, HTTPS and TCP, TLS/SSL	IP
Use case	Control of HTTP and HTTPS traffic.	TCP traffic when high performance is required.	Classic Load Balancer can do what others do	Gateway for distributing traffic across multiple virtual appliances

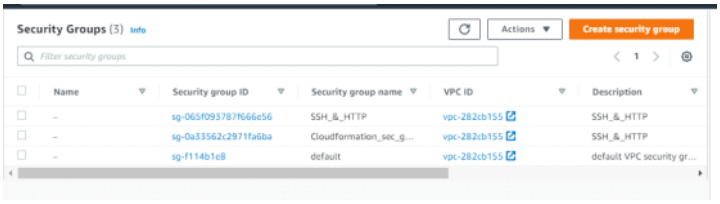
Elastic Load Balancing (ELB)

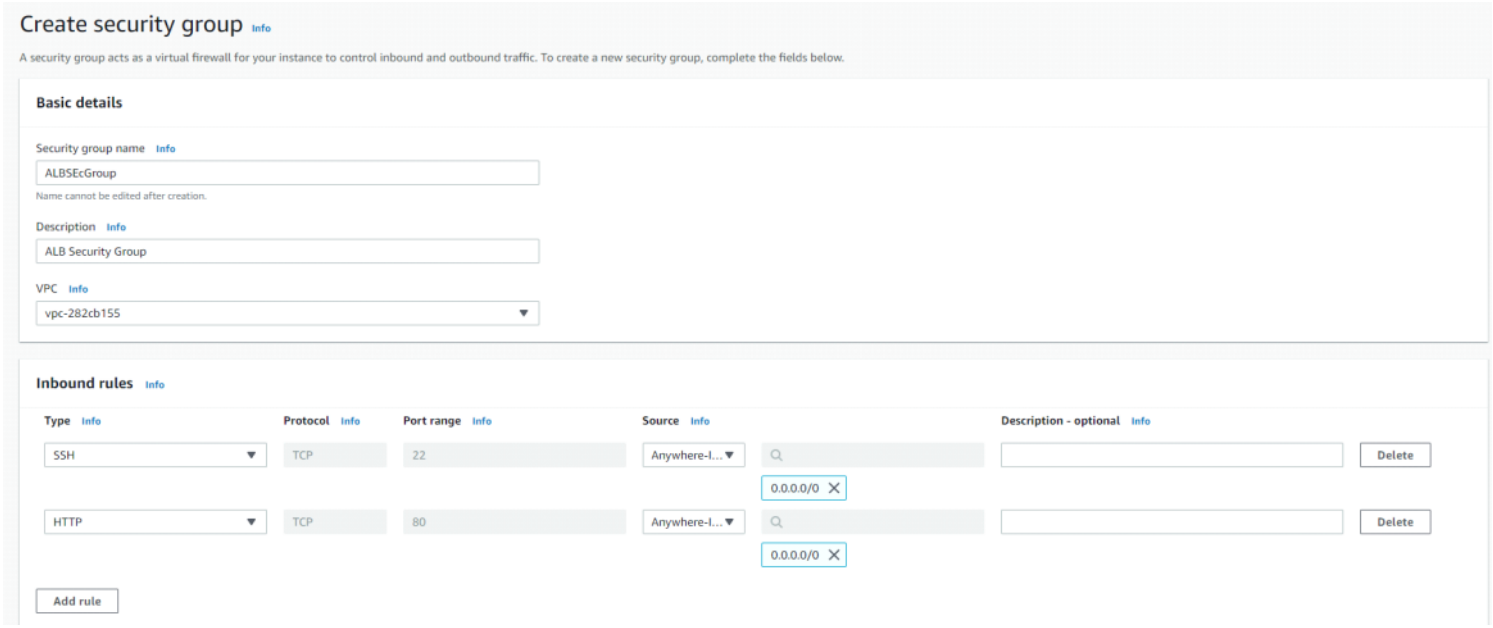
Health Checking

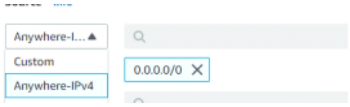
Application Load Balancer ==> target group olusturacagiz. Daha sonra launch template olusturacagiz yani kalip uretimi yapacagiz. 7 adim yerine tek kalipla belli bir sayida instance olusturup target group icerisine ekleyecegiz ve load balancer create edecegiz.

- EC2 lari
  - Tek tek yani manuel olarak
  - Launch template ile
  - Cloud formation
  - CLI ile olusturabiliyoruz

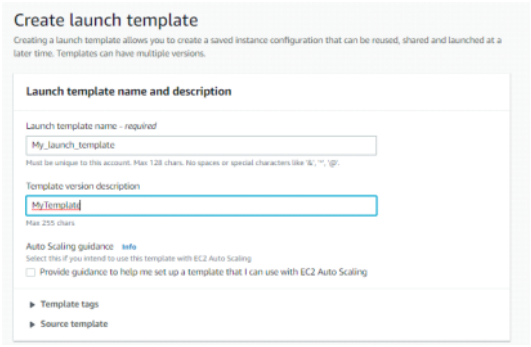
- Oncelikle security group olusturacagiz. HTTP ve SSH inbound lari anywhere olarak secip create ediyoruz. Inbound otomatik outbound olarak geciyor.

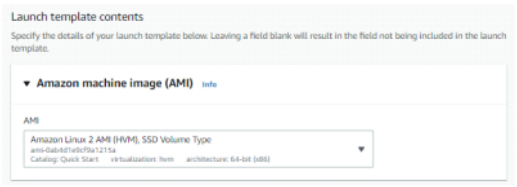


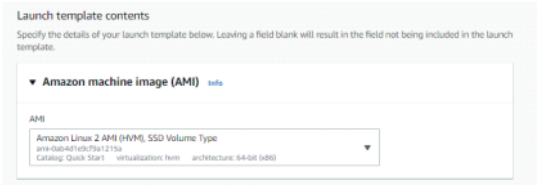


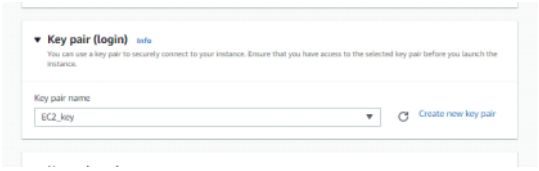


- Launch template olusturacagiz. Bizim instancelerimize belirli bir kaliba sokuyoruz
- Cloudformation ile launchtemplate arasindaki fark?

