Project-503 : Blog Page Application (Django) deployed on AWS Application Load Balancer with Auto Scaling, S3, Relational Database Service(RDS), VPC's Components, Lambda, DynamoDB and Cloudfront with Route 53

Description

The Clarusway Blog Page Application aims to deploy blog application as a web application written Django Framework on AWS Cloud Infrastructure. This infrastructure has Application Load Balancer with Auto Scaling Group of Elastic Compute Cloud (EC2) Instances and Relational Database Service (RDS) on defined VPC. Also, The Cloudfront and Route 53 services are located in front of the architecture and manage the traffic in secure. User is able to upload pictures and videos on own blog page and these are kept on S3 Bucket. This architecture will be created by Firms DevOps Guy.

Problem Statement

- **Your company has recently ended up a project that aims to serve as Blog web application on isolated VPC environment. You and your colleagues have started to work on the project. Your Developer team has developed the application and you are going to deploy the app in production environment.**
- **Application is coded by Clarusway Fullstack development team and given you as DevOps team. App allows users to write their own blog page to whom user registration data should be kept in separate MySQL database in AWS RDS service and pictures or videos should be kept in S3 bucket. The object list of S3 Bucket containing movies and videos is recorded on DynamoDB table.**
- **The web application will be deployed using Django framework.**
- **The Web Application should be accessible via web browser from anywhere in secure.**
- **You are requested to push your program to the project repository on the Github. You are going to pull it into the webservers in the production environment on AWS Cloud.**

In the architecture, you can configure your infrastructure using the followings,

- **The application stack should be created with new AWS resources.**
- **Specifications of VPC:**
  - **VPC has two AZs and every AZ has 1 public and 1 private subnets.**
  - **VPC has Internet Gateway**
  - **One of public subnets has NAT Instance.**
  - **You might create new instance as Bastion host on Public subnet or you can use NAT instance as Bastion host.**
  - **There should be managed private and public route tables.**
  - **Route tables should be arranged regarding of routing policies and subnet associations based on public and private subnets.**
- **You should create Application Load Balancer with Auto Scaling Group of Ubuntu 18.04 EC2 Instances within created VPC.**
- **You should create RDS instance within one of private subnets on created VPC and configure it on application.**
- **The Auto Scaling Group should use a Launch Template in order to launch instances needed and should be configured to;**
  - **use all Availability Zones on created VPC.**
  - **set desired capacity of instances to 2**
  - **set minimum size of instances to 2**
  - **set maximum size of instances to 4**
  - **set health check grace period to 90 seconds**
  - **set health check type to ELB**
  - **Scaling Policy --> Target Tracking Policy**
    - **Average CPU utilization (set Target Value %70)**
    - **seconds warm up before including in metric ---> 200**
    - **Set notification to your email address for launch, terminate, fail to launch, fail to terminate instance situations**
- **ALB configuration;**

        - Application Load Balancer should be placed within a security group which allows HTTP (80) and HTTPS (443) connections from anywhere.
        - Certification should be created for secure connection (HTTPS)
            - To create certificate, AWS Certificate Manager can be utilized.
        - ALB redirects to traffic from HTTP to HTTPS
        - Target Group
            - Health Check Protocol is going to be HTTP
    - The Launch Template should be configured to;
        - Prepare Django environment on EC2 instance based on Developer Notes,
        - Download the "clarusway_aws_capstone" folder from Github repository,
        - Install the requirements using requirements.txt in 'clarusway_aws_capstone' folder
        - Deploy the Django application on port 80.
        - Launch Template only allows HTTP (80) and HTTPS (443) ports coming from ALB Security Group and SSH (22) connections from anywhere.
        - EC2 Instances type can be configured as t2.micro.
        - Instance launched should be tagged Clarusway AWS Capstone Project
        - Since Django App needs to talk with S3, S3 full access role must be attached EC2s.
    - For RDS Database Instance;
        - Instance type can be configured as db.t2.micro
        - Database engine can be MySQL with version of 8.0.20.
        - RDS endpoint should be addressed within settings file of blog application that is explained developer notes.
        - Please read carefully "Developer notes" to manage RDS sub settings.
    - Cloudfront should be set as a cache server which points to Application Load Balance with following configurations;
        - The cloudfront distribution should communicate with ALB securely.
        - Origin Protocol policy can be selected as HTTPS only.
        - Viewer Protocol Policy can be selected as Redirect HTTP to HTTPS
    - As cache behavior;
        - GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE methods should be allowed.
        - Forward Cookies must be selected All.
        - Newly created ACM Certificate should be used for securing connections. (You can use same certificate with ALB)
    - Route 53
        - Connection must be secure (HTTPS).
        - Your hostname can be used to publish website.
        - Failover routing policy should be set while publishing application
            - Primary connection is going to be Cloudformation
            - Secondary connection is going to be a static website placed another S3 bucket. This S3 bucket has just basic static website that has a picture said "the page is under construction" given files within S3_static_Website folder
            - Healthcheck should check If Cloudfront is healthy or not.
    - As S3 Bucket
        - First S3 Bucket
            - It should be created within the Region that you created VPC
            - Since development team doesn't prefer to expose traffic between S3 and EC2s on internet, Endpoint should be set on created VPC.
            - S3 Bucket name should be addressed within configuration file of blog application that is explained developer notes.
        - Second S3 Bucket

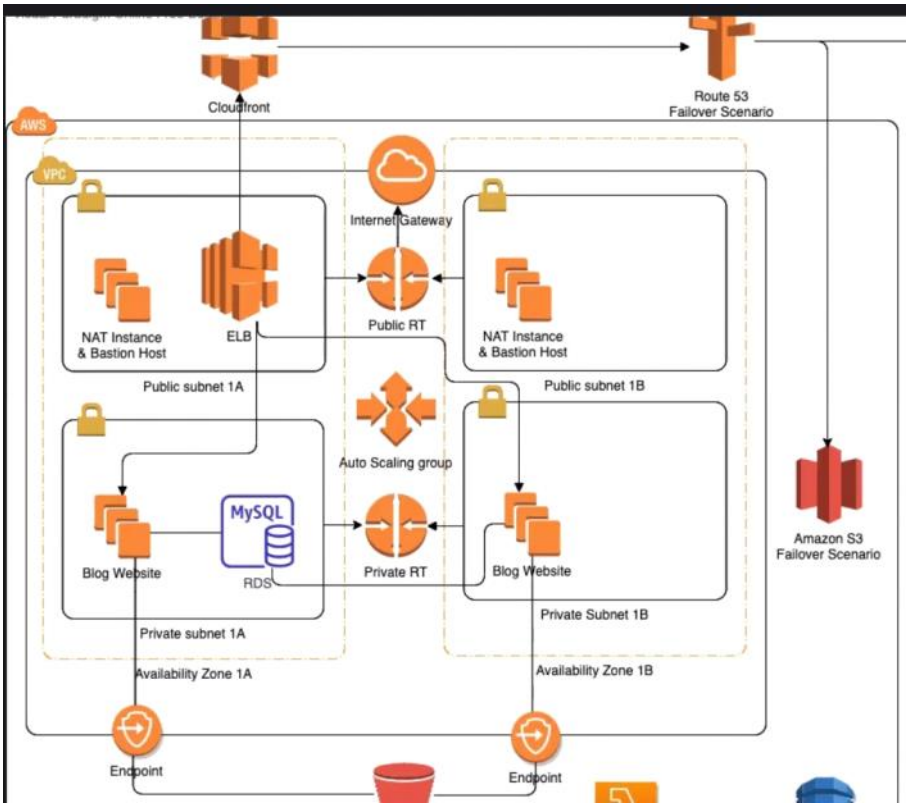
        - This Bucket is going to be used for failover scenario. It has just a basic static website that has a picture said "the page is under construction"
    - To write the objects of S3 on DynamoDB table
        - Lambda Function
            - Lambda function is going to be Python 3.8
            - Python Function can be found in github repo
            - S3 event is set as trigger
            - Since Lambda needs to talk S3 and DynamoDB and to run on created VPC, S3, DynamoDB full access policies and NetworkAdministrator policy must be attached it
            - S3 Event must be created first S3 Bucket to trigger Lambda function
        - DynamoDB Table
            - Create a DynamoDB table which has primary key that is id
            - Created DynamoDB table's name should be placed on Lambda function.

Project Skeleton

```
clarusway_blog_proj (folder)
|
|----Readme.md               # Given to the students (Definition of the project)
|----src (folder)           # Given to the students (Django Application's )
|----requirements.txt       # Given to the students (txt file)
|----lambda_function.py     # Given to the students (python file)
|----developer_notes.txt    # Given to the students (txt file)
```

Expected Outcome

From < https://github.com/clarusway/clarusway-aws-8-21/tree/main/aws/projects/Project-503-Capstone-Project-Blog-Page-App-(Django)-on-AWS-Environment>



(Janko flask in gelimis hali diyebiliriz bir framworktur.)

Developerlar bir jango framworku ile block aplicationu yazdilar. Blok sayfasi ile herkes kendi blogunu sayfasini yazabiliyor. Fotograflarini ve videolarini paylasabiliyor.

Videolarini paylasacaklari icin bi login olunmasi ve bir yerde de user ve passwordlerin tutulmasi store edilmes gerekiyor.

Store denilince aklimiza AWS de s3 geliyor. Developerlar s3 de tutulmasini istiyorlar. S3 obje tabanli bir storeage idi. Ne kadar cagirilir ise o kadar ucretlendirmeye tabiydi.

Login datalarda database de tutulacak. Bunun icinde Mysql tercih edilmis .

Bunun icin Custom VPC kuracagiz . Production ortaminda da custom vpc kullanilmasi tercih ediliyor. Cunku VPC lerde Database gibi bazi makinalrimizin private subnetlerde kurulmasi gerekiyor ki bir bakima bunlar bierer firewal olarak nitelendiriliyor.

Custom VPC nin isterlerine bakalim;

- **2 adet Availibity zonumuz var ve bu Availibility Zon un icerisinde bir private ve birde public subnetlerimiz olacak (Availibity Zone : Bir region içerisinde bulunan, birbirlerine yakın konumlandırılmış, küçük sunucu tarlalarına verilen isimdir. Bir region içerisinde birden fazla availability zone bulunabilir ve herbir zone harf ile adlandırılır. Örneğin Oregon region'unda us-west-2a, us-west-2b ve us-west-2c olmak üzere üç adet AZ (availability-zone) bulunmaktadır.**
  From < https://www.mobilhanem.com/aws-dersleri-aws-nedir/ >)

Private ve public subnetler vpc icerisine kurulan izole network çözumleridir.

- **Bi izole network icerisine farkli resourcelari koruma altina alabiliyoruz.**
- **Ortak bir subnet kumesini bir subnet gurubunda toplayabiliyorsunuz.**
- **Ayrica farkli subnetlerde farkli replicalar bulundurarak failover durumlarinda afet yangin olasi tabi durumlarda mevcut subnetmize bir sey olmasi durumunda diger subneti devreye sokabiliyoruz.**

Subnetler in kendi aralarinda irtibati kurmalari icn veyahut subnetlerin icerisinde ki resourcelarin dis dunyay ya ulasmak istediklerinde, veya her hangi bir porta ulasmak

istediklerinde, Rote Tablet I kullaniriz.Route Table ler subnet icerisinde bir novigasyon cihazidir. Rota Tableler icerisinde olusturdugumuz Rule lar ile bir tarafa gitmesini veyahut da gitmemesini saglariz.

Özellikle de private Route Tablet yapilmasinin sebebi bir firewal yapilarak dis dunyadan ulasilamayacak bir hale girmesini saglamaktir.

Ayrica videolar ve resimlerin s3 de saklanacagini ve dis dunyay ya expoz olmsaini istemedigimizi dusunursek subnetler ile S3 bucket lar arsinda da bir gateway end point kuracagiz

Ardindan Jango web sitemiz bir EC2 lar gurubu uzerinde tutulacak bunuda Autoscaling group saglayacak. Bunun öncesi de bir Launch Template kuracagiz ki L .Template ile Auto Scalinge makinalrin kurulumunu salayalim . Bizim istedigimiz desire kapasitede minimum ve max seviyede makinalar ayaga kalkacak . Burda bir Target Tracing Policy olusturacagiz.

Ayni zaman bu Ec2 lar ELB ye bagli olacak ELB nin en önemli özelligi EC2 lardaki yoguniuga bakip HelatCheck ler ile her hangir yoguniuk olmasi durumunda esit olarak bu yuku EC2 lar arasinda dagitmak. Hem Trafigi yayiniiyor hem tek elden dagilimi sagliyor. Hemde Dis dunyadan gelen User in her hangi bir aksama olmadan EC2 ya ulasilmasini sagliyor.

ALB nin de önune bir CloudFront koyacagiz ki bu da br cash hizmeti sunacak. Videolar ve fotograflar S3 den cekilecek her seferinde ucret ödenmemesi icin bunu yapacaklar Cash hizmeti ile sagalanacak.

Production oirtaminda Cloudfront s3 un önune de kurulabiliyor Eger burda tanimlamak istersek s3 u orjin olarak gösterecektir.

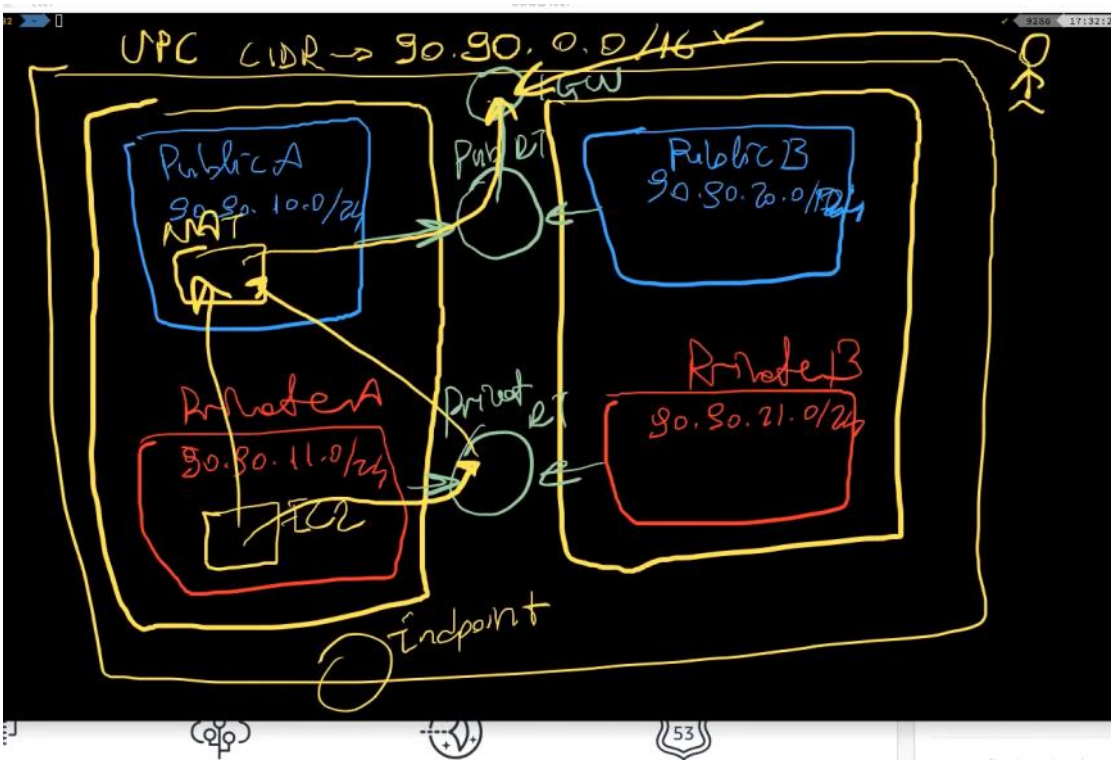
Developerlar kodunu kullanici karsisina cikartmak icin Route53 failover senaryosu ile karsimiza cikacak.

Burada Failover senaryosu Route 53 olarakta bir ister atanmis durumdadir. Route53 de iki adet ent pointi takip eder. Birisi static website olur. Degiride uniq olarak caalisan web sitemizdir.basina herhangi bir durum gelmesi halinde secondary e trafigi yönlendirir.

Userlar bu haliyle karsilarina bir sey cikarki mevcut olarak bir sorunun oldugunu ve bu sorunun bir sure sonra cözelecegi gibi seyler icerebilir.

Birde Deevelopeiatr S3 un icerisine konulan her seyi takip etmek istiyorlar. S3 e konan her hangi bir obje oldugunda , bir event olusacak bu event bir lambda function olusturacak bu Lambda Functon da Dynamodb icerisine bilgileri yazdiracak.

Dolayisiyla yazilan ve silinen bilgileri Dynamodb de grebilecegiz.



Konsolumuza geciyoruz :

Vpc de CIDR Bloklari vardir. Biz bunlari Private ipler icin tanimliyoruz. Public oalnlari EC2 kendisi veriyor.

Tanimlyacagimiz vpc de AWS 90.90.0.0/16 toplamda 65 bin ip tanimlayabilecegimiz BIR SINIRLAMA TANIYOR

Bir subneti private yada Public yapan disariya acilmasidir.

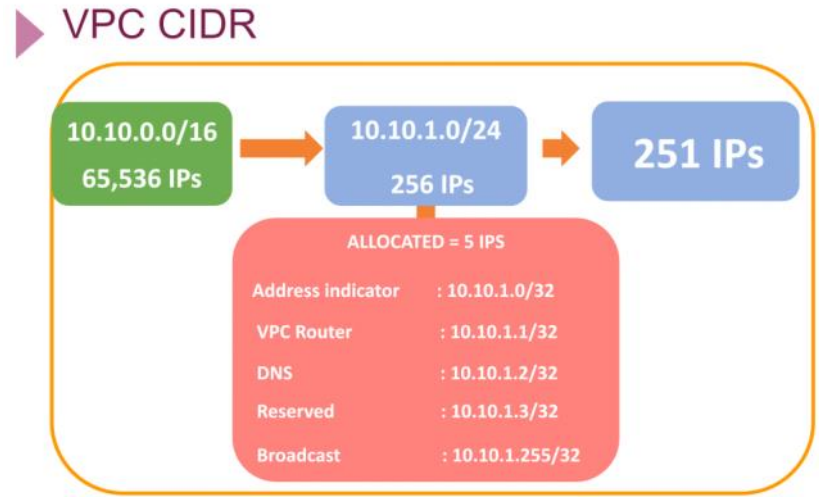
Birde bunlara Private ve Public Rote tablet ayarlayacagiz ki Routa tablelere bir kural tanimlar ve dis dunnya ya acilmasini isterseniz Public istemez iseniz Private olurlar

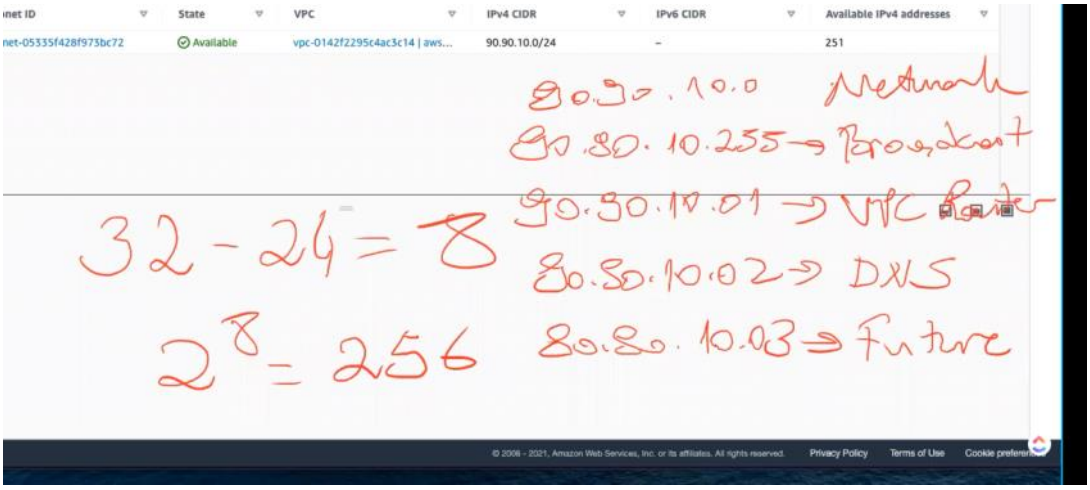
Ayrica Bir de end Point ekleyecegiz S3 ler ve DYNAMODB Cloud Watch lar icin kullanilir ve burada s3 e irtibat saglamk iclin kullanacagiz.

Biz Internet Gate way araciligiyla Öncelikle Public Iplerimize giris yapabiliyoruz. Ancak Private gecis yapmak icin Publiclede kuracagimiz Buston Hostlar ile sibiama tahtalari ile Gecis yapacagiz.

Ayrica icerdeki makinanin disariyla acilmasi icin ne yapmamiz gerekiyordu dersek bunun icinde Nat instance veýahut nat Gateway kurmamiz gerekiyor Bastion Host oarak da göruluyor Ancak Nat Instance da dikkat edilmesi gereken konu http, HTTPS ve SSh anywhere olarak her yerden acmamiz gerekiyor

Ayrica birde icerdeki makinanin disardaki ile görusmesi icin ne yapmamiz gerekiyor.; Nat instance veyahut Nat Gateway kurmaiz gerekiyor NatGateway AWS manage bir sistemdir. Yani biz kontrol etmiyorduk. Nat instance ise özel bir instance tipidir. Public Subnete yerlestiririz. Private instance da bulunan EC2 muz Private Routa Table araciligiyla Nat instance uzerinden dis dunya ile irtibat saglayabiliyoruz. Ayrica Bastion Host olarakta kullanilabilinir ancak HTTP,HTTPS ve ssh in anywere olarak her yöne acilmasi gerekiyor.

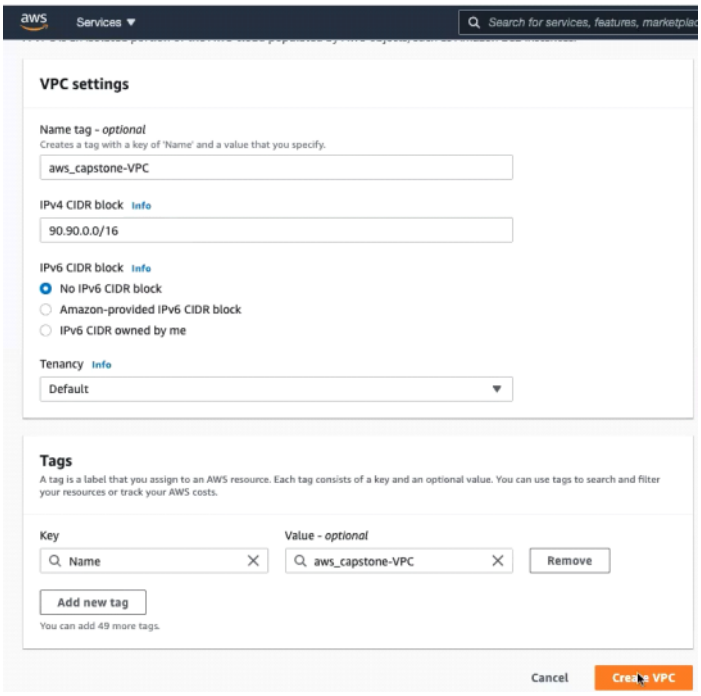




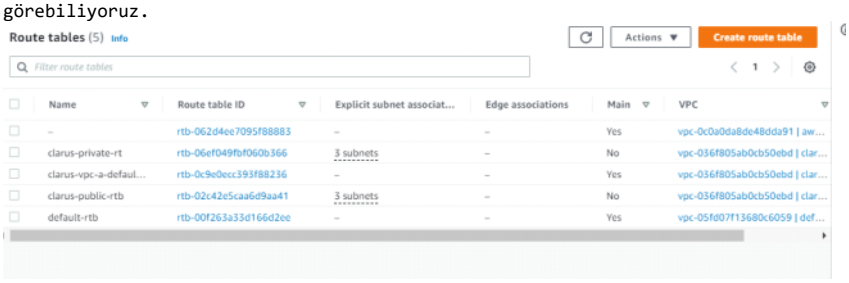
256 - 1 = 255 kullanilabilir 5 ntanesini AWS kendisine Reeverse eder.

Ilk olarak VPC mizi yapmak icin konsola gidiyoruz

```
# Project-503 : Blog Page Application (Django) deployed on AWS Appl
ication Load Balancer with Auto Scaling, S3, Relational Database Se
rvice(RDS), VPC's Components, DynamoDB and Cloudfront with Route 53
 (STUDENT_SOLUTION)
## Description

The Clarusway Blog Page Application aims to deploy blog application
 as a web application written Django Framework on AWS Cloud Infrast
ructure. This infrastructure has Application Load Balancer with Aut
o Scaling Group of Elastic Compute Cloud (EC2) Instances and Relati
onal Database Service (RDS) on defined VPC. Also, The Cloudfront an
d Route 53 services are located in front of the architecture and ma
nage the traffic in secure. User is able to upload pictures and
videos on own blog page and these are kept on S3 Bucket. Th
is architecture will be created by Firms DevOps Guy.


# Steps to Solution

### Step 1: Create dedicated VPC and whole components

    ### VPC
    - Create VPC.
        create a vpc named `aws_capstone-
VPC` CIDR blok is `90.90.0.0/16`
        no ipv6 CIDR block
        tenancy: default
    - select `aws_capstone-
VPC` VPC, click `Actions` and `enable DNS hostnames` for th
e `aws_capstone-VPC`.
```



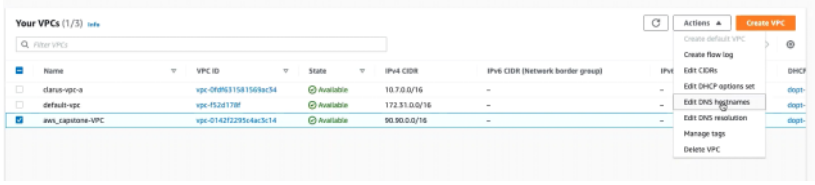
vpc ile birlikte route tablet in default olarak olustugunu görebiliyoruz.



Buna da public Routa tablet deyip bir taNE DE PRIVATE OLUSTurabiliriz

Vpc icerisinde butun resource slarin bir birleri ile ,konusmasini istiyorsak edit DNS Hostaname den ENABLE YAPIYORUZ

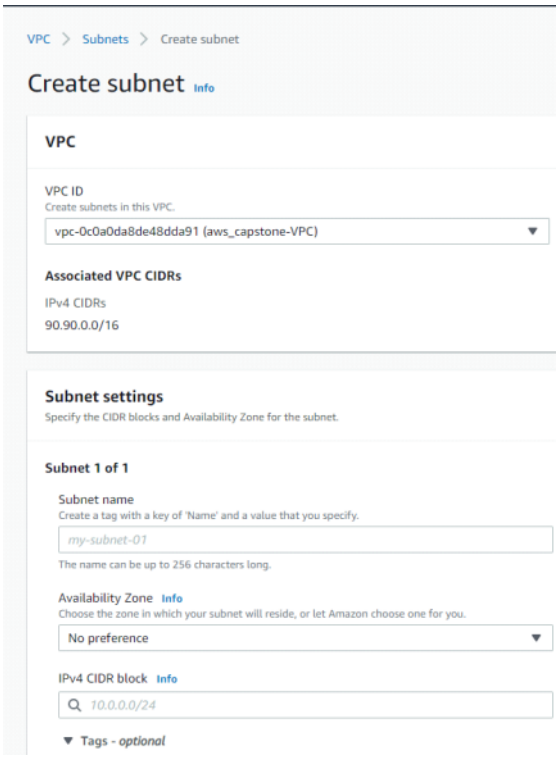
Enable olmaz is e bu vpc icerrisine bir DNS atanmaz ve bir bir leri ile konusmalari zora girer.



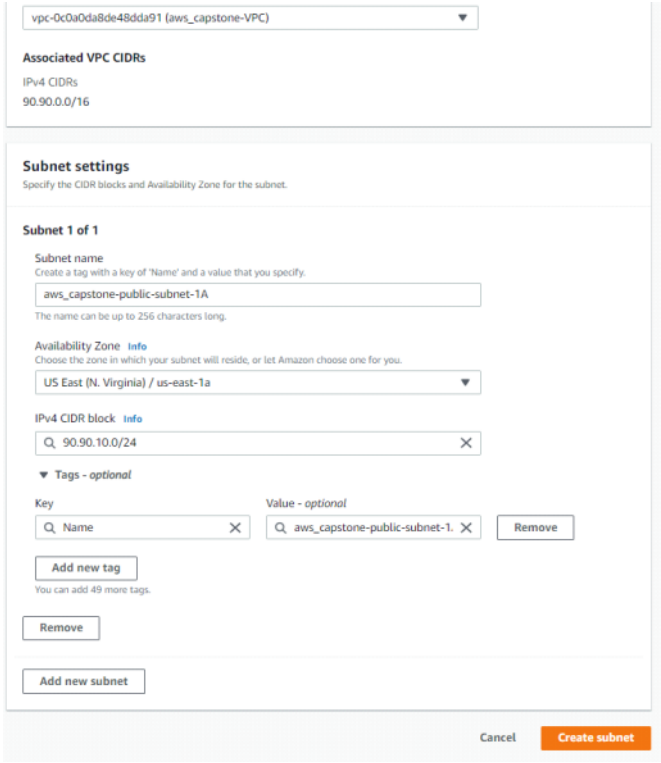
Simdid e subnet olusturacagiz

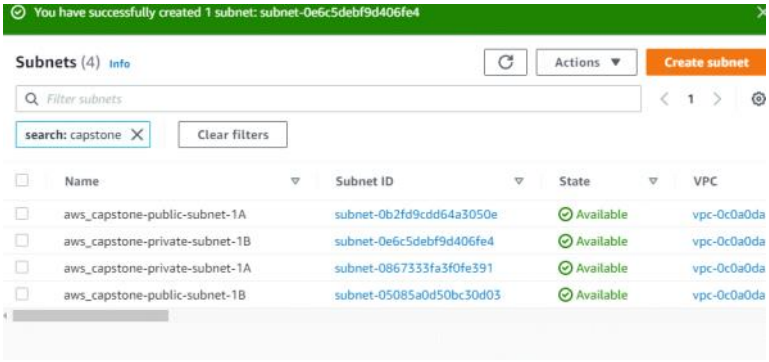
Subnets
- Create Subnets
    - Create a public subnet named `aws_capstone-public-subnet-1A` under the vpc aws_capstone-VPC in AZ us-east-1a with 90.90.10.0/24
    - Create a private subnet named `aws_capstone-private-subnet-1A` under the vpc aws_capstone-VPC in AZ us-east-1a with 90.90.11.0/24
    - Create a public subnet named `aws_capstone-public-subnet-1B` under the vpc aws_capstone-VPC in AZ us-east-1b with 90.90.20.0/24
    - Create a private subnet named `aws_capstone-private-subnet-1B` under the vpc aws_capstone-VPC in AZ us-east-1b with 90.90.21.0/24
- Set `auto-assign IP` up for public subnets. **Select each public subnets and click Modify "auto-**

assign IP settings" and select "Enable auto-assign public IPv4 address"

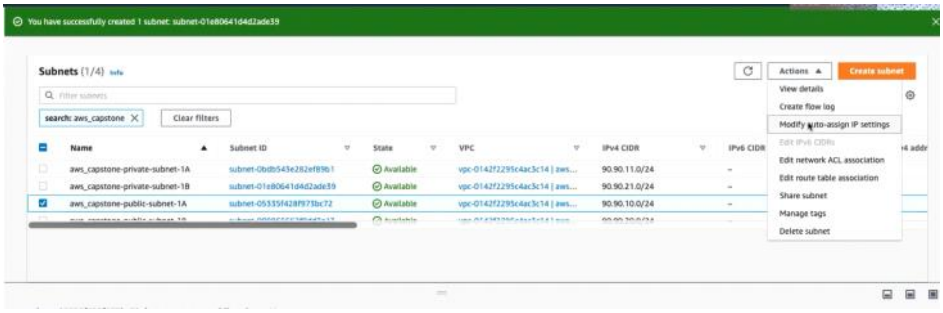


Vpc Id olarak daha önce olusturdugumuz vpc yi seciyoruz.





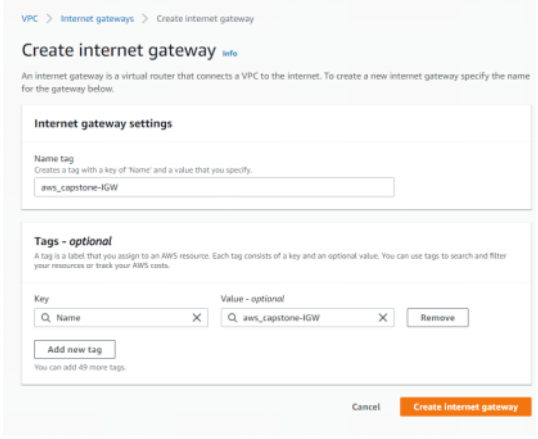
Public Subnetler icin Public ip atanmasi gerekiyor. Bunu bizde kontroledebiliyoruz Bunun icin Public Ipleri Actions bölumunden Modify deyip daha sonra enable ediyoruz.



Subnetleri Private ve Public Yapmak icin Route Tabletleri de duzenleyecegiz

Public routa Tabletleri Public yapmak icin interne Gate Way tanimlayacagiz

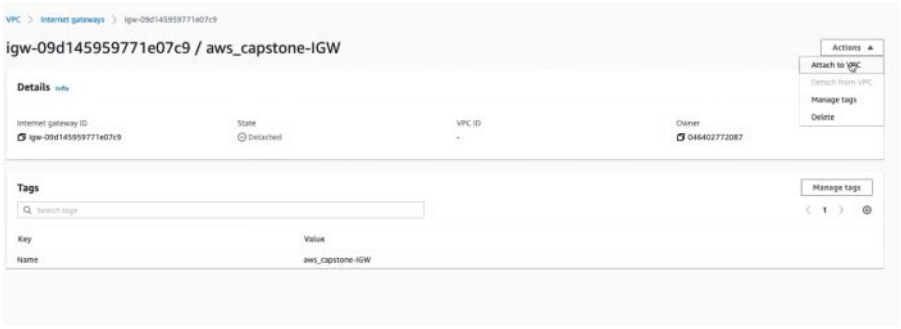
```
## Internet Gateway
    - Click Internet gateway section on left hand side. Cre
ate an internet gateway named `aws_capstone-
IGW` and create.
```



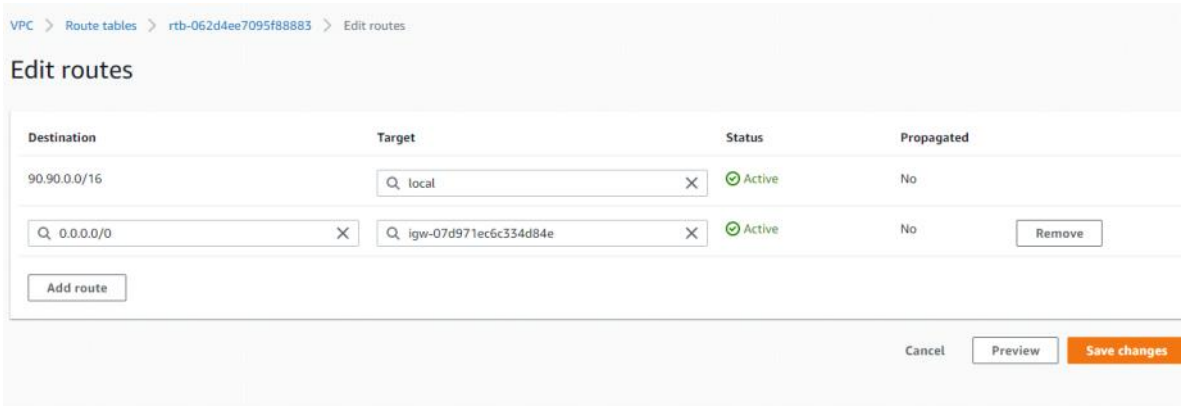

```
    - ATTACH the internet gateway `aws_capstone-
IGW` to the newly created VPC `aws_capstone-
VPC`. Go to VPC and select newly created VPC and click acti
on ---> Attach to VPC ---> Select `aws_capstone-VPC` VPC
```

Internet Gateway olusturduk ancak bunu attach etemeiz gerekiyor su an Detach olarak gözukuyor

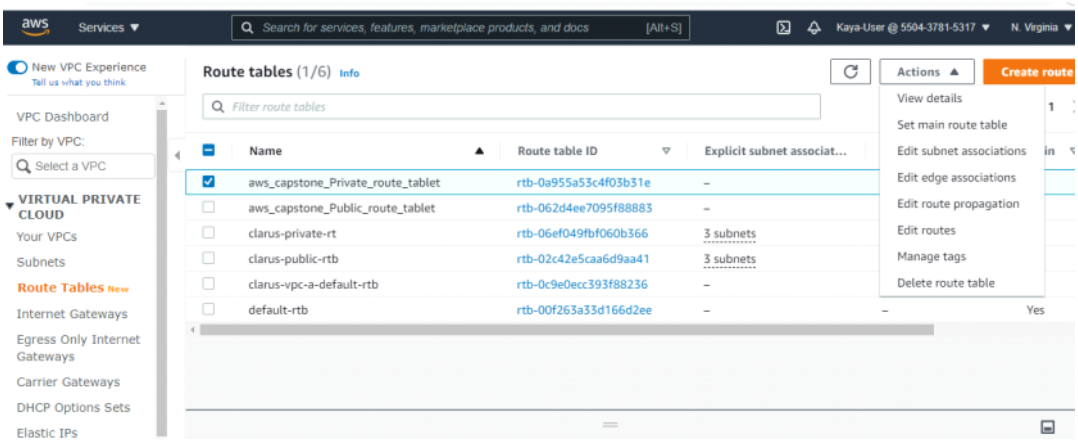
Her bir VPC ye Ancak bir InternetGateWay Attach edilebiliyor.



Son olarak Route Tablelerimize internet Gate Way i ekliyoruz.



Ve ardindan Private Routa Tablelar ile Private Subnetleri, Public leri Public olan ile Associations edecegiz

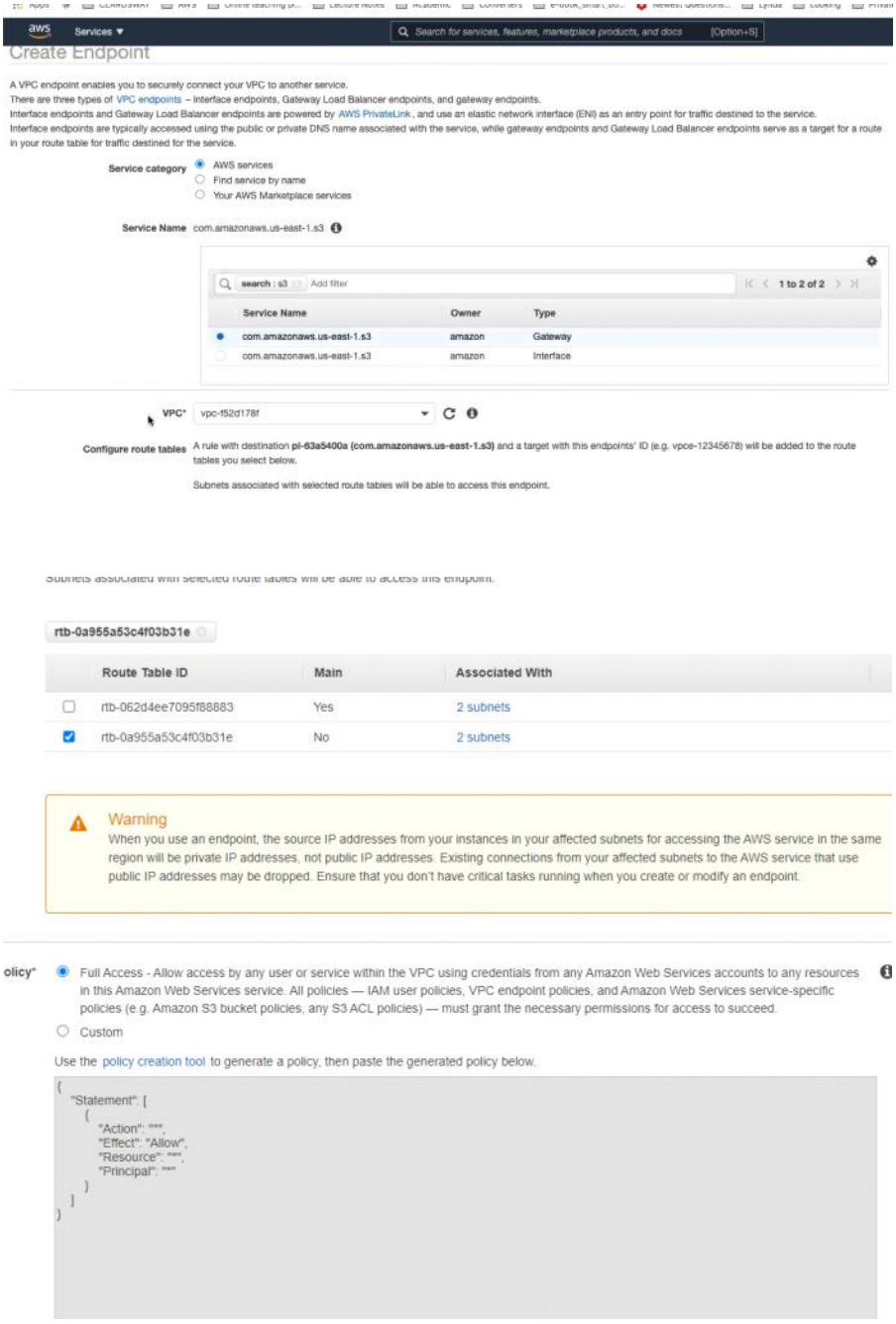


ss

Ve simdi de endpoint olusturacagiz. Ent pointleri Private Subnetler icin tanimlayacagiz.

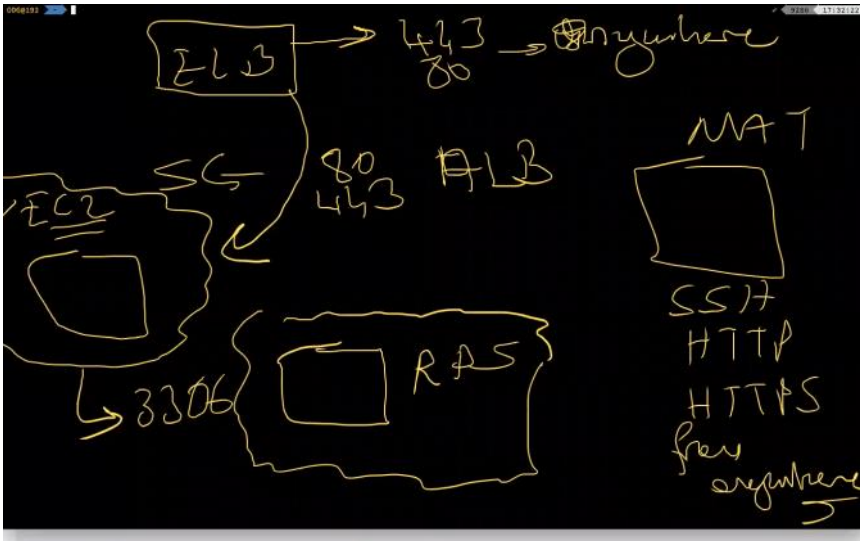
Vpc olarak kendi vpc mizi seciyoruz

Ardindan hangi Rote Tablelari sececegimiz karsimiza geliyor



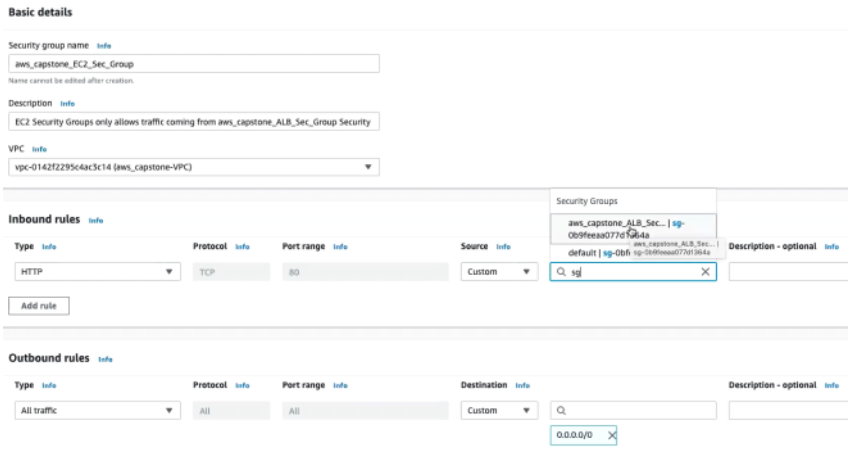
Ve simdi de security Grouplarimizi olusturacagiz. Ve daha sonra olusturdugumuz makinalrda kolay bir sekilde sececegiz.

Rds e her makinani ulasmasini istemiyoruz sadece application unumuzun oldugu EC2 nun ulasmasini istiyoruz

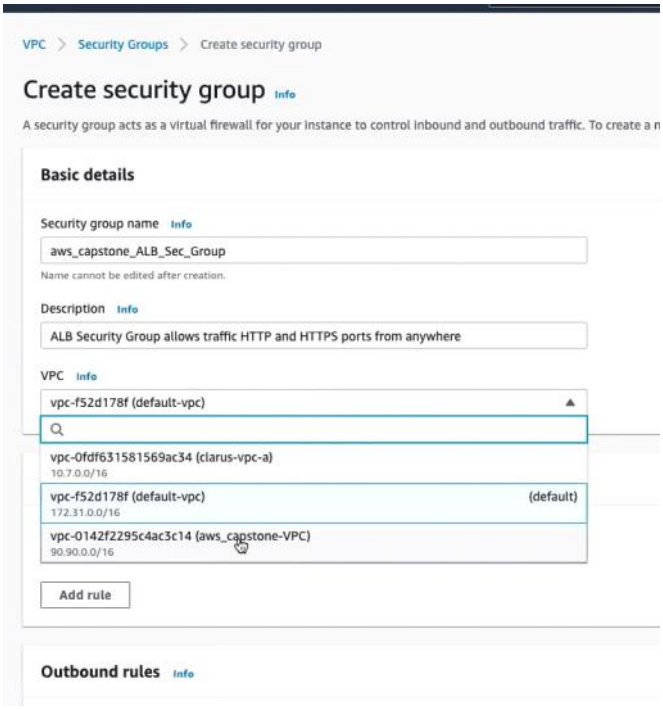


- **ELB dis dunya ile irtibat sagladigi icin ona 443 ve 80 portundan her yere yani any where olarak aciyoruz**
- **Ancak Ec2 ya sadece ELB uzerinden ulasilmasini istedigimiz icin EC2 icin olusturdugumuz Sec Grupta 80 ve 443 portundan sadece EC2 yu belirtecegiz**
- **RDS sadece EC2 dan ulasilmasini istedigimiz icin RDS e 3306 portundan sadec EC2 yu tanimliyoruz.**
- **Nat instance dis dunyadan her istegi ic teki Private icin yerine getiren oldugu icin burda any where ile http HTTPS I Tnimliyoruz**

```
### Step 2: Create Security Groups (ALB ---> EC2 ---> RDS)

1. ALB Security Group
Name              : aws_capstone_ALB_Sec_Group
Description       : ALB Security Group allows traffic HTTP an
d HTTPS ports from anywhere
Inbound Rules
VPC               : AWS_Capstone_VPC
HTTP(80)     ----> anywhere
HTTPS (443) ----> anywhere
```




```
2. EC2 Security Groups
Name              : aws_capstone_EC2_Sec_Group
Description       : EC2 Security Groups only allows traffic c
oming from aws_capstone_ALB_Sec_Group Security Groups for H
TTP and HTTPS ports. In addition, ssh port is allowed from
anywhere
VPC               : AWS_Capstone_VPC
Inbound Rules
HTTP(80)     ----> aws_capstone_ALB_Sec_Group
HTTPS (443) ----> aws_capstone_ALB_Sec_Group
ssh          ----> anywhere
```

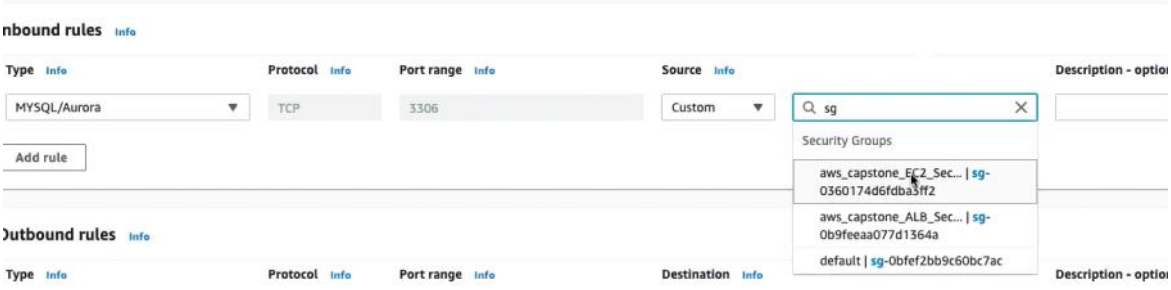


Bu su anlama geliyor 80 ve 443 portu ile attach ettigimiz (her ALB nin) makinenin bu vpc ye ulasabilecegini belirtiyoruz.

3. RDS Security Groups

```
Name            : aws_capstone_RDS_Sec_Group
Description     : EC2 Security Groups only allows traffic c
oming from aws_capstone_EC2
_Sec_Group Security Groups for MYSQL/Aurora port.
VPC             : AWS_Capstone_VPC
Inbound Rules
MYSQL/Aurora(3306)  ----> aws_capstone_EC2_Sec_Group
```

Sadece EC2 kardan geleni kabul edecek



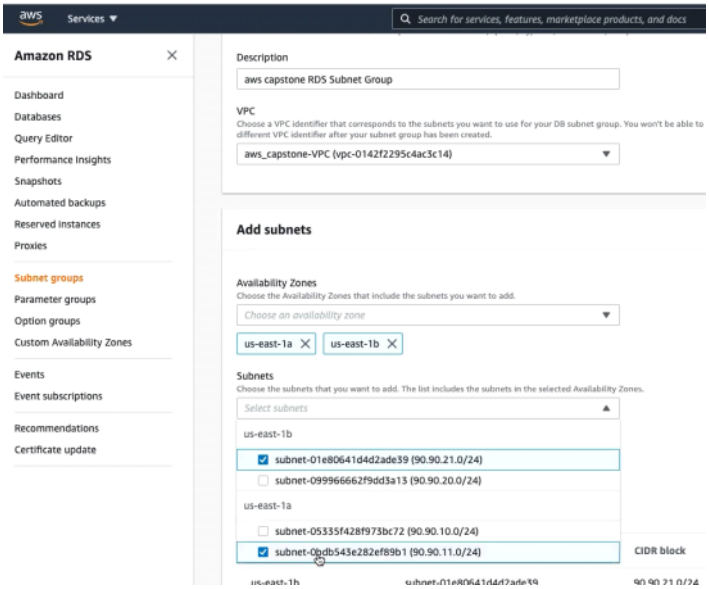

```
4. NAT Instance Security Group
Name            : aws_capstone_NAT_Sec_Group
Description     : ALB Security Group allows traffic HTTP an
d HTTPS and SSH ports from anywhere
Inbound Rules
VPC             : AWS_Capstone_VPC
HTTP(80)    ----> anywhere
HTTPS (443) ----> anywhere
SSH (22)    ----> anywhere
```

Step 3: Create RDS

Custom VPC lerde RDS kurmadan önce bu RDS lerin hangi VPC de calismasini istiyorsak o sekilde bir Subnet Group olusturuuz

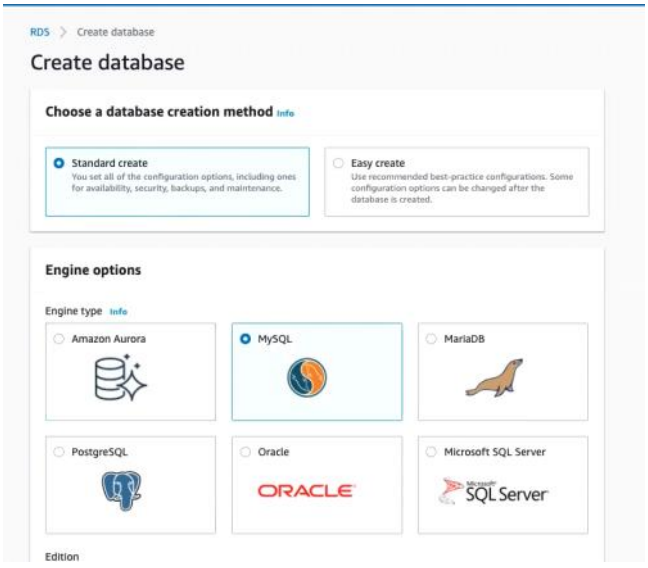
First we create a subnet group for our custom VPC. Click `subnet Groups` on the left hand menu and click `create DB Subnet Group`

```text
Name                : aws_capstone_RDS_Subnet_Group
Description         : aws capstone RDS Subnet Group
VPC                 : aws_capstone_VPC
Add Subnets
Availability Zones  : Select 2 AZ in aws_capstone_VPC
Subnets             : Select 2 Private Subnets in these subnets
```

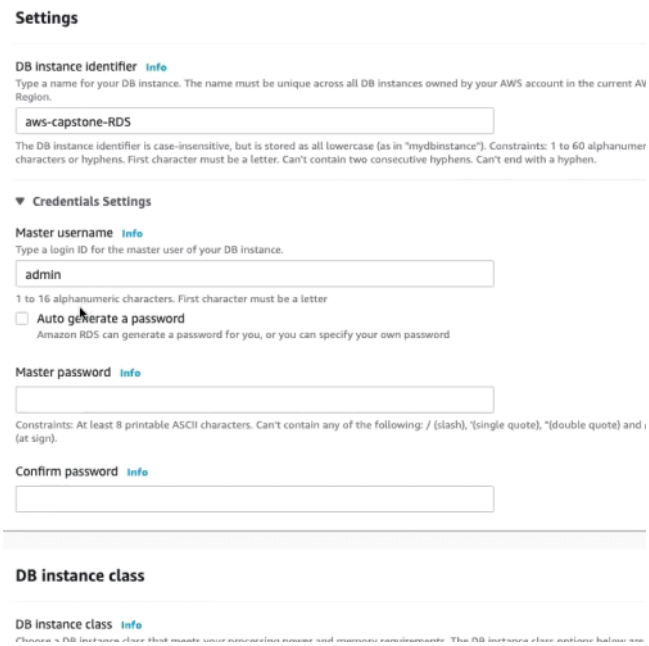


- Go to the RDS console and click `create database` button

```text
Choose a database creation method : Standart Create
Engine Options  : Mysql
```




```text
Version          : 8.0.20
Templates        : Free Tier
Settings         :
    - DB instance identifier : aws-capstone-RDS
    - Master username        : admin
    - Password               : Clarusway1234
DB Instance Class            : Burstable classes (includes t classes) ---> db.t2.micro
Storage                      : 20 GB and enable autoscaling (up to 40GB)
```



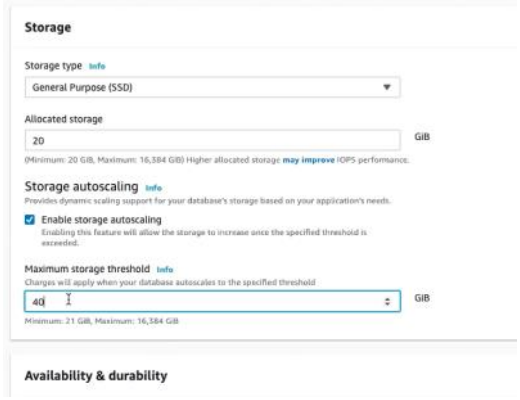
Connectivity:

VPC : aws_capstone_VPC

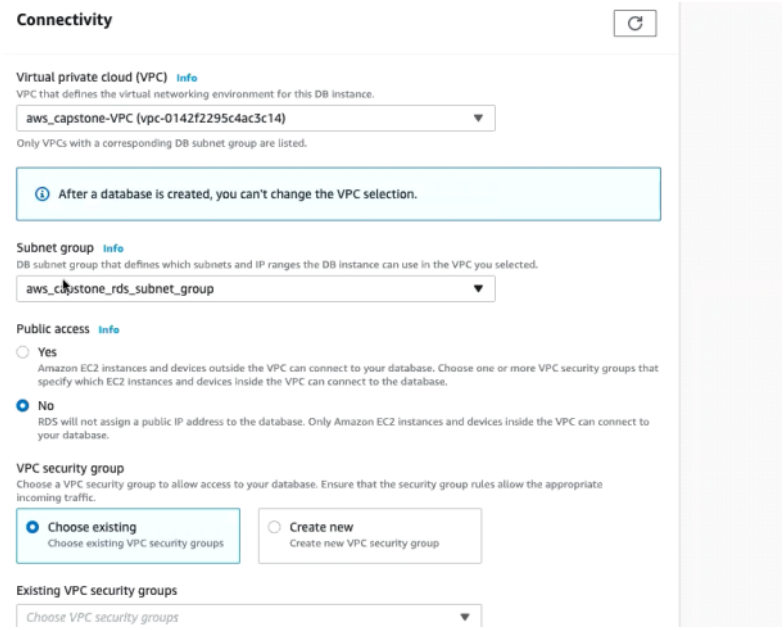
Subnet Group : aws_capstone_RDS_Subnet_Group

Public Access : No

VPC Security Groups : Choose existing ---> aws_capstone_RDS_Sec_Group

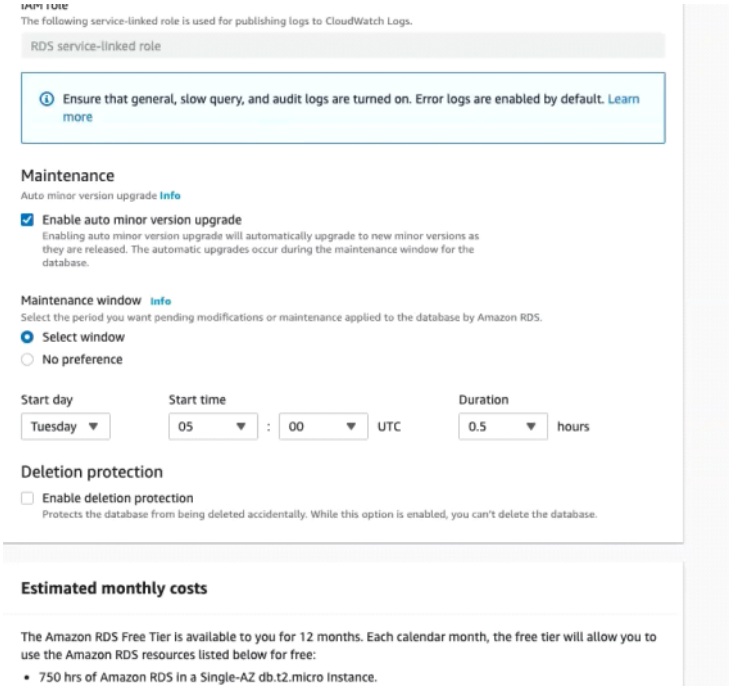


Storaage kismini RDS in kapasitesi max kac arttirabiliyor max olarak 16,384 GIB yapabiliyor



Public acces no diyoruz VPC nin disindan verilen bir izindir.

```
Availability Zone        : No preference
    Additional Configuration : Database port ---> 3306
Database authentication ---> Password authentication
Additional Configuration:
    - Initial Database Name  : database1
    - Backup ---> Enable automatic backups
    - Backup retention period ---> 7 days
    - Select Backup Window ---> Select 03:00 (am) Duration
1 hour
    - Maintance window : Select window    ---> 04:00(am) Du
ration:1 hour
create instance
```



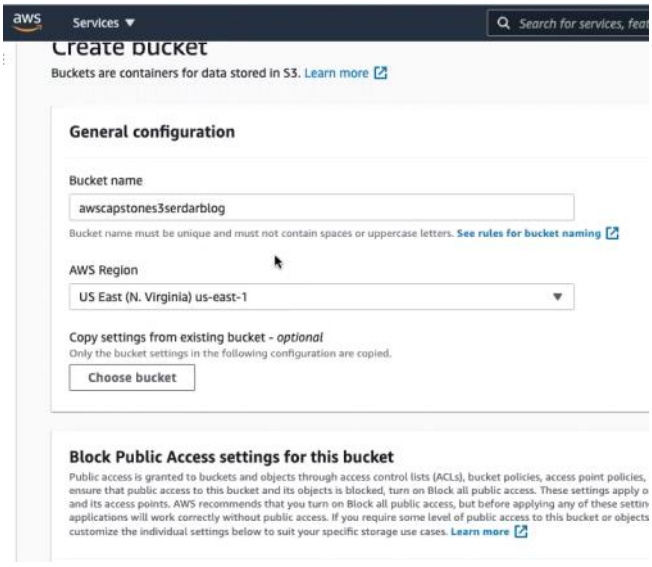
Enable deleion dersek silmek icin bir kac basamli bir sart getiri protection ortanminda bu öneriir.

Iki bucket icin olusturacagiz bir failover icin digerri resim ve metinleri saklamak icin olusturacagiz

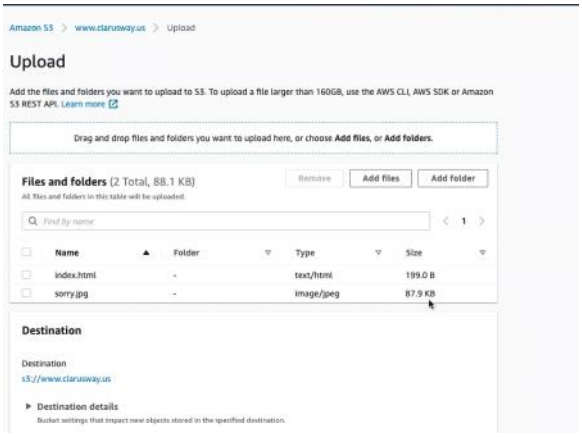
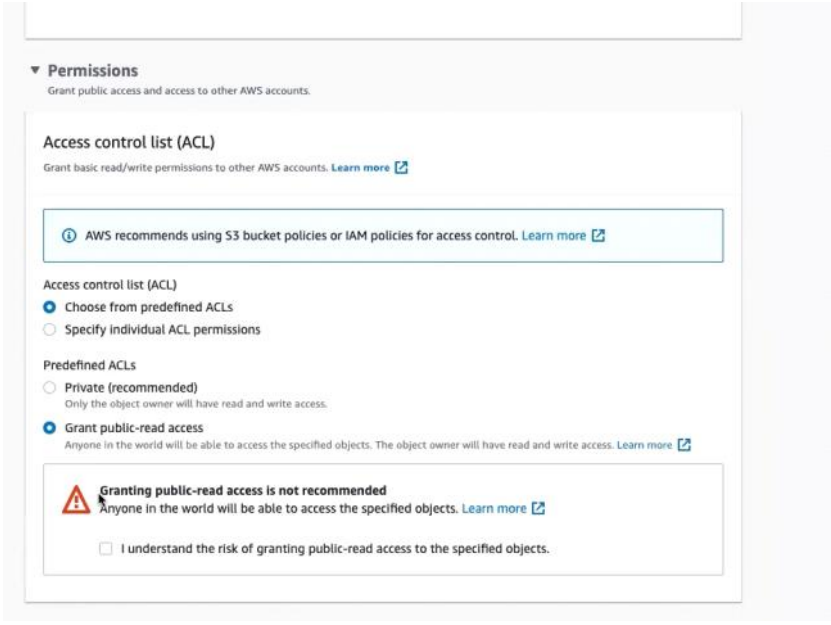
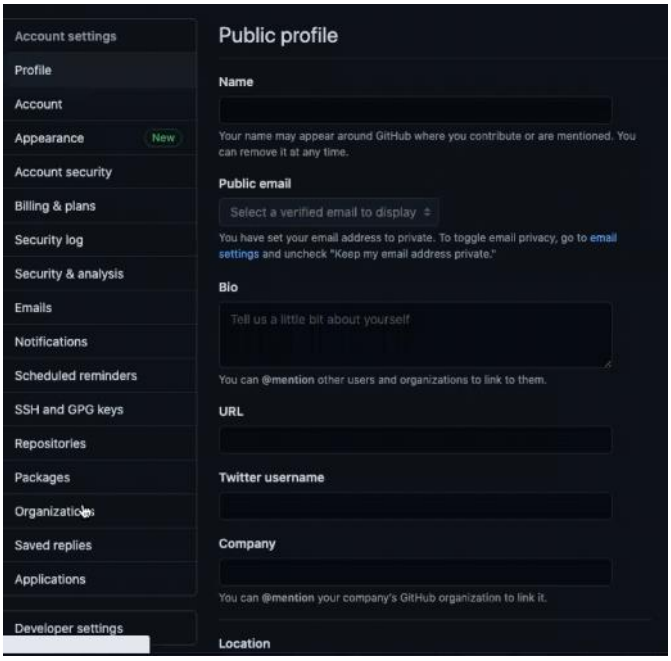
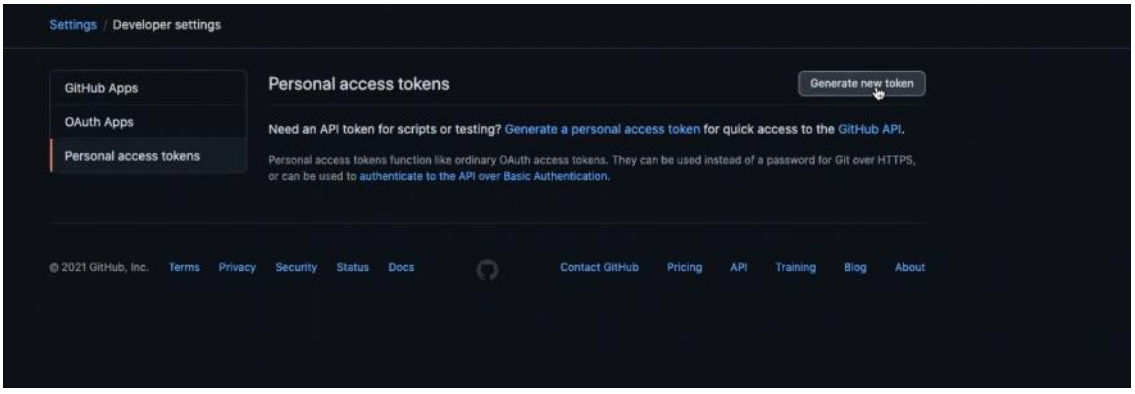
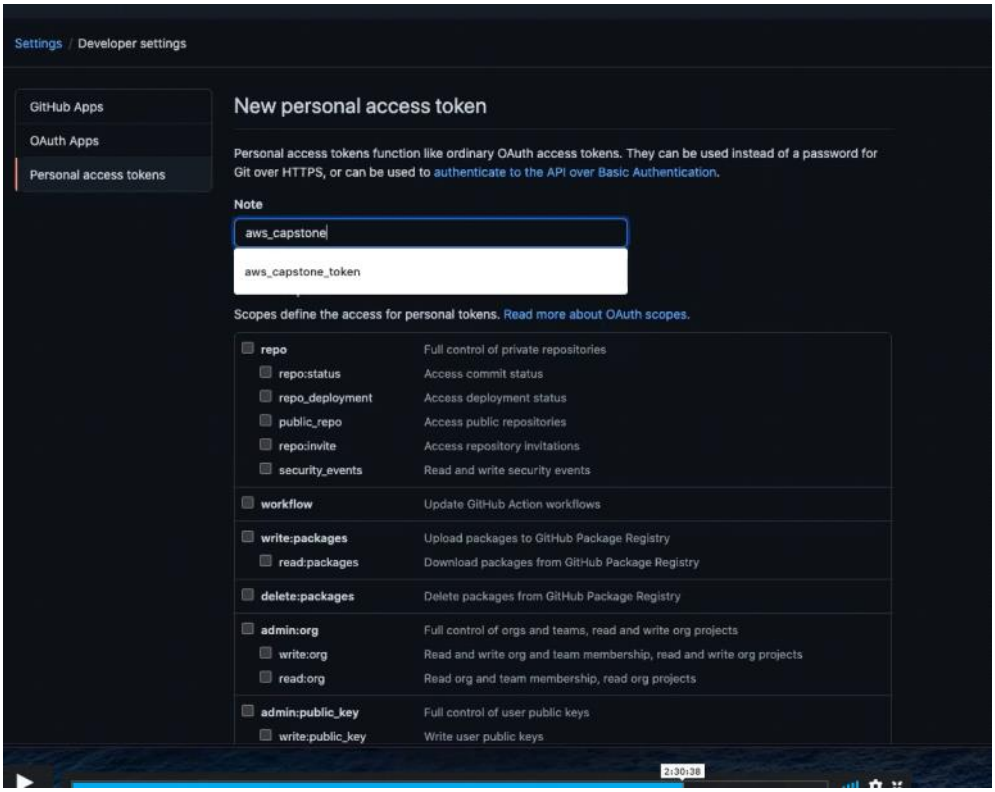
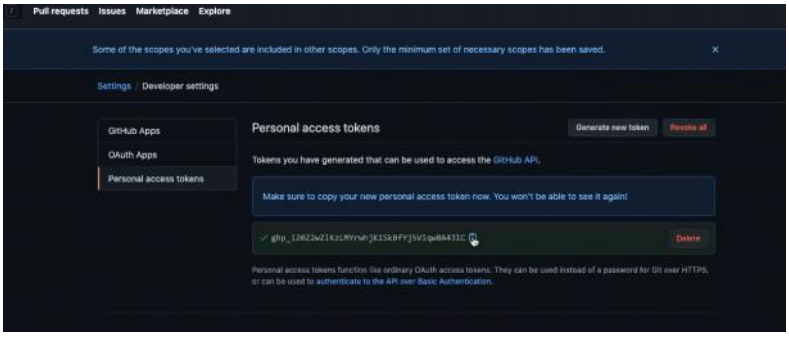
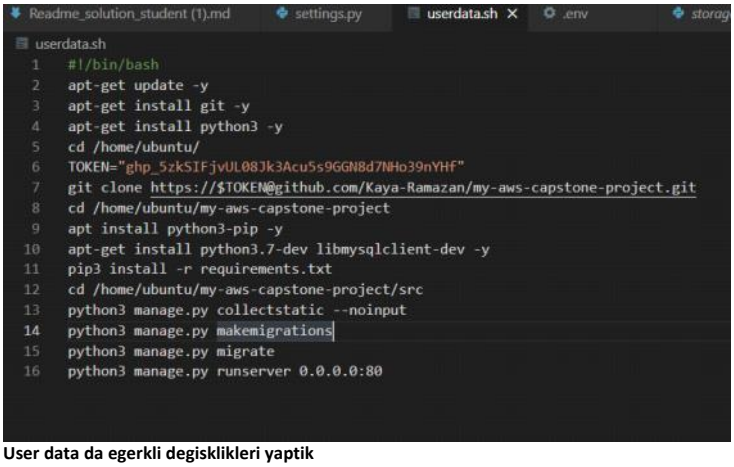
Step 4: Create two S3 Buckets and set one of these as s tatic website.

Go to the S3 Consol and lets create two buckets.

1. Blog Website's S3 Bucket
- Click Create Bucket
```text
Bucket Name : awscapstones3<YOUR NAME>blog------uniq dikkat edilmesi gereken
Region      : N.Virginia
```



Block all public access : Unchecked
Other Settings are keep them as are
create bucket
```

2. S3 Bucket for failover scenario
- Click Create Bucket
```text

Bucket Name : www.<YOUR DNS NAME>
Region      : N.Virginia
Block all public access : Unchecked
Please keep other settings as are
```
- create bucket
- Selects created `www.<YOUR DNS NAME>` bucket ---> Properties ---> Static website hosting
```text
Static website hosting : Enable
Hosting Type : Host a static website
Index document : index.html
save changes
```
- Select `www.<YOUR DNS NAME>` bucket ---> select Upload and upload `index.html` and `sorry.jpg` files from given folder---> Permissions ---> Grant public-
read access ---> Checked warning massage
```



Ayrica static web site olarak kullanmak icin de

Ayrica WWW. Ve DNS name ile olusturdugumuz subnette Acces Control sekmesini Grant Public olarak aciyoruz upload ettigimiz file in dis dunyadan okunmasini sagliyoruz



Step 5: Copy files downloaded or cloned from `Clarusway_project` repo on Github

Git hub dan
S3 ler ayarlandi RDS ayarlandi ve simdi Githubda bulnan private repomuzu clonlamistik daha önce simdi de git commit - m ve git push komutlari ile ile lokalde bulunan dosyalarimizi git hub dosyamiza gönderdik.

Simdi de git hub hesabimiz dan TOKEN alacagiz;

Avatar----Settin-----Developer settings--Personel Access tokens----



Ve Generate new token



Enter the passsword



Ve Token hesabimiz olustu





User data da egerkli degisklikleri yaptik

Son dört komut Jangoyu calistiriyor.

```
## Step 8: Write RDS database endpoint and S3 Bucket name i
n settings file given by Clarusway Fullstack Developer team
 and push your application into your own public repo on Git
hub
Please follow and apply the instructions in the developer_n
otes.txt.
```text
- Movie and picture files are kept on S3 bucket named aws_c
```
```

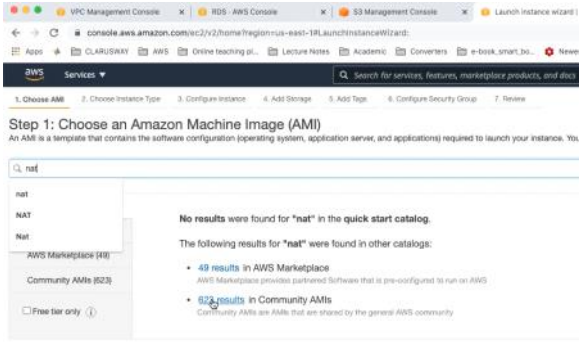
```
apstone_S3
_<name>_Blog as object. You should create an S3 bucket and
write name of it on "/src/cblog/settings.py" file as AWS_ST
ORAGE_BUCKET_NAME variable. In addition, you must assign re
gion of S3 as AWS_S3_REGION_NAME variable
- Users credentials and blog contents are going to be kept
on RDS database. To connect EC2 to RDS, following variables
 must be assigned on "/src/cblog/settings.py" file after yo
u create RDS;
    a. Database name - "NAME" variable
    b. Database endpoint - "HOST" variables
    c. Port - "PORT"
    d. PASSWORD variable must be written on "/src/.env" fil
e not to be exposed with settings file
```
- Please check if this userdata is working or not. to do th
is create new instance in public subnet and show to student
s that it is working

Step 9: Create NAT Instance in Public Subnet

Public Subnette Nat instance olusturacagiz. Nat instance olusturmamizin sebebi Auto scaling olusturacagiz ki bunun icin öncelikle bir Launch Template olusturacaktik . AutoScalin EC2 olusturacak ve Ec2 larda Data cekmek isteyecekler onun icin Nat instance ihtiyac var.

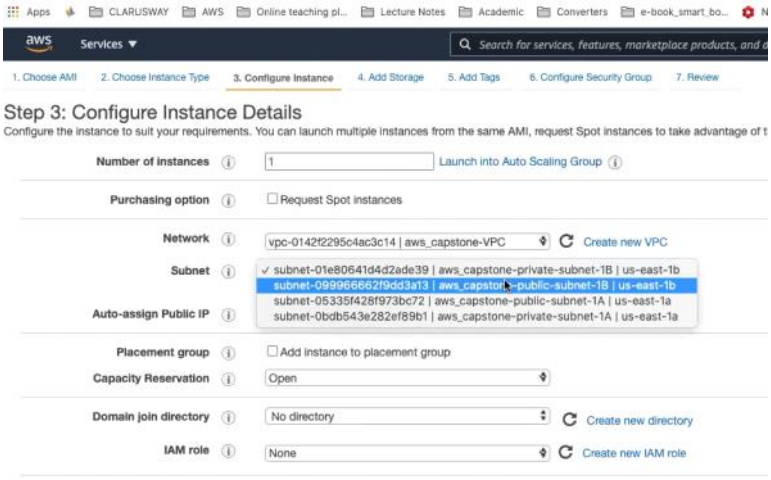
To launch NAT instance, go to the EC2 console and click the create button.

```text
write "NAT" into the filter box
select NAT Instance `amzn-ami-vpc-nat-
hvm-2018.03.0.20181116-x86_64-ebs`
```

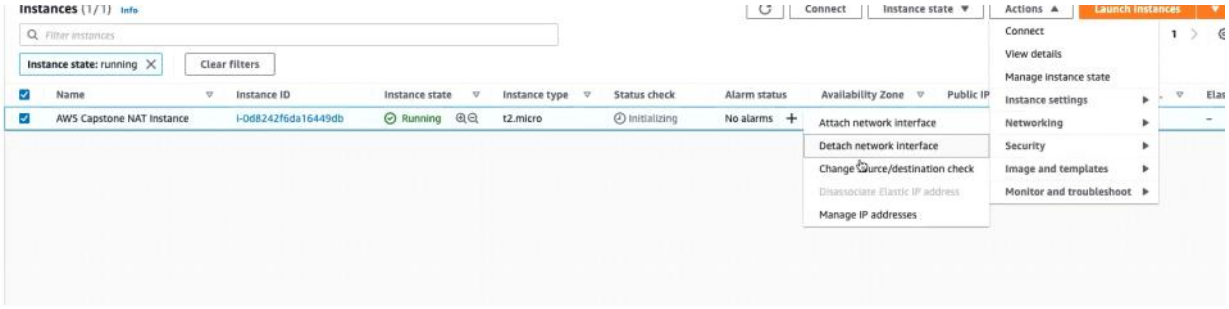



```
Instance Type: t2.micro
Configure Instance Details
    - Network : aws_capstone_VPC
    - Subnet  : aws_capstone-public-
subnet-1A (Please select one of your Public Subnets)
    - Other features keep them as are
Storage ---> Keep it as is
Tags: Key: Name     Value: AWS Capstone NAT Instance
Configure Security Group
    - Select an existing security group: aws_capstone_NAT_S
ec_Group
Review and select our own pem key
```
```
!!!IMPORTANT!!!
- select newly created NAT instance and enable stop source/
destination check
- go to private route table and write a rule
```
```
Destination : 0.0.0.0/0
Target      : instance ---> Select NAT Instance
Save
```

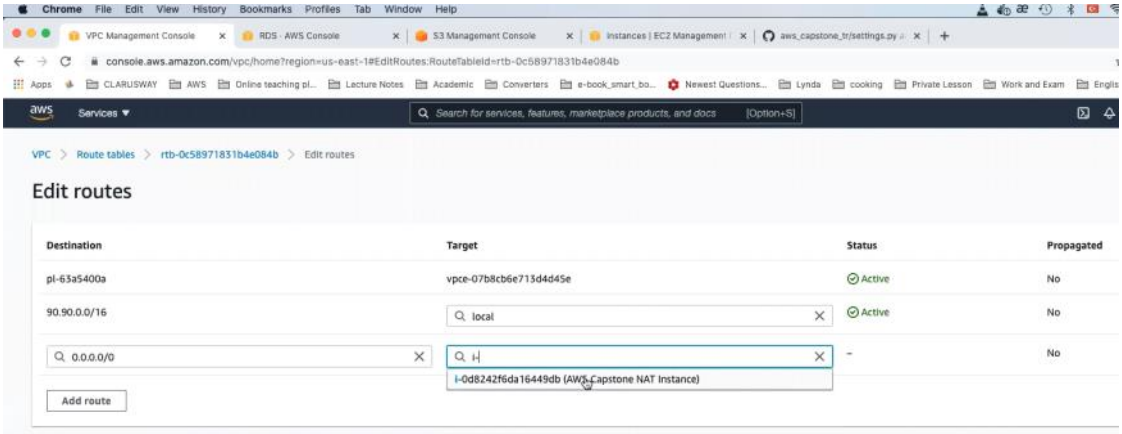
Public Subnetlerde olusturacagiz



Nat instance olusturduktan sonra Actions ---Change Source destination Check edecegiz ----stop edecegiz



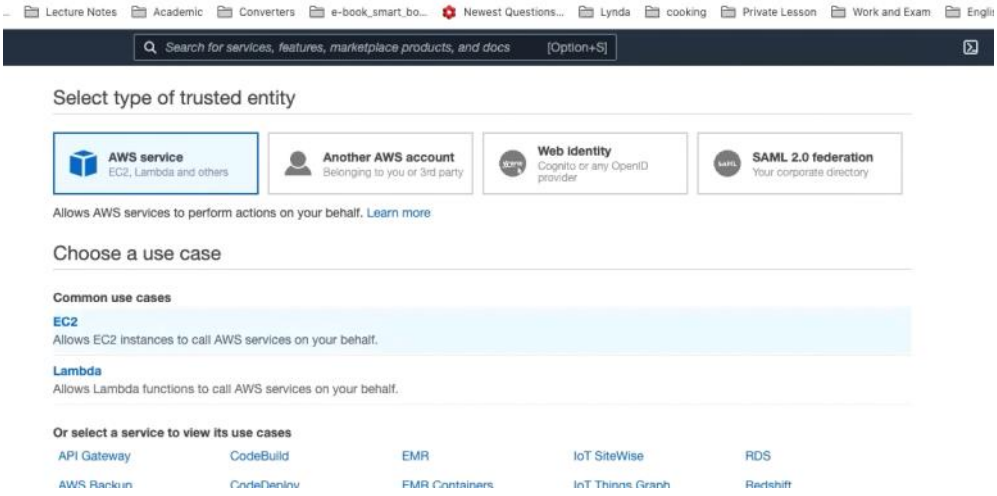
Daha sonra Private Rote Tabletelere bir Raute Table eklememiz gerekiyor.
Private
Makinalrda dis dunyaya gitmek istediklerinde bi istegi nat instance aktarmasini sagla diyoruzz.



Simdi de Launch Template olusturacagiz Launch Template tanimlamak icin EC2 ile konusabilmesi icin 2 alteranatif var bunlardan ilki boto3 ile oluyor. Veyahut bu ulasmayi Iam Role tanimlayarak da yapabiliyoruz.

Full access yetkisi verecek bir IAM Role tanimlayacagiz ki bunun ile

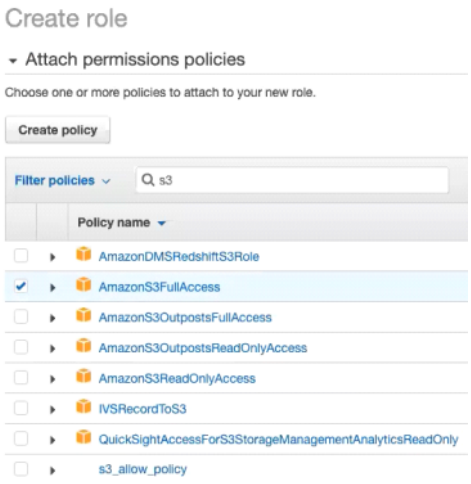
Ilk olarak EC2 seciyoruz



Step 10: Create Launch Template and IAM role for it

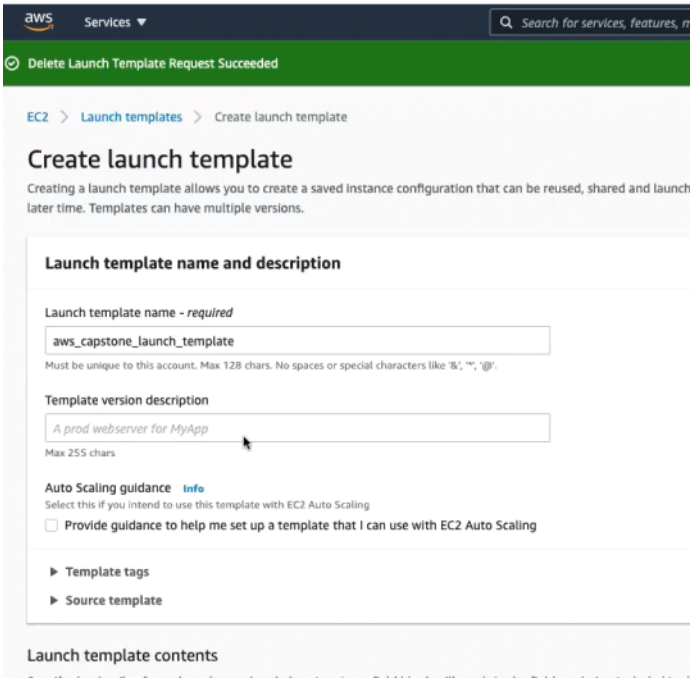
Go to the IAM role console click role on the right hand men u than create role

```text
trusted entity  : EC2 as  ---> click Next:Permission
Policy          : AmazonS3FullAccess policy
Tags            : No tags
Role Name       : aws_capstone_EC2_S3_Full_Access
Description     : For EC2, S3 Full Access Role
```



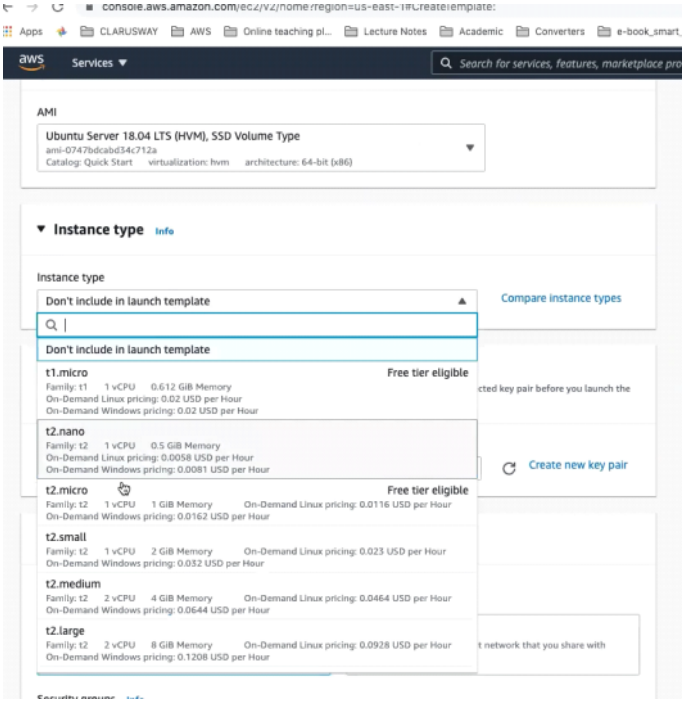
Rol Create edildikten sonra Launch Template olusturacagiz;

```
- create launch template
```

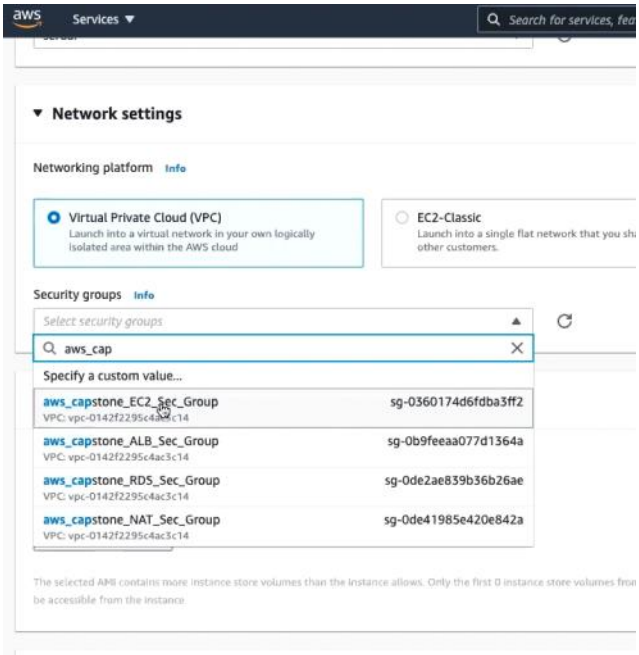


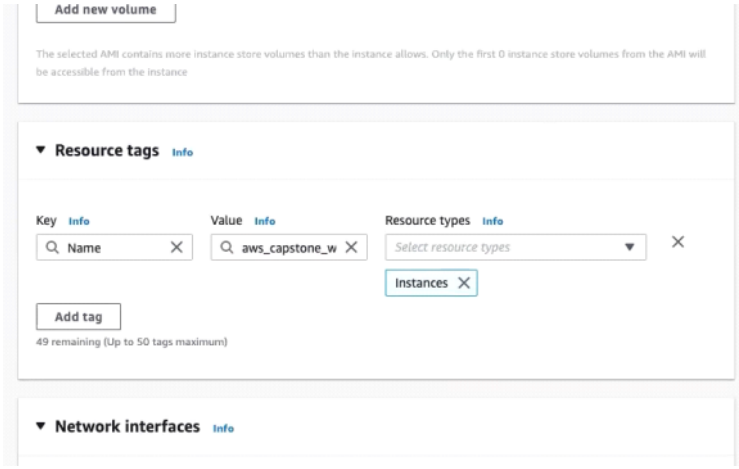
To create Launch Template, go to the EC2 console and select `Launch Template` on the left hand menu. Tab the Create Launch Template button.