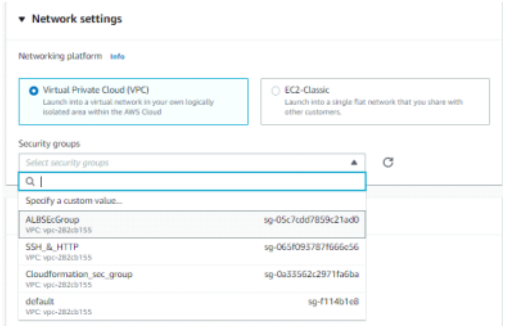


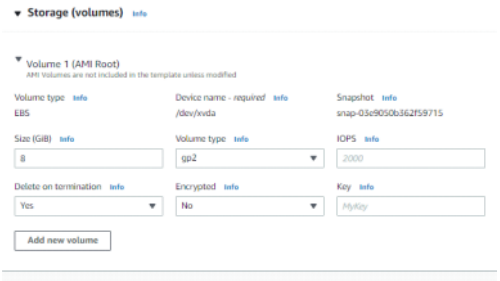


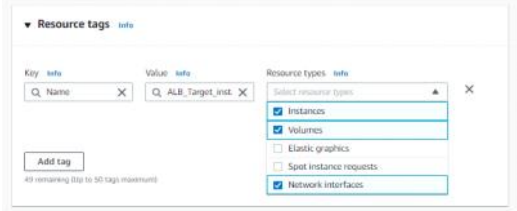




Daha once olusturdugumuz secgroup u seciyoruz



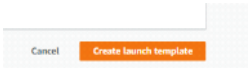




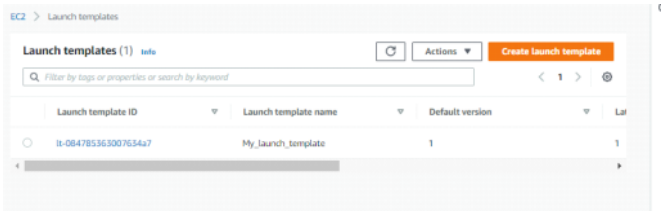
- Userdata (Advanced details icerisine) kismina readme deki kod blogunu yapistiriyoruz
- Html dosyasinda yourname kismina kendi ismimizi yazabiliriz
- Diger ayarlarimiz default olarak kalacak

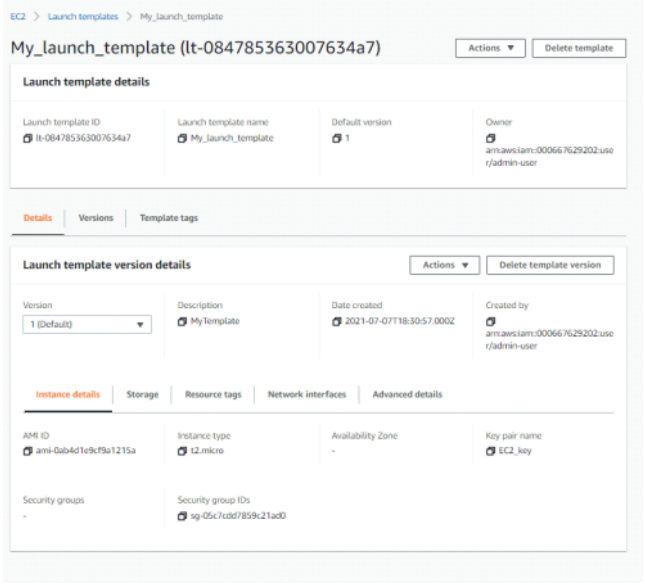
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instancedata-data-retrieval.html ==> metadata hakkinda bilgi

```
#!/bin/bash
#update os
yum update -y
#install apache server
yum install -y httpd
# get private ip address of ec2 instance using instance metadata
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-
metadata-token-ttl-seconds: 21600"` \
&& PRIVATE_IP=`curl -H "X-aws-ec2-metadata-
token: $TOKEN" http://169.254.169.254/latest/meta-data/local-ipv4`
# get public ip address of ec2 instance using instance metadata
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-
metadata-token-ttl-seconds: 21600"` \
&& PUBLIC_IP=`curl -H "X-aws-ec2-metadata-
token: $TOKEN" http://169.254.169.254/latest/meta-data/public-ipv4`
# get date and time of server
DATE_TIME=`date`
# set all permissions
chmod -R 777 /var/www/html
# create a custom index.html file
echo "<html>
<head>
    <title> Congratulations! You have created an instance from Launch Template
</title>
</head>
<body>
    <h1>This web server is launched from launch template by YOUR_NAME</h1>
    <p>This instance is created at <b>$DATE_TIME</b></p>
    <p>Private IP address of this instance is <b>$PRIVATE_IP</b></p>
    <p>Public IP address of this instance is <b>$PUBLIC_IP</b></p>
</body>
</html>" > /var/www/html/index.html
# start apache server
systemctl start httpd
systemctl enable httpd
```

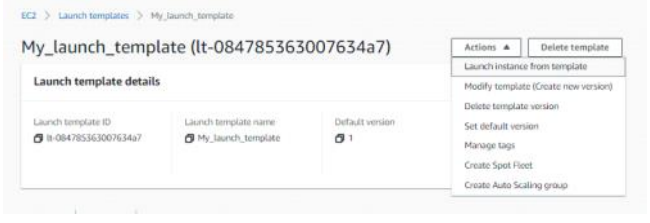


- Olusturdugumuz templatemiz gorselde;

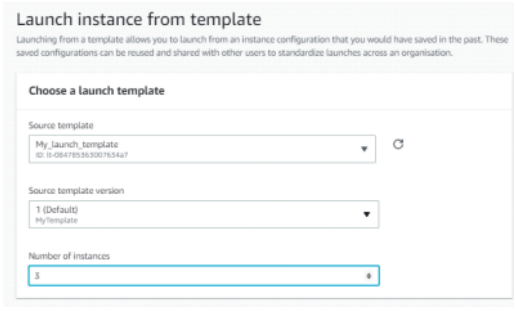


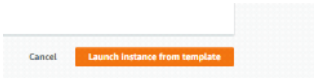


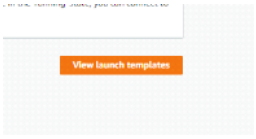
- ACTIONS / LAUNCH ... SECIYORUZ



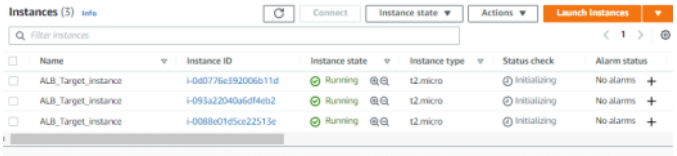
- 3 TEMPLATE OLUSTURACAGIZ



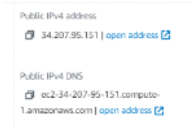


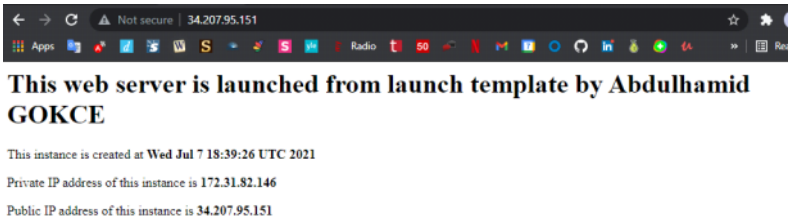


- Olusturduguz 3 adet instance;



- Instance' nin punlic adresini aliyoruz. Ve browsera yapistiriyoruz. Linkte guvenli olmasi https olarak default acilacaktir ve bastaki 's' silinebilir.

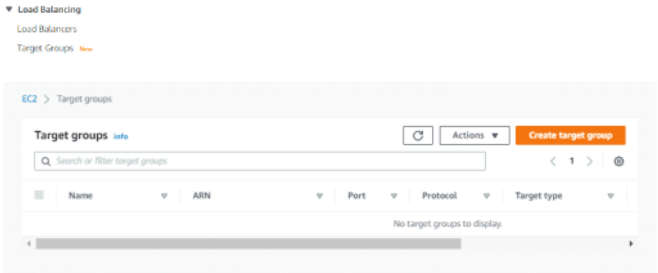


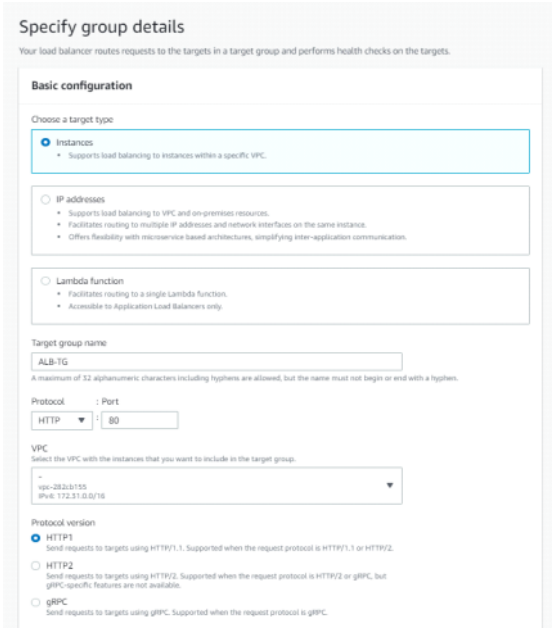


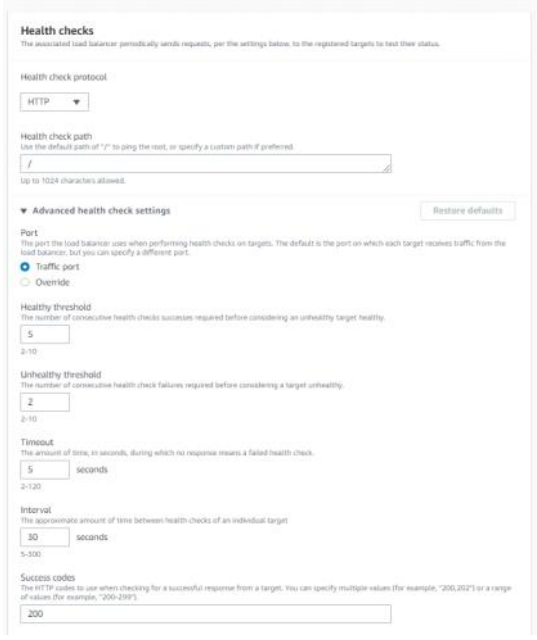
-

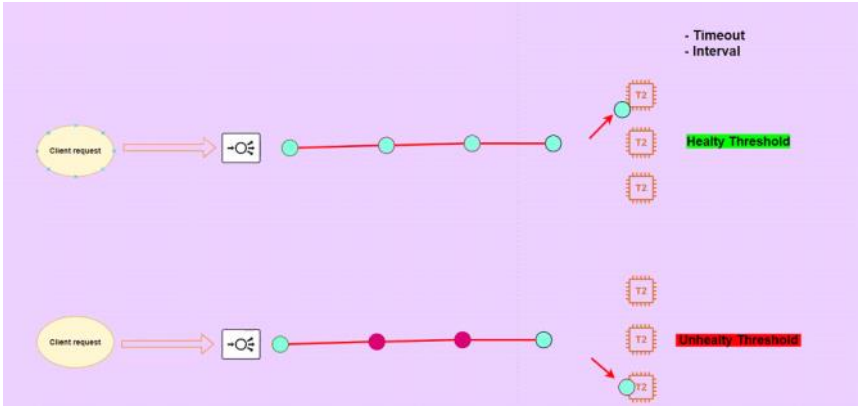
- Bir tane instance'a VS coddan baglanalim asagidaki kodu yapistiralim
  ==> curl http://169.254.169.254/latest/meta-data/
  Bu kod bize ... Gosterir

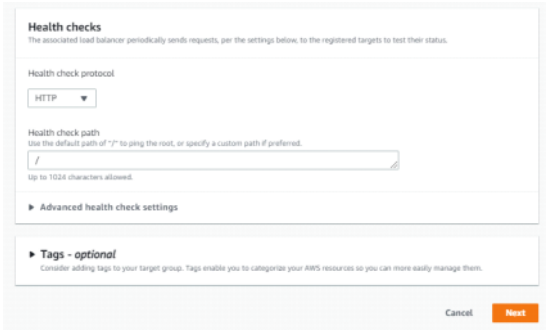
- Olusturdugumuz 3 instance yi target grubu olusturarak icerisine ekleyecegiz. AWS dashboard da gorulebilir.