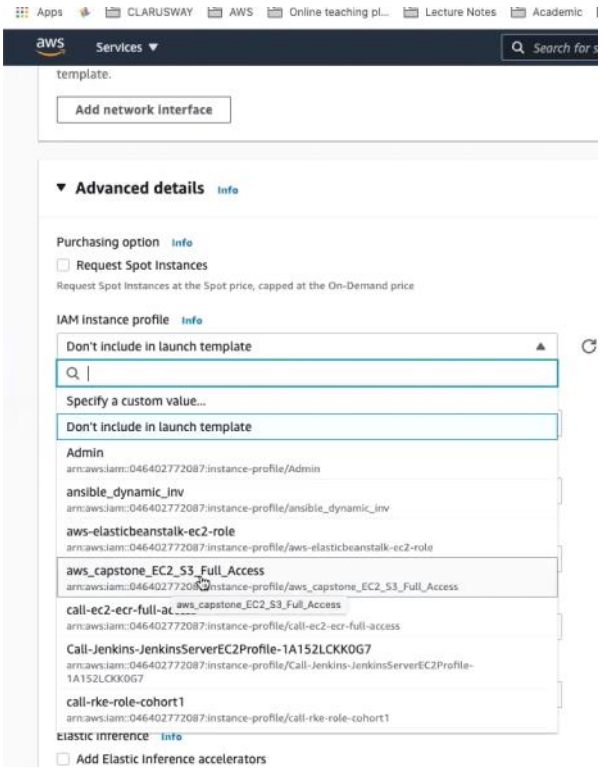
```bash
Launch template name                : aws_capstone_launch_template
Template version description        : Blog Web Page version 1
Amazon machine image (AMI)          : Ubuntu 18.04
Instance Type                       : t2.micro
Key Pair                            : mykey.pem
Network Platform                    : VPC
```



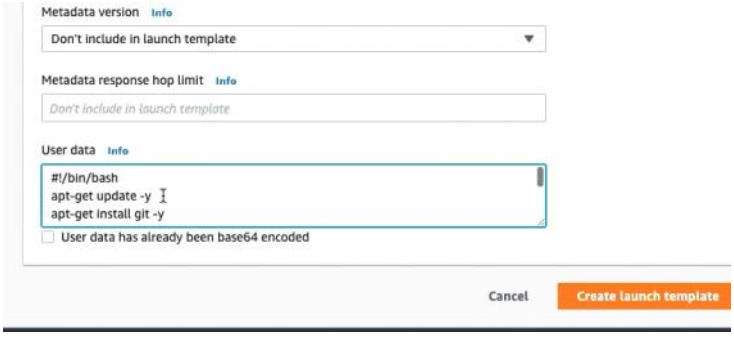

```bash
Security Groups                     : aws_capstone_EC2_sec_group
Storage (Volumes)                   : keep it as is
Resource tags                       : Key: Name   Value: aws_capstone_web_server
Advance Details:
    - IAM instance profile          : aws_capstone_EC2_S3_Full_Access
    - Termination protection        : Enable
```


aws
Services
Search for services, fea
Network settings
Networking platform Info
Virtual Private Cloud (VPC)
Launch into a virtual network in your own logically isolated area within the AWS cloud
EC2-Classic
Launch into a single flat network that you sh other customers.
Security groups Info
Select security groups
aws_cap
Specify a custom value...
aws_capstone_EC2_Sec_Group
VPC: vpc-0142f2295c4ac3c14
sg-0360174d6fdba3ff2
aws_capstone_ALB_Sec_Group
VPC: vpc-0142f2295c4ac3c14
sg-0b9feeaa077d1364a
aws_capstone_RDS_Sec_Group
VPC: vpc-0142f2295c4ac3c14
sg-0de2ae839b36b26ae
aws_capstone_NAT_Sec_Group
VPC: vpc-0142f2295c4ac3c14
sg-0de41985e420e842a
The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes fro
be accessible from the instance


Add new volume
The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance
Resource tags Info
Key Info
Name
Value Info
aws_capstone_w
Resource types Info
Select resource types
Instances
Add tag
49 remaining (Up to 50 tags maximum)
Network interfaces Info




```
```     - User Data
#!/bin/bash
apt-get update -y
apt-get install git -y
apt-get install python3 -y
cd /home/ubuntu/
TOKEN="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"
git clone https://$TOKEN@github.com/Kaya-Ramazan/my-aws-capstone-project.git
cd /home/ubuntu/my-aws-capstone-project
apt install python3-pip -y
apt-get install python3.7-dev default-libmysqlclient-dev -y
pip3 install -r requirements.txt
cd /home/ubuntu/my-aws-capstone-project/src
python3 manage.py collectstatic --noinput
python3 manage.py makemigrations
python3 manage.py migrate
python3 manage.py runserver 0.0.0.0:80
```



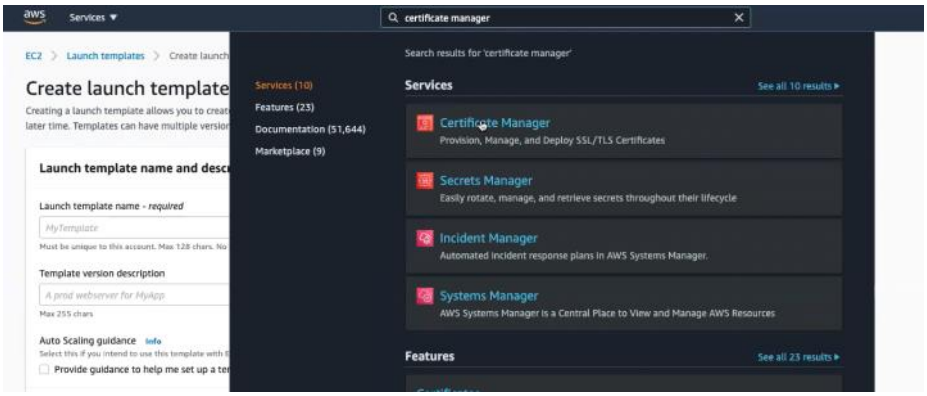

**Production Ortaminda en basit bir site adahil secur bir baglansti yapmak ister. Browserlar sec baglanti var ise**

**HTTP den yayin Yapmazlar daha secure olan HTTPS den yayin yaparlar increp etmek icin sertifikalari kullanirlar**

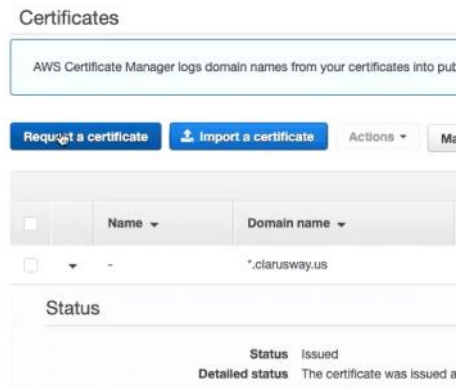
## Step 11: Create certification for secure connection

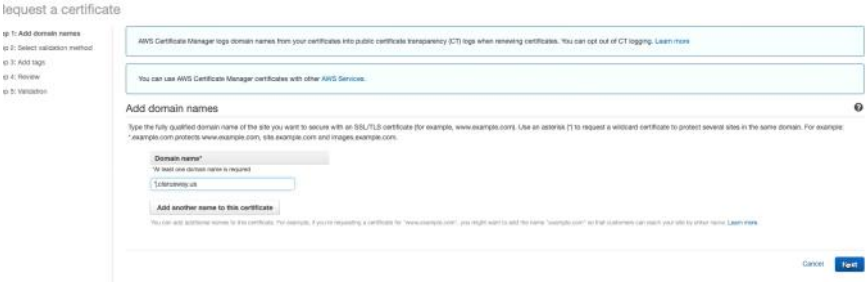
**Go to the certification manager console** and click `request a certificate` button. Select `Request a public certificate
```

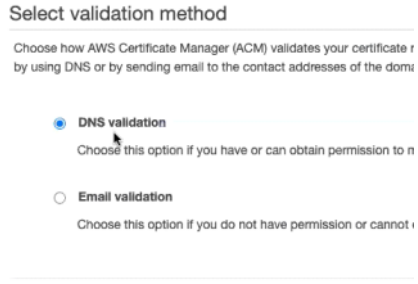
`, then `request a certificate` ---> `*.<YOUR DNS NAME>` --
-> DNS validation ---> No tag ---> Review ---> click confir
m and request button. Then it takes a while to be activated
.

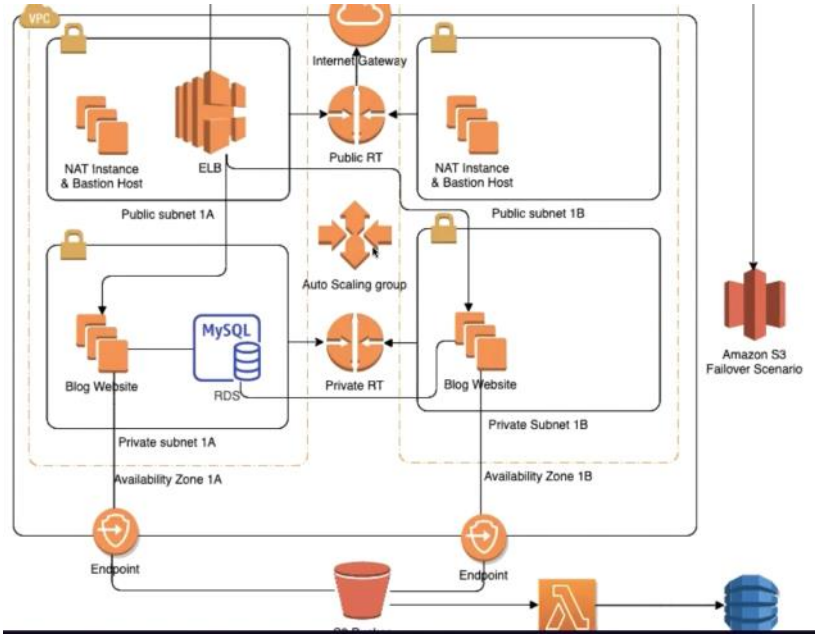


Yeni alamak icin;









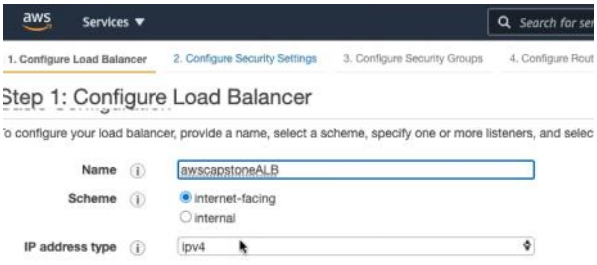
Simdiye kadar yaptiklarimizi özetler isek ;

- Vpc leri kurduk
- Availibility zonalar ve Subnetleri ayarladik
- internetGateway baglandi
- Public Subnetleri Asocate ettik
- RDS kurduk
- Endpointimizi verdik

Step 12: Create ALB and Target Group

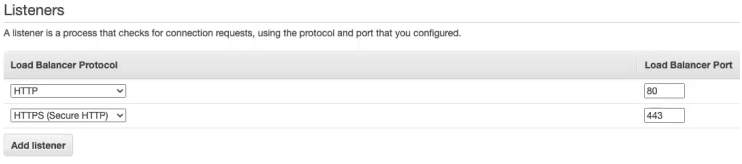
Go to the Load Balancer section on the left hand side menu of EC2 console. Click `create Load Balancer` button and select Application Load Balancer

```text
Name                    : awscapstoneALB
Schema                  : internet-facing
```

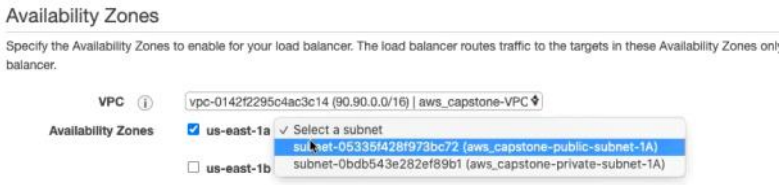


Listeners

Listeners : HTTPS, HTTP

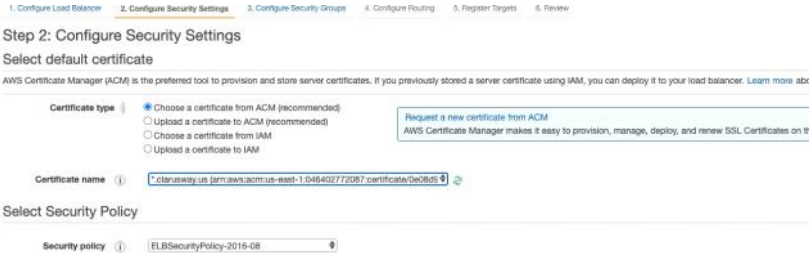


Availability Zones :
- VPC : aws_capstone_VPC
- **Availability zones:**
  1. aws_capstone-public-subnet-1A
  2. aws_capstone-public-subnet-1B

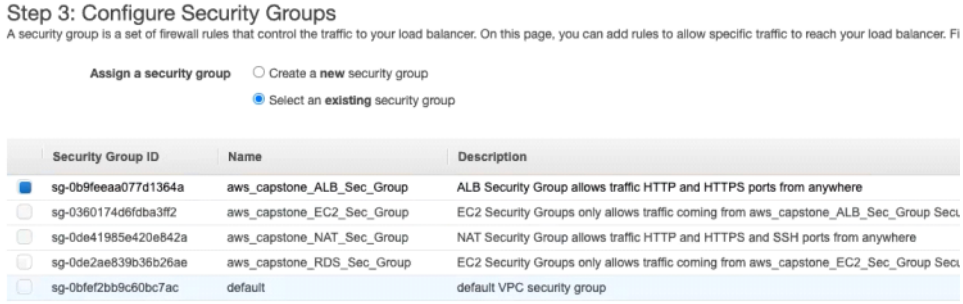


Public e yerlestiriyoruz cunku yayuin yapacak

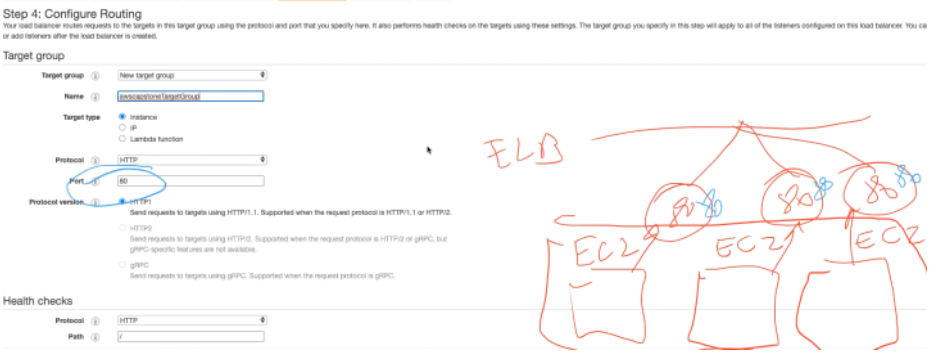
Step 2 - Configure Security Settings
Certificate type ---> Choose a certificate from ACM (recommended)
- Certificate name : "*.clarusway.us" certificate
- Security policy : keep it as is



Step 3 - Configure Security Groups : aws_capstone_ALB_Sec_group



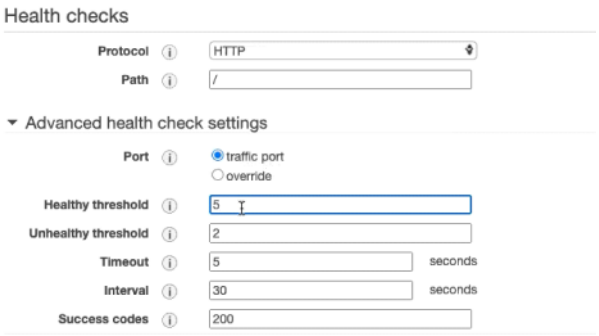
Step 4 - Configure Routing
- **Target group** : New target group
- Name : awscapstoneTargetGroup
- Target Type : Instance
- Protocol : HTTP
- Port : 80



Instance nerede calisiyorsa o portta calistirmamiz gerekiyor

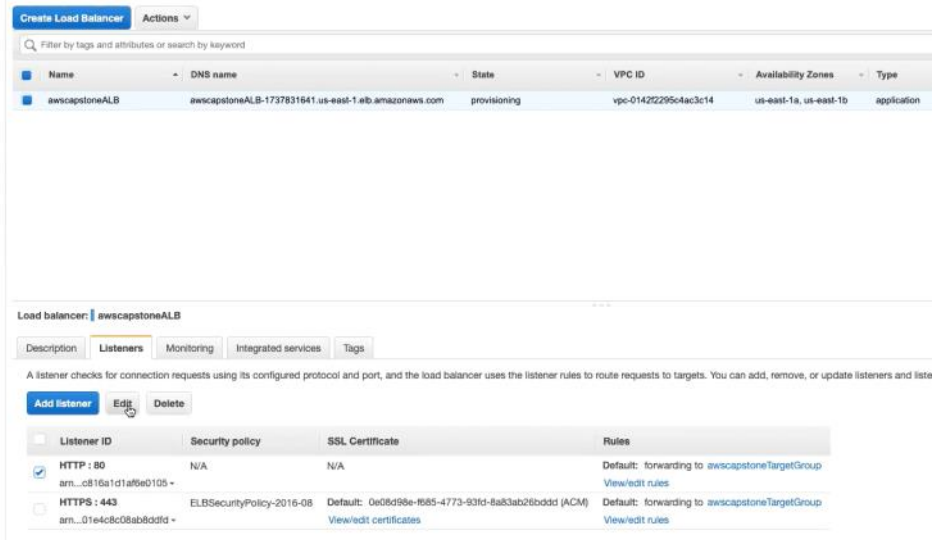
- Protocol version : HTTP1
- Health Check :
  - Protocol : HTTP
  - Path : /
  - Port : traffic port
  - Healthy threshold : 5 ------ ardi ardina gönderilen 3 saglik sinyalde sagliklin olarak muhurluyor
  - Unhealthy threshold : 2-------ard ardina 2 kez

gönerdi ve cevap alamadi sagliksiz olarak nitelendiriyor
      - Timeout   : 5   ------sinyali gönderdikten sonra bekleme suresi
      - Interval   : 20
      - Success Code   : 200 ----- herhangi problem yok demejtir.

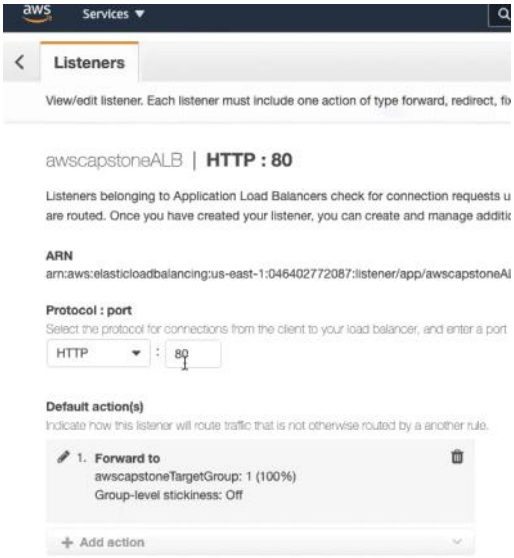


Step 5 - Register Targets
without register any target click Next: Review
```
- **click create**
**To redirect traffic from HTTP to HTTPS, go to the ALB conso le and select Listeners sub-section.**
```text
select HTTP: 80 rule ---> click edit
- Default action(s)
 - Remove existing rule and create new rule which is
    - Redirect to HTTPS 443
    - Original host, path, query
    - 301 - permanently moved
```
Lets go ahead and look at our ALB DNS --> it going to say " it is not safe", however, it will be fixed after settings o f Route 53

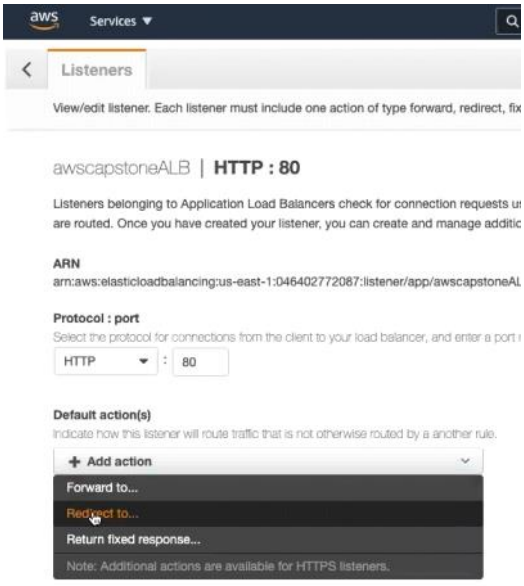
Load balanceri olusturduktan sonra yayini
80 portundan gelen yayini mevcutta Target Guruba gönderiyor iken Biz Önce 443 HTTPS e göndermesini ve sonmrasinda Target guruba gitmrisni istiyorduk. Bu sekilde daha secure olacakti.



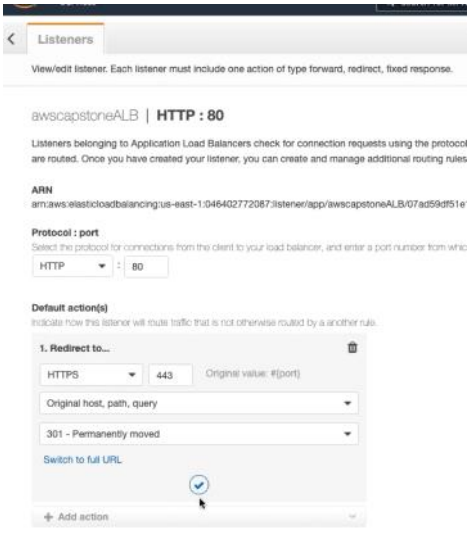
Bunun icin 80 baglantisini tiklayip edit diyoruz.
```



Ardindan default olan Forwardi siliyoruz



Ve Redirection ekliyoruz ve 443 secip tikliyoruz.

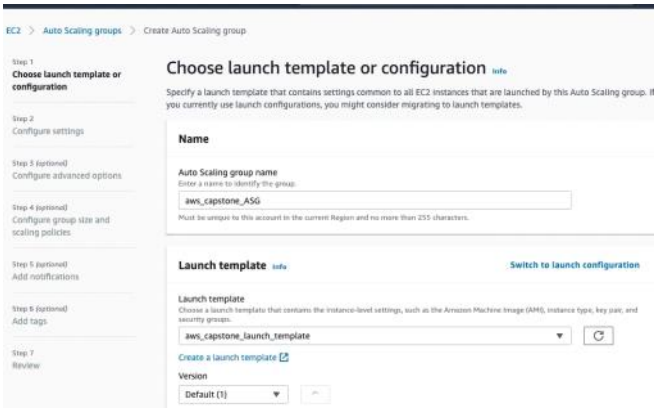


Step 13: Create Autoscaling Group with Launch Template

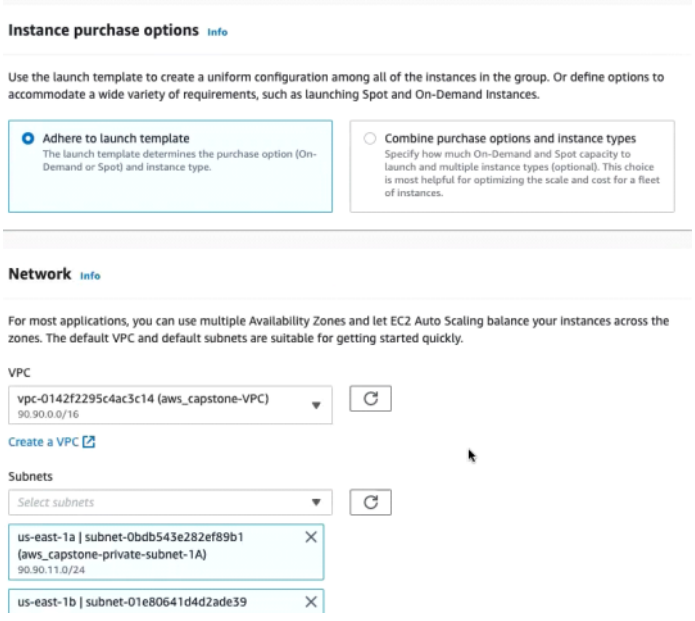
Go to the Autoscaling Group on the left hand side menu. Cli

ck create Autoscaling group.

- Choose launch template or configuration
```text
Auto Scaling group name          : aws_capstone_ASG
Launch Template                  : aws_capstone_launch_templ
ate
```

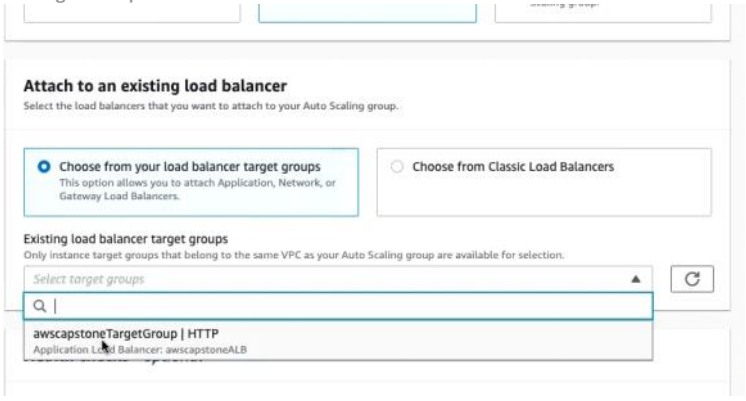


- Configure settings
```text
Instance purchase options        : Adhere to launch template
Network                          :
    - VPC                        : aws-capstone-VPC
    - Subnets                    : Private 1A and Private 1B
```

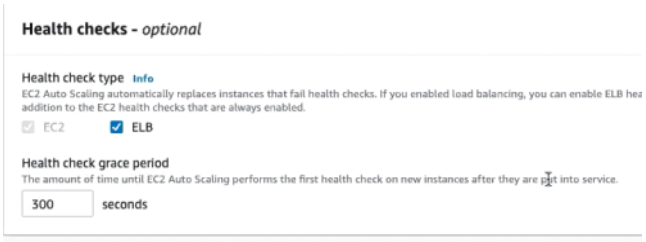


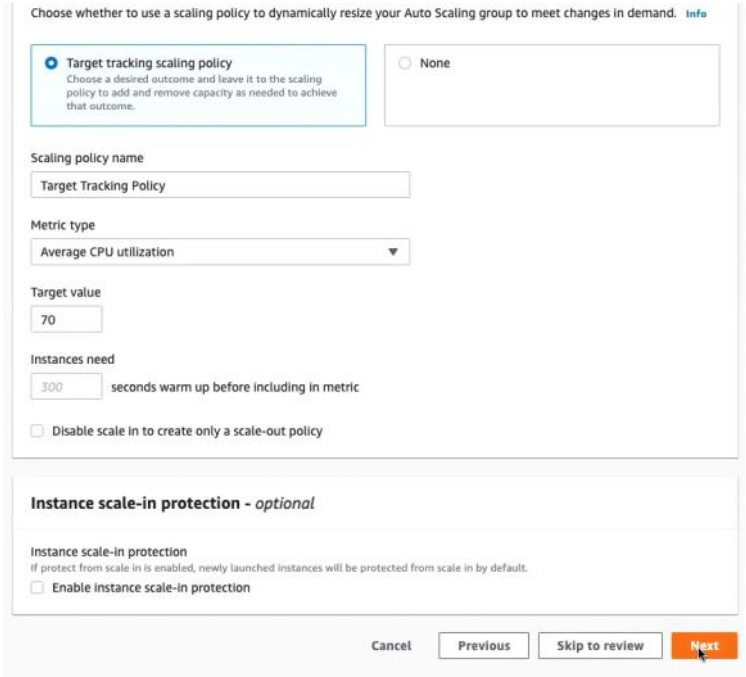
- Configure advanced options
```text

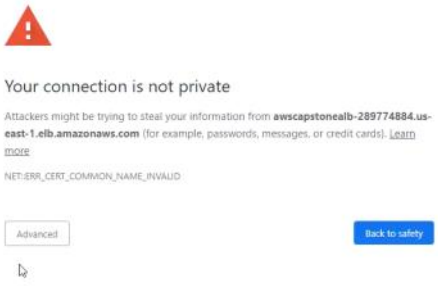
- Load balancing                 : Attach to
 an existing load balancer
- Choose from your load balancer target groups  : awscapsto
neTargetGroup
```



```
- Health Checks
    - Health Check Type             : ELB
    - Health check grace period     : 300
```

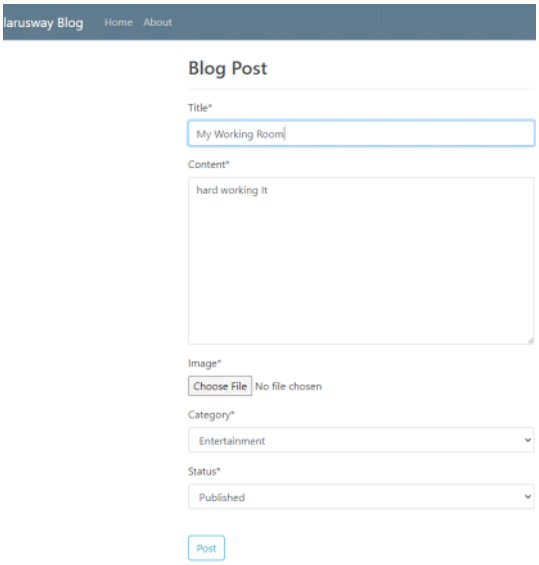



```
- Configure group size and scaling policies
```text
Group size
    - Desired capacity  : 2
    - Minimum capacity  : 2
    - Maximum capacity  : 4
Scaling policies


    - Target tracking scaling policy
        - Scaling policy name       : Target Tracking Polic
y
        - Metric Type               : Average CPU utilizati
on
        - Target value              : 70
```
- Add notifications
```text
Create new Notification
    - Notification1
        - Send a notification to    : aws-capstone-SNS
        - with these recipients     : serdar@clarusway.com
        - event type                : select all
```
```



Target Traning Policy Ec2 instance olusmasi veyahut Terminate ettirilmesi icin belirtilir.

CPU Utilazition ortalama Cpu kullanimi bizim belirledigimiz degere ulasirsa yeni bir instance kuracak altinda kalirsa mevcudu koruyacaktir.

```
<!-- WARNING!!! Sometimes your EC2 has a problem after you
create autoscaling group, If you need to look inside one of
```

```
 your instance to make sure where the problem is, please fo
llow these steps...
```bash
eval "$(ssh-agent)" (your local)
ssh-add <pem-key>   (your local )
ssh -A ec2-
user@<Public IP or DNS name of NAT instance> (your local)
ssh ubuntu@<Public IP or DNS name of private instance>  (NA
T instance)
You are in the private EC2 instance
```  -->
```



Load balancer in DNS name ini alip Consola yapistirinca karsimiza cikan ekran da sol alt kism da ADVANCE tikliyoruz
Hala not secure gözukmesinin sebebi biz sertifikalari Host Name icin tanimladik Road Balancer in cikarttigi adres icin degil Dolayisiyka bunu oraya yönkendirmemiz gerekiyor

Azcilan sayfada ilk oalrak Register edecegiz ve kendimize bir Mail ve sifre belirleyip sisteme girecegiz



Ve daha sonra

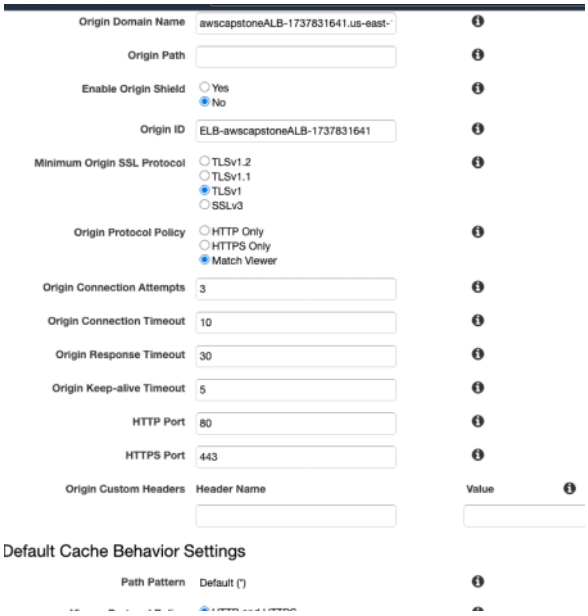
Step 14: Create Cloudfront in front of ALB

Go to the cloudfront menu and click start
- Origin Settings

Load balancer in Domain Name ini isretliyoruz

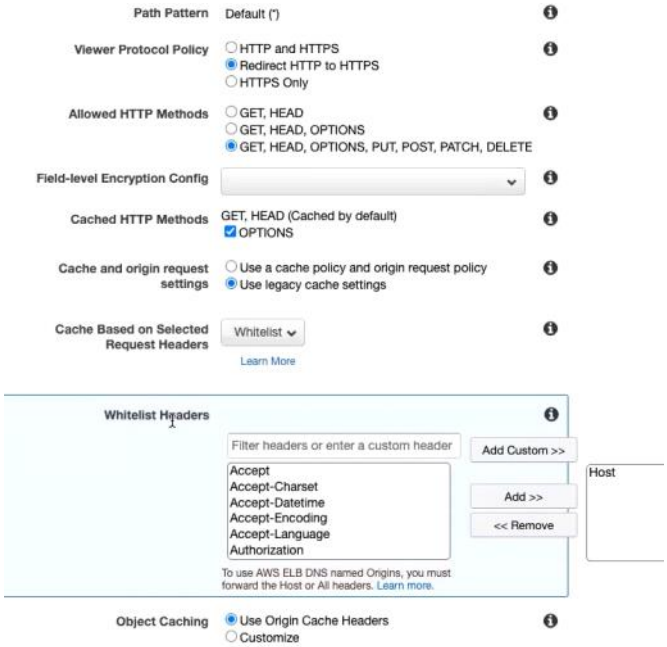
```text
Origin Domain Name          : aws-capstone-
ALB-1947210493.us-east-2.elb.amazonaws.com
Origin Path                 : Leave empty (this means, defi
ne for root '/')
```


```
Protocol                    : Match Viewer
HTTP Port                   : 80
HTTPS                       : 443
```




```
Minimum Origin SSL Protocol : Keep it as is
Name                        : Keep it as is
Add custom header           : No header
Enable Origin Shield        : No
Additional settings         : Keep it as is
```

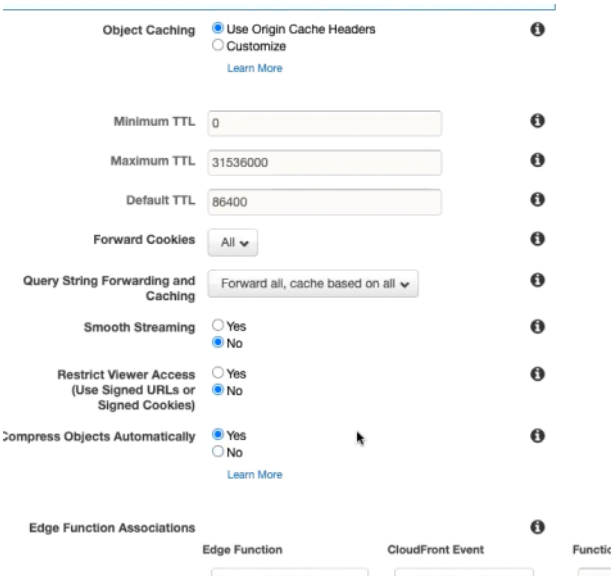
```
Default Cache Behavior Settings
```text
Path pattern                                : Defaul
t (*)
Compress objects automatically              : Yes
Viewer Protocol Policy                      : Redire
ct HTTP to HTTPS
Allowed HTTP Methods                        : GET, H
EAD, OPTIONS, PUT, POST, PATCH, DELETE
Cached HTTP Methods                         : Select
 OPTIONS
Cache key and origin requests
- Use legacy cache settings
```



Headerlarda neleri cashlemesini istiyorsak onlari secebiliyoruz

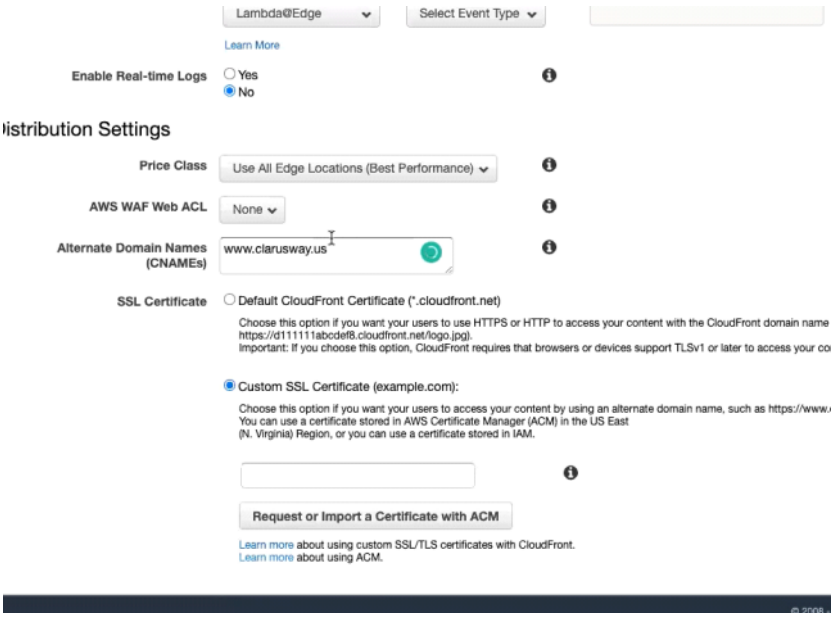
```
 Headers     : Include the following headers
```
```

```
 Add Header
    - Accept
    - Accept-Charset
    - Accept-Datetime
    - Accept-Encoding
    - Accept-Language
    - Authorization
    - Cloudfront-Forwarded-Proto
    - Host
    - Origin
    - Referrer
Forward Cookies                          : All
Query String Forwarding and Caching      : All
Other stuff                              : Keep them as are
```



- Distribution Settings

```text
Price Class                              : Use all edge loca
tions (best performance)
Alternate Domain Names                   : www.clarusway.us
SSL Certificate                          : Custom SSL Certif
icate (example.com) ---> Select your certificate creared be
fore
Other stuff                              : Keep them as are
```



Ayni zaman bu Ec2 lar ELB ye bagli olacak ELB nin en önemli özelligi EC2 lardaki yoguniuga bakip HelatCheck ler ile her hangir yoguniuk olmasi durumunda esit olarak bu

yuku EC2 lar arasinda dagitmak. Hem Trafigi yayiniliyor hem tek elden dagilimi sagliyor. Hemde Dis dunyadan gelen User in her hangi bir aksama olmadan EC2 ya ulasilmasini sagliyor.

ALB nin de önune bir CloudFront koyacagiz ki bu da br cash hizmeti sunacak. Videolar ve fotograflar S3 den cekilecek her seferinde ucret ödenmemesi icin bunu yapacaklar Cash hizmeti ile sagalanacak.

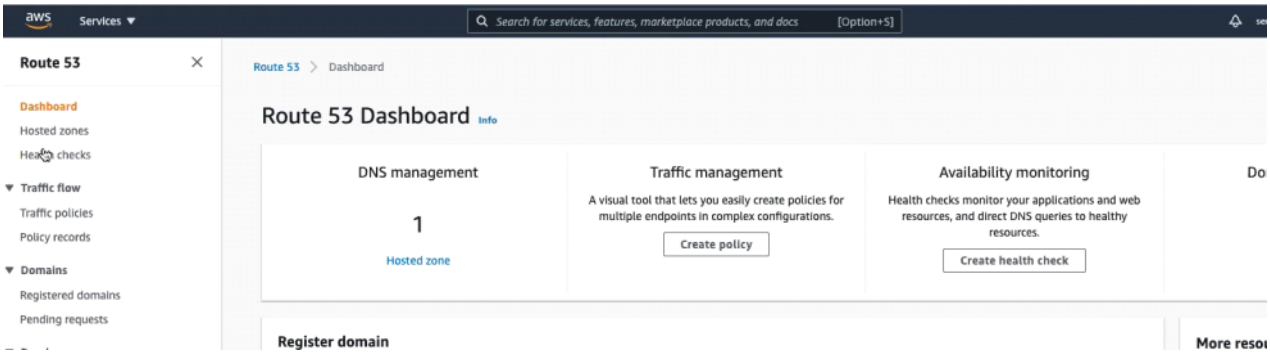
Production oirtaminda Cloudfront s3 un önune de kurulabiliyor Eger burda tanimlamak istersek s3 u orjin olarak gösterecektir.

Developerlar kodunu kullanici karsisina cikartmak icin Route53 failover senaryosu ile karsimiza cikacak.

Burada Failover senaryosu Route 53 olarakta bir ister atanmis durumdadir. Route53 de iki adet ent pointi takip eder. Birisi static website olur. Degiride uniq olarak caalisan web sitemizdir.basina herhangi bir durum gelmesi halinde secondary e trafigi yönlendirir.

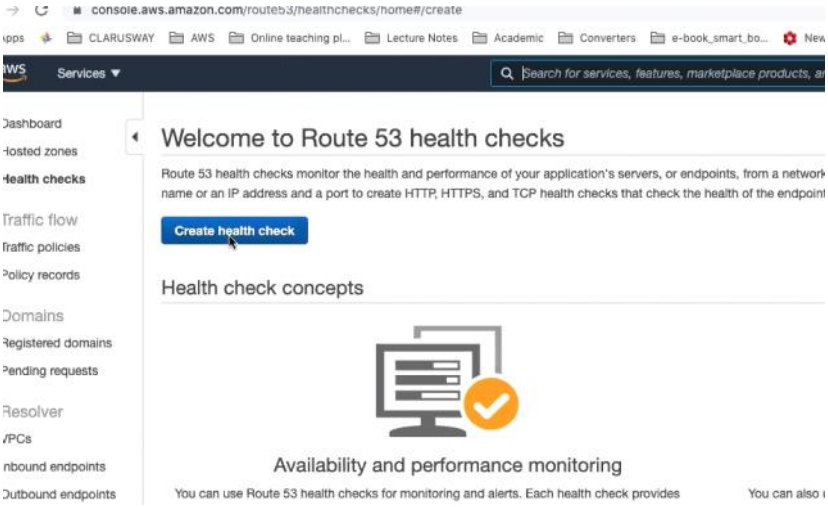
Userlar bu hailyle karsilarina bir sey cikarki mevcut olarak bir sorunun oldugunu ve bu sorunun bir sure sonra cözelecegi gibi seyler icerebilir.

Simdide Rote53 e gidecegziz Failover senaryosunu olustursmak icin.

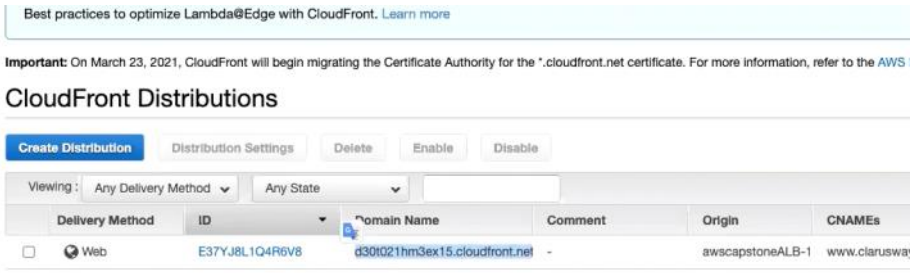


Step 15: Create Route 53 with Failover settings

Come to the Route53 console and select Health checks on the left hand menu. Click create health check

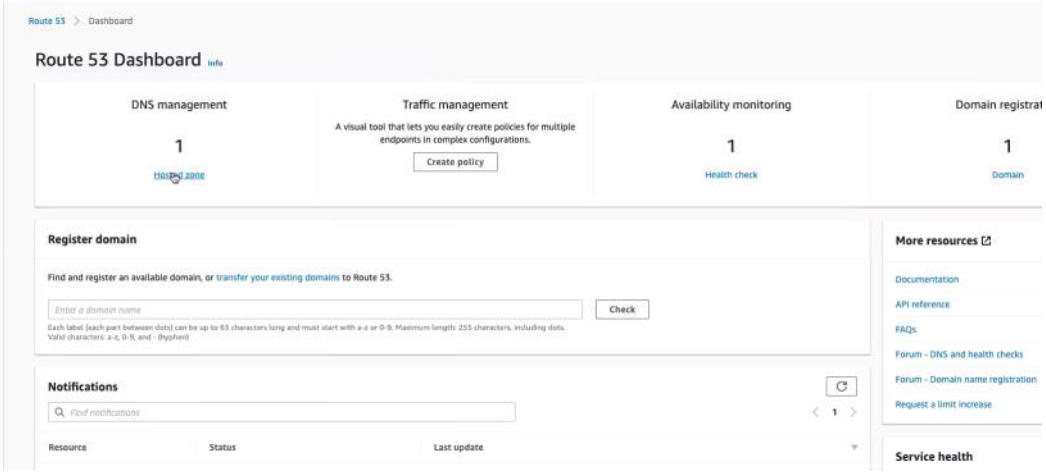


Configure health check
```text
Name                : aws capstone health check
What to monitor     : Endpoint
Specify endpoint by : Domain Name
```

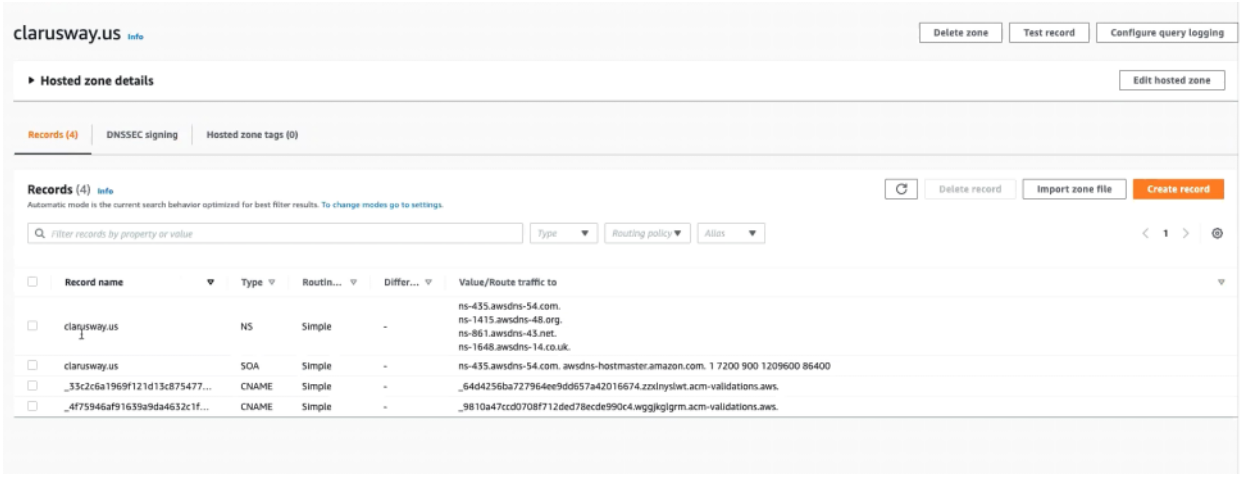


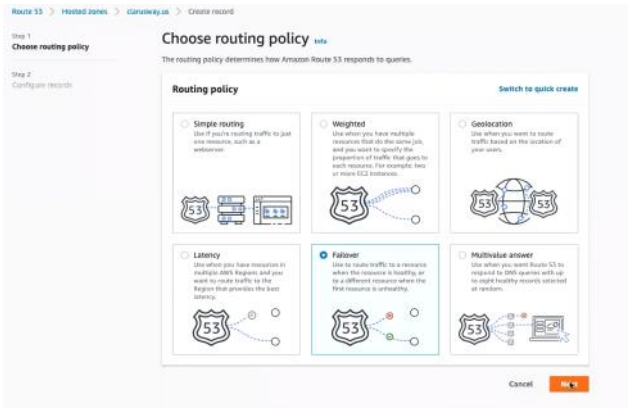
CloudFront taki Domain Name alip buraya kopyaliyoruz.

```text
Protocol            : HTTP
Domain Name         : Write cloudfront domain name
Port                : 80
Path                : leave it blank
Other stuff         : Keep them as are
```

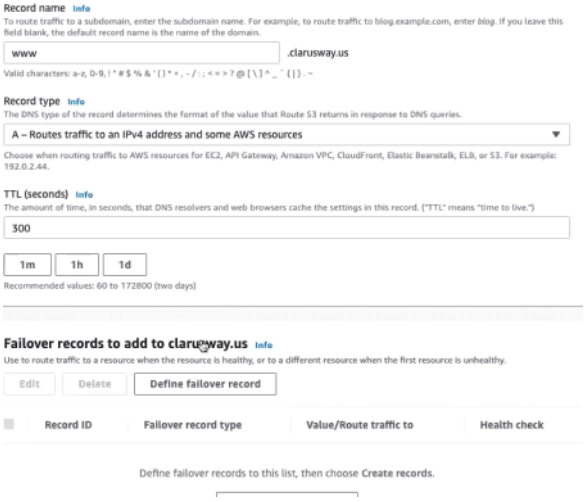


- Click Hosted zones on the left hand menu
- click your Hosted zone   : <YOUR DNS NAME>
- Create Failover scenario
- Click Create Record
- Select Failover ---> Click Next

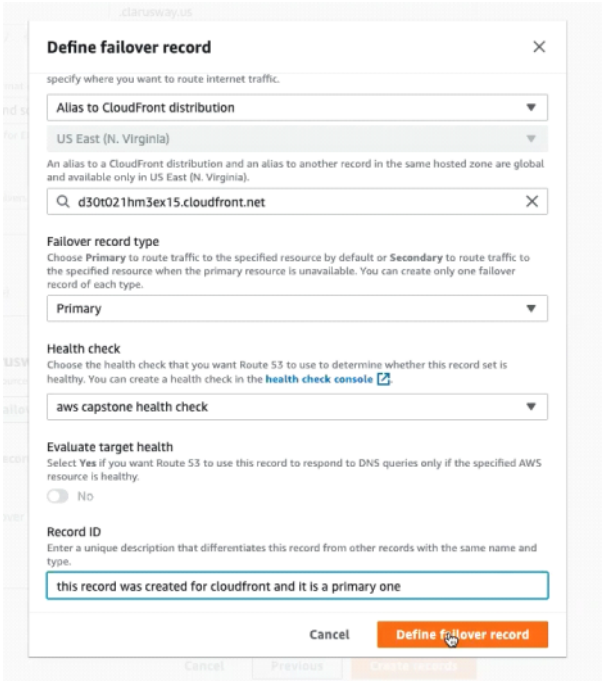





```text
Configure records
Record name              : www.<YOUR DNS NAME>
Record Type              : A - Routes traffic to an IPv4 add
ress and some AWS resources
TTL
```

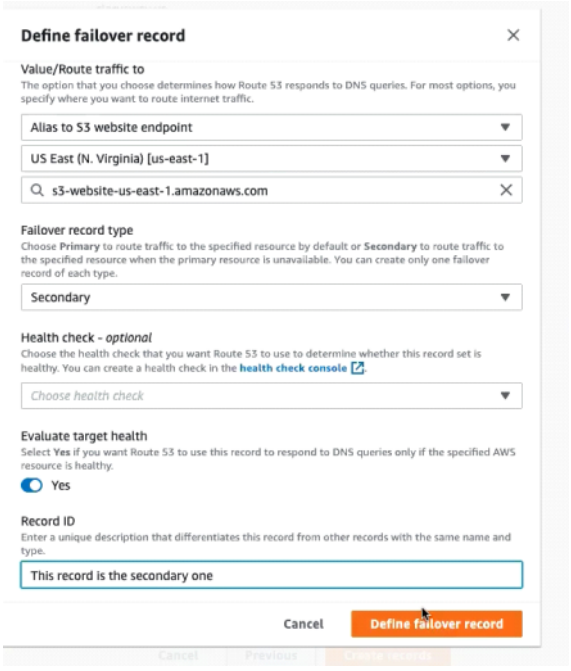



```text
                         : 300
---> First we'll create a primary record for cloudfront
Failover record to add to your DNS ---> Define failover rec
ord
Value/Route traffic to   : Alias to cloudfront distribution
                           - Select created cloudfront DNS
Failover record type     : Primary
Health check             : aws capstone health check
Record ID                : Cloudfront as Primary Record
```

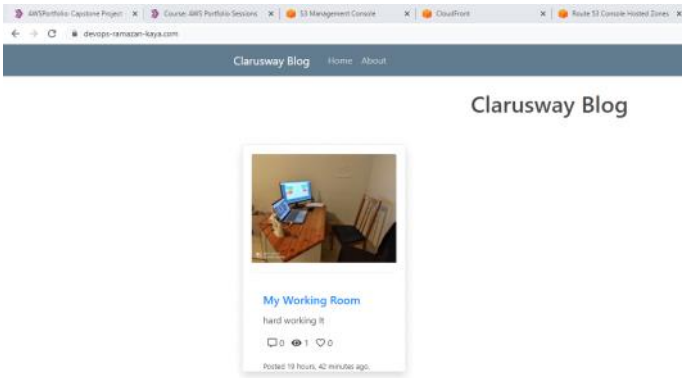



```
------------------------------------------------------------
----
---> Second we'll create secondary record for S3
Failover another record to add to your DNS ---> Define fail
over record
Value/Route traffic to  : Alias to S3 website endpoint
                          - Select Region
                          - Your created bucket name emerge
s ---> Select it
Failover record type    : Secondary
Health check            : No health check
Record ID               : S3 Bucket for Secondary record ty
pe
```

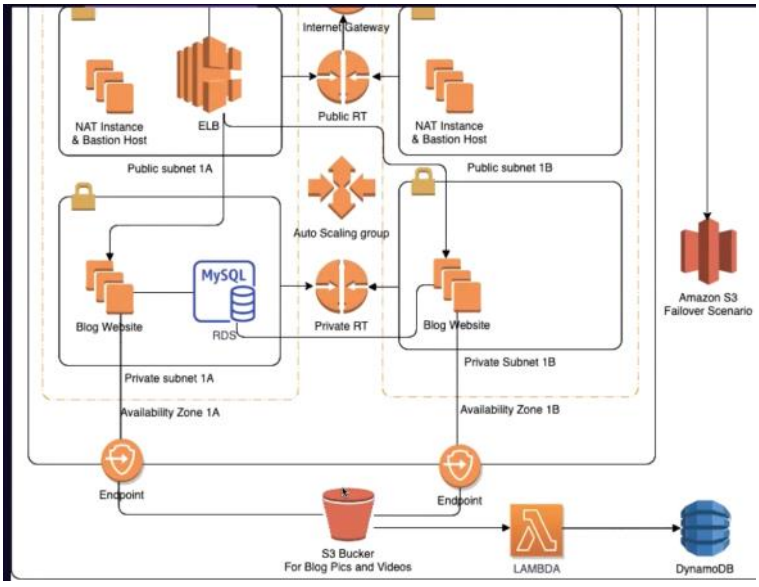
- click create records



Artikn sitemiz yayin yapmaya basladi ve secure bir yayindir .
Clou´dFront ile olusturdugumuz yayi su anda secure bir bagaalanti



Simdi secure bir baglanti yapmis oldulk

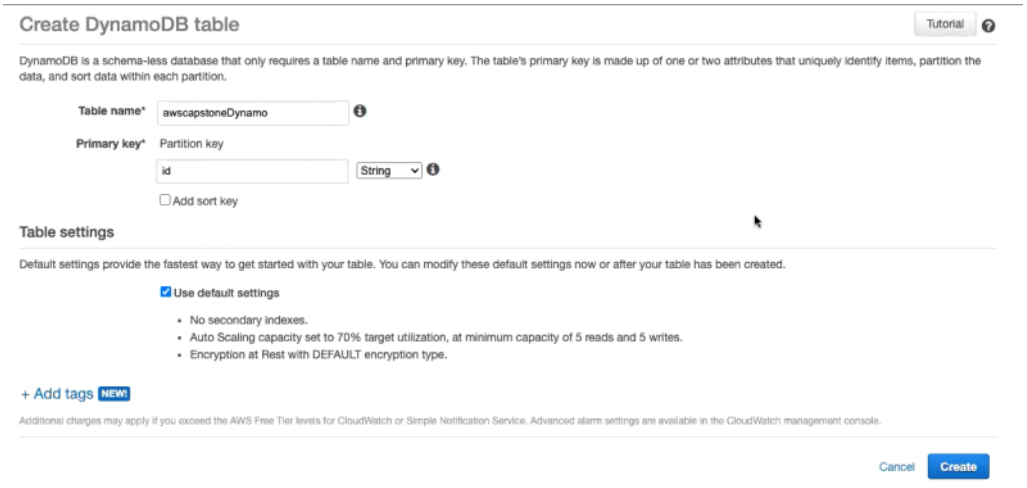


Developerlar s3 de atilan objelerin takibini yapmak icin DYNAMODB de liste haliunde bunun takibini yapmak istiyor lar

Step 16: Create DynamoDB Table

Go to the Dynamo Db table and click create table button
- Create DynamoDB table

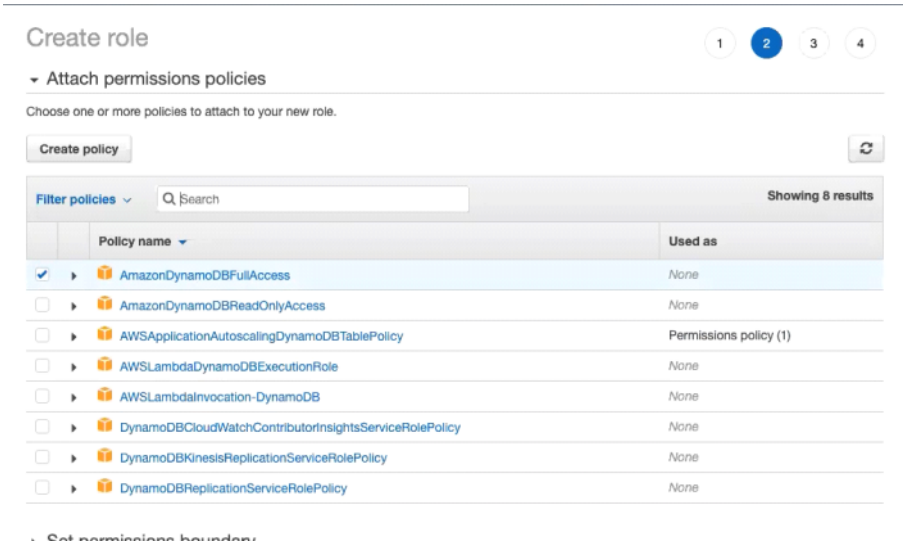
```text
Name            : awscapstoneDynamo
Primary key     : id
Other Stuff     : Keep them as are
click create
```



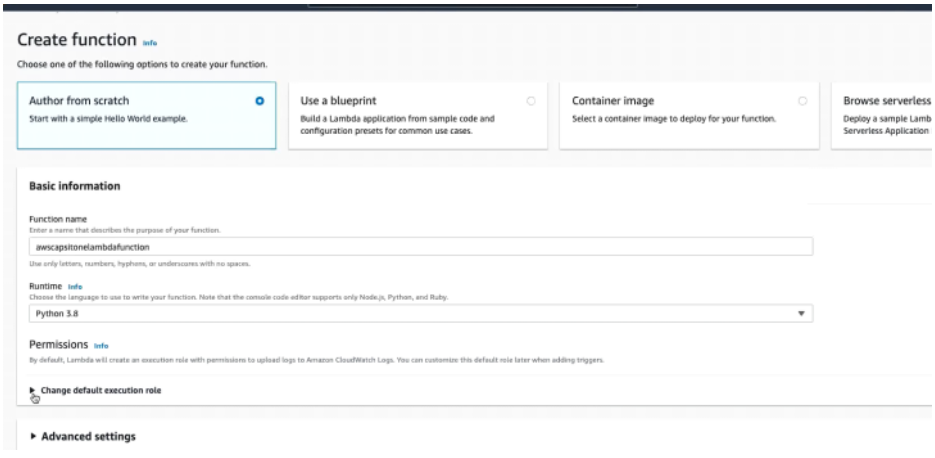
```
## Step 17-18: Create Lambda function

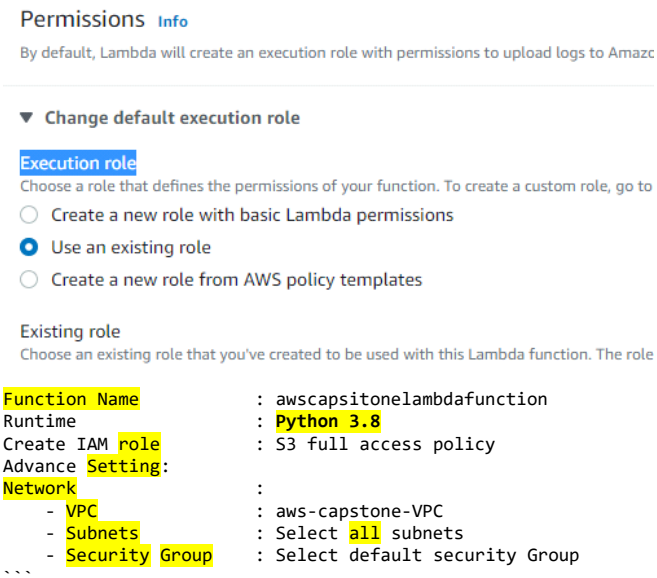
**Lambda Fonksiyonunun s3 e ulasmasi lazim yani konusmasi
lazim ayrica Dynamodb ile konususyor olamsi a+lazim ayrica
Network  altyapisinda irtibatalanmasi gerekiyor**

Before we create our Lambda function, we should create IAM
role that we'll use for Lambda function. Go to the IAM cons
ole and select role on the left hand menu, then create role
 button
```text
Select Lambda as trusted entity ---> click Next:Permission
Choose: - LambdaS3fullaccess,
        - Network Administrator
        - DynamoDBFullAccess
```



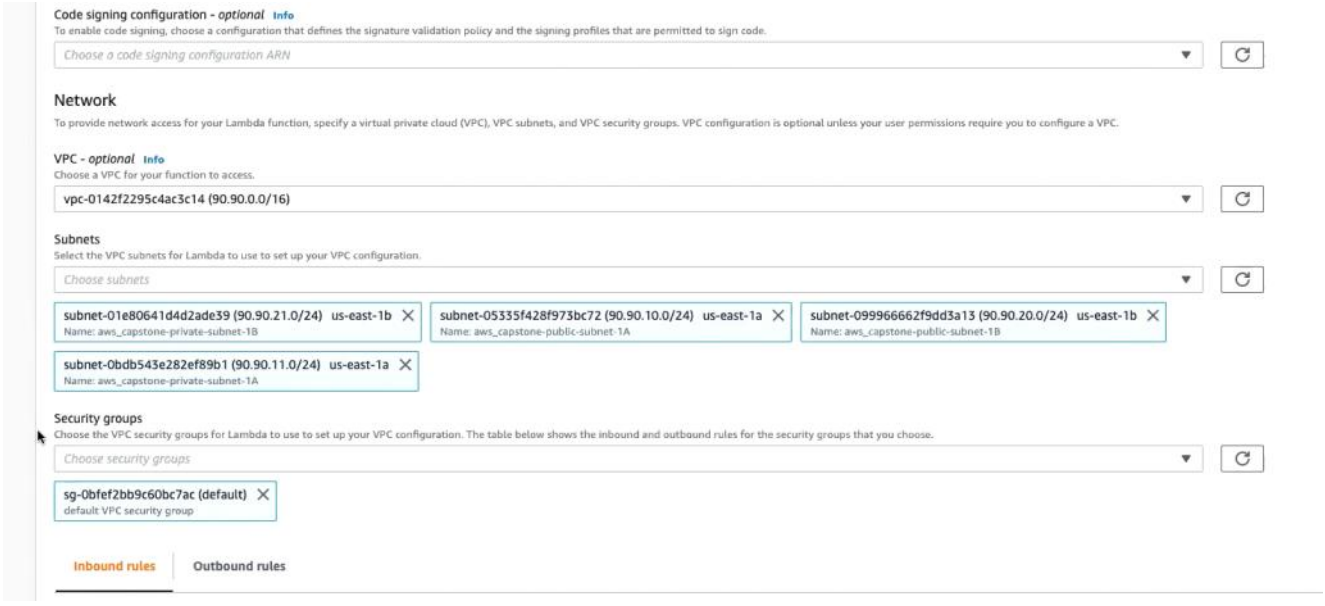

```
No tags
Role Name           : aws_capstone_lambda_Role
Role description    : This role give a permission to lambda
 to reach S3 and DynamoDB on custom VPC
```
then, go to the Lambda Console and click create function
- Basic Information
```text
```


```

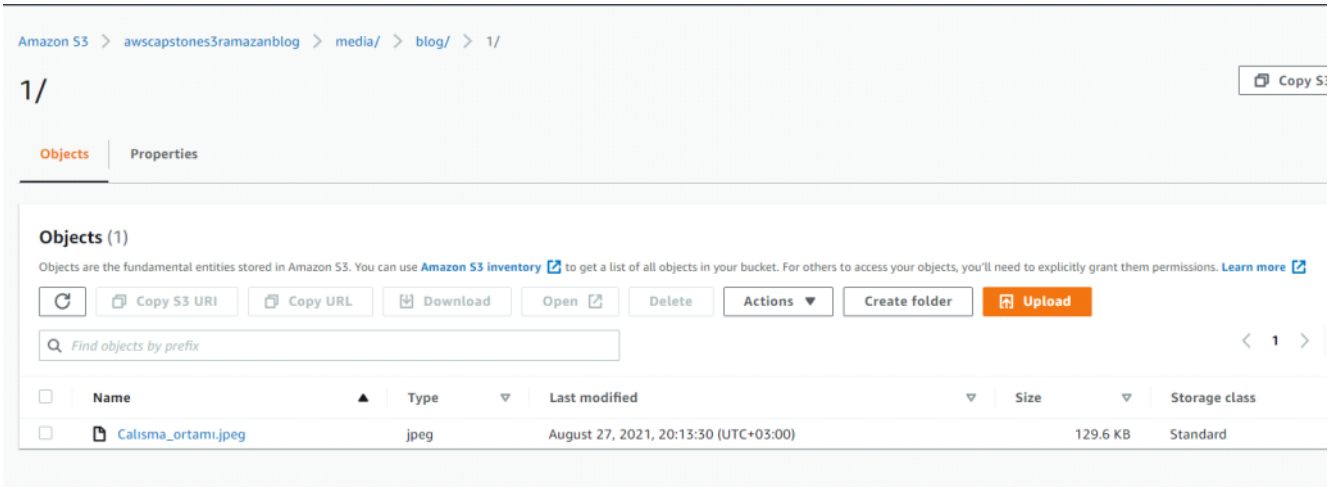



```
Function Name           : awscapsitonelambdafunction
Runtime                 : Python 3.8
Create IAM role         : S3 full access policy
Advance Setting:
Network                 :
    - VPC               : aws-capstone-VPC
    - Subnets           : Select all subnets
    - Security Group    : Select default security Group
```

- Now we'll go to the S3 bucket belongs our website and cre
ate an event to trigger our Lambda function.



Ve simdi de s3 de event olusturacagiz; bu s3 bucket imiizin altina ne dosya atilirsa bir event olsuturmasini isteyecegiz



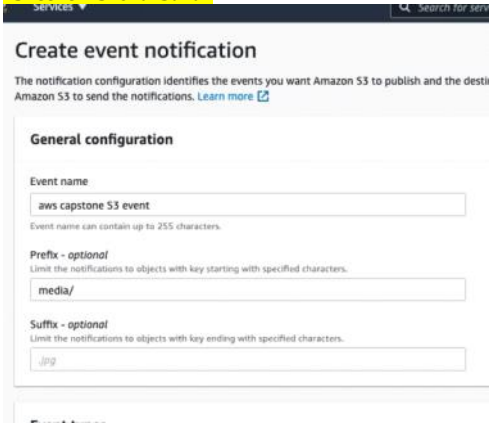
Bucketimizin-------- Properties kismina tiklayip daha somra--------- event notifications kismina tiklayacagiz

Evevnt Notifications S3 un bucktin altinda her hangi bir sey olusstugu taktirde onu event olusturur
Ve bu event i Lambda functiona tanitacagiz

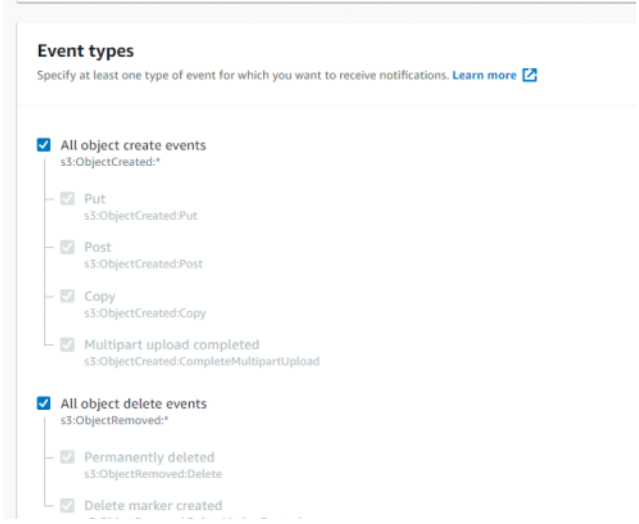
Step 17-18: Create S3 Event and set it as trigger for Lambda Function

Go to the S3 console and select the S3 bucket named `awscapstonec3<name>blog`.
- Go to the properties menu ---> Go to the Event notifications part
- Click create event notification for creating object
```text
Event Name              : aws capstone S3 event
Prefix                  : media/
```

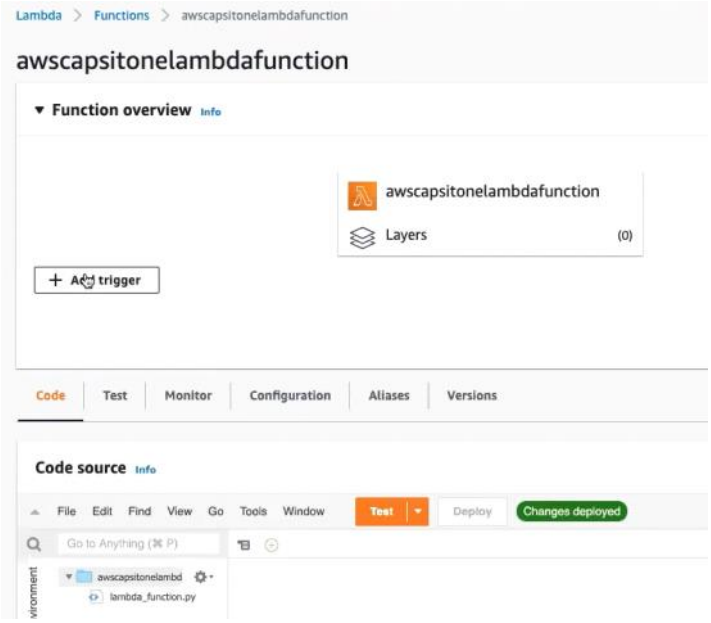
Egerki Suffix kismina jpeg belirtirseniz sedece jpegleri lsitelemis olursunuz



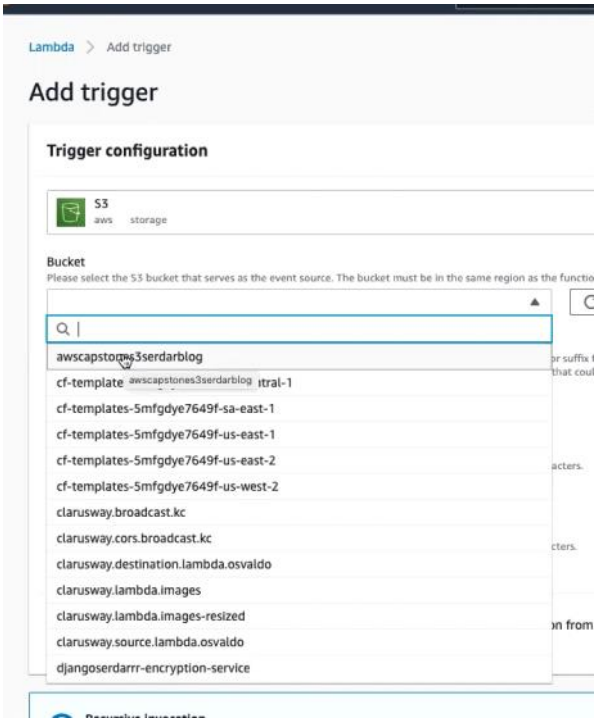

```text
Select                  :
    - All object create events
Destination             : Lambda Function
Specify Lambda function : Choose from your Lambda functions

Lambda funstion         : awscapstonelambdafunction
click save
```



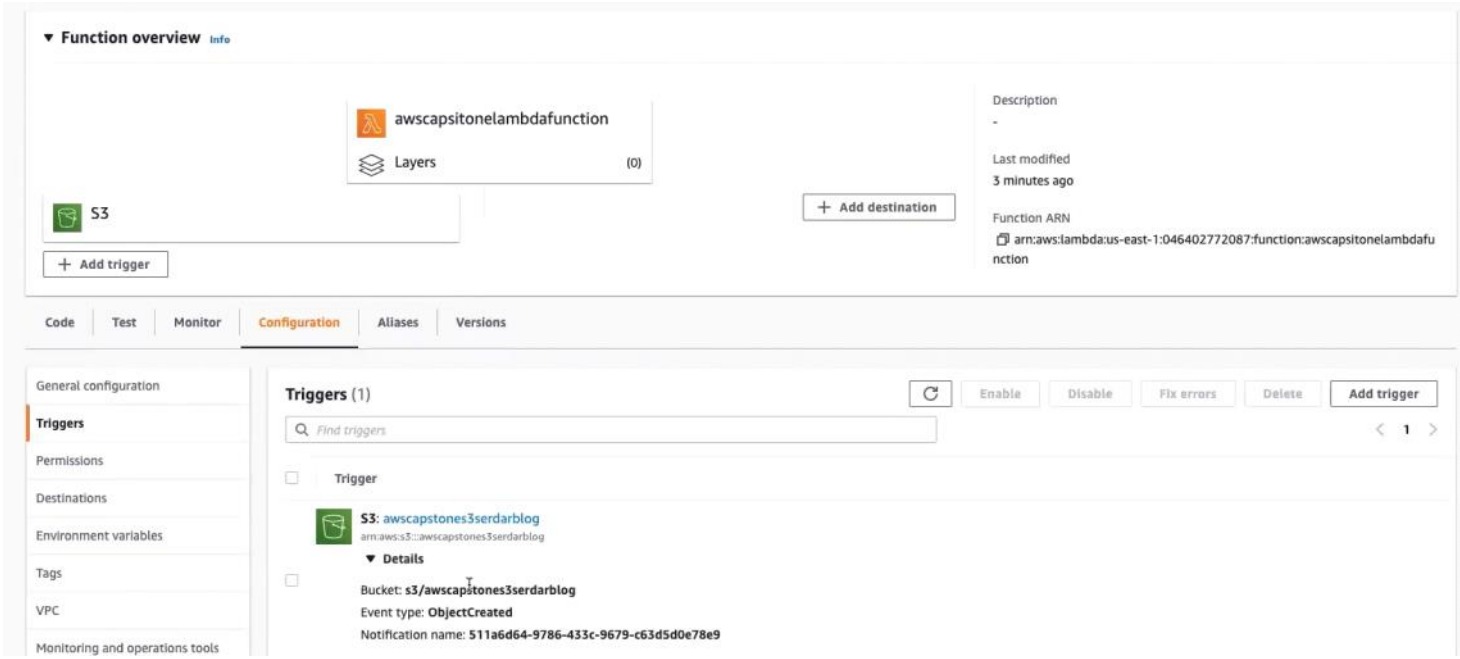
```text
```
**- After create an event go to the `awscapstonelambdafunctio
n` lambda Function and click add trigger on the top left ha
nd side.
- For defining trigger for creating objects**




```text
Trigger configuration   : S3
Bucket                  : awscapstonec3<name>blog
Event type              : All object create events
Check the warning message and click add ---> sometimes it s
ays overlapping situation. When it occurs, try refresh page
 and create a new trigger or remove the s3 event and recrea
te again. then again create a trigger for lambda function
```

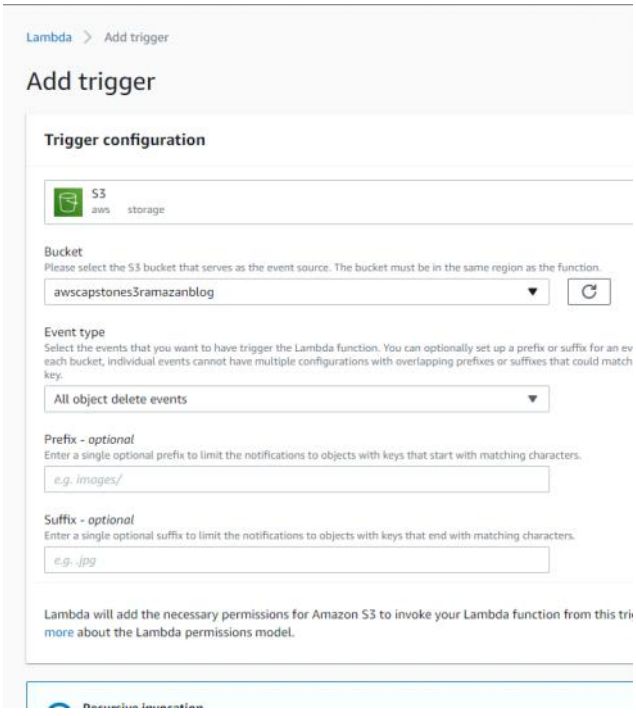


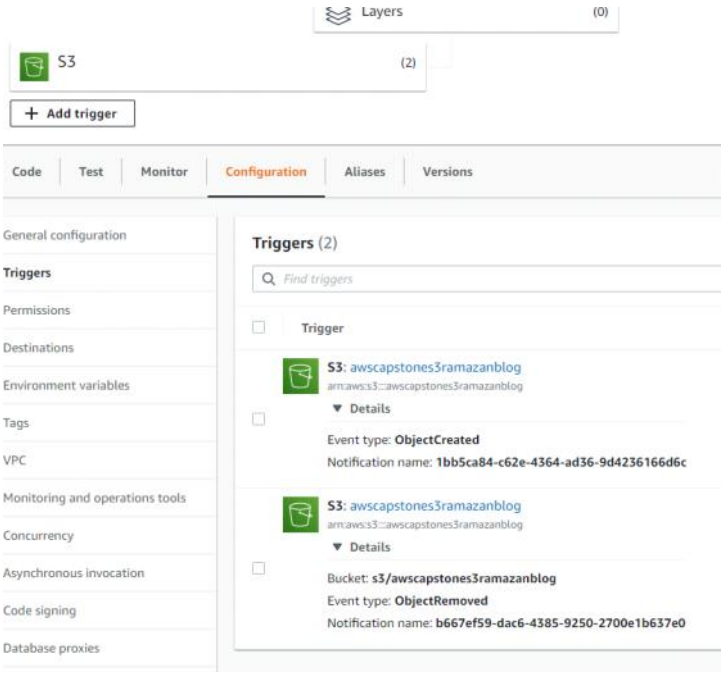
Hata verebiliyor eger hata verir ise lamda function s una
gidip silip tekarar Trigger edecegiz



- For defining trigger for **deleting objects**

```bash
Trigger configuration   : S3
Bucket                  : awscapstonec3<name>blog
Event type              : All object delete events
Check the warning message and click add ---> sometimes it s
ays overlapping situation. When it occurs, try refresh page
 and create a new trigger or remove the s3 event and recrea
te again. then again create a trigger for lambda function
```

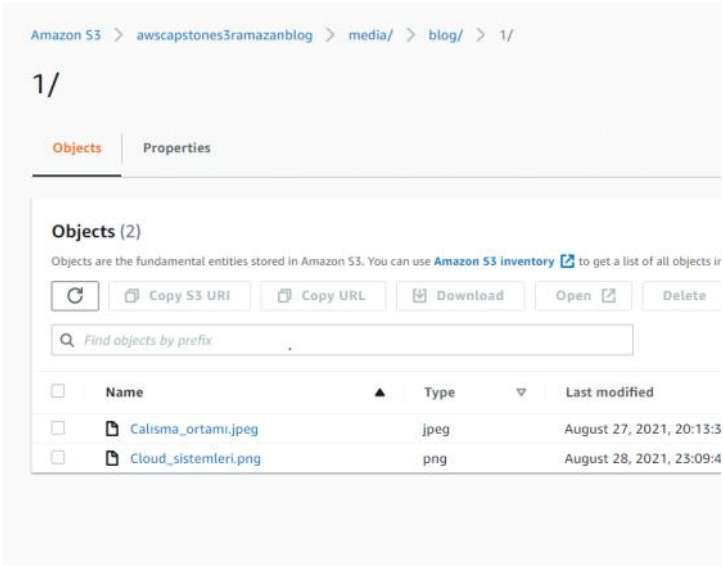


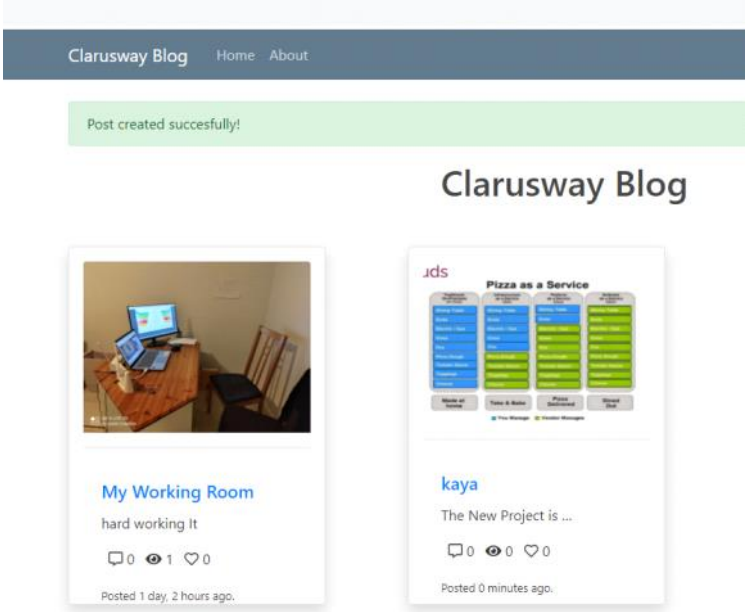


- **Go to the code part and select lambda_function.py** ---> remove default code and paste a code on below. If you give DynamoDB a different name, please make sure to change it into the code.

```python
import json
import boto3
def lambda_handler(event, context):
    s3 = boto3.client("s3")

    if event:
        print("Event: ", event)
        filename = str(event['Records'][0]['s3']['object']['key'])
        timestamp = str(event['Records'][0]['eventTime'])
        event_name = str(event['Records'][0]['eventName']).split(':')[0][6:]

        filename1 = filename.split('/')
        filename2 = filename1[-1]

        dynamo_db = boto3.resource('dynamodb')
        dynamoTable = dynamo_db.Table('awscapstoneDynamo')

        dynamoTable.put_item(Item = {
            'id': filename2,
            'timestamp': timestamp,
            'Event': event_name,
        })

    return "Lammda success"
```

- Click deploy and all set. go to the website and add a new post with photo, then control if their record is written on DynamoDB.
- WE ALL SET
- Congratulations!! You have finished your AWS Capstone Project





Yeni yukledigimiz nesnenin geldigini görebiliyoruz

Project-503 : Blog Page Application (Django) deployed on AWS Application Load Balancer with Auto Scaling, S3, Relational Database Service(RDS), VPC's Components, Lambda, DynamoDB and Cloudfront with Route 53

Description

The Clarusway Blog Page Application aims to deploy blog application as a web application written Django Framework on AWS Cloud Infrastructure. This infrastructure has Application Load Balancer with Auto Scaling Group of Elastic Compute Cloud (EC2) Instances and Relational Database Service (RDS) on defined VPC. Also, The Cloudfront and Route 53 services are located in front of the architecture and manage the traffic in secure. User is able to upload pictures and videos on own blog page and these are kept on S3 Bucket. This architecture will be created by Firms DevOps Guy.