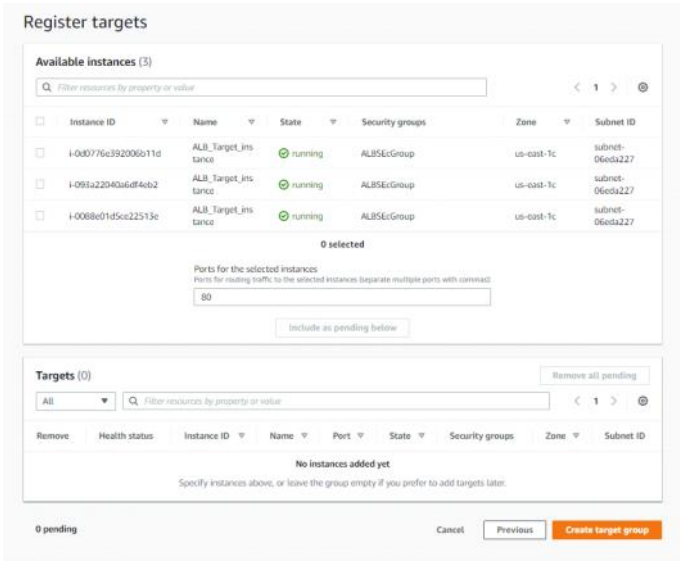



Specify group details
Your load balancer routes requests to the targets in a target group and performs health checks on the targets.
Basic configuration
Choose a target type
Instances
Supports load balancing to instances within a specific VPC.
IP addresses
Supports load balancing to VPC and on-premises resources.
Facilitates routing to multiple IP addresses and network interfaces on the same instance.
Offers flexibility with microservice based architectures, simplifying inter-application communication.
Lambda function
Facilitates routing to a single Lambda function.
Accessible to Application Load Balancers only.
Target group name
ALB-TG
A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.
Protocol
: Port
HTTP
80
VPC
Select the VPC with the instances that you want to include in the target group.
vpc-282cb155
IPv4: 172.31.0.0/16
Protocol version
HTTP1
Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.
HTTP2
Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.
gRPC
Send requests to targets using gRPC. Supported when the request protocol is gRPC.

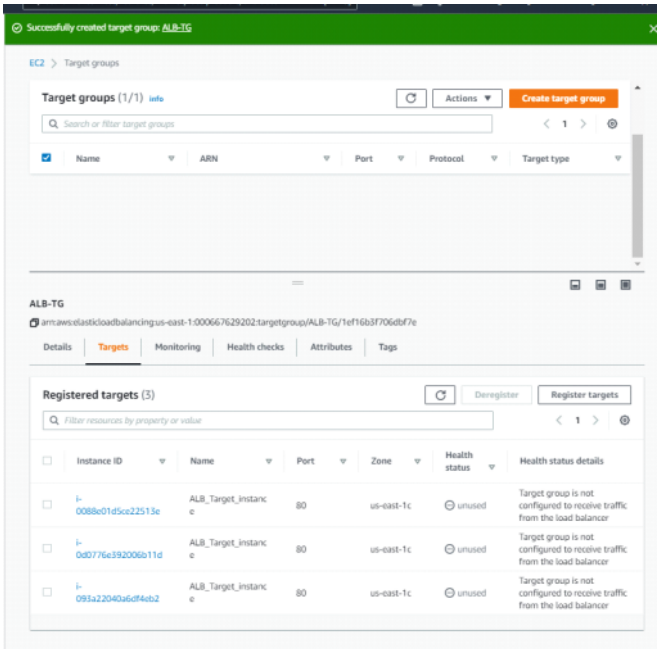

Health checks
The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.
Health check protocol
HTTP
Health check path
Use the default path of "/" to ping the root, or specify a custom path if preferred.
/
Up to 1024 characters allowed.
Advanced health check settings
Restore defaults
Port
The port the load balancer uses when performing health checks on targets. The default is the port on which each target receives traffic from the load balancer, but you can specify a different port.
Traffic port
Override
Healthy threshold
The number of consecutive health checks successes required before considering an unhealthy target healthy.
5
2-10
Unhealthy threshold
The number of consecutive health check failures required before considering a target unhealthy.
2
2-10
Timeout
The amount of time, in seconds, during which no response means a failed health check.
5
seconds
2-120
Interval
The approximate amount of time between health checks of an individual target
30
seconds
5-300
Success codes
The HTTP codes to use when checking for a successful response from a target. You can specify multiple values (for example, "200,202") or a range of values (for example, "200-299").
200


- Timeout
- Interval
Client request
T2
T2
T2
Healty Threshold
Client request
T2
T2
T2
Unhealty Threshold

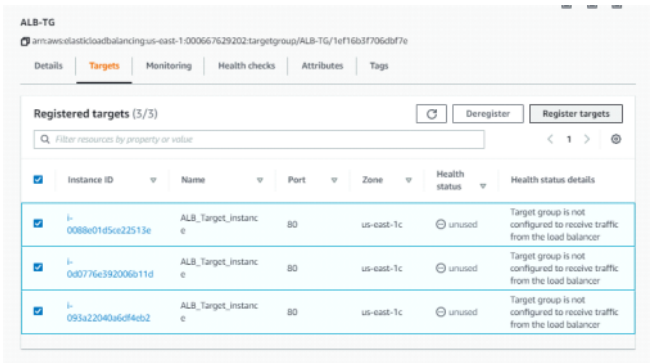




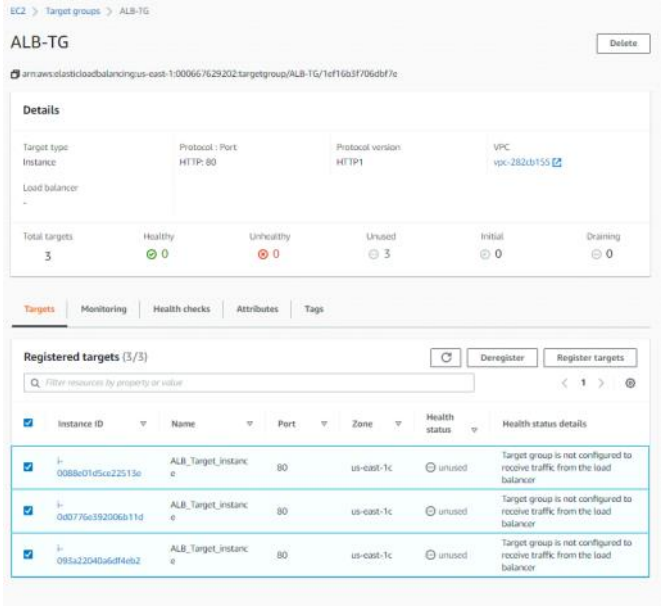
Available instances deki cihazlarin basina tik koyup include as pending below tikla ve sonra create target group tikla



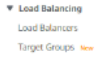
Bu islemle instanceleri target gruplarina eklemis olduk.



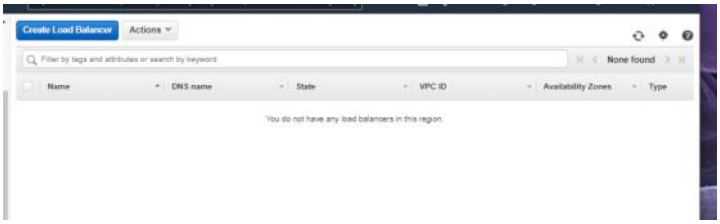
- Secip register targets diyoruz

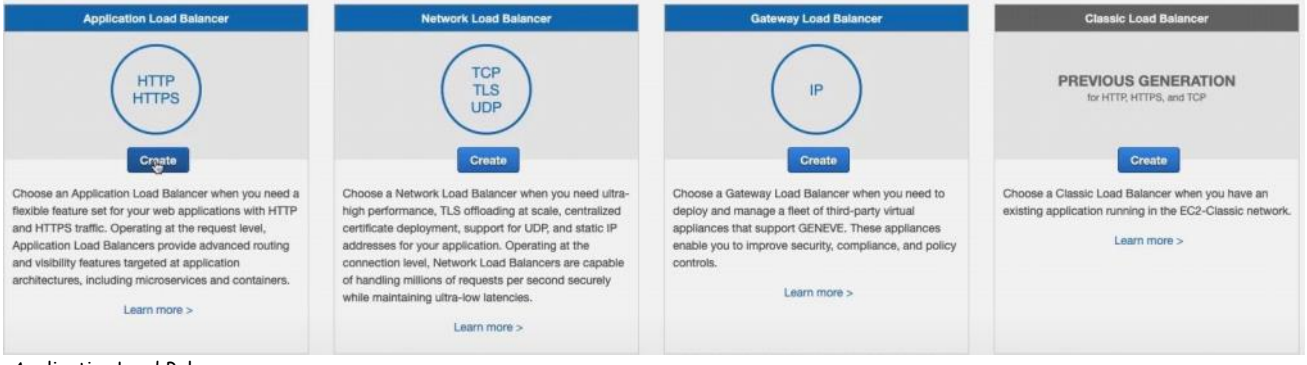


- Target group da hazir simdi loadbalancer olusturacagiz. AWS konsolu icerisindedir.

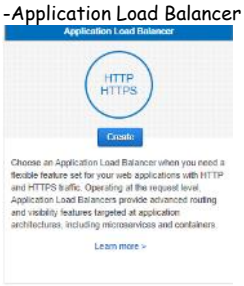


- Create load balancer

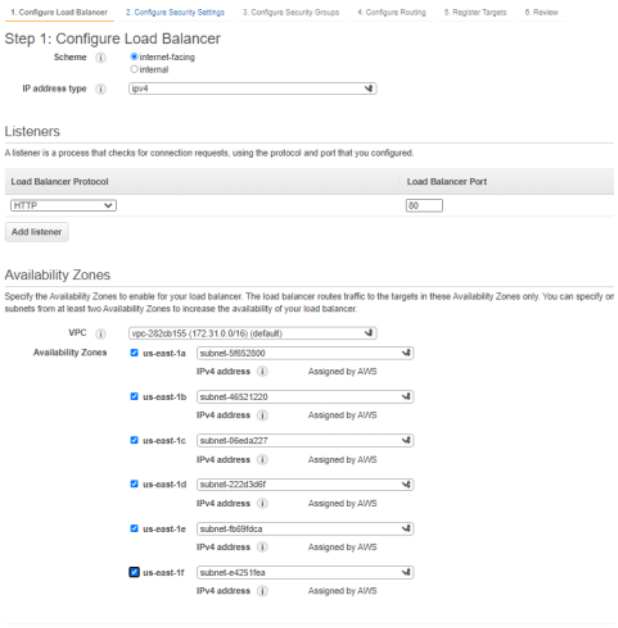




-Application Load Balancer



- Bizim yapacagimiz HTTP uzerinden bir dagitim olacagi icin bu sebeple Application Load Balancer kullanacagiz
- Name atadik (alttaki resimde Scheme bolumunun bir ustunde kalmistir)
- Availability zones bolumundeki zonelarin hepsini seciyoruz



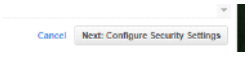
Internet facing ==> eger disardan gelen bir trafigi load balancer in dagitmasini istiyorsak;

Internal ==> bizim yapimiz internal bir yapi ise yani icte bir vpc(virtual private cloud) icerisinde yuk dagitimi istiyorsak

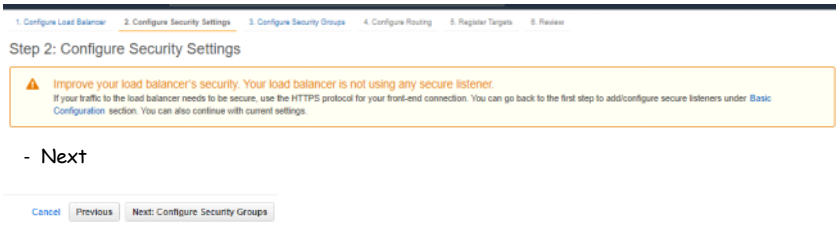
Listeners ==> hangi porttan bizim trafigimiz izlenecek ayrica https sertifikamiz varsa add listener kismindan ekleme yapabiliriz

Aws en az iki tane Availability Zones zorunlu tutmaktadir.