Problem Statement

- **Your company has recently ended up a project that aims to serve as Blog web application on isolated VPC environment. You and your colleagues have started to work on the project. Your Developer team has developed the application and you are going to deploy the app in production environment.**
- **Application is coded by Clarusway Fullstack development team and given you as DevOps team. App allows users to write their own blog page to whom user registration data should be kept in separate MySQL database in AWS RDS service and pictures or videos should be kept in S3 bucket. The object list of S3 Bucket containing movies and videos is recorded on DynamoDB table.**
- **The web application will be deployed using Django framework.**
- **The Web Application should be accessible via web browser from anywhere in secure.**
- **You are requested to push your program to the project repository on the Github. You are going to pull it into the webservers in the production environment on AWS Cloud.**

In the architecture, you can configure your infrastructure using the followings,

- **The application stack should be created with new AWS resources.**
- **Specifications of VPC:**
  - **VPC has two AZs and every AZ has 1 public and 1 private subnets.**
  - **VPC has Internet Gateway**
  - **One of public subnets has NAT Instance.**
  - **You might create new instance as Bastion host on Public subnet or you can use NAT instance as Bastion host.**
  - **There should be managed private and public route tables.**
  - **Route tables should be arranged regarding of routing policies and subnet associations based on public and private subnets.**
- **You should create Application Load Balancer with Auto Scaling Group of Ubuntu 18.04 EC2 Instances within created VPC.**
- **You should create RDS instance within one of private subnets on created VPC and configure it on application.**
- **The Auto Scaling Group should use a Launch Template in order to launch instances needed and should be configured to;**
  - **use all Availability Zones on created VPC.**
  - **set desired capacity of instances to 2**
  - **set minimum size of instances to 2**
  - **set maximum size of instances to 4**
  - **set health check grace period to 90 seconds**
  - **set health check type to ELB**
  - **Scaling Policy --> Target Tracking Policy**
    - **Average CPU utilization (set Target Value %70)**
    - **seconds warm up before including in metric ---> 200**
    - **Set notification to your email address for launch, terminate, fail to launch, fail to terminate instance situations**
- **ALB configuration;**

        - Application Load Balancer should be placed within a security group which allows HTTP (80) and HTTPS (443) connections from anywhere.
        - Certification should be created for secure connection (HTTPS)
            - To create certificate, AWS Certificate Manager can be utilized.
        - ALB redirects to traffic from HTTP to HTTPS
        - Target Group
            - Health Check Protocol is going to be HTTP
    - The Launch Template should be configured to;
        - Prepare Django environment on EC2 instance based on Developer Notes,
        - Download the "clarusway_aws_capstone" folder from Github repository,
        - Install the requirements using requirements.txt in 'clarusway_aws_capstone' folder
        - Deploy the Django application on port 80.
        - Launch Template only allows HTTP (80) and HTTPS (443) ports coming from ALB Security Group and SSH (22) connections from anywhere.
        - EC2 Instances type can be configured as t2.micro.
        - Instance launched should be tagged Clarusway AWS Capstone Project
        - Since Django App needs to talk with S3, S3 full access role must be attached EC2s.
    - For RDS Database Instance;
        - Instance type can be configured as db.t2.micro
        - Database engine can be MySQL with version of 8.0.20.
        - RDS endpoint should be addressed within settings file of blog application that is explained developer notes.
        - Please read carefully "Developer notes" to manage RDS sub settings.
    - Cloudfront should be set as a cache server which points to Application Load Balance with following configurations;
        - The cloudfront distribution should communicate with ALB securely.
        - Origin Protocol policy can be selected as HTTPS only.
        - Viewer Protocol Policy can be selected as Redirect HTTP to HTTPS
    - As cache behavior;
        - GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE methods should be allowed.
        - Forward Cookies must be selected All.
        - Newly created ACM Certificate should be used for securing connections. (You can use same certificate with ALB)
    - Route 53
        - Connection must be secure (HTTPS).
        - Your hostname can be used to publish website.
        - Failover routing policy should be set while publishing application
            - Primary connection is going to be Cloudformation
            - Secondary connection is going to be a static website placed another S3 bucket. This S3 bucket has just basic static website that has a picture said "the page is under construction" given files within S3_static_Website folder
            - Healthcheck should check If Cloudfront is healthy or not.
    - As S3 Bucket
        - First S3 Bucket
            - It should be created within the Region that you created VPC
            - Since development team doesn't prefer to expose traffic between S3 and EC2s on internet, Endpoint should be set on created VPC.
            - S3 Bucket name should be addressed within

configuration file of blog application that is explained developer notes.

- Second S3 Bucket
  - This Bucket is going to be used for failover scenario. It has just a basic static website that has a picture said "the page is under construction"

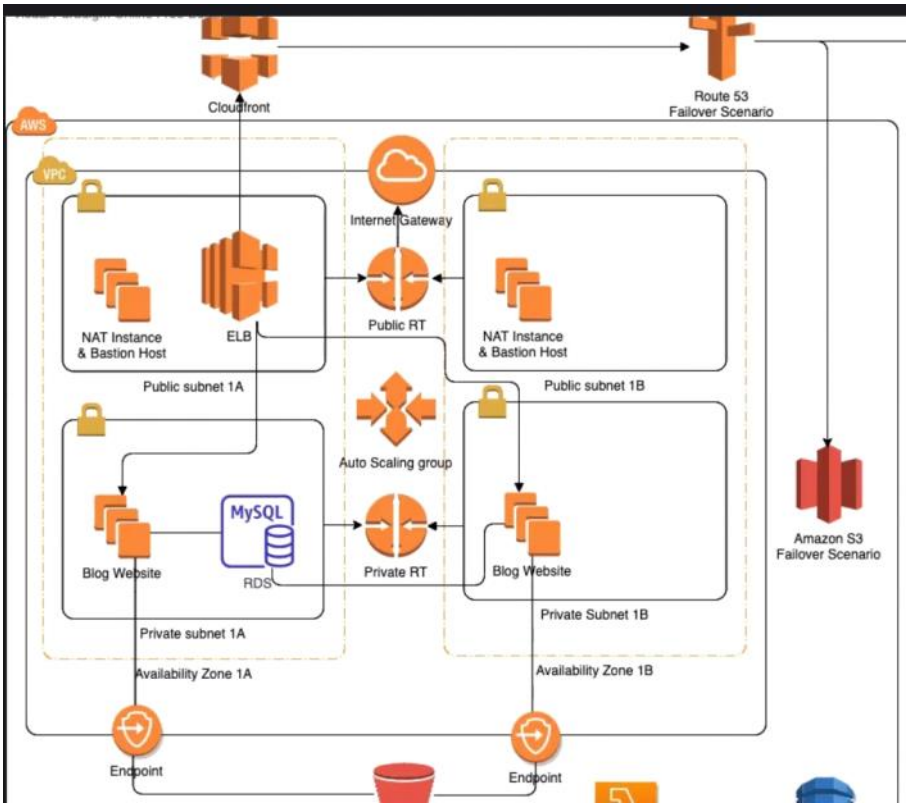
- To write the objects of S3 on DynamoDB table
  - Lambda Function
    - Lambda function is going to be Python 3.8
    - Python Function can be found in github repo
    - S3 event is set as trigger
    - Since Lambda needs to talk S3 and DynamoDB and to run on created VPC, S3, DynamoDB full access policies and NetworkAdministrator policy must be attached it
    - S3 Event must be created first S3 Bucket to trigger Lambda function
  - DynamoDB Table
    - Create a DynamoDB table which has primary key that is id
    - Created DynamoDB table's name should be placed on Lambda function.

Project Skeleton

```
clarusway_blog_proj (folder)
|
|----Readme.md               # Given to the students (Definition of the project)
|----src (folder)           # Given to the students (Django Application's )
|----requirements.txt       # Given to the students (txt file)
|----lambda_function.py     # Given to the students (python file)
|----developer_notes.txt    # Given to the students (txt file)
```

Expected Outcome

From < https://github.com/clarusway/clarusway-aws-8-21/tree/main/aws/projects/Project-503-Capstone-Project-Blog-Page-App-(Django)-on-AWS-Environment>



(Janko flask in gelimis hali diyebiliriz bir framworktur.)

Developerlar bir jango framworku ile block aplicationu yazdilar. Blok sayfasi ile herkes kendi blogunu sayfasini yazabiliyor. Fotograflarini ve videolarini paylasabiliyor.

Videolarini paylasacaklari icin bi login olunmasi ve bir yerde de user ve passwordlerin tutulmasi store edilmes gerekiyor.

Store denilince aklimiza AWS de s3 geliyor. Developerlar s3 de tutulmasini istiyorlar. S3 obje tabanli bir storeage idi. Ne kadar cagirilir ise o kadar ucretlendirmeye tabiydi.

Login datalarda database de tutulacak. Bunun icinde Mysql tercih edilmis .

Bunun icin Custom VPC kuracagiz . Production ortaminda da custom vpc kullanilmasi tercih ediliyor. Cunku VPC lerde Database gibi bazi makinalrimizin private subnetlerde kurulmasi gerekiyor ki bir bakima bunlar bierer firewal olarak nitelendiriliyor.

Custom VPC nin isterlerine bakalim;

- **2 adet Availibity zonumuz var ve bu Availibility Zon un icerisinde bir private ve birde public subnetlerimiz olacak (Availibity Zone : Bir region içerisinde bulunan, birbirlerine yakın konumlandırılmış, küçük sunucu tarlalarına verilen isimdir. Bir region içerisinde birden fazla availability zone bulunabilir ve herbir zone harf ile adlandırılır. Örneğin Oregon region'unda us-west-2a, us-west-2b ve us-west-2c olmak üzere üç adet AZ (availability-zone) bulunmaktadır.**
  From < https://www.mobilhanem.com/aws-dersleri-aws-nedir/>)

Private ve public subnetler vpc icerisine kurulan izole network çözumleridir.

- **Bi izole network icerisine farkli resourclari koruma altina alabiliyoruz.**
- **Ortak bir subnet kumesini bir subnet gurubunda toplayabiliyorsunuz.**
- **Ayrica farkli subnetlerde farkli replicalar bulundurarak failover durumlarinda afet yangin olasi tabi durumlarda mevcut subnetmize bir sey olmasi durumunda diger subneti devreye**

sokabiliyoruz.

Subnetler in kendi aralarinda irtibati kurmalari icin veyahut subnetlerin icerisinde ki resourcelarin dis dunyay ya ulasmak istediklerinde, veya her hangi bir porta ulasmak istediklerinde, Rote Tablet I kullaniriz.Route Table ler subnet icerisinde bir novigasyon cihazidir. Rota Tableler icerisinde olusturdugumuz Rule lar ile bir tarafa gitmesini veyahut da gitmemesini sagliriz.

Özellikle de private Route Tablet yapilmasinin sebebi bir firewal yapilarak dis dunyadan ulasilamayacak bir hale girmesini sagiamaktir.

Ayrica videolar ve resimlerin s3 de saklanacagini ve dis dunyay ya expoz olmsaini istemedigimizi dusunursek subnetler ile S3 bucket lar arsinda da bir gateway end point kuracagiz

Ardindan Jango web sitemiz bir EC2 lar gurubu uzerinde tutulacak bunuda Autoscaling group saglayacak. Bunun öncesi de bir Launch Template kuracagiz ki L .Template ile Auto Scalinge makinalrin kurulumunu salayalim . Bizim istedigimiz desire kapasitede minimum ve max seviyede makinalar ayaga kalkacak . Burda bir Target Tracing Policy olusturacagiz.

Ayni zaman bu Ec2 lar ELB ye bagli olacak ELB nin en önemli özelligi EC2 lardaki yoguniuga bakip HelatCheck ler ile her hangir yoguniuk olmasi durumunda esit olarak bu yuku EC2 lar arasinda dagitmak. Hem Trafigi yayiniiyor hem tek elden dagiiimi sagiiyor. Hemde Dis dunyadan gelen User in her hangi bir aksama olmadan EC2 ya ulasiimasini sagiiyor.

ALB nin de önune bir CloudFront koyacagiz ki bu da br cash hizmeti sunacak. Videolar ve fotograflar S3 den cekilecek her seferinde ucret ödenmemesi icin bunu yapacaklar Cash hizmeti ile sagalanacak.

Production oirtaminda Cloudfront s3 un önune de kurulabiliyor Eger burda tanimlamak istersek s3 u orjin olarak gösterecektir.

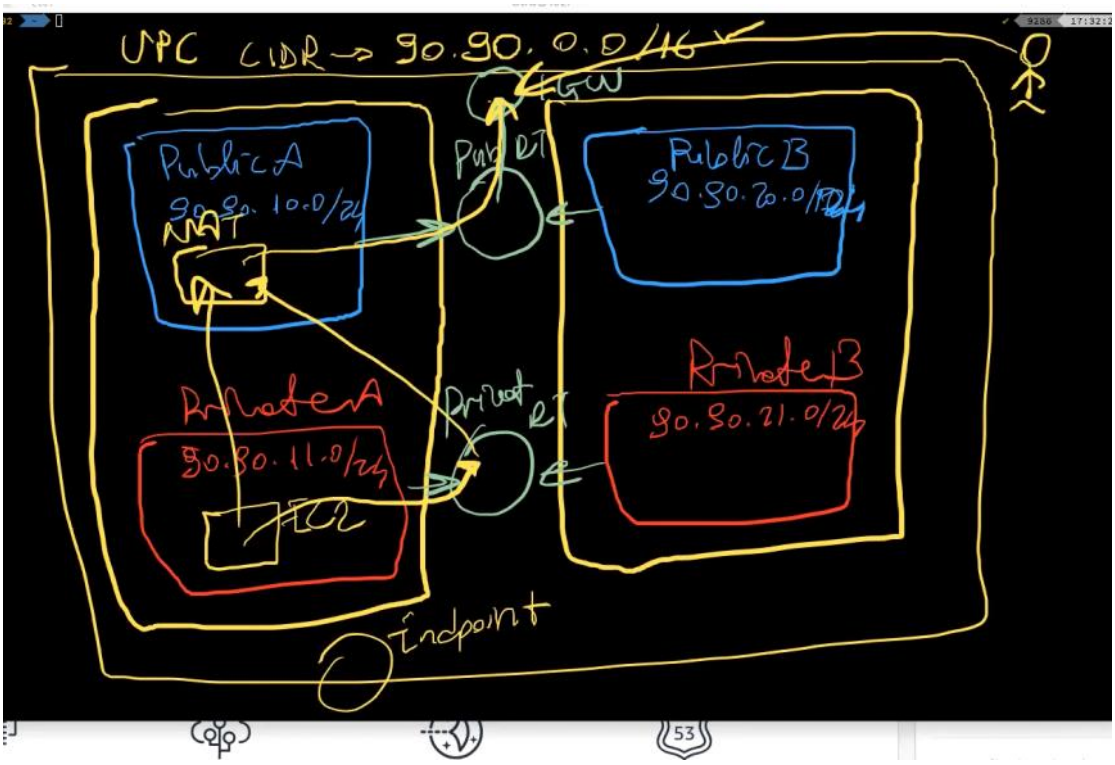
Developerlar kodunu kullanici karsisina cikartmak icin Route53 failover senaryosu ile karsimiza cikacak.

Burada Failover senaryosu Route 53 olarakta bir ister atanmis durumdadir. Route53 de iki adet ent pointi takip eder. Birisi statlc website olur. Degiride uniq olarak caalisan web sitemizdir.basina herhangi bir durum gelmesi halinde secondary e trafigi yönlendirir.

Userlar bu haliyle karsilarina bir sey cikarki mevcut olarak bir sorunun oldugunu ve bu sorunun bir sure sonra cözelecegi gibi seyler icerebilir.

Birde Deeveloperiatr S3 un icerisine konulan her seyi takip etmek istiyorlar. S3 e konan her hangi bir obje oldugunda , bir event olusacak bu event bir lambda function olusturacak bu Lambda Functon da Dynamodb icerisine bilgileri yazdiracak.

Dolayisiyla yazilan ve silinen bilgileri Dynamodb de grebilecegiz.



Konsolumuza geciyoruz :

Vpc de CIDR Bloklari vardir. Biz bunlari Private ipler icin tanimliyoruz. Public oalnlari EC2 kendisi veriyor.

Tanimlyacagimiz vpc de AWS 90.90.0.0/16 toplamda 65 bin ip tanimlayabilecegimiz BIR SINIRLAMA TANIYOR

Bir subneti private yada Public yapan disariya acilmasidir.

Birde bunlara Private ve Public Rote tablet ayarlayacagiz ki Routa tabletlere bir kural tanimlar ve dis dunnya ya acilmasini isterseniz Public istemez iseniz Private olurlar

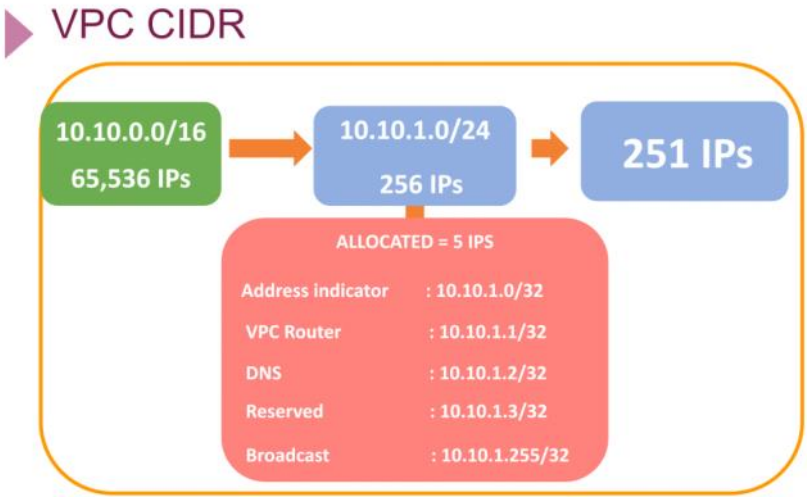
Ayrica Bir de end Point ekleyecegiz S3 ler ve DYNAMODB Cloud Watch lar icin kullanilir ve burada s3 e irtibat saglamk iclin kullanacagiz.

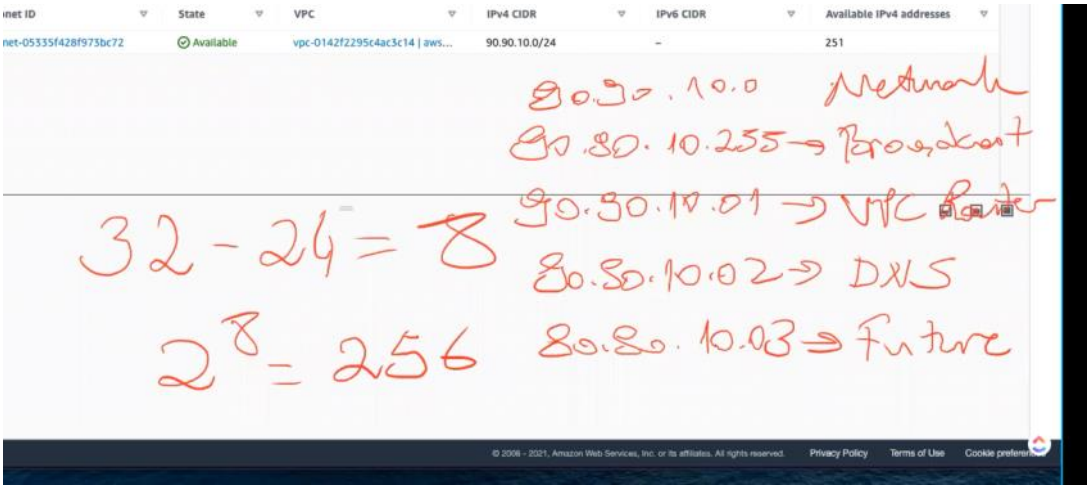
Biz Internet Gate way araciligiyla Öncelikle Public Iplerimize giris yapabiliyoruz. Ancak Private gecis yapmak icin Publiclede kuracagimiz Buston Hostlar ile siblama tahtalari ile Gecis yapacagiz.

Ayrica icerdeki makinanin disariyla acilmasi icin ne yapmamiz gerekiyordu dersek bunun icinde Nat Instance veýahut nat Gateway kurmamiz gerekiyor Bastion Host oarak da göruluyor Ancak Nat Instance da dikkat edilmesi gereken konu http, HTTPS ve SSh anywhere olarak her yerden acmamiz gerekiyor

Ayrica birde icerdeki makinanin disardaki ile görusmesi icin ne yapmamiz gerekiyor.; Nat instance veyahut Nat Gateway kurmaiz gerekiyor NatGateway AWS manage bir sistemdir. Yani biz kontrol etmiyorduk. Nat Instance ise özel bir instance tipidir. Public Subnete yerlestiririz. Private Instance da bulunan EC2 muz Private Routa Table araciligiyla Nat Instance uzerinden dis dunya ile irtibat saglayabiliyoruz. Ayrica Bastion Host olarakta

kullanilabilinir ancak HTTP,HTTPS ve ssh in anywere olarak her yöne acilmasi gerekiyor.



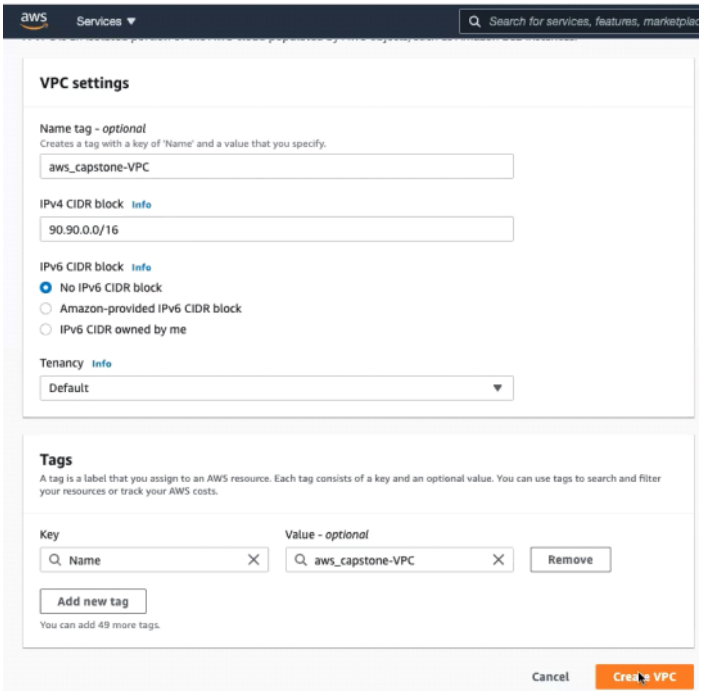


256 - 1 = 255 kullanilabilir 5 ntanesini AWS kendisine Reeverse eder.

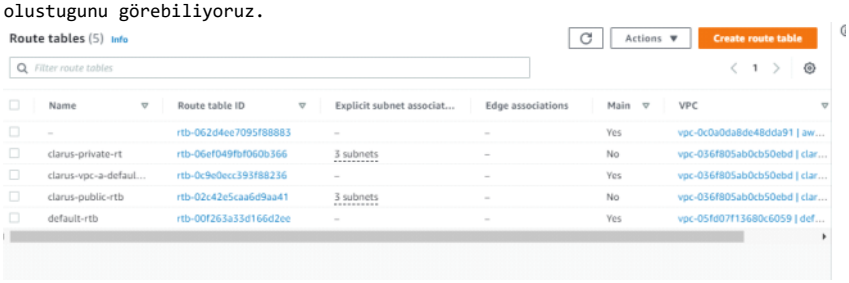
Ilk olarak VPC mizi yapmak icin konsola gidiyoruz

```
# Project-503 : Blog Page Application (Django) deployed on A
WS Application Load Balancer with Auto Scaling, S3, Relation
al Database Service(RDS), VPC's Components, DynamoDB and Clo
udfront with Route 53 (STUDENT_SOLUTION)
## Description

The Clarusway Blog Page Application aims to deploy blog appl
ication as a web application written Django Framework on AWS
 Cloud Infrastructure. This infrastructure has Application L
oad Balancer with Auto Scaling Group of Elastic Compute Clou
d (EC2) Instances and Relational Database Service (RDS) on d
efined VPC. Also, The Cloudfront and Route 53 services are l
ocated in front of the architecture and manage the traffic i
n secure. User is able to upload pictures and videos on
own blog page and these are kept on S3 Bucket. This a
rchitecture will be created by Firms DevOps Guy.



# Steps to Solution

### Step 1: Create dedicated VPC and whole components

    ### VPC
    - Create VPC.
        create a vpc named `aws_capstone-
VPC` CIDR blok is `90.90.0.0/16`
        no ipv6 CIDR block
```

tenancy: default
- select `aws_capstone-VPC` VPC, click `Actions` and `enable DNS hostnames` for the `aws_capstone-VPC`.



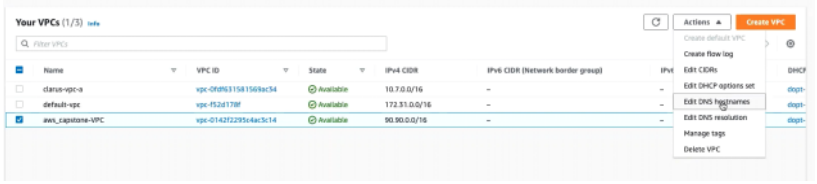
vpc ile birlikte route tablet in default olarak olustugunu görebiliyoruz.



Buna da public Routa tablet deyip bir taNE DE PRIVATE OLUSTurabiliriz

Vpc icerisinde butun resource slarin bir birleri ile ,konusmasini istiyorsak edit DNS Hostaname den ENABLE YAPIYORUZ

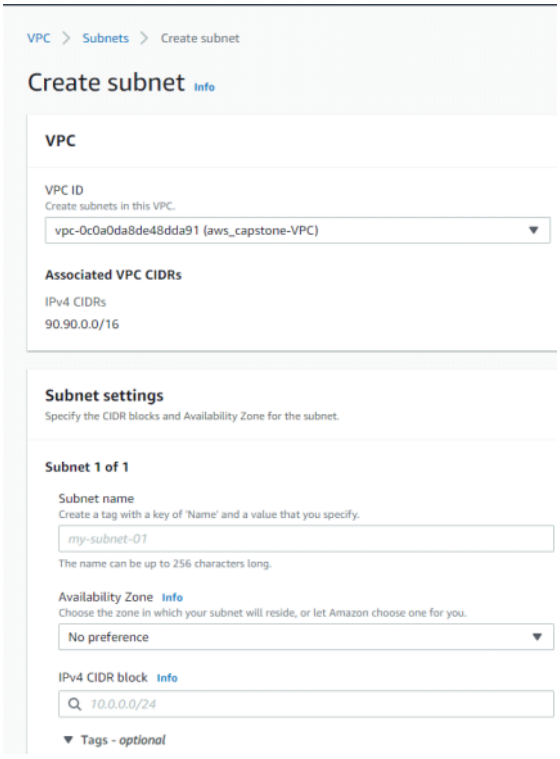
Enable olmaz is e bu vpc icerrisine bir DNS atanmaz ve bir bir leri ile konusmalari zora girer.



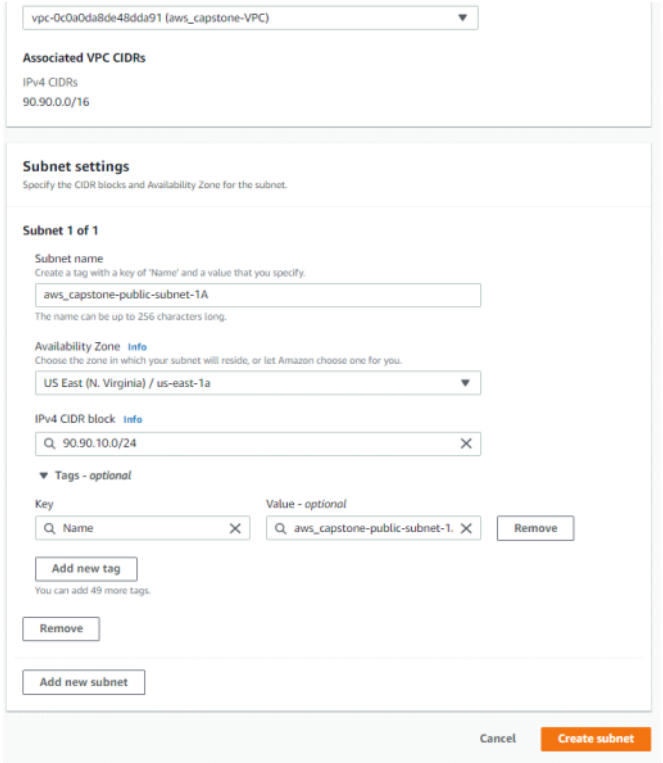
Simdid e subnet olusturacagiz

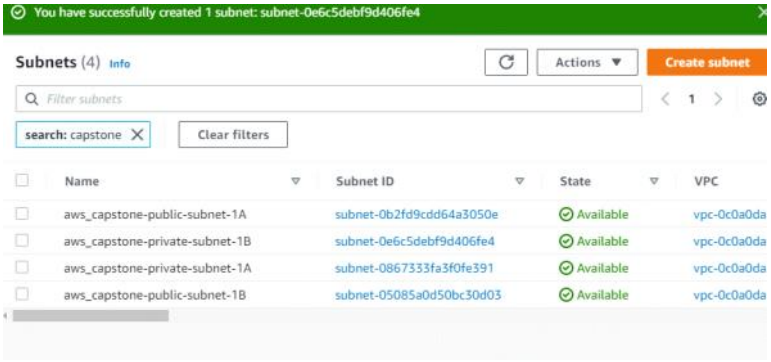
Subnets
- Create Subnets
- Create a public subnet named `aws_capstone-public-subnet-1A` under the vpc aws_capstone-VPC in AZ us-east-1a with 90.90.10.0/24
- Create a private subnet named `aws_capstone-private-subnet-1A` under the vpc aws_capstone-VPC in AZ us-east-1a with 90.90.11.0/24
- Create a public subnet named `aws_capstone-public-subnet-1B` under the vpc aws_capstone-VPC in AZ us-east-1b with 90.90.20.0/24
- Create a private subnet named `aws_capstone

-private-subnet-1B` under the vpc aws_capstone-VPC in AZ us-east-1b with 90.90.21.0/24
    - Set `auto-assign IP` up for public subnets. **Select each public subnets and click Modify "auto-assign IP settings**" and select "Enable auto-assign public IPv4 address"

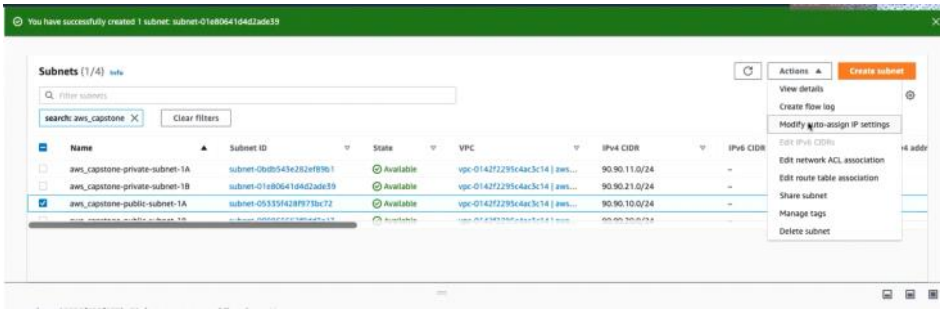


Vpc id olarak daha önce olusturdugumuz vpc yi seciyoruz.





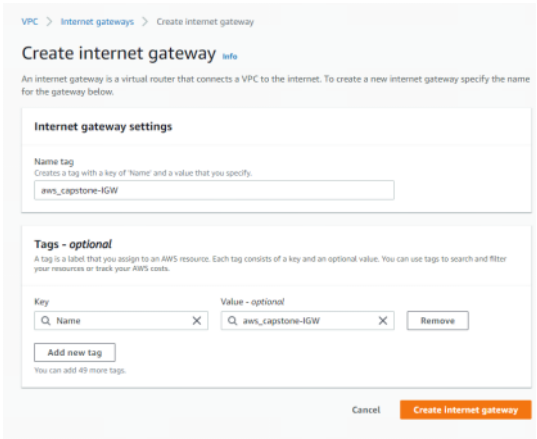
Public Subnetler icin Public ip atanmasi gerekiyor. Bunu bizde kontroledebiliyoruz Bunun icin Public Ipleri Actions bölumunden Modify deyip daha sonra enable ediyoruz.



Subnetleri Private ve Public Yapmak icin Route Tabletleri de duzenleyecegiz

Public routa Tabletleri Public yapmak icin Interne Gate Way tanimlayacagiz

```
## Internet Gateway
    - Click Internet gateway section on left hand sid
e. Create an internet gateway named `aws_capstone-
IGW` and create.
```

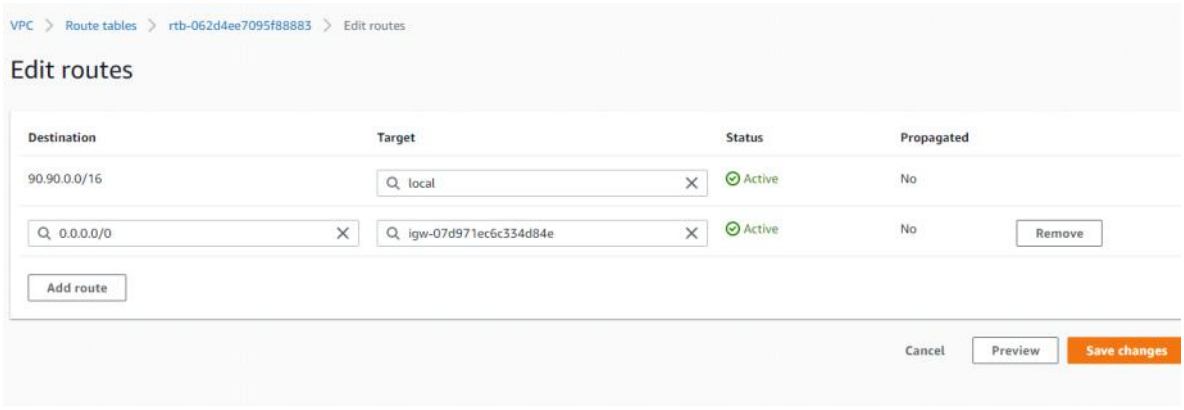



```
    - ATTACH the internet gateway `aws_capstone-
IGW` to the newly created VPC `aws_capstone-
VPC`. Go to VPC and select newly created VPC and clic
k action ---> Attach to VPC ---> Select `aws_capstone
-VPC` VPC
```

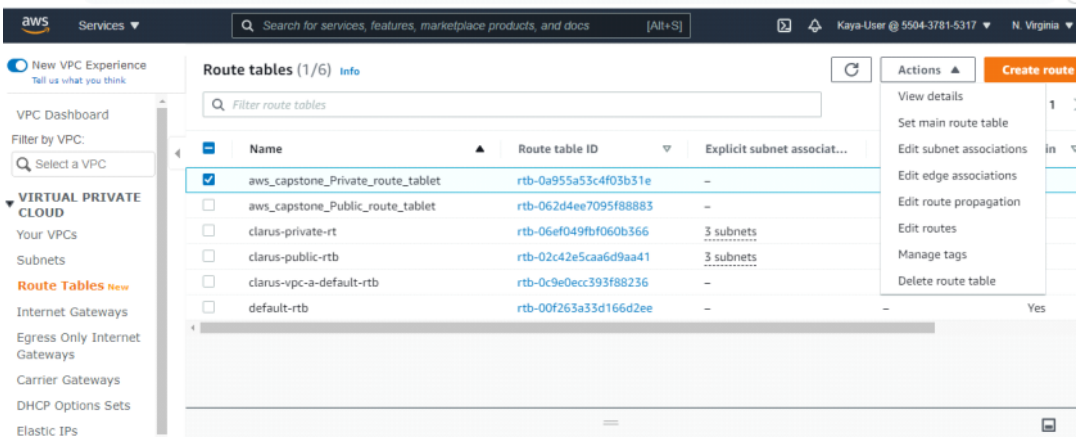
Internet Gateway olusturduk ancak bunu attach etemelz gerekiyor su an Detach olarak gözukuyor

Her bir VPC ye Ancak bir InternetGateWay Attach edilebiliyor.

Son olarak Route Tablelerimize internet Gate Way i ekliyoruz.



Ve ardindan Private Routa Tablelier ile Private Subnetleri, Public leri Public olan ile Associations edecegiz

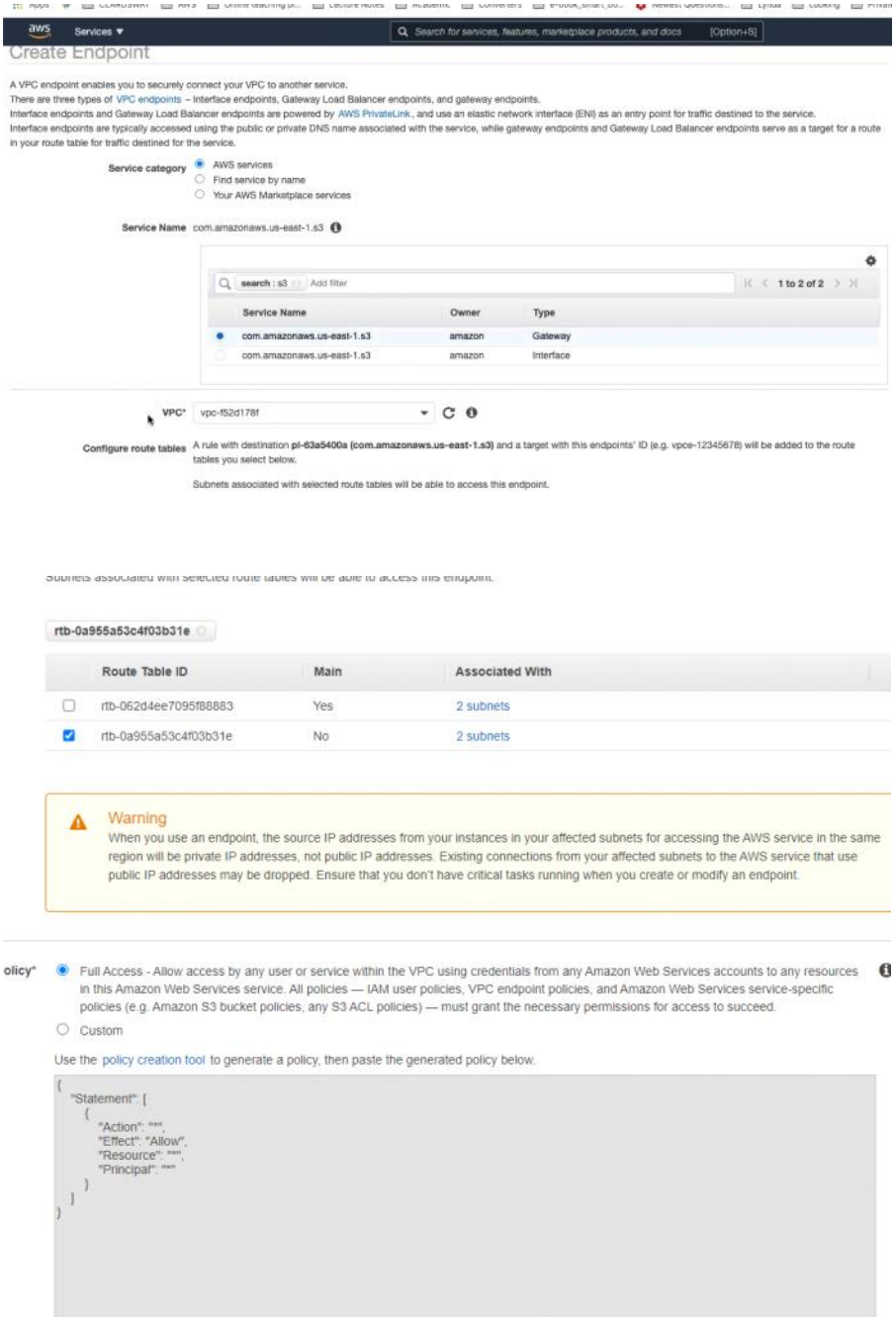


ss

Ve simdi de endpoint olusturacagiz. Ent pointleri Private Subnetler icin tanimlayacagiz.

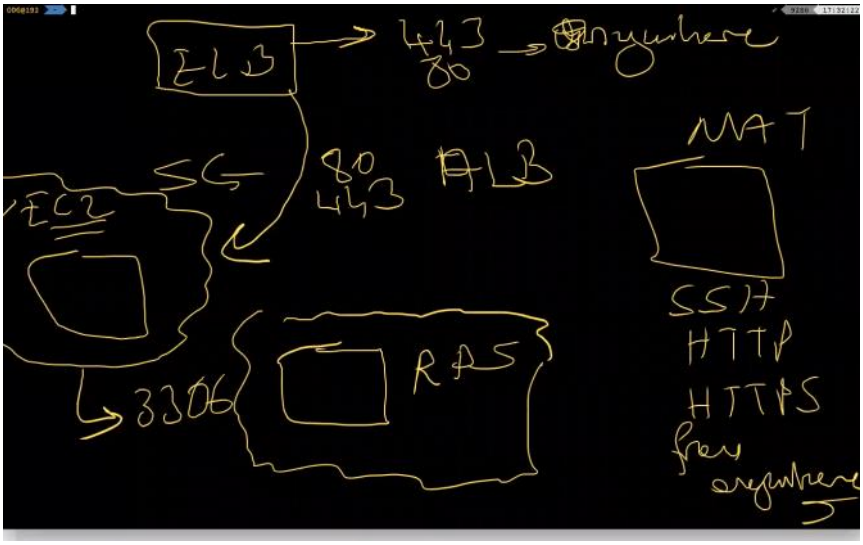
Vpc olarak kendi vpc mizi seciyoruz

Ardindan hangi Rote Tablelleri sececegimiz karsimiza geliyor



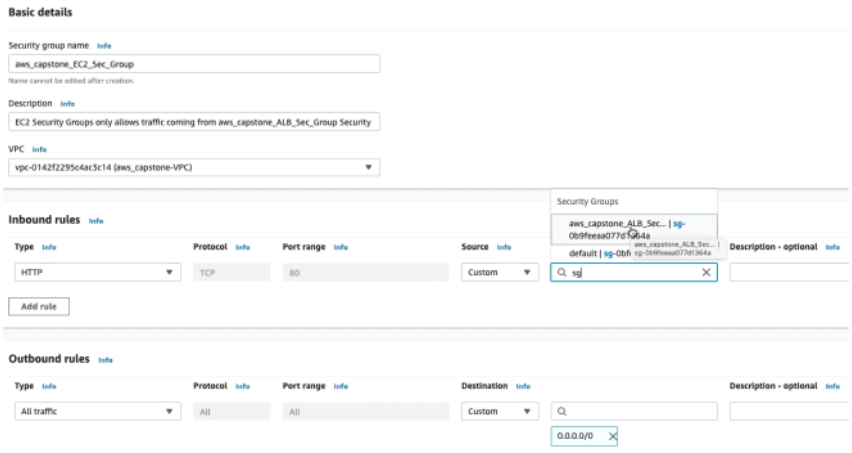
Ve simdi de security Grouplarimizi olusturacagiz. Ve daha sonra olusturdugumuz makinalrda kolay bir sekilde sececegiz.