- Next

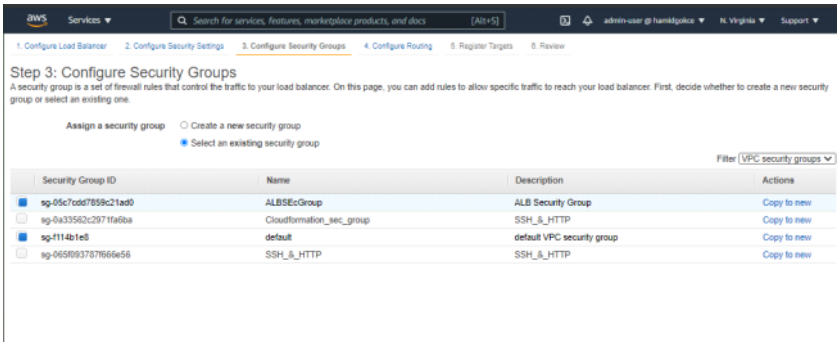


- Guvenli bir listener kullanmadigimiz icin uyari alacagiz. Http ile actigimiz icin uyari metni aldik. Ama https secseydik bu uyari almayacaktik.



- Next

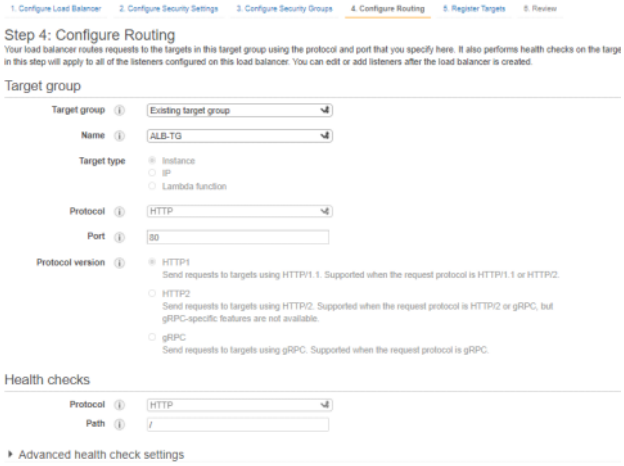
- Daha once olusturdugumuz security group seciyoruz.



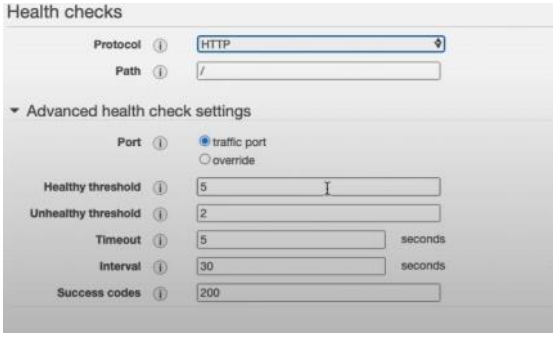
- Next

- Asagida trafigi nasil yonlendirecegimize dair degiskenler girecegiz ve bu sebeple de target grouplar kuracagiz
  - Daha oncesinde bir target group olusturdugumuz icin ALB-TG grubunu seciyoruz. Ama elimizde bir group olmasaydi new

target group secenegi ile yeni bir group olusturabilecektik.
    - Trafigi takip edebilecegimiz 3 secenek sunulmustur biz instanceler uzerinden takip yapacagimiz icin target type e instance yi seciyoruz.
    - Protocol kismi trafigi load balancer yönlendirirken kullandigi protokolu gostermektedir.
    - Health check : Load balancerlarin instancelerin duzgun calisip calismadigini kontrol eden bir fonksiyon. Load balancer target in duzgun calisip calismadigini kontrol eder. Http uzerinden health check yapacagiz. Path kismida health checkin yapilacagi target path dir ve default olarak root gorunmektedir.

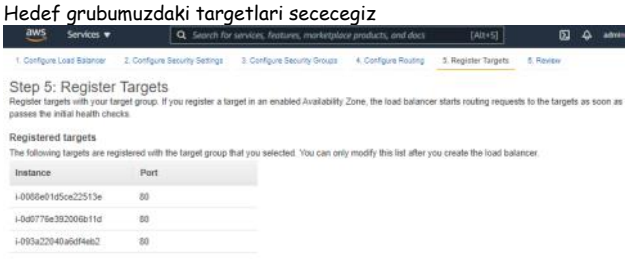


- Healthy threshold ile arka planda target a gonderilecek request sayisidir defaut deger 5 tir. 5 sorgunun hepsi olumlu ise healthy karari verilecektir.
- Bir alt secenekte ise ard arda 2 defa request basarisiz olursa target unhealthy olarak belirlenecektir.
- Timeout : connestion failure durumunda beklenecek sure
- Interval : her deneme arasindaki bekleme suresi
- Success codes : response olarak olumlu sonuc alinirsa 200 kodu denmektedir.

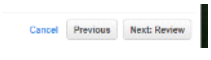


- Next

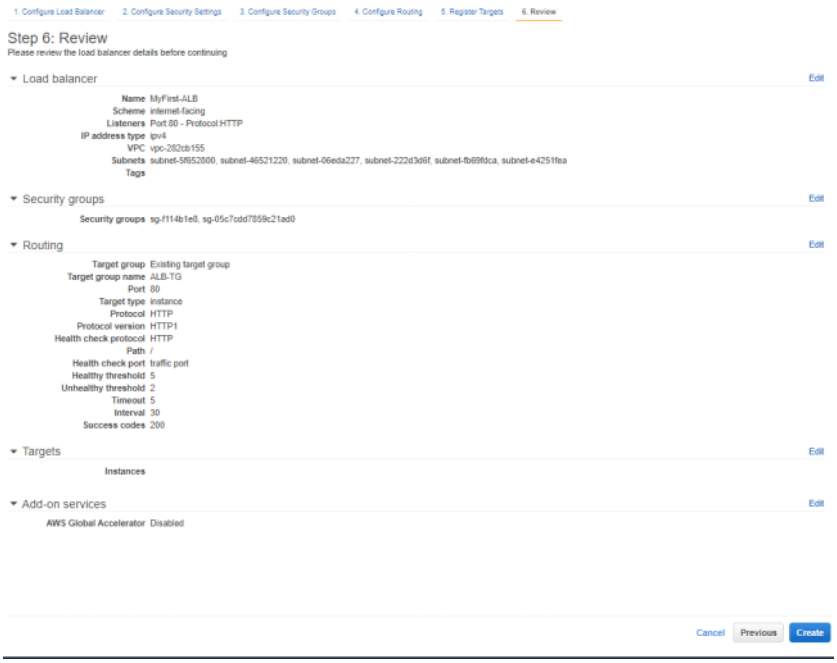
Hedef grubumuzdaki targetlari sececegiz



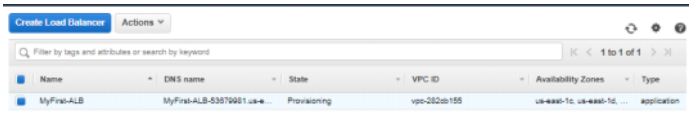
- Next



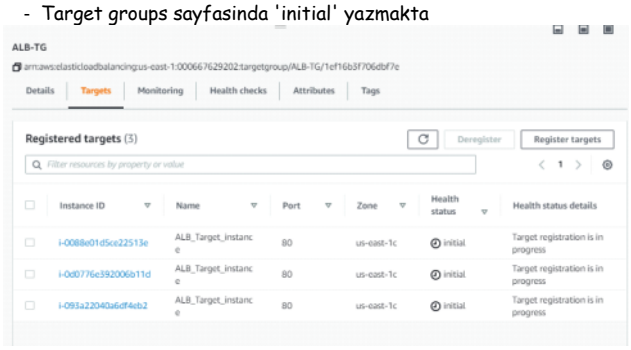
- Create : review sayfamiz



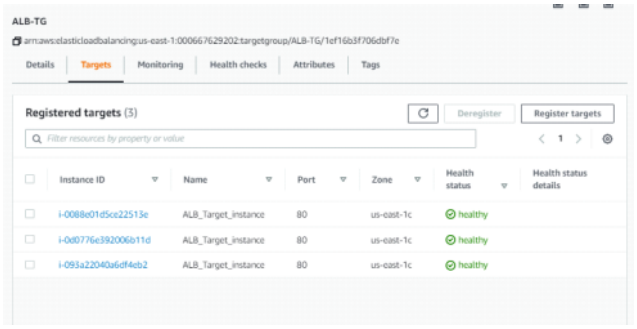
- Olusturuluyor asamasinda
- Road balancer tum targetlar icin bize bir dns name atamaktadir.



- Target groups sayfasinda 'initial' yazmakta



- Ayni sayfada healty gorecegiz



- Uc instance nin DNS i ni kopyalayip browser a yapistiralim

This web server is launched from launch template by Abdulhamid GOKCE

This instance is created at **Wed Jul 7 18:39:28 UTC 2021**

Private IP address of this instance is **172.31.95.15**

Public IP address of this instance is **34.207.126.40**

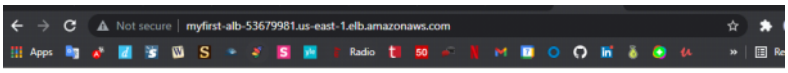
This web server is launched from launch template by Abdulhamid GOKCE

This instance is created at **Wed Jul 7 18:39:29 UTC 2021**

Private IP address of this instance is **172.31.88.125**

Public IP address of this instance is **52.91.206.43**

- DNS i actigimizda yukaridaki gibi uc ayri sayfa acmaya gerek kalmayacaktir ve sayfayi her yenilediğimizde diger sayfalara gececektir



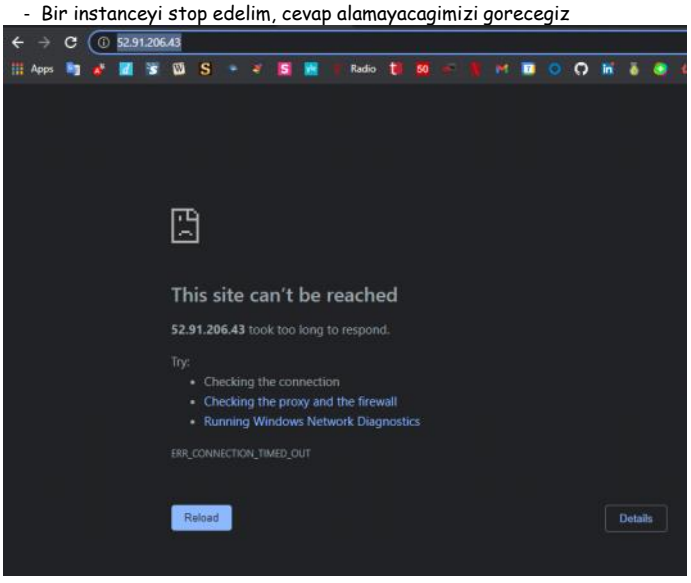
This web server is launched from launch template by Abdulhamid GOKCE

This instance is created at **Wed Jul 7 18:39:29 UTC 2021**

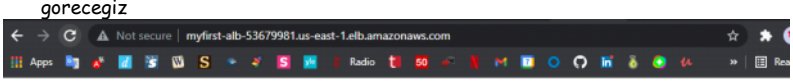
Private IP address of this instance is **172.31.88.125**

Public IP address of this instance is **52.91.206.43**

- Bir instanceyi stop edelim, cevap alamayacagimizi gorecegiz



- Asagidaki iki sayfada iki instance arasina yuku gonderdigini gorecegiz



This web server is launched from launch template by Abdulhamid GOKCE

This instance is created at **Wed Jul 7 18:39:28 UTC 2021**

Private IP address of this instance is **172.31.95.15**

Public IP address of this instance is **34.207.126.40**

This web server is launched from launch template by Abdulhamid GOKCE

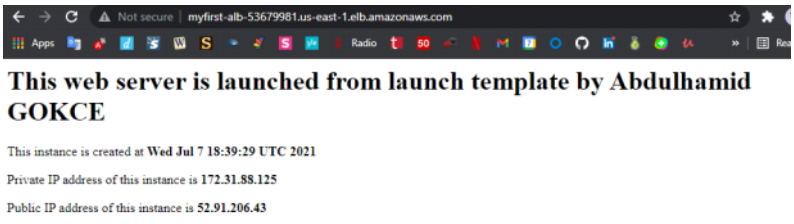
This instance is created at **Wed Jul 7 18:39:26 UTC 2021**