Rds e her makinani ulasmasini istemiyoruz sadece application unumuzun oldugu EC2 nun ulasmasini istiyoruz

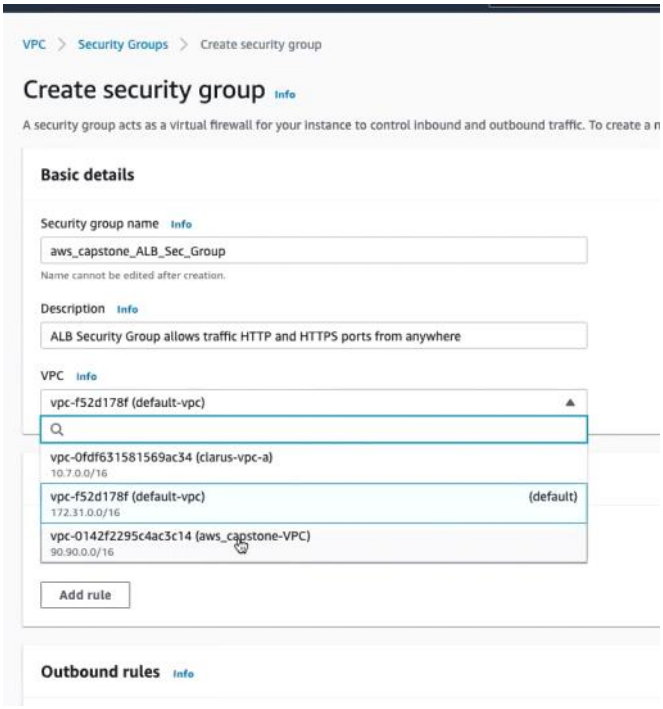


- ELB dis dunya ile irtibat sagladigi icin ona 443 ve 80 portundan her yere yani any where olarak aciyoruz
- Ancak Ec2 ya sadece ELB uzerinden ulasilmasini istedigimiz icin EC2 icin olusturdugumuz Sec Grupta 80 ve 443 portundan sadece EC2 yu belirtecegiz
- RDS sadece EC2 dan ulasilmasini istedigimiz icin RDS e 3306 portundan sadec EC2 yu tanimliyoruz.
- Nat instance dis dunyadan her istegi ic teki Private icin yerine getiren oldugu icin burda any where ile http HTTPS I Tnimliyoruz

```
### Step 2: Create Security Groups (ALB ---> EC2 --->
 RDS)

1. ALB Security Group
Name            : aws_capstone_ALB_Sec_Group
Description     : ALB Security Group allows traffic H
TTP and HTTPS ports from anywhere
Inbound Rules
VPC             : AWS_Capstone_VPC
HTTP(80)    ----> anywhere
HTTPS (443) ----> anywhere
```




```
2. EC2 Security Groups
Name            : aws_capstone_EC2_Sec_Group
Description     : EC2 Security Groups only allows tra
ffic coming from aws_capstone_ALB_Sec_Group Security
Groups for HTTP and HTTPS ports. In addition, ssh por
t is allowed from anywhere
VPC             : AWS_Capstone_VPC
Inbound Rules
HTTP(80)    ----> aws_capstone_ALB_Sec_Group
HTTPS (443) ----> aws_capstone_ALB_Sec_Group
ssh         ----> anywhere
```

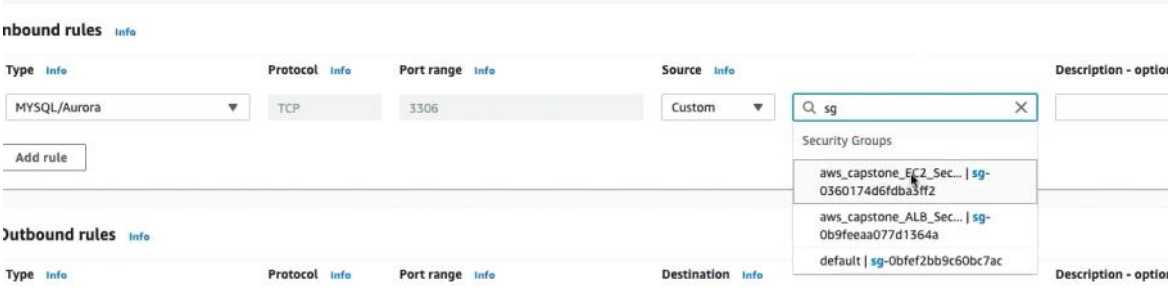


Bu su anlama geliyor 80 ve 443 portu ile attach ettigimiz (her ALB nin) makinenin bu vpc ye ulasabilecegini belirtiyoruz.

3. RDS Security Groups

```
Name            : aws_capstone_RDS_Sec_Group
Description     : EC2 Security Groups only allows tra
ffic coming from aws_capstone_EC2
_Sec_Group Security Groups for MYSQL/Aurora port.
VPC             : AWS_Capstone_VPC
Inbound Rules
MYSQL/Aurora(3306)  ----> aws_capstone_EC2_Sec_Group
```

Sadece EC2 kardan geleni kabul edecek



4. NAT Instance Security Group

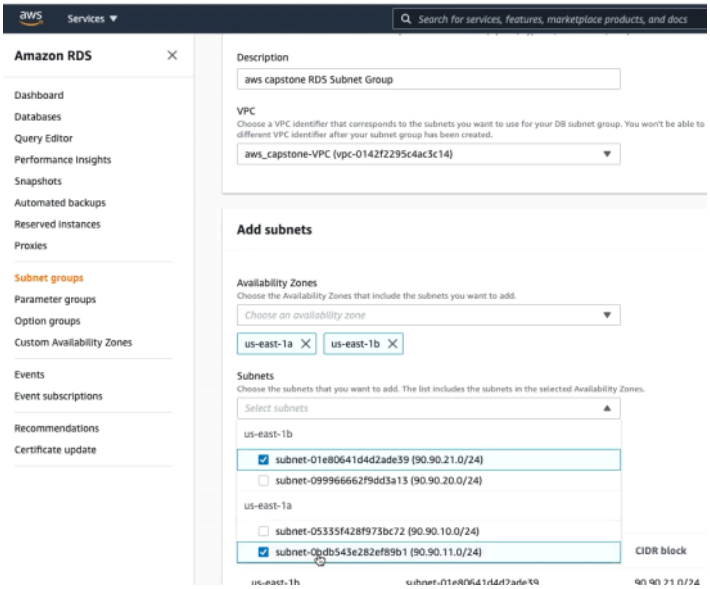
```
Name            : aws_capstone_NAT_Sec_Group
Description     : ALB Security Group allows traffic H
TTP and HTTPS and SSH ports from anywhere
Inbound Rules
VPC             : AWS_Capstone_VPC
HTTP(80)    ----> anywhere
HTTPS (443) ----> anywhere
SSH (22)    ----> anywhere
```

Step 3: Create RDS

Custom VPC lerde RDS kurmadan önce bu RDS lerin hangi VPC de calismasini istiyorsak o sekilde bir Subnet Group olusturuuz

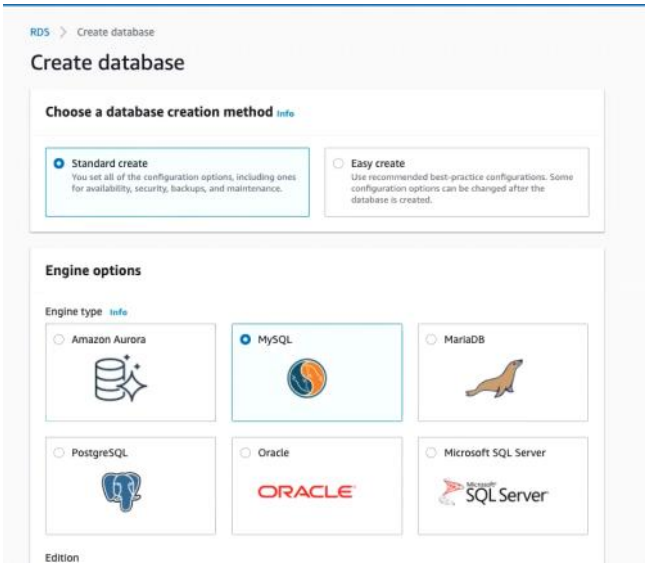
First we create a subnet group for our custom VPC. Click `subnet Groups` on the left hand menu and click `create DB Subnet Group`

```text
Name                : aws_capstone_RDS_Subnet_Group
Description         : aws capstone RDS Subnet Group
VPC                 : aws_capstone_VPC
Add Subnets
Availability Zones : Select 2 AZ in aws_capstone_VPC
Subnets             : Select 2 Private Subnets in these subnets
```

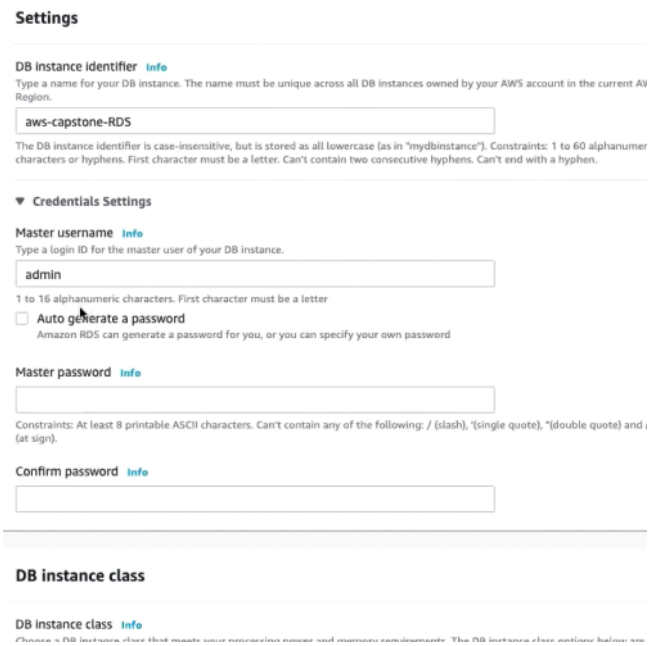


- Go to the RDS console and click `create database` button

```text
Choose a database creation method : Standart Create
Engine Options  : Mysql
```




```text
Version         : 8.0.20
Templates       : Free Tier
Settings        :
    - DB instance identifier : aws-capstone-RDS
    - Master username        : admin
    - Password               : Clarusway1234
DB Instance Class            : Burstable classes (includes t classes) ---> db.t2.micro
Storage                      : 20 GB and enable autoscaling(up to 40GB)
```



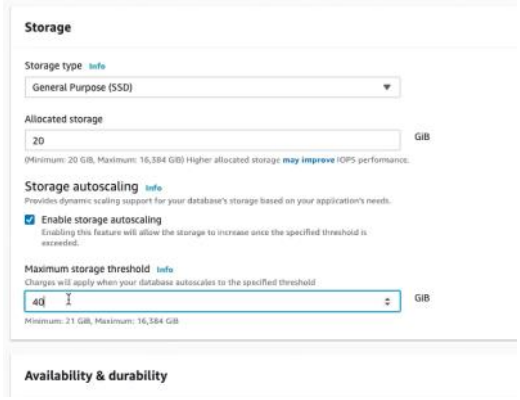
Connectivity:

VPC : aws_capstone_VPC

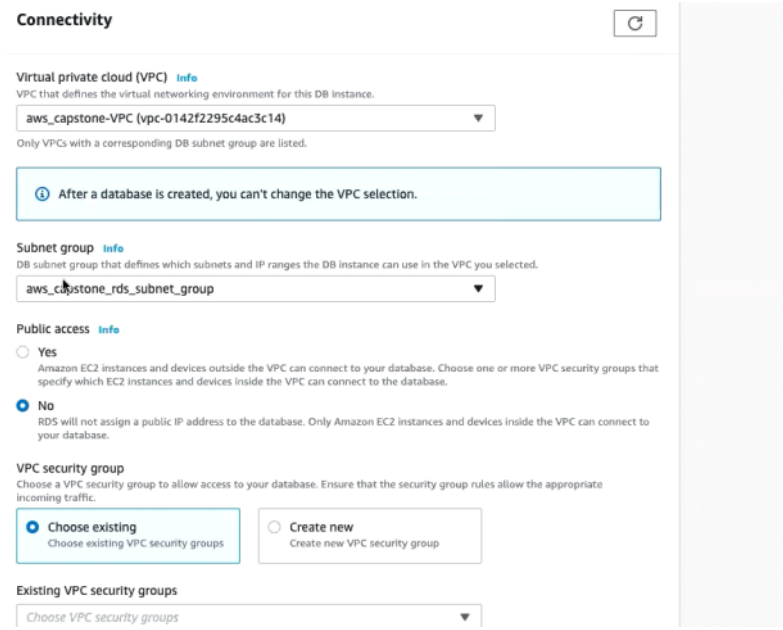
Subnet Group : aws_capstone_RDS_Subnet_Group

Public Access : No

VPC Security Groups : Choose existing ---> aws_capstone_RDS_Sec_Group

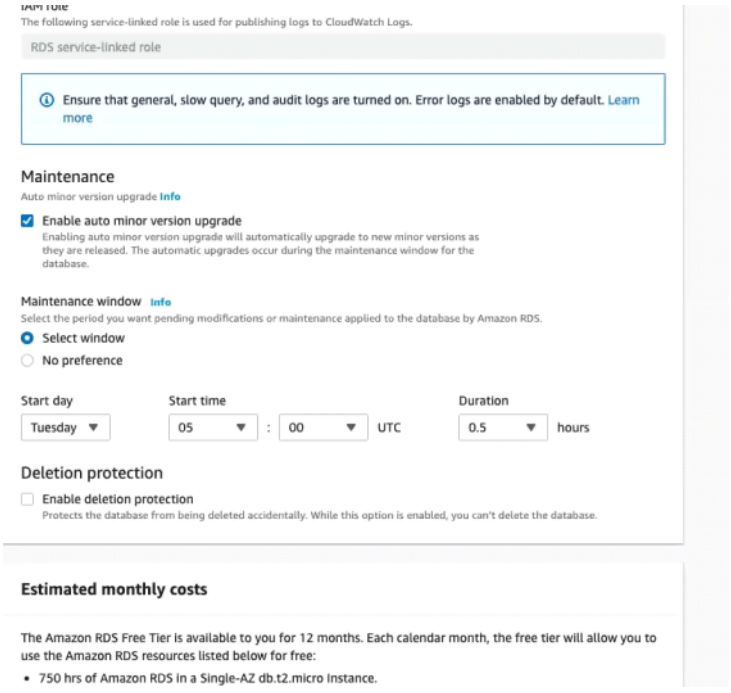


Storaage kismini RDS in kapasitesi max kac arttirabiliyor max olarak 16,384 GIB yapabiliyor



Public acces no diyoruz VPC nin disindan verilen bir izindir.

```
Availability Zone        : No preference
    Additional Configuration : Database port ---> 330
6
Database authentication ---> Password authentication
Additional Configuration:
    - Initial Database Name  : database1
    - Backup ---> Enable automatic backups
    - Backup retention period ---> 7 days
    - Select Backup Window ---> Select 03:00 (am) Dur
ation 1 hour
    - Maintance window : Select window    ---> 04:00(
am) Duration:1 hour
create instance
```



Enable deleion dersek silmek icin bir kac basamli bir sart getiri protection ortanminda bu öneriir.

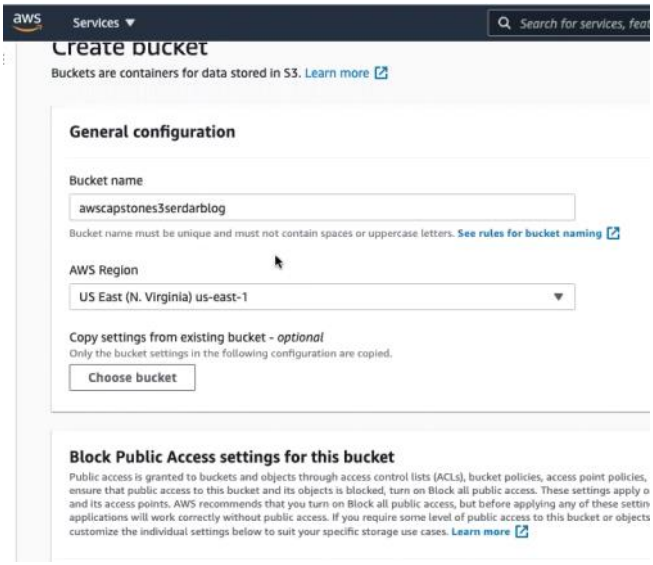
Iki bucket icin olusturacagiz bir failover icin digerri resim ve metinleri saklamak icin olusturacagiz

Step 4: Create two S3 Buckets and set one of thes

e as static website.

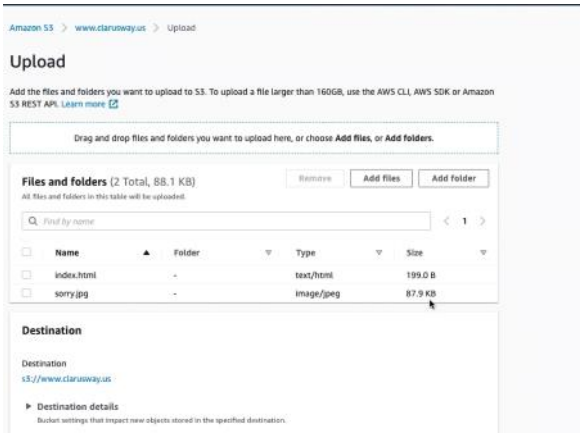
Go to the S3 Consol and lets create two buckets.

1. Blog Website's S3 Bucket
- Click Create Bucket
```text
Bucket Name : awscapstones3<YOUR NAME>blog------uniq
dikkat edilmesi gereken
Region      : N.Virginia
```



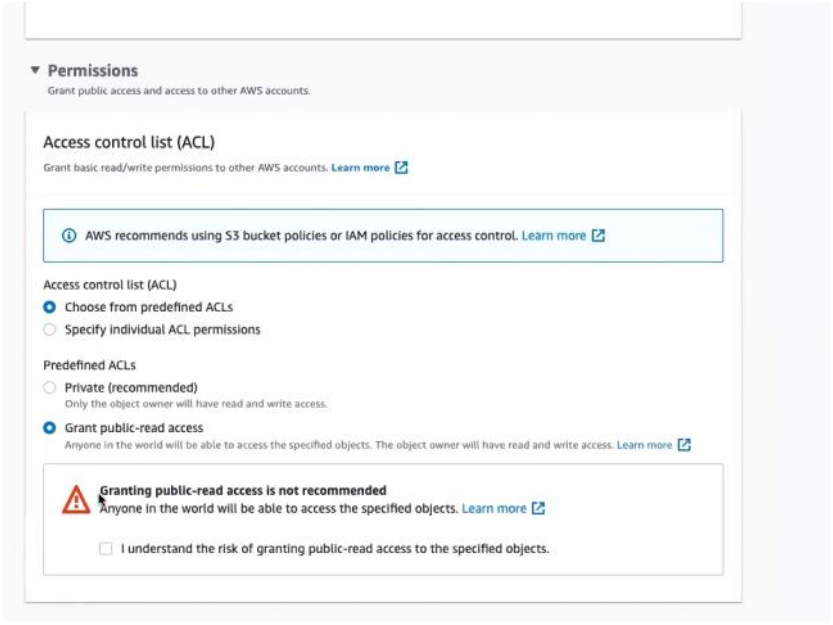
Block all public access : Unchecked
Other Settings are keep them as are
create bucket
```

2. S3 Bucket for failover scenario
- Click Create Bucket
```text

Bucket Name : www.<YOUR DNS NAME>
Region      : N.Virginia
Block all public access : Unchecked
Please keep other settings as are
```
- create bucket
- Selects created `www.<YOUR DNS NAME>` bucket ---> P
roperties ---> Static website hosting
```text
Static website hosting : Enable
Hosting Type : Host a static website
Index document : index.html
save changes
```
- Select `www.<YOUR DNS NAME>` bucket ---> select Upl
oad and upload `index.html` and `sorry.jpg` files fro
m given folder---> Permissions ---> Grant public-
read access ---> Checked warning massage
```



Ayrica static web site olarak kullanmak icin de

Ayrica WWW. Ve DNS name ile olusturdugumuz subnette Acces Control sekmesini Grant Public olarak aciyoruz upload ettigimiz file in dis dunyadan okunmasini sagliyoruz

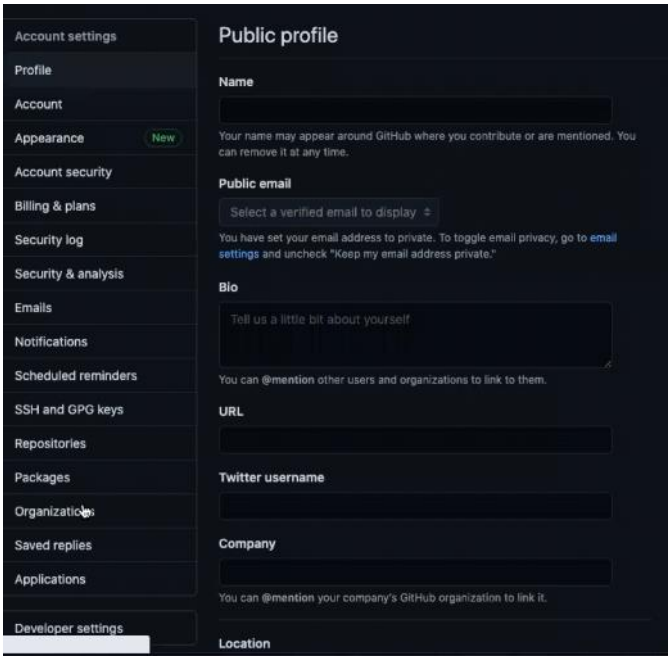


Step 5: Copy files downloaded or cloned from `Clarusway_project` repo on Github

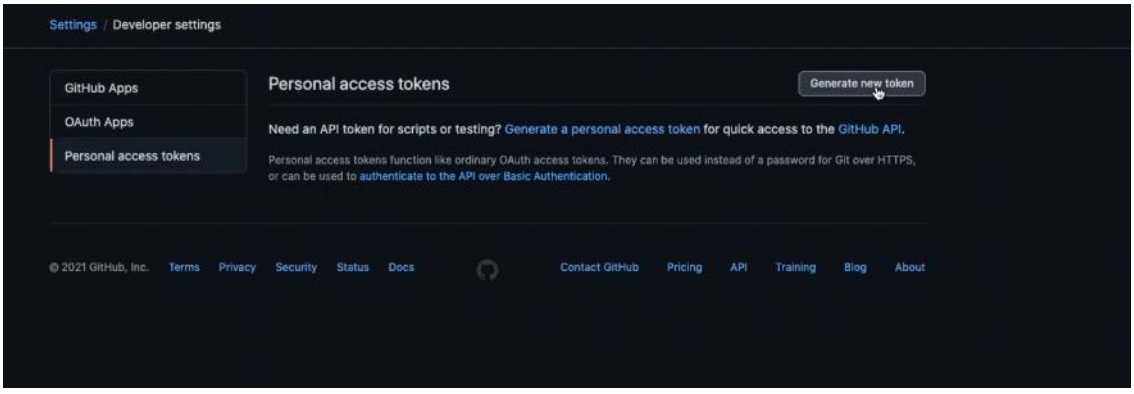
Git hub dan
S3 ler ayarlandi RDS ayarlandi ve simdi Githubda bulnan private repomuzu clonlamistik daha önce simdi de git commit - m ve git push komutlari ile ile lokalde bulunan dosyalarimizi git hub dosyamiza gönderdik.

Simdi de git hub hesabimiz dan TOKEN alacagiz;

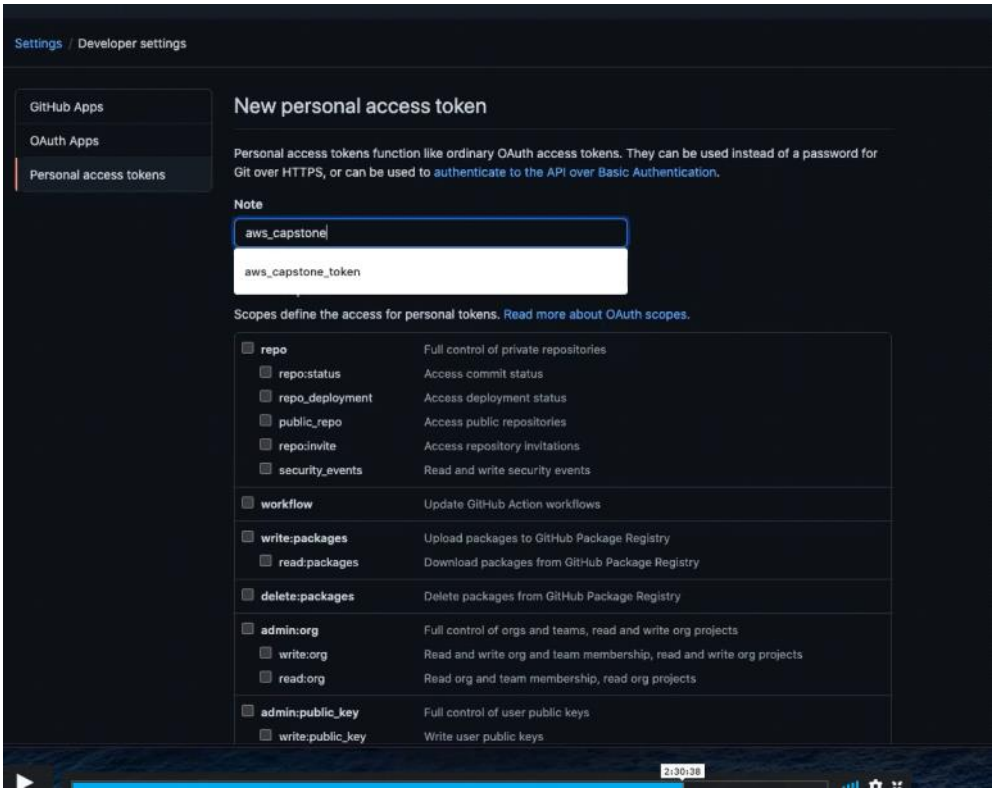
Avatar----Settin-----Developer settings--Personel Access tokens----



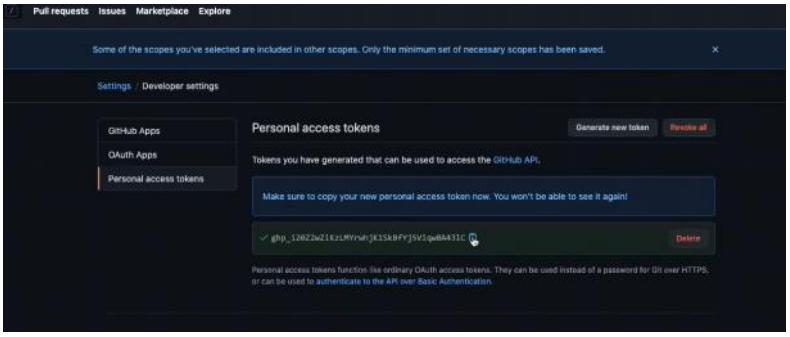
Ve Generate new token

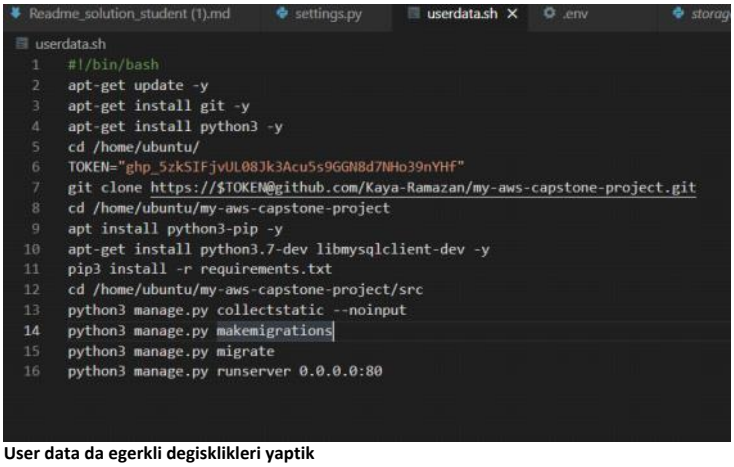


Enter the passsword



Ve Token hesabimiz olustu





User data da egerkli degisklikleri yaptik

Son dört komut Jangoyu calistiriyor.

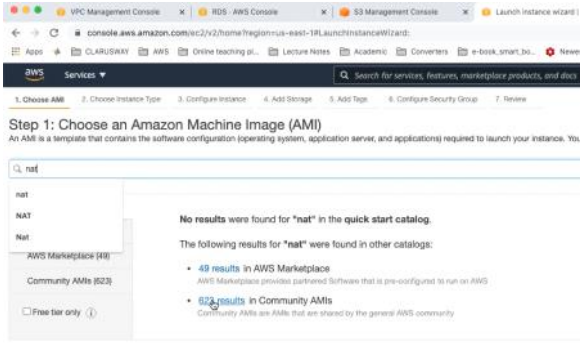
Step 8: Write RDS database endpoint and S3 Bucket name in settings file given by Clarusway Fullstack De
veloper team and push your application into your own
public repo on Github
Please follow and apply the instructions in the devel
oper_notes.txt.
```text
- Movie and picture files are kept on S3 bucket named
```

aws_capstone_S3
_<name>_Blog as object. You should create an S3 bucke
t and write name of it on "/src/cblog/settings.py" fi
le as AWS_STORAGE_BUCKET_NAME variable. In addition,
you must assign region of S3 as AWS_S3
_REGION_NAME variable
- Users credentials and blog contents are going to be
kept on RDS database. To connect EC2 to RDS, followi
ng variables must be assigned on "/src/cblog/settings
.py" file after you create RDS;
    a. Database name - "NAME" variable
    b. Database endpoint - "HOST" variables
    c. Port - "PORT"
    d. PASSWORD variable must be written on "/src/.en
v" file not to be exposed with settings file
```
- Please check if this userdata is working or not. to
do this create new instance in public subnet and sho
w to students that it is working

## Step 9: Create NAT Instance in Public Subnet

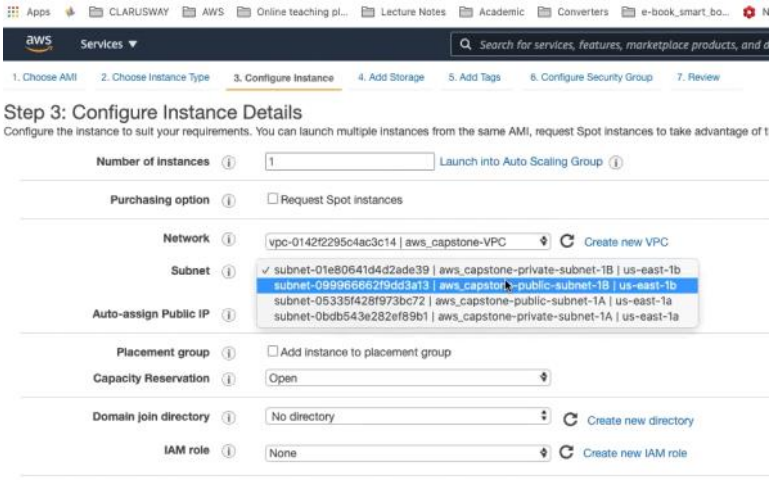
Public Subnette Nat instance olusturacagiz. Nat instance olusturmamizin sebebi Auto scaling olusturacagiz ki bunun icin öncelikle bir Launch Template olusturacaktik . AutoScalin EC2 olusturacak ve Ec2 larda Data cekmek isteyecekler onun icin Nat instance ihtiyac var.

**To launch NAT instance, go to the EC2 console and cli
ck the create button.**
```text
write "NAT" into the filter box
select NAT Instance `amzn-ami-vpc-nat-
hvm-2018.03.0.20181116-x86_64-ebs`
```

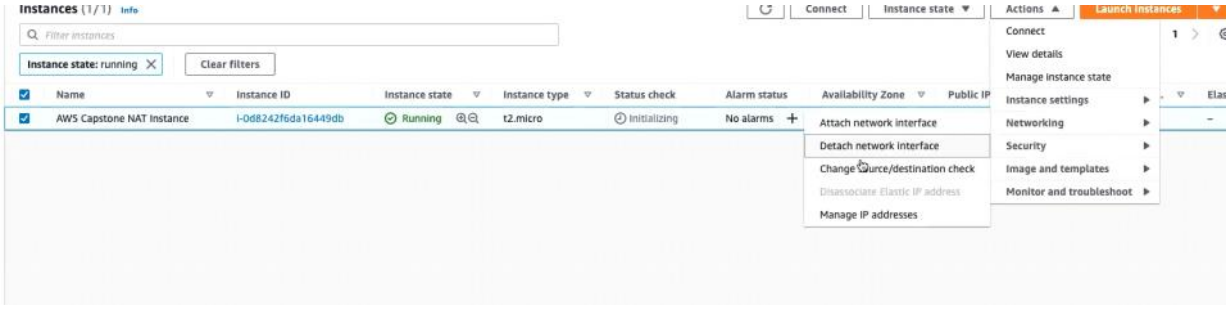


**Instance Type: t2.micro
Configure Instance Details
    - Network : aws_capstone_VPC
    - Subnet : aws_capstone-public-
subnet-1A (Please select one of your Public Subnets)
    - Other features keep them as are
Storage ---> Keep it as is
Tags: Key: Name   Value: AWS Capstone NAT Instance
Configure Security Group
    - Select an existing security group: aws_capstone
_NAT_Sec_Group
Review and select our own pem key**
```
**!!!IMPORTANT!!!
- select newly created NAT instance and enable stop s
ource/destination check
- go to private route table and write a rule**
```
**Destination : 0.0.0.0/0
Target   : instance ---> Select NAT Instance
Save**
```

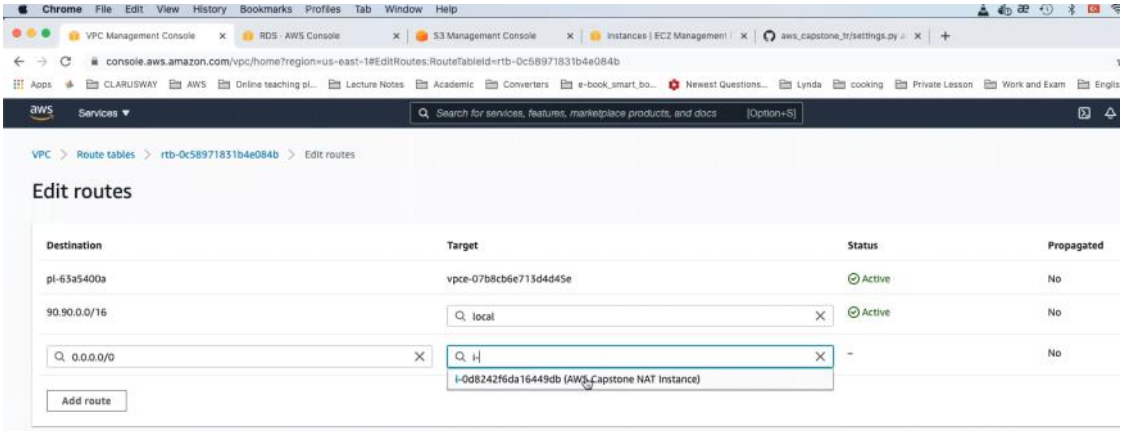
Public Subnetlerde olusturacagiz



Nat instance olusturduktan sonra Actions ---Change Source destination Check edecegiz ----stop edecegiz



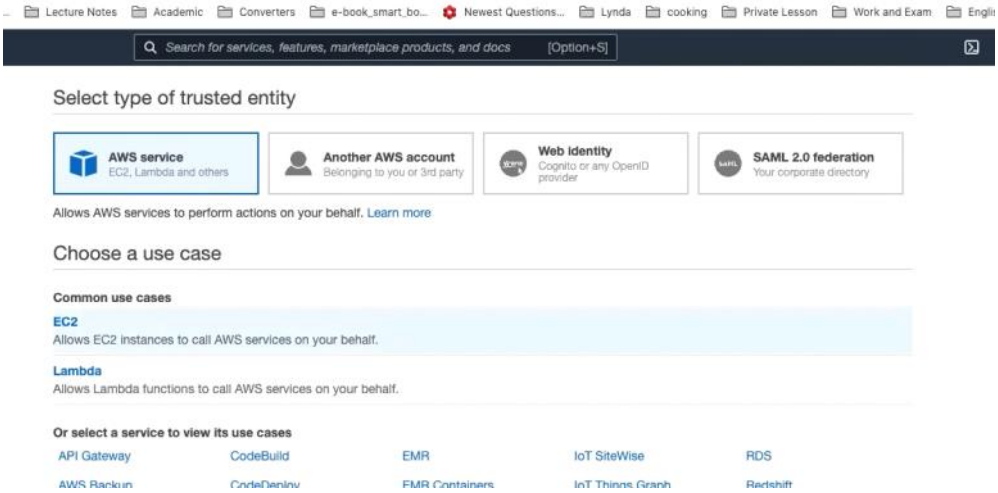
Daha sonra Private Rote Tabletelere bir Raute Table eklememiz gerekiyor. Private Makinalrda dis dunyaya gitmek istediklerinde bi istegi nat instance aktarmasini sagla diyoruzz.



Simdi de Launch Template olusturacagiz Launch Template tanimlamak icin EC2 ile konusabilmesi icin 2 alteranatif var bunlardan ilki boto3 ile oluyor. Veyahut bu ulasmayi Iam Role tanimlayarak da yapabiliyoruz.

Full access yetkisi verecek bir IAM Role tanimlayacagiz ki bunun ile

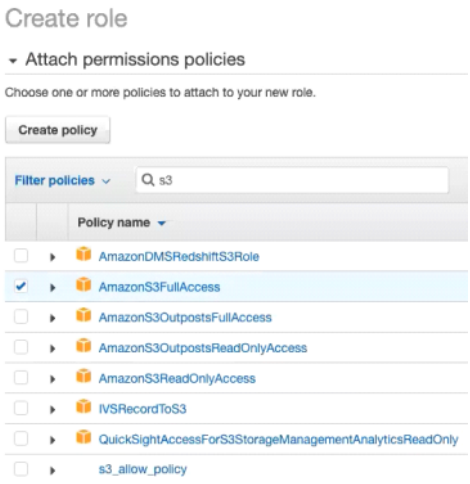
Ilk olarak EC2 seciyoruz



Step 10: Create Launch Template and IAM role for it

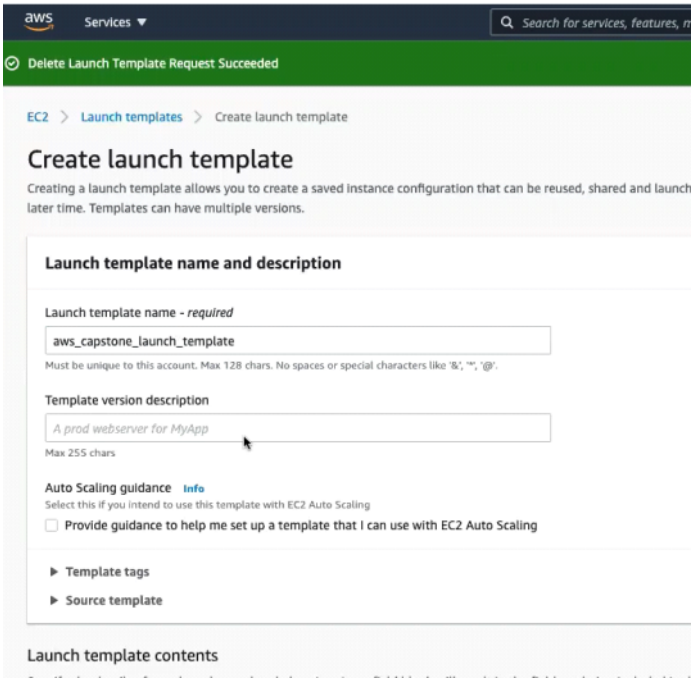
Go to the IAM role console click role on the right hand menu than create role

```text
trusted entity  : EC2 as  ---> click Next:Permission
Policy          : AmazonS3FullAccess policy
Tags            : No tags
Role Name       : aws_capstone_EC2_S3_Full_Access
Description     : For EC2, S3 Full Access Role
```



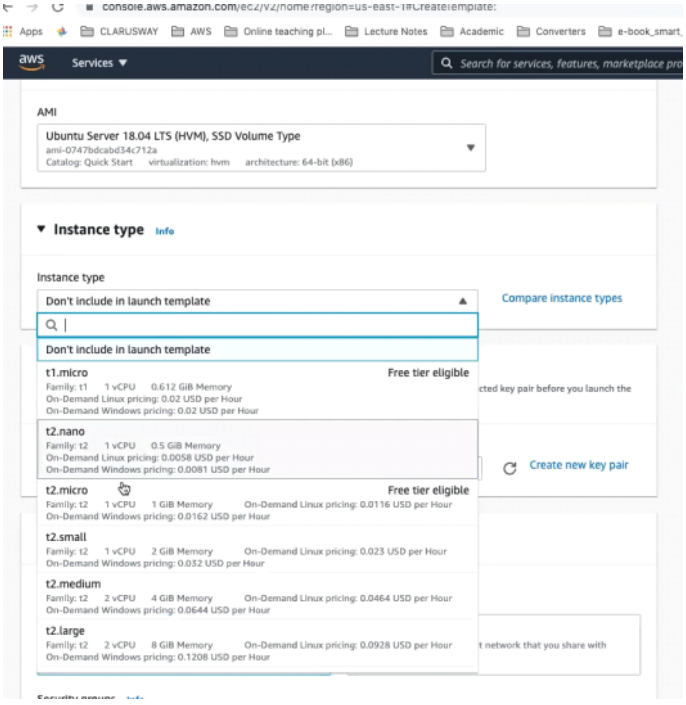
Rol Create edildikten sonra Launch Template olusturacagiz;

```
- create launch template
```

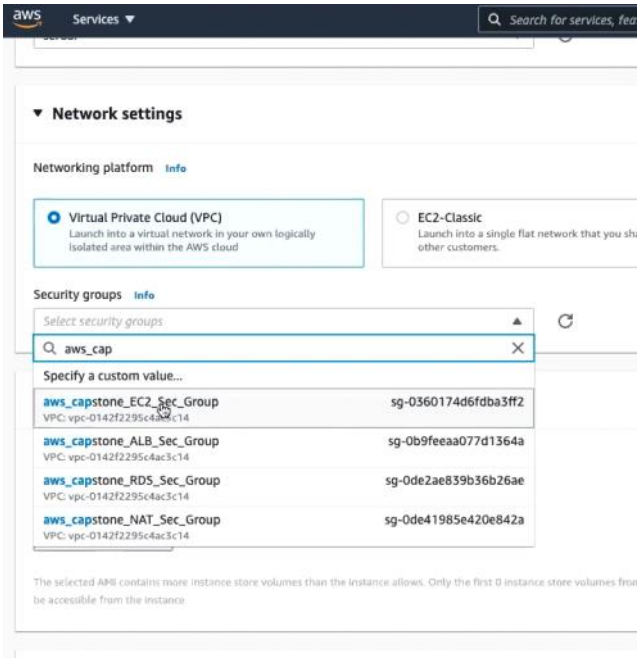


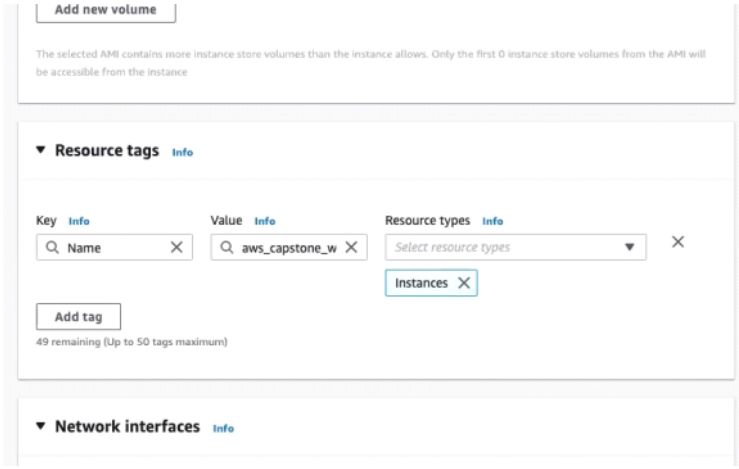
To create Launch Template, go to the EC2 console and select `Launch Template` on the left hand menu. Tab the Create Launch Template button.