Private IP address of this instance is **172.31.82.146**

Public IP address of this instance is **34.207.95.151**

- Kapattigimiz instance yi tekrar acarsak DNS in calistigini

gorecegiz



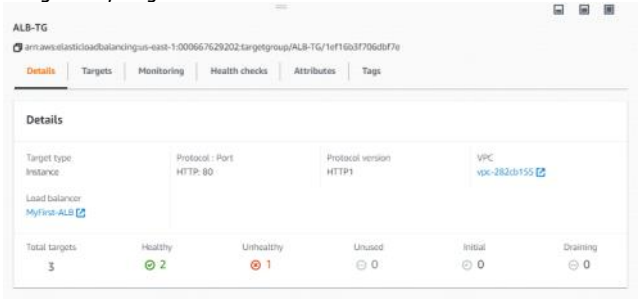
- Instance ye baglanalim ve index.html dosyasini gorelim

```
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-88-125 ~]$ cd /var/www/html
[ec2-user@ip-172-31-88-125 html]$ pwd
/var/www/html
[ec2-user@ip-172-31-88-125 html]$ ls
index.html
[ec2-user@ip-172-31-88-125 html]$ cat index.html
<html>
<head>
    <title> Congratulations! You have created an instance from Launch Template</title>
</head>
<body>
    <h1>This web server is launched from launch template by Abdulhamid GOKCE</h1>
    <p>This instance is created at <b>Wed Jul  7 18:39:29 UTC 2021</b></p>
    <p>Private IP address of this instance is <b>172.31.88.125</b></p>
    <p>Public IP address of this instance is <b>52.91.206.43</b></p>
</body>
</html>
[ec2-user@ip-172-31-88-125 html]$
```

Cihazimizi durdurdugumuzda inaktif pozisyonda oldugunu gorecegiz

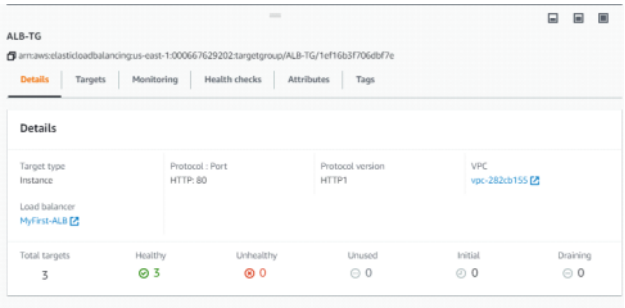
```
[ec2-user@ip-172-31-88-125 html]$ sudo systemctl stop httpd
[ec2-user@ip-172-31-88-125 html]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: inactive (dead) since Wed 2021-07-07 19:46:49 UTC; 1min 2s ago
     Docs: man:httpd.service(8)
  Process: 2798 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND (code=exited, status=0/SUCCESS)
 Main PID: 2798 (code=exited, status=0/SUCCESS)
   Status: "Total requests: 17; Idle/Busy workers 100/0;Requests/sec: 0.0742; Bytes served/sec:  66 B/sec"

Jul 07 19:42:59 ip-172-31-88-125.ec2.internal systemd[1]: Starting The Apache HTTP Server...
Jul 07 19:42:59 ip-172-31-88-125.ec2.internal systemd[1]: Started The Apache HTTP Server.
Jul 07 19:46:48 ip-172-31-88-125.ec2.internal systemd[1]: Stopping The Apache HTTP Server...
Jul 07 19:46:49 ip-172-31-88-125.ec2.internal systemd[1]: Stopped The Apache HTTP Server.
[ec2-user@ip-172-31-88-125 html]$
```

- bu sayede browserlarda da vs coddan kapattigimiz makinayi goremeyecegiz.

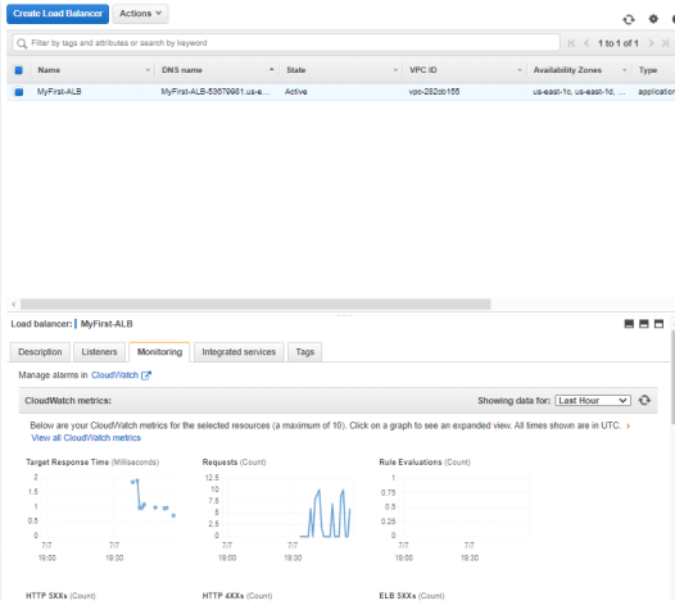


- Makinayi calistirdigimizda tekrar healty konumuna geldigini gorecegiz

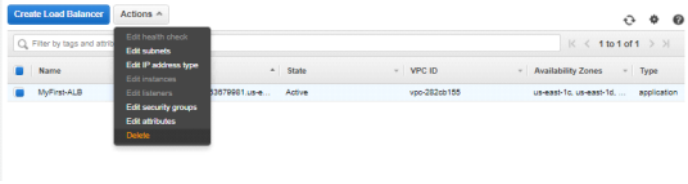
```
[ec2-user@ip-172-31-88-125 html]$ sudo systemctl start httpd
[ec2-user@ip-172-31-88-125 html]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2021-07-07 19:50:46 UTC; 4s ago
     Docs: man:httpd.service(8)
 Main PID: 3263 (httpd)
   Status: "Processing requests..."
   CGroup: /system.slice/httpd.service
           ├─3263 /usr/sbin/httpd -DFOREGROUND
           ├─3264 /usr/sbin/httpd -DFOREGROUND
           ├─3265 /usr/sbin/httpd -DFOREGROUND
           ├─3266 /usr/sbin/httpd -DFOREGROUND
           ├─3267 /usr/sbin/httpd -DFOREGROUND
           └─3268 /usr/sbin/httpd -DFOREGROUND

Jul 07 19:50:46 ip-172-31-88-125.ec2.internal systemd[1]: Starting The Apache HTTP Server...
Jul 07 19:50:46 ip-172-31-88-125.ec2.internal systemd[1]: Started The Apache HTTP Server.
```



- Monitoring bolumunden kapali olan cihazimizin genel durumu hakkinda bilgi alabiliyoruz



- Actigimiz
  - Sec group ==> free
  - Launch template ==> free
  - Target group ==> free
  - Elastic load balancer ==> ucretli ve silinme sekli gorseldeki gibidir



- Instance leri de terminate edebiliriz

https://docs.nginx.com/nginx/admin-guide/load-balancer/http-load-balancer/

https://medium.com/dazn-tech/aws-application-load-balancer-algorithms-765be2eca158

https://aws.amazon.com/elasticloadbalancing/features/

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instancedata-data-retrieval.html