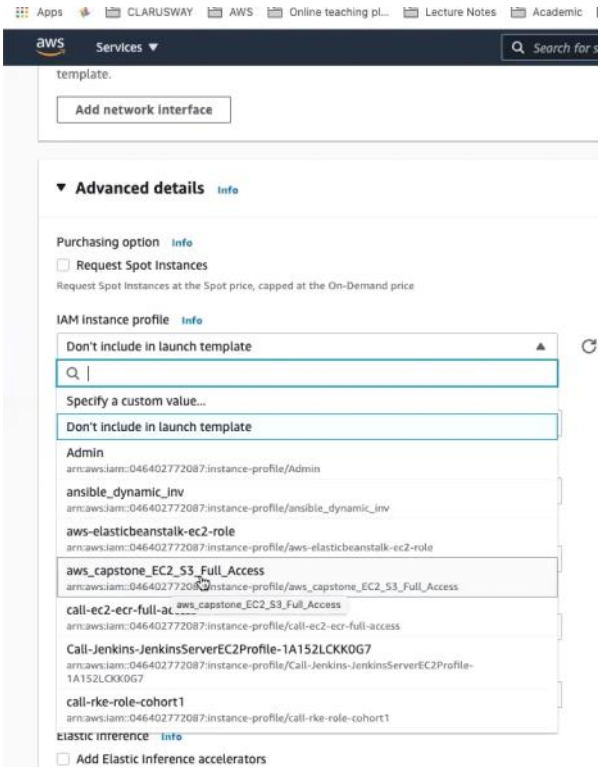
```bash
Launch template name                : aws_capstone_launch_template
Template version description        : Blog Web Page version 1
Amazon machine image (AMI)          : Ubuntu 18.04
Instance Type                       : t2.micro
Key Pair                            : mykey.pem
Network Platform                    : VPC
```



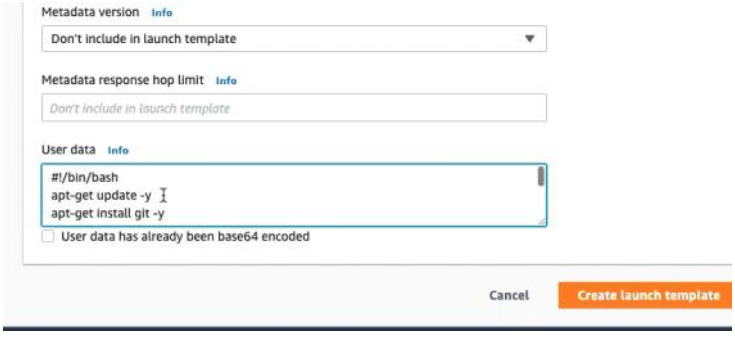

```bash
Security Groups                     : aws_capstone_EC2_sec_group
Storage (Volumes)                   : keep it as is
Resource tags                       : Key: Name   Value: aws_capstone_web_server
Advance Details:
    - IAM instance profile          : aws_capstone_EC2_S3_Full_Access
    - Termination protection        : Enable
```


aws
Services
Search for services, fea
Network settings
Networking platform Info
Virtual Private Cloud (VPC)
Launch into a virtual network in your own logically isolated area within the AWS cloud
EC2-Classic
Launch into a single flat network that you sh
other customers.
Security groups Info
Select security groups
aws_cap
Specify a custom value...
aws_capstone_EC2_Sec_Group
VPC: vpc-0142f2295c4ac3c14
sg-0360174d6fdba3ff2
aws_capstone_ALB_Sec_Group
VPC: vpc-0142f2295c4ac3c14
sg-0b9feeaa077d1364a
aws_capstone_RDS_Sec_Group
VPC: vpc-0142f2295c4ac3c14
sg-0de2ae839b36b26ae
aws_capstone_NAT_Sec_Group
VPC: vpc-0142f2295c4ac3c14
sg-0de41985e420e842a
The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes fro
be accessible from the instance


Add new volume
The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance
Resource tags Info
Key Info
Name
Value Info
aws_capstone_w
Resource types Info
Select resource types
Instances
Add tag
49 remaining (Up to 50 tags maximum)
Network interfaces Info




```
- User Data
#!/bin/bash
apt-get update -y
apt-get install git -y
apt-get install python3 -y
cd /home/ubuntu/
TOKEN="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"
git clone  https://$TOKEN@github.com/Kaya-Ramazan/my-aws-capstone-project.git
cd /home/ubuntu/my-aws-capstone-project
apt install python3-pip -y
apt-get install python3.7-dev default-libmysqlclient-dev -y
pip3 install -r requirements.txt
cd /home/ubuntu/my-aws-capstone-project/src
python3 manage.py collectstatic --noinput
python3 manage.py makemigrations
python3 manage.py migrate
python3 manage.py runserver 0.0.0.0:80
```



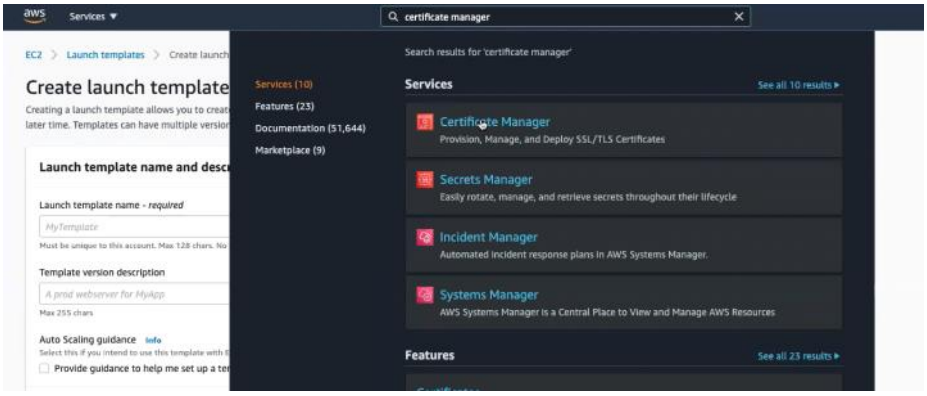
Production Ortaminda en basit bir site adahil secur bir baglansti yapmak ister. Browserlar sec baglanti var ise

HTTP den yayin Yapmazlar daha secure olan HTTPS den yayin yaparlar increp etmek icin sertifikalari kullanirlar

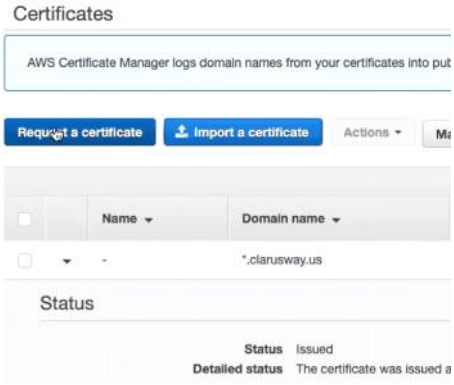
Step 11: Create certification for secure connection

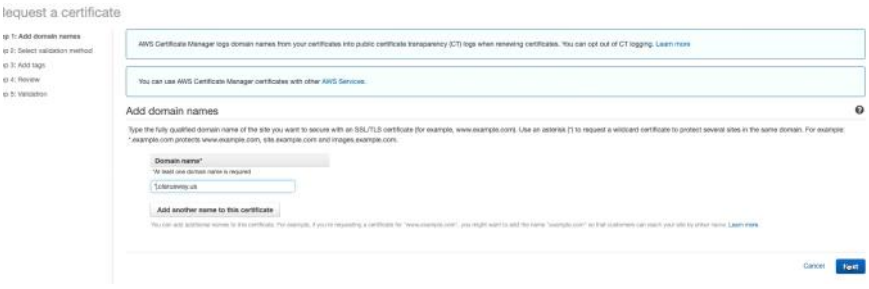
Go to the certification manager console and click `re

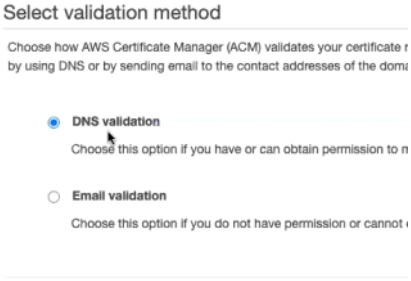
quest a certificate` button. Select `Request a public
 certificate`, then `request a certificate` ---> `*.<
YOUR DNS NAME>` ---> DNS validation ---> No tag --->
Review ---> click confirm and request button. Then it
 takes a while to be activated.

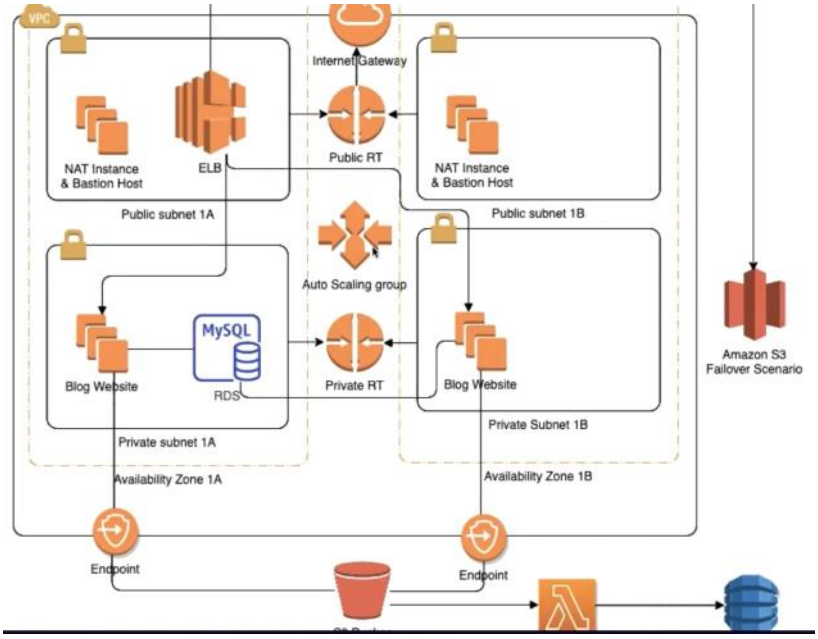


Yeni alamak icin;







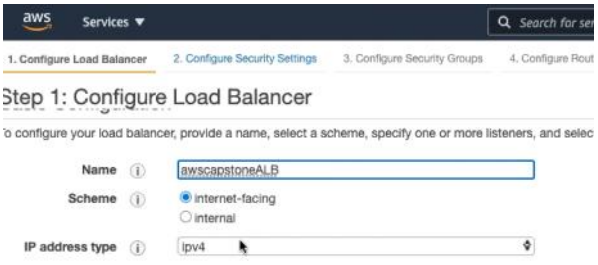


Simdiye kadar yaptiklarimizi özetler isek ;

- Vpc leri kurduk
- Availibility zonalar ve Subnetleri ayarladik
- internetGateway baglandi
- Public Subnetleri Asocate ettik
- RDS kurduk
- Endpointimizi verdik

Step 12: Create ALB and Target Group

Go to the Load Balancer section on the left hand side
 menu of EC2 console. Click `create Load Balancer` bu
tton and select Application Load Balancer
```text
Name                    : awscapstoneALB
Schema                  : internet-facing
```



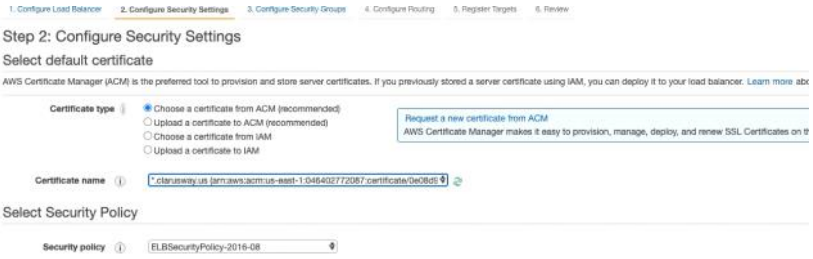
Listeners

Listeners   : HTTPS, HTTP

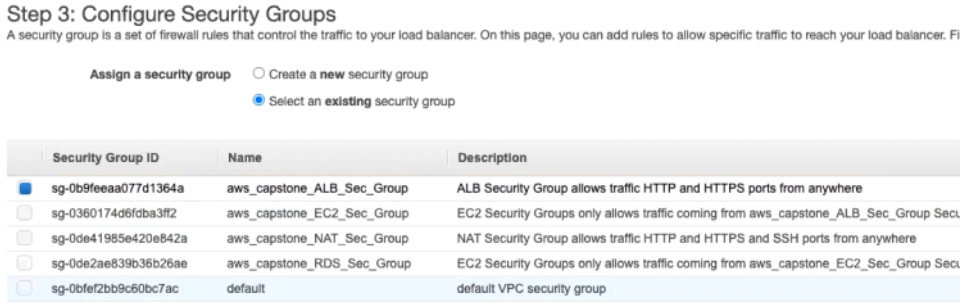
Availability Zones   :
    - VPC   : aws_capstone_VPC
    - **Availability zones:**
        1. aws_capstone-public-subnet-1A
        2. aws_capstone-public-subnet-1B

Public e yerlestiriyoruz cunku yayuin yapacak

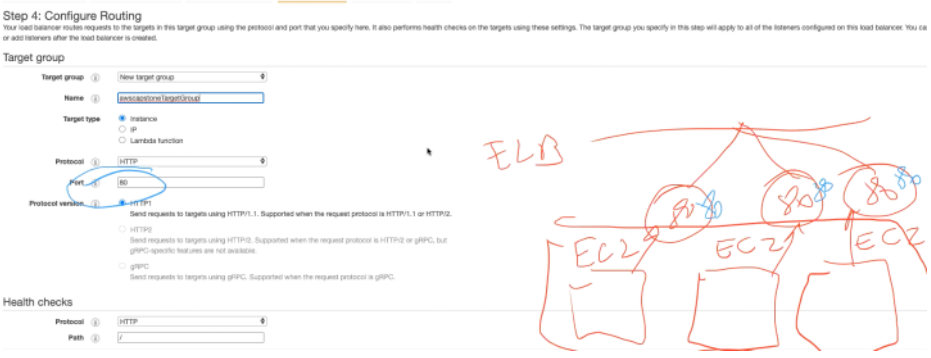
Step 2 - Configure Security Settings
Certificate type ---> Choose a certificate from ACM (recommended)
- Certificate name : "*.clarusway.us" certific ate
- Security policy : keep it as is



Step 3 - Configure Security Groups : aws_capstone_ALB _Sec_group



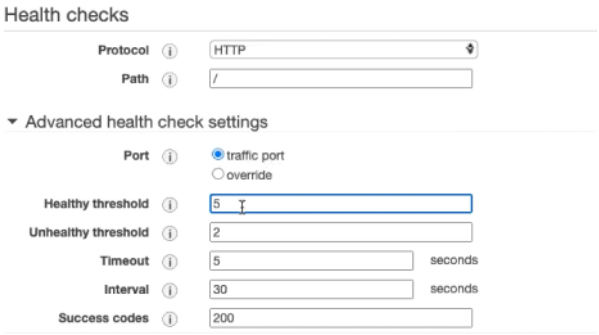
Step 4 - Configure Routing
- Target group : New target group
- Name : awscapstoneTargetGroup
- Target Type : Instance
- Protocol : HTTP
- Port : 80



Instance nerede calisiyorsa o portta calistirmamiz gerekiyor

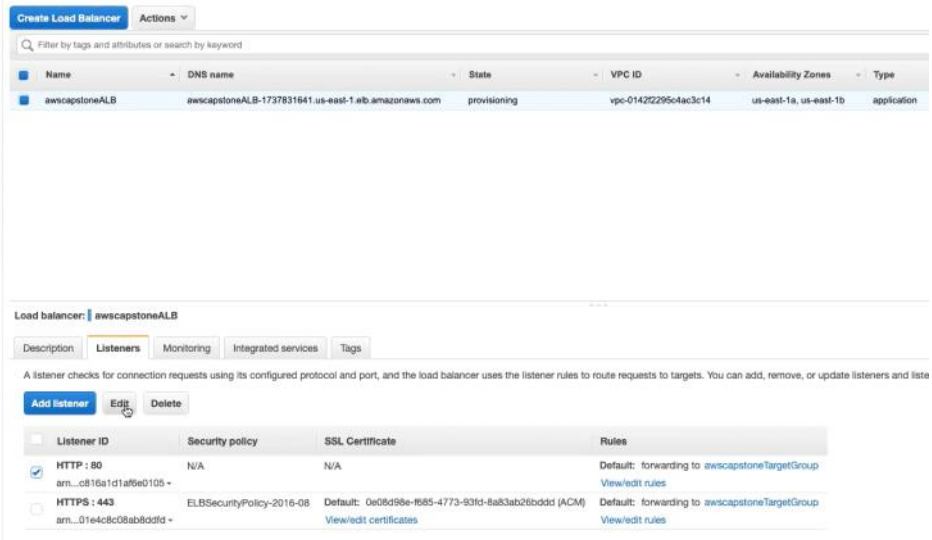
- Protocol version : HTTP1
- Health Check :
  - Protocol : HTTP
  - Path : /
  - Port : traffic port
  - Healthy threshold : 5 ------ ardi ardina gönderilen 3 saglik sinyalde sagliklin olarak

muhurluyor
      - Unhealthy threshold : 2-------ard ardina 2 kez gönerdi ve cevap alamadi sagliksiz olarak nitelendiriyor
      - Timeout   : 5   ------sinyali gönderdikten sonra bekleme suresi
      - Interval   : 20
      - Success Code   : 200 ----- herhangi problem yok demejtir.

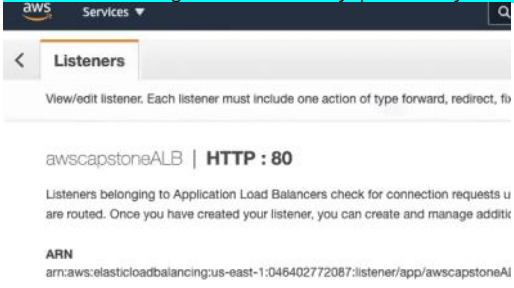


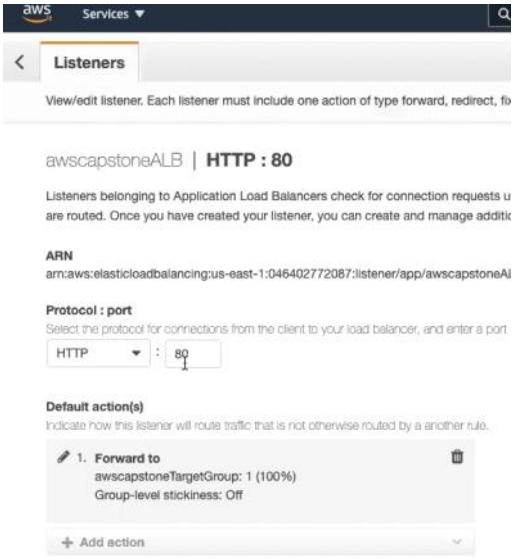
Step 5 - Register Targets
without register any target click Next: Review
```
- **click create**
**To redirect traffic from HTTP to HTTPS, go to the ALB console and select Listeners sub-section.**
```text
select HTTP: 80 rule ---> click edit
- Default action(s)
 - Remove existing rule and create new rule which is
    - Redirect to HTTPS 443
    - Original host, path, query
    - 301 - permanently moved
```
Lets go ahead and look at our ALB DNS --> it going to say "it is not safe", however, it will be fixed after settings of Route 53

Load balanceri olusturduktan sonra yayini 80 portundan gelen yayini mevcutta Target Guruba gönderiyor iken Biz Önce 443 HTTPS e göndermesini ve sonmrasinda Target guruba gitmrisni istiyorduk. Bu sekilde daha secure olacakti.

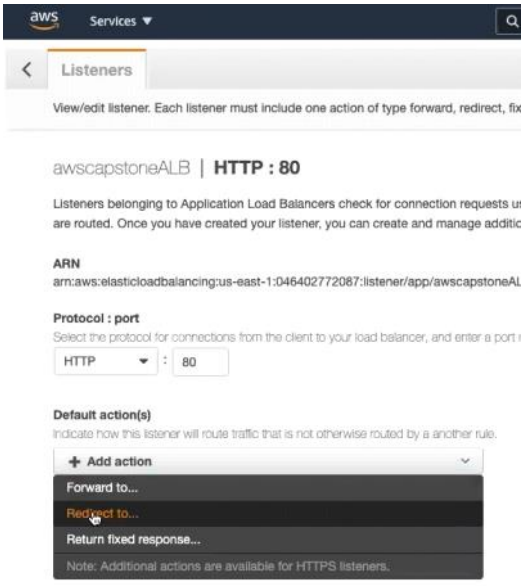


Bunun icin 80 baglantisini tiklayip edit diyoruz.

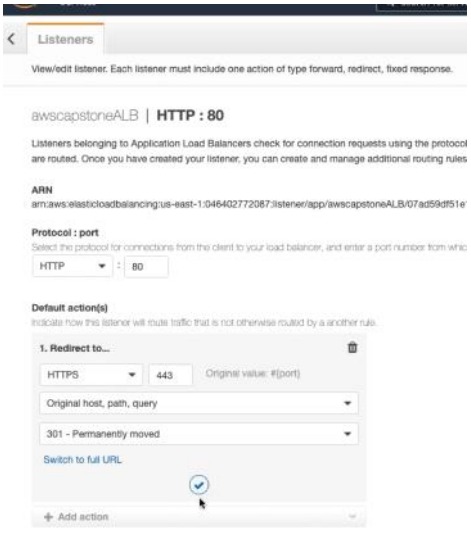

```



Ardindan default olan Forwardi siliyoruz



Ve Redirection ekliyoruz ve 443 secip tikliyoruz.

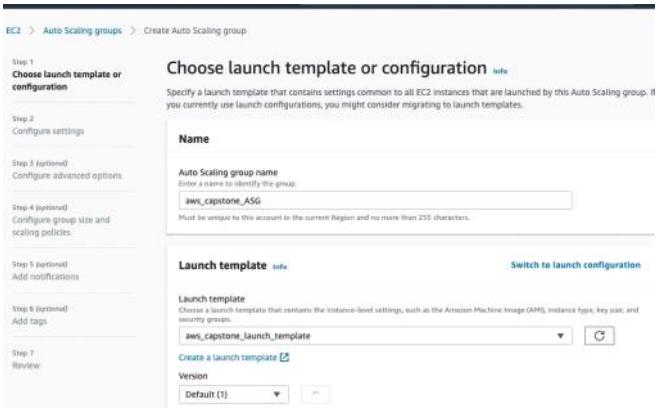


Step 13: Create Autoscaling Group with Launch Template

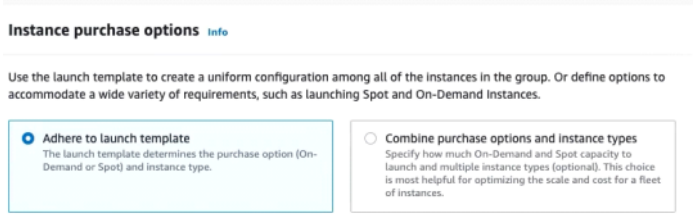
Go to the Autoscaling Group on the left hand side men

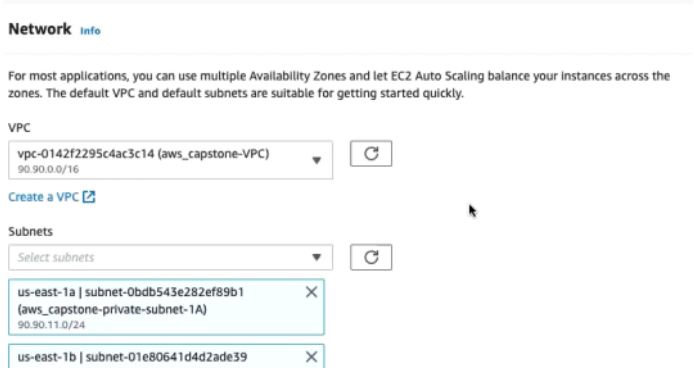
u. Click create Autoscaling group.

- Choose launch template or configuration
```text
Auto Scaling group name          : aws_capstone_ASG
Launch Template                  : aws_capstone_launch
_template
```

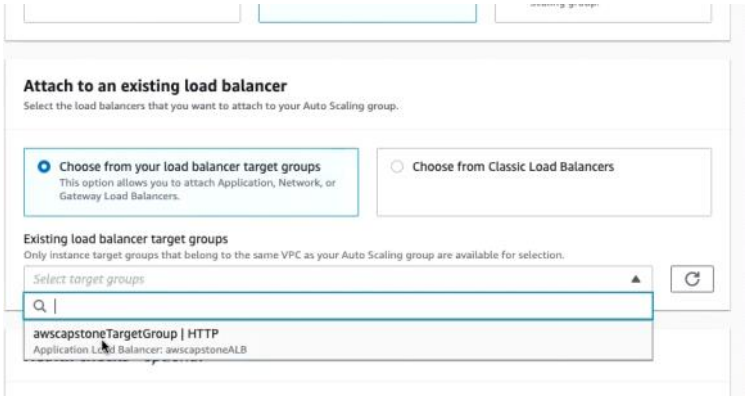


- Configure settings
```text
Instance purchase options        : Adhere to launch te
mplate
Network                          :
    - VPC                        : aws-capstone-VPC
    - Subnets                    : Private 1A and Priv
ate 1B
```

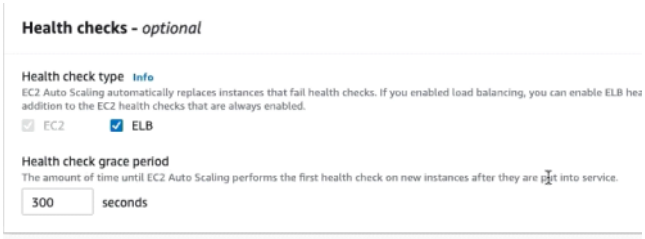




- Configure advanced options
```text

- Load balancing                         : Att
ach to an existing load balancer
- Choose from your load balancer target groups  : aws
capstoneTargetGroup
```




```
- Health Checks
    - Health Check Type             : ELB
    - Health check grace period     : 300
```



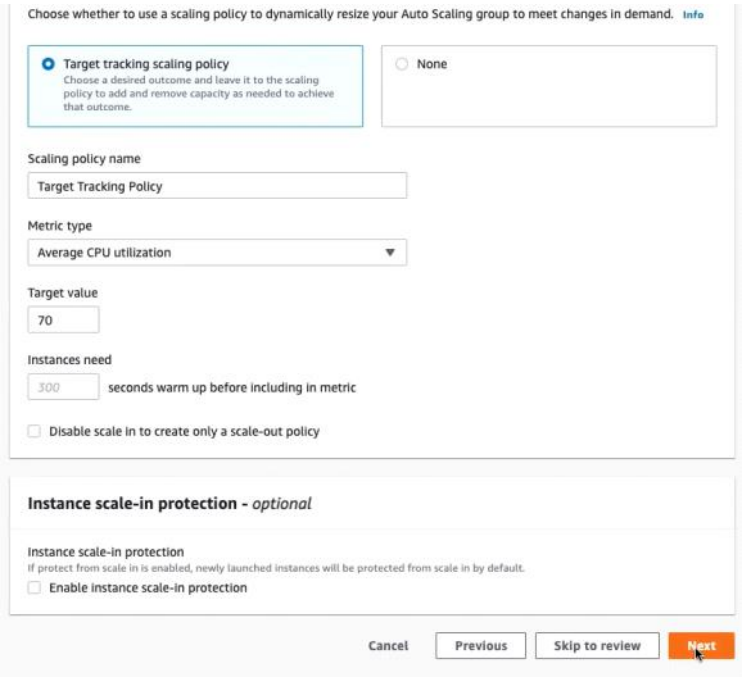
- Configure group size and scaling policies

```text
Group size
    - Desired capacity  : 2
    - Minimum capacity  : 2
    - Maximum capacity  : 4
Scaling policies
```

```
    - Target tracking scaling policy
        - Scaling policy name       : Target Tracking
 Policy
        - Metric Type               : Average CPU uti
lization
        - Target value              : 70
```

- Add notifications

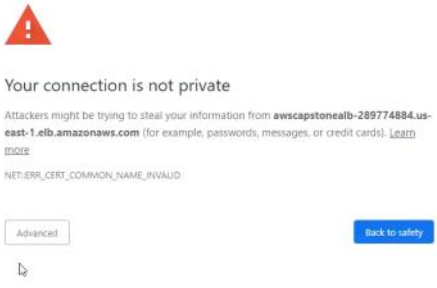
```text
Create new Notification
    - Notification1
        - Send a notification to    : aws-capstone-
SNS
        - with these recipients     : serdar@claruswa
y.com
        - event type                : select all
```



Target Traning Policy Ec2 instance olusmasi veyahut Terminate ettirilmesi icin belirtilir.

CPU Utilazition ortalama Cpu kullanimi bizim belirledigimiz degere ulasirsa yeni bir instance kuracak altinda kalirsa mevcudu koruyacaktir.

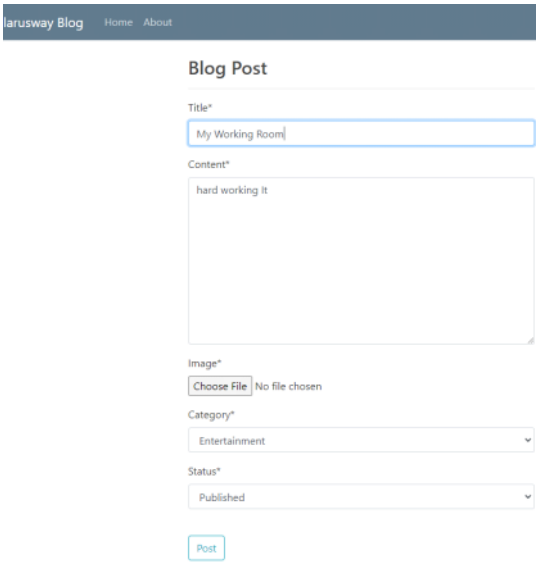
```
<!-- WARNING!!! Sometimes your EC2 has a problem afte
r you create autoscaling group, If you need to look i
nside one of your instance to make sure where the pro
blem is, please follow these steps...
```bash
eval "$(ssh-agent)" (your local)
ssh-add <pem-key>   (your local )
ssh -A ec2-
user@<Public IP or DNS name of NAT instance> (your lo
cal)
ssh ubuntu@<Public IP or DNS name of private instance
>  (NAT instance)
You are in the private EC2 instance
``` -->
```



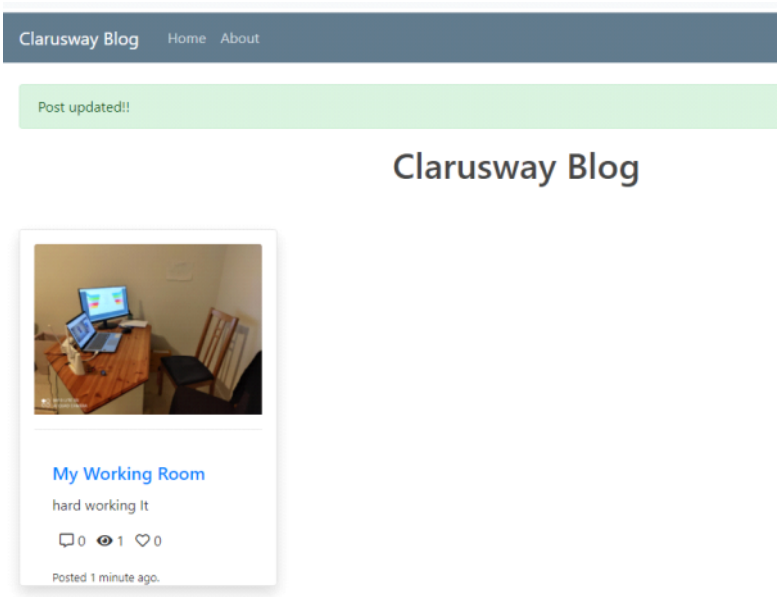
Load balancer in DNS name ini alip Consola yapistirinca karsimiza cikan ekran da sol alt kism da ADVANCE tikliyoruz
Hala not secure gözukmesinin sebebi biz sertifikalari Host Name icin tanimladik Road Balancer in cikarttigi adres icin degil Dolayisiyka bunu oraya yönkendirmemiz gerekiyor

Azcilan sayfada ilk oalrak Register edecegiz ve kendimize bir Mail ve

sifre belirleyip sisteme girecegiz



Ve daha sonra

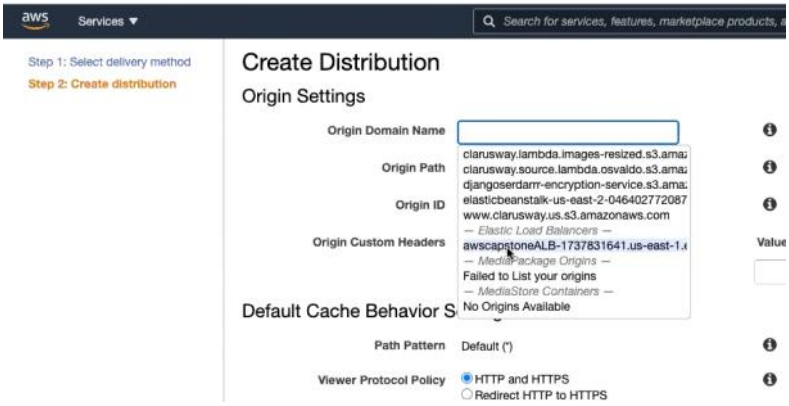


Step 14: Create Cloudfront in front of ALB

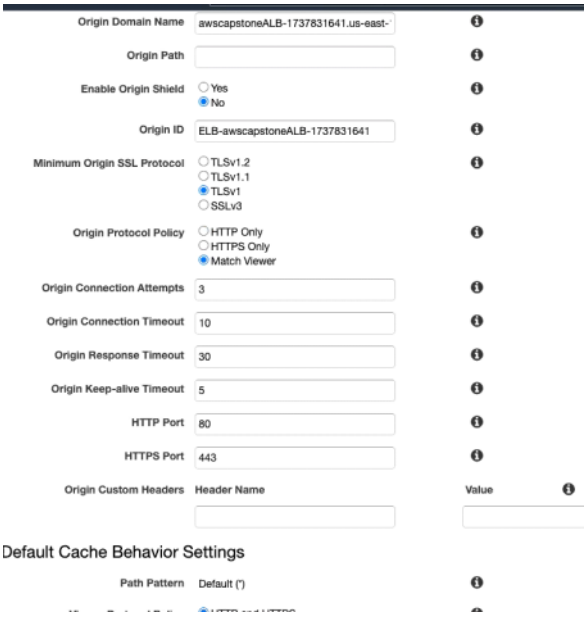
Go to the cloudfront menu and click start
- Origin Settings

Load balancer in Domain Name ini isretliyoruz

```text
Origin Domain Name          : aws-capstone-
ALB-1947210493.us-east-2.elb.amazonaws.com
Origin Path                 : Leave empty (this means
, define for root '/')
```

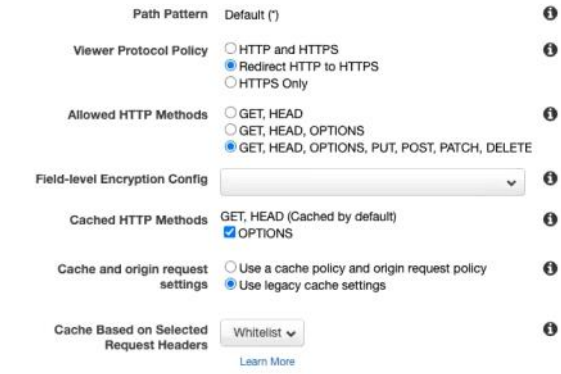


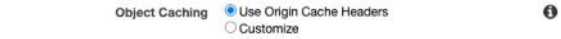

```
Protocol                    : Match Viewer
HTTP Port                   : 80
HTTPS                       : 443
```




```
Minimum Origin SSL Protocol : Keep it as is
Name                        : Keep it as is
Add custom header           : No header
Enable Origin Shield        : No
Additional settings         : Keep it as is
```

```
Default Cache Behavior Settings
```text
Path pattern                                : D
efault (*)
Compress objects automatically              : Y
es
Viewer Protocol Policy                      : R
edirect HTTP to HTTPS
Allowed HTTP Methods                        : G
ET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE
Cached HTTP Methods                         : S
elect OPTIONS
Cache key and origin requests
- Use legacy cache settings
```
```

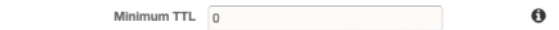


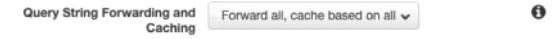


Headerlarda neleri cashlemesini istiyorsak onlari secebiliyoruz

```
 Headers     : Include the following headers

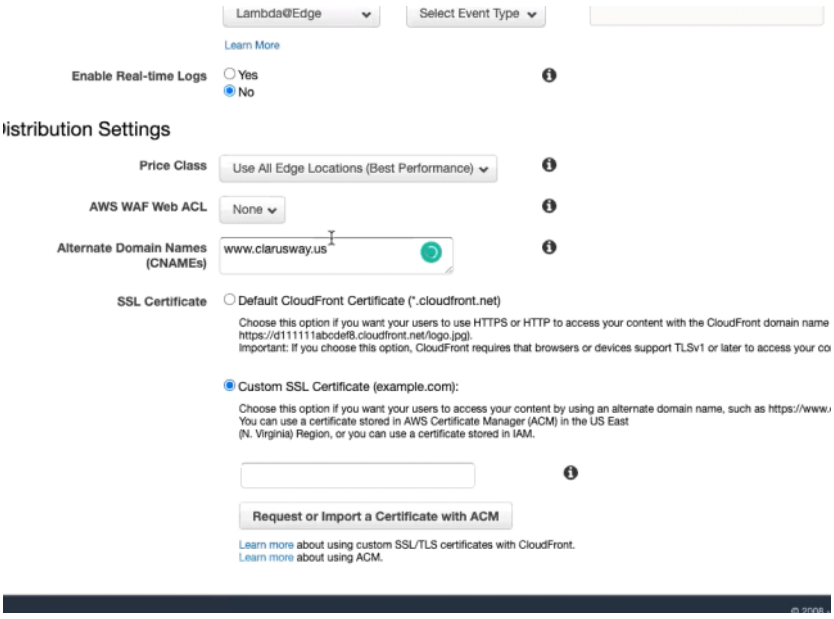
 Add Header
    - Accept
    - Accept-Charset
    - Accept-Datetime
    - Accept-Encoding
    - Accept-Language
    - Authorization
    - Cloudfront-Forwarded-Proto
    - Host
    - Origin
    - Referrer
Forward Cookies                          : All
Query String Forwarding and Caching      : All
Other stuff                              : Keep them a
s are
```





- Distribution Settings

```text
Price Class                              : Use all edg
e locations (best performance)
Alternate Domain Names                   : www.clarusw
ay.us
SSL Certificate                          : Custom SSL
Certificate (example.com) ---> Select your certificat
e creared before
Other stuff                              : Keep them a
s are
```



Ayni zaman bu Ec2 lar ELB ye bagli olacak ELB nin en önemli özelligi EC2 lardaki yoguniuga bakip HelatCheck ler ile her hangir yoguniuk olmasi durumunda esit olarak bu yuku EC2 lar arasinda dagitmak. Hem Trafigi yayiniliyor hem tek elden dagilimi sagliyor. Hemde Dis dunyadan gelen User in her hangi bir aksama olmadan EC2 ya ulasilmasini sagliyor.

ALB nin de önune bir CloudFront koyacagiz ki bu da br cash hizmeti sunacak. Videolar ve fotograflar S3 den cekilecek her seferinde ucret ödenmemesi icin bunu yapacaklar Cash hizmeti ile sagalanacak.

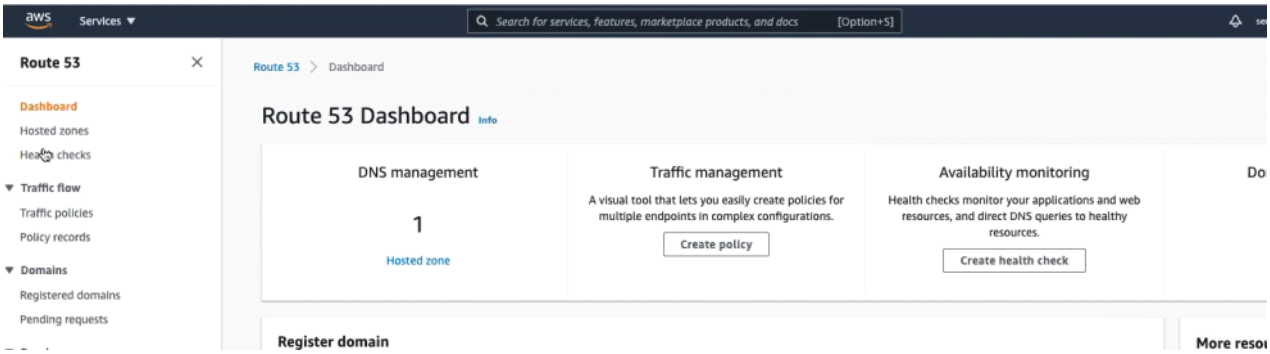
Production oirtaminda Cloudfront s3 un önune de kurulabiliyor Eger burda tanimlamak istersek s3 u orjin olarak gösterecektir.

Developerlar kodunu kullanici karsisina cikartmak icin Route53 failover senaryosu ile karsimiza cikacak.

Burada Failover senaryosu Route 53 olarakta bir ister atanmis durumdadir. Route53 de iki adet ent pointi takip eder. Birisi static website olur. Degiride uniq olarak caalisan web sitemizdir.basina herhangi bir durum gelmesi halinde secondary e trafigi yönlendirir.

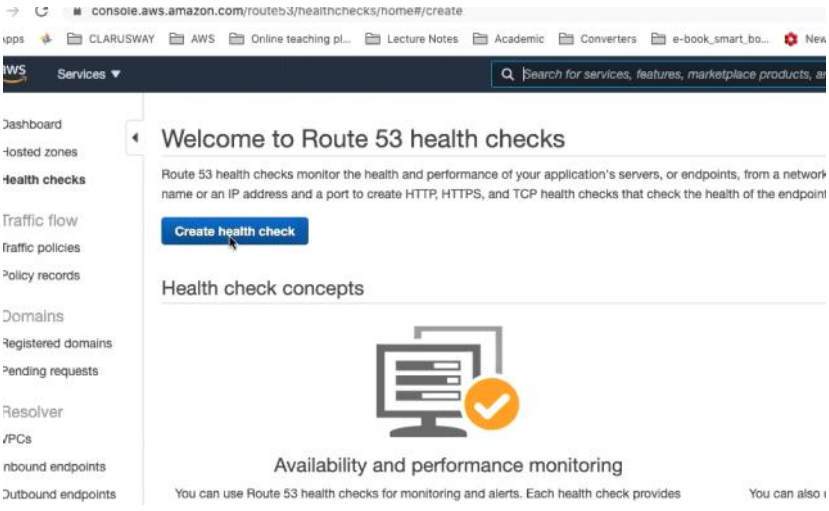
Userlar bu haliyle karsilarina bir sey cikarki mevcut olarak bir sorunun oldugunu ve bu sorunun bir sure sonra cözelecegi gibi seyler icerebilir.

Simdide Rote53 e gidecegziz Failover senaryosunu olustursmak icin.

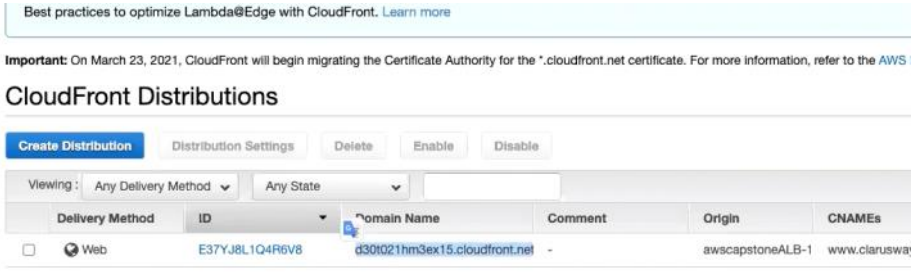


Step 15: Create Route 53 with Failover settings

Come to the Route53 console and select Health checks on the left hand menu. Click create health check

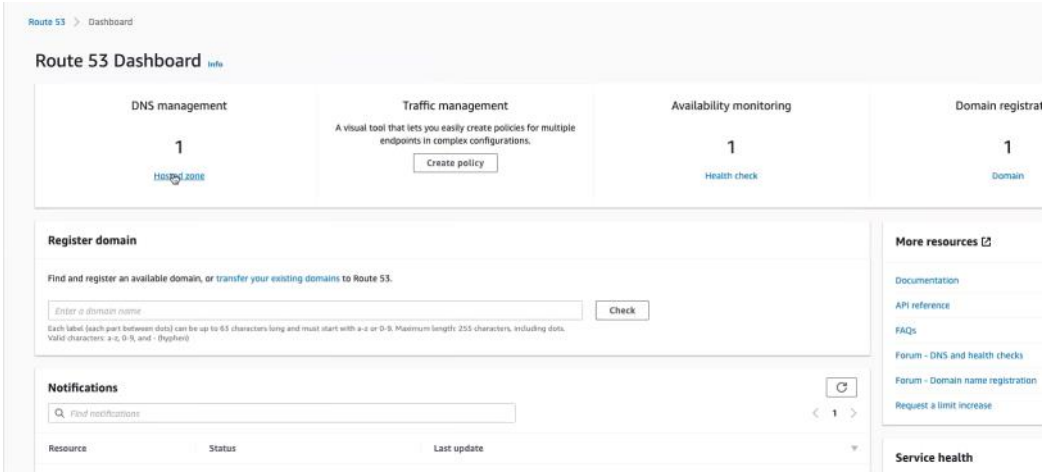


Configure health check
```text
Name                : aws capstone health check
What to monitor     : Endpoint
Specify endpoint by : Domain Name
```

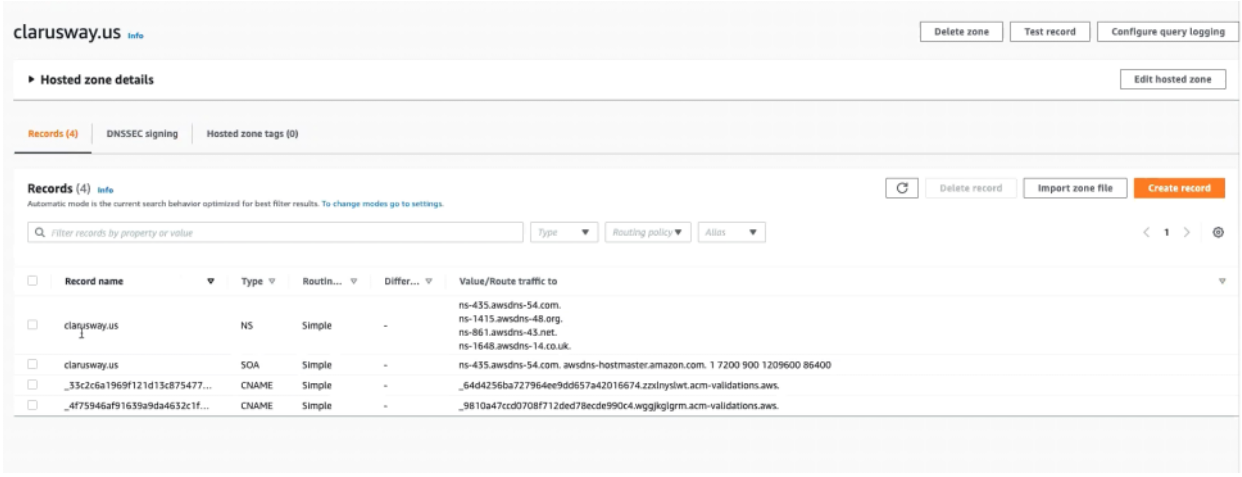


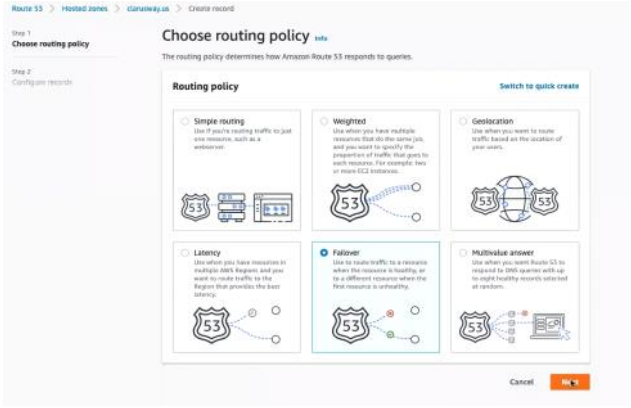
CloudFront taki Domain Name alip buraya kopyaliyoruz.

```text
Protocol            : HTTP
Domain Name         : Write cloudfront domain name
Port                : 80
Path                : leave it blank
Other stuff         : Keep them as are
```

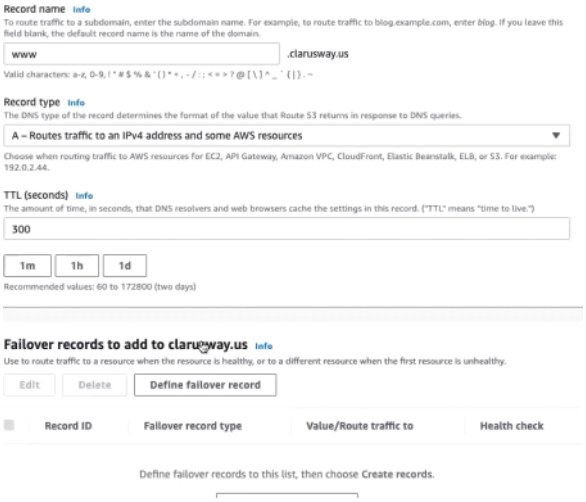


- Click Hosted zones on the left hand menu
- click your Hosted zone : <YOUR DNS NAME>
- Create Failover scenario
- Click Create Record
- Select Failover ---> Click Next

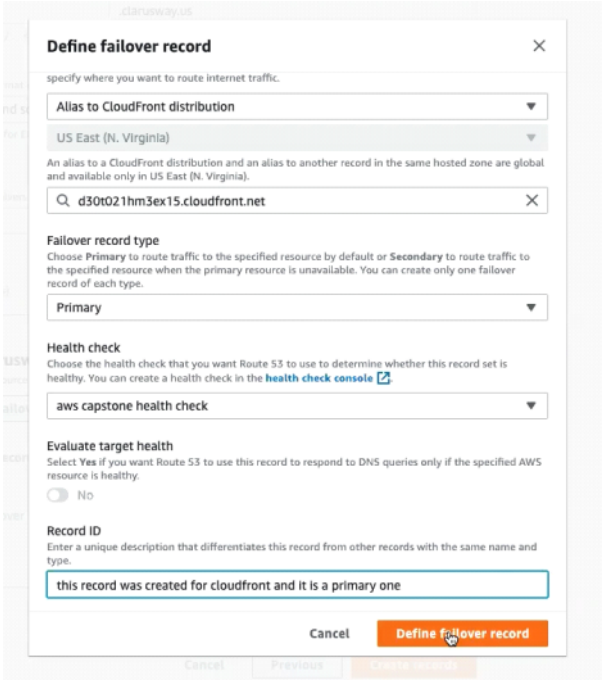





```text
Configure records
Record name             : www.<YOUR DNS NAME>
Record Type             : A - Routes traffic to an IP
v4 address and some AWS resources
TTL
```

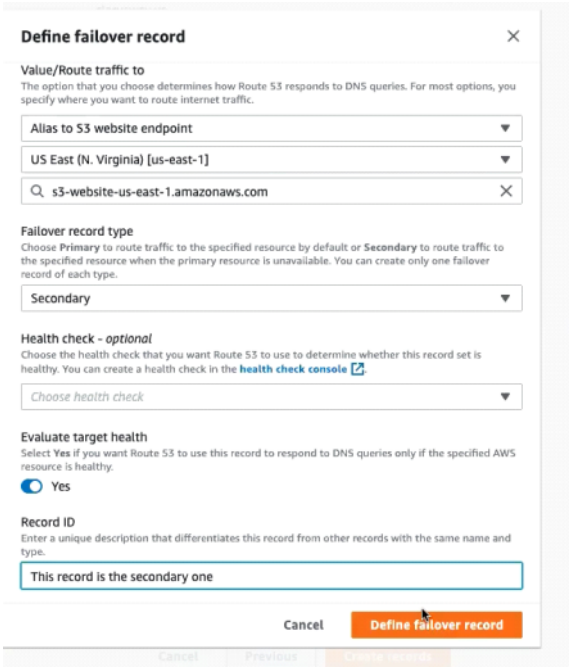



```
                    : 300
---> First we'll create a primary record for cloudfro
nt
Failover record to add to your DNS ---> Define failov
er record
Value/Route traffic to  : Alias to cloudfront distrib
ution
                          - Select created cloudfront
 DNS
Failover record type    : Primary
Health check            : aws capstone health check
Record ID               : Cloudfront as Primary Recor
d
```

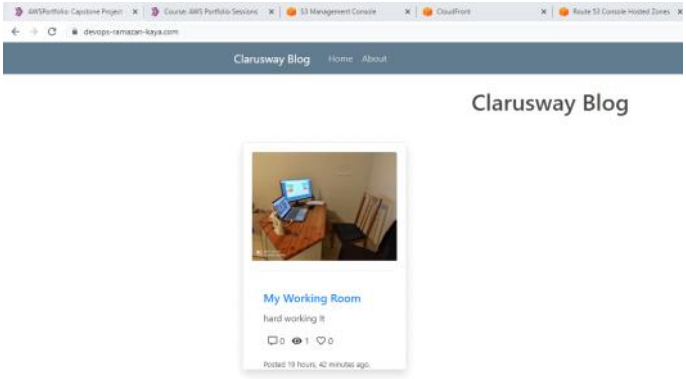



```
----------------------------------------------------
----------
---> Second we'll create secondary record for S3
Failover another record to add to your DNS ---> Defin
e failover record
Value/Route traffic to  : Alias to S3 website endpoin
t
                          - Select Region
                          - Your created bucket name
emerges ---> Select it
Failover record type    : Secondary
Health check            : No health check
Record ID               : S3 Bucket for Secondary rec
ord type
```

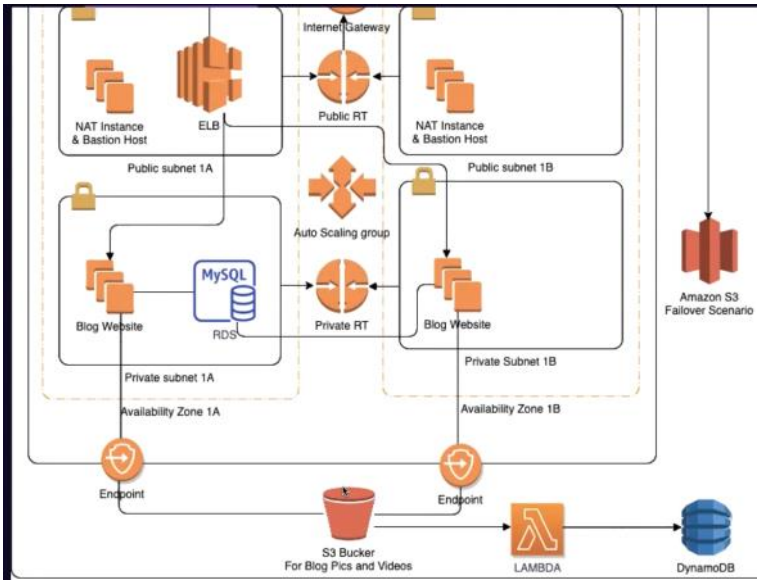
- click create records



Artikn sitemiz yayin yapmaya basladi ve secure bir yayindir .
Clou´dFront ile olusturdugumuz yayi su anda secure bir bagaalanti



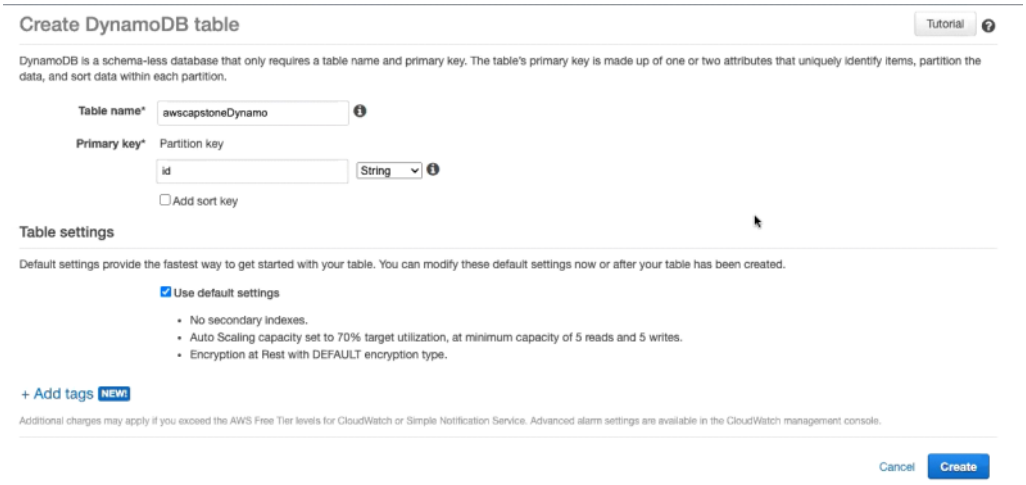
Simdi secure bir baglanti yapmis oldulk



Developerlar s3 de atilan objelerin takibini yapmak icin DYNAMODB de liste haliunde bunun takibini yapmak istiyor lar

Step 16: Create DynamoDB Table

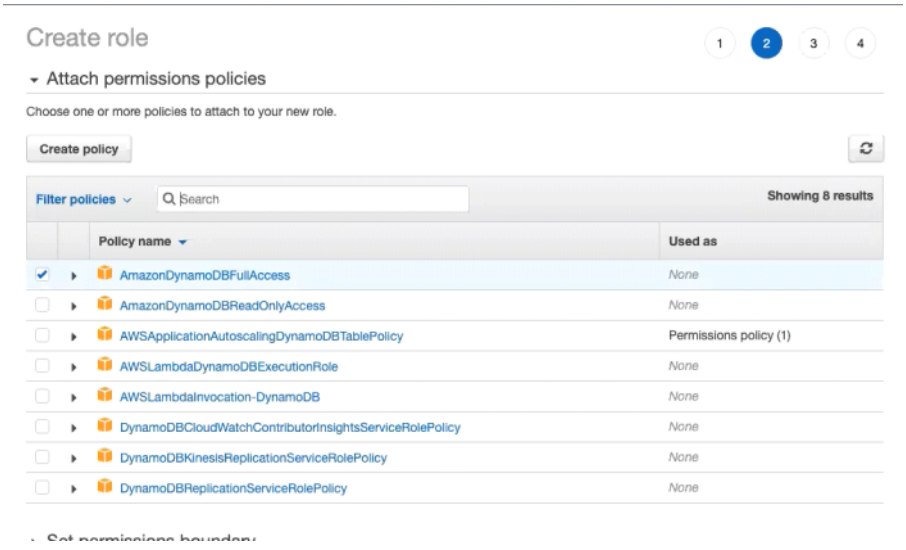
Go to the Dynamo Db table and click create table butt on
- Create DynamoDB table
```text
Name            : awscapstoneDynamo
Primary key     : id
Other Stuff     : Keep them as are
click create
```



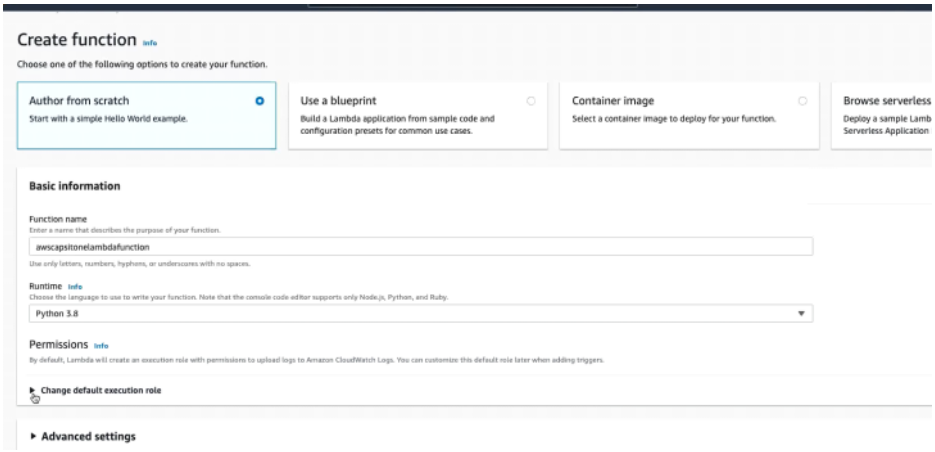

```
## Step 17-18: Create Lambda function

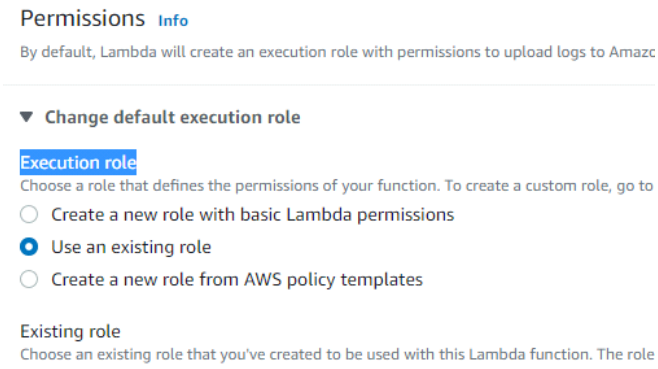
**Lambda Fonksiyonunun s3 e ulasmasi lazim yani konusmasi lazim ayrica Dynamodb ile konususyor olamsi a+lazim ayrica Network  altyapisinda irtibatalanmasi gerekiyor**

Before we create our Lambda function, **we should creat e IAM role t**hat we'll use for Lambda function. Go to the IAM console and select role on the left hand menu , then create role button
```text
Select Lambda as trusted entity ---> click Next:Permi ssion
Choose: - LambdaS3fullaccess,
        - Network Administrator
        - DynamoDBFullAccess
```

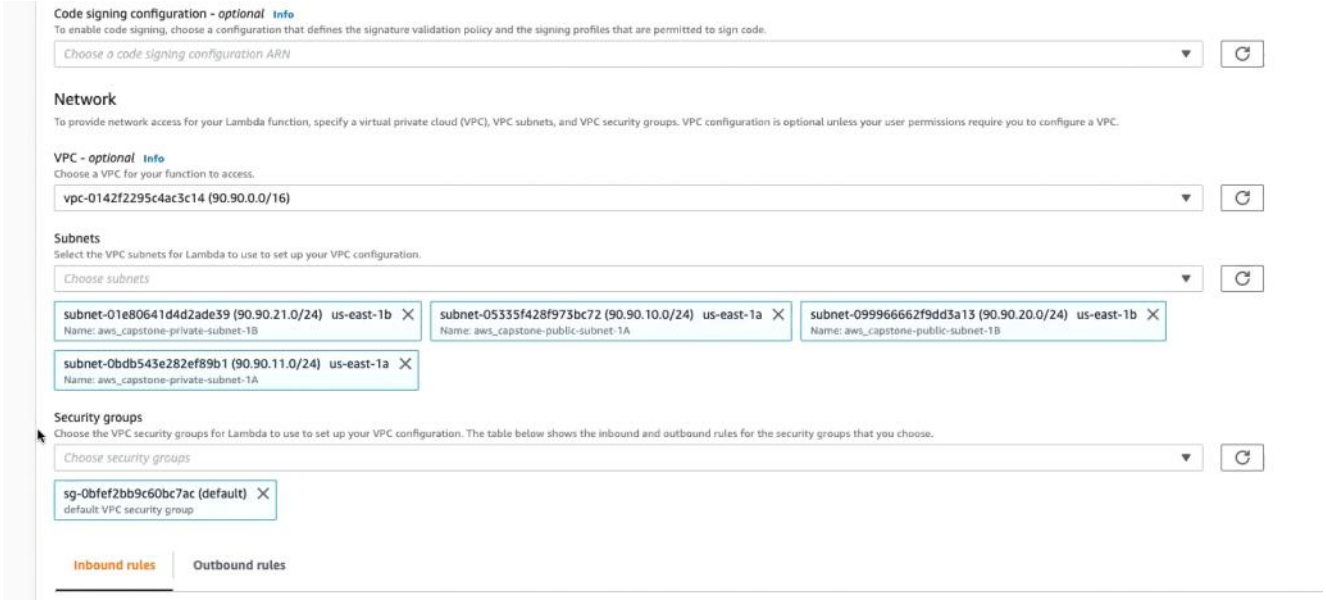



```
No tags
Role Name           : aws_capstone_lambda_Role
Role description    : This role give a permission to
```
```

lambda to reach S3 and DynamoDB on custom VPC
```
then, go to the Lambda Console and click create funct
ion
- Basic Information
```text
```



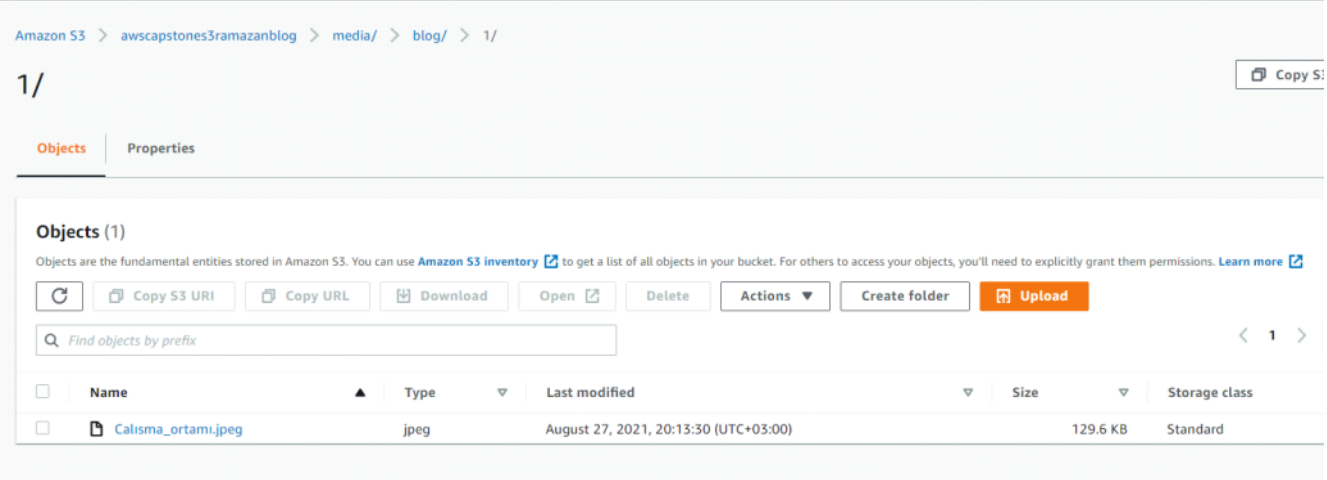



```
Function Name           : awscapsitonelambdafunction
Runtime                 : Python 3.8
Create IAM role         : S3 full access policy
Advance Setting:
Network                 :
    - VPC               : aws-capstone-VPC
    - Subnets           : Select all subnets
    - Security Group    : Select default security Gro
up
```
- Now we'll go to the S3 bucket belongs our website a
nd create an event to trigger our Lambda function.



Ve simdi de s3 de event olusturacagiz; bu s3 bucket imiizin altina ne
```

dosya atilirsa bir event olsuturmasini isteyecegiz



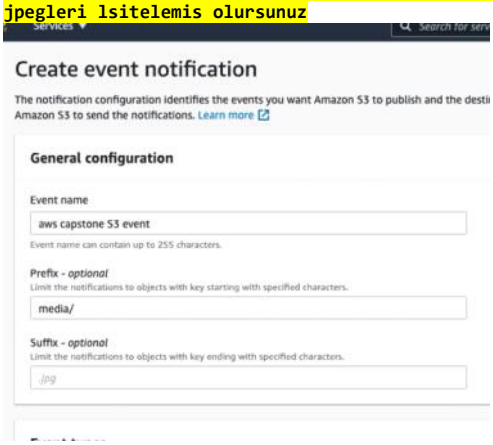
**Bucketimizin-------- Properties kismina tiklayip daha somra---------
event notifications kismina tiklayacagiz**

Evevnt Notifications S3 un bucktin altinda her hangi bir sey olusstugu taktirde onu event olusturur
Ve bu event i Lambda functiona tanitacagiz

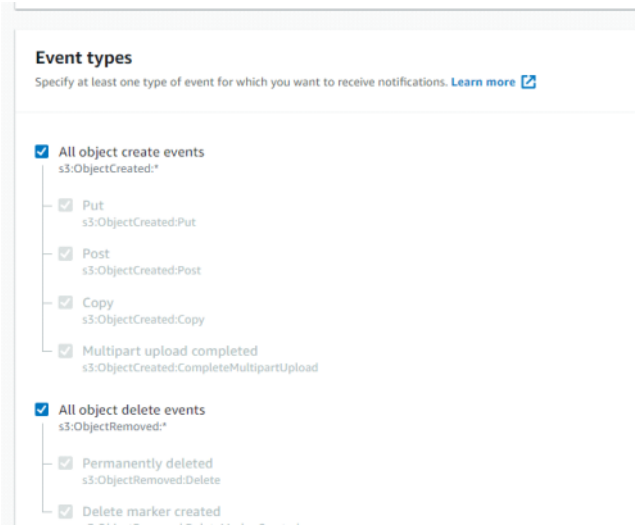
Step 17-18: Create S3 Event and set it as trigger for Lambda Function

Go to the S3 console and select the S3 bucket named `awscapstonec3<name>blog`.
- Go to the properties menu ---> Go to the Event notifications part
- Click create event notification for creating object
```text
Event Name              : aws capstone S3 event
Prefix                  : media/
```

Egerki Suffix kismina jpeg belirtirseniz sedece jpegleri lsitelemis olursunuz

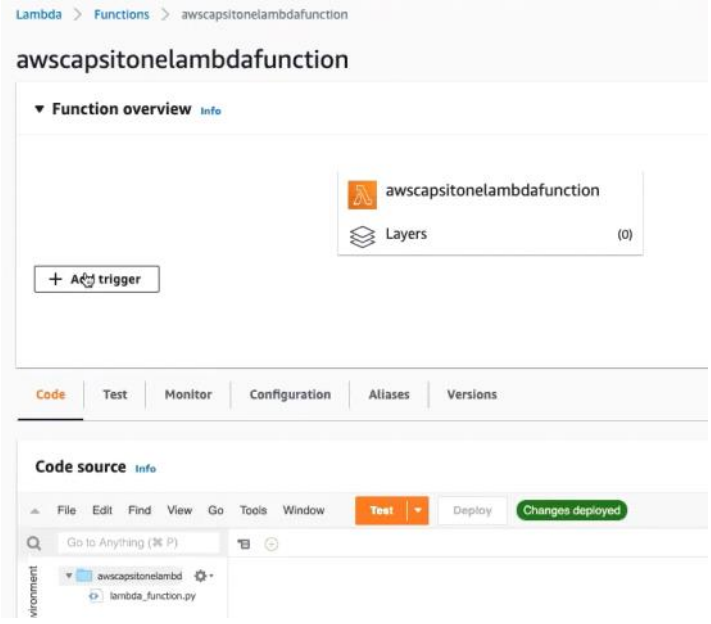



```
Select                  :
    - All object create events
Destination             : Lambda Function
Specify Lambda function : Choose from your Lambda functions
Lambda funstion         : awscapstonelambdafunction
click save
```

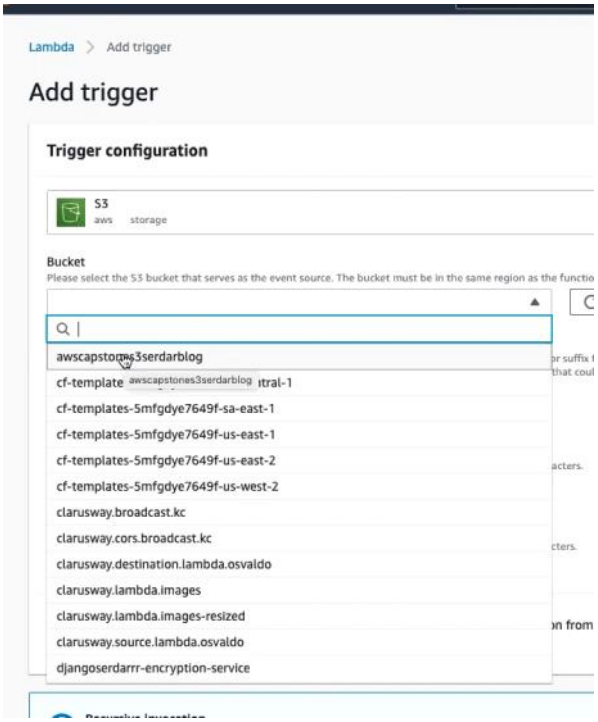



```text
```

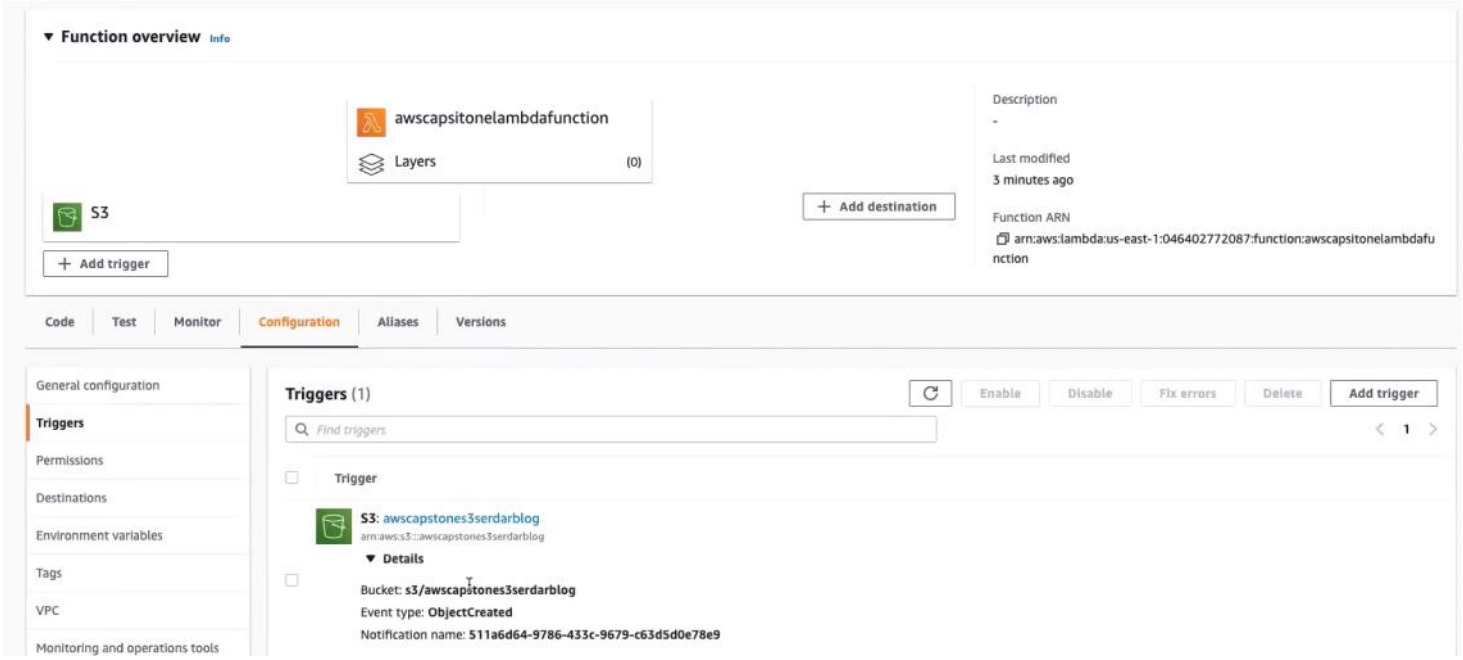
- After create an event go to the `awscapstonelambdaf unction` lambda Function and click add trigger on the top left hand side.
- For defining trigger for creating objects



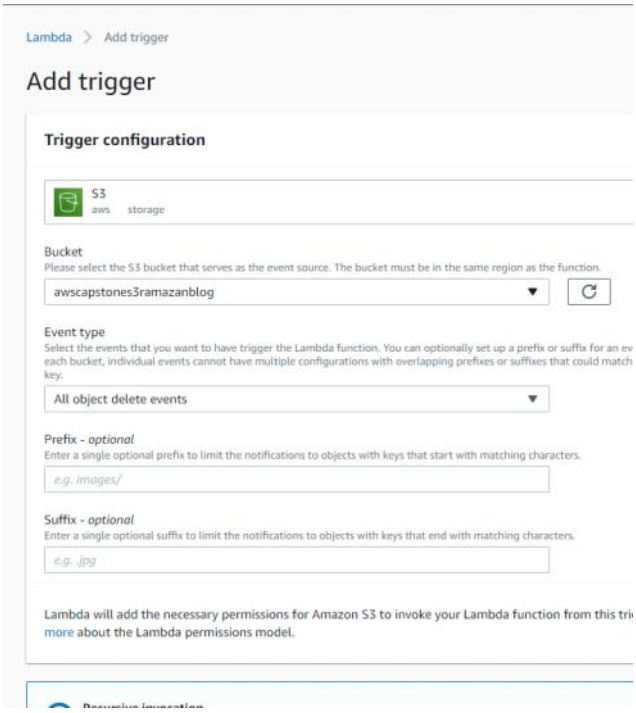

```text
Trigger configuration   : S3
Bucket                  : awscapstonec3<name>blog
Event type              : All object create events
Check the warning message and click add ---> sometime
s it says overlapping situation. When it occurs, try
refresh page and create a new trigger or remove the s
3 event and recreate again. then again create a trigg
er for lambda function
```

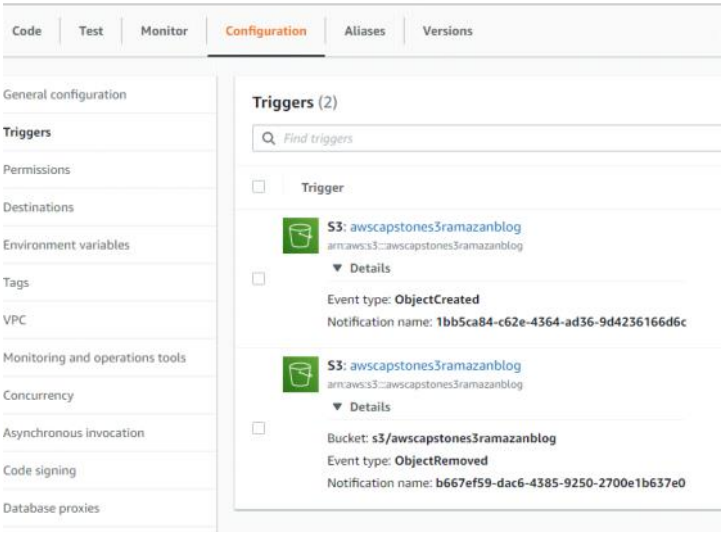


Hata verebiliyor eger hata verir ise lamda function s
una gidip silip tekarar Trigger edecegiz



- For defining trigger for **deleting objects**
```bash
Trigger configuration   : S3
Bucket                  : awscapstonec3<name>blog
Event type              : All object delete events
Check the warning message and click add ---> sometime
s it says overlapping situation. When it occurs, try
refresh page and create a new trigger or remove the s
3 event and recreate again. then again create a trigg
er for lambda function
```

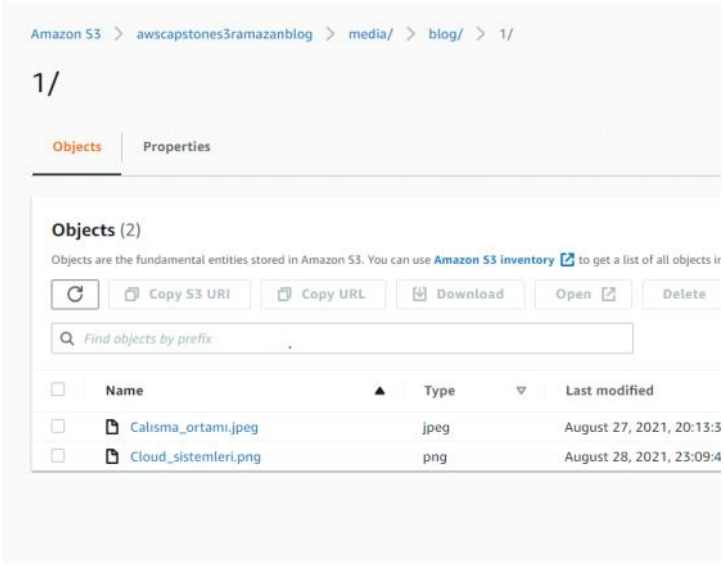


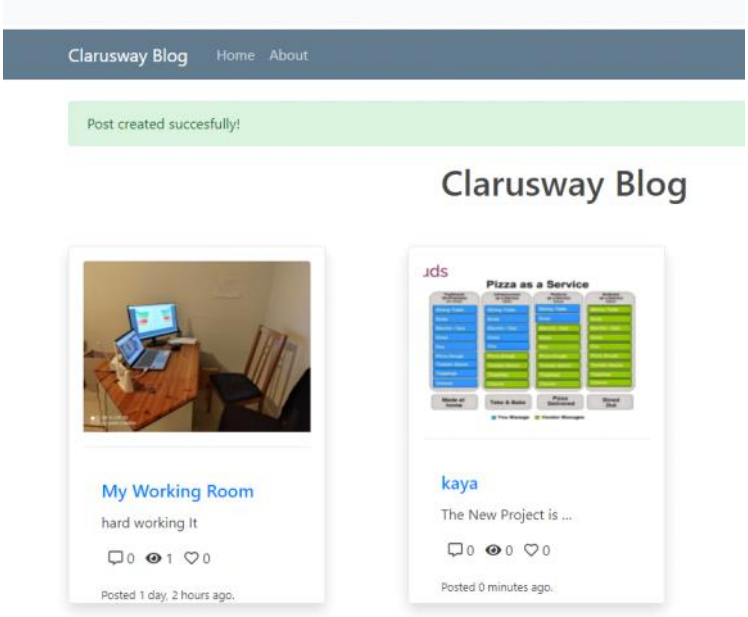


- Go to the code part and select lambda_function.py -
--> remove default code and paste a code on below. If
you give DynamoDB a different name, please make sure
to change it into the code.

```python
import json
import boto3
def lambda_handler(event, context):
    s3 = boto3.client("s3")

    if event:
        print("Event: ", event)
        filename = str(event['Records'][0]['s3']
['object']['key'])
        timestamp = str(event['Records'][0]
['eventTime'])
        event_name = str(event['Records'][0]
['eventName']).split(':')[0][6:]

        filename1 = filename.split('/')
        filename2 = filename1[-1]

        dynamo_db = boto3.resource('dynamodb')
        dynamoTable = dynamo_db.Table('awscapstoneDyn
amo')

        dynamoTable.put_item(Item = {
            'id': filename2,
```

```
            'timestamp': timestamp,
            'Event': event_name,
        })

    return "Lammda success"
```
- Click deploy and all set. go to the website and add
 a new post with photo, then control if their record
is written on DynamoDB.
- WE ALL SET
- Congratulations!! You have finished your AWS Capsto
ne Project





Yeni yukledigimiz nesnenin geldigini görebiliyoruz