# Amazon Machine Image (AMI) & Snapshot

Saturday, 10 July 2021 11.01


## Today's Takeaways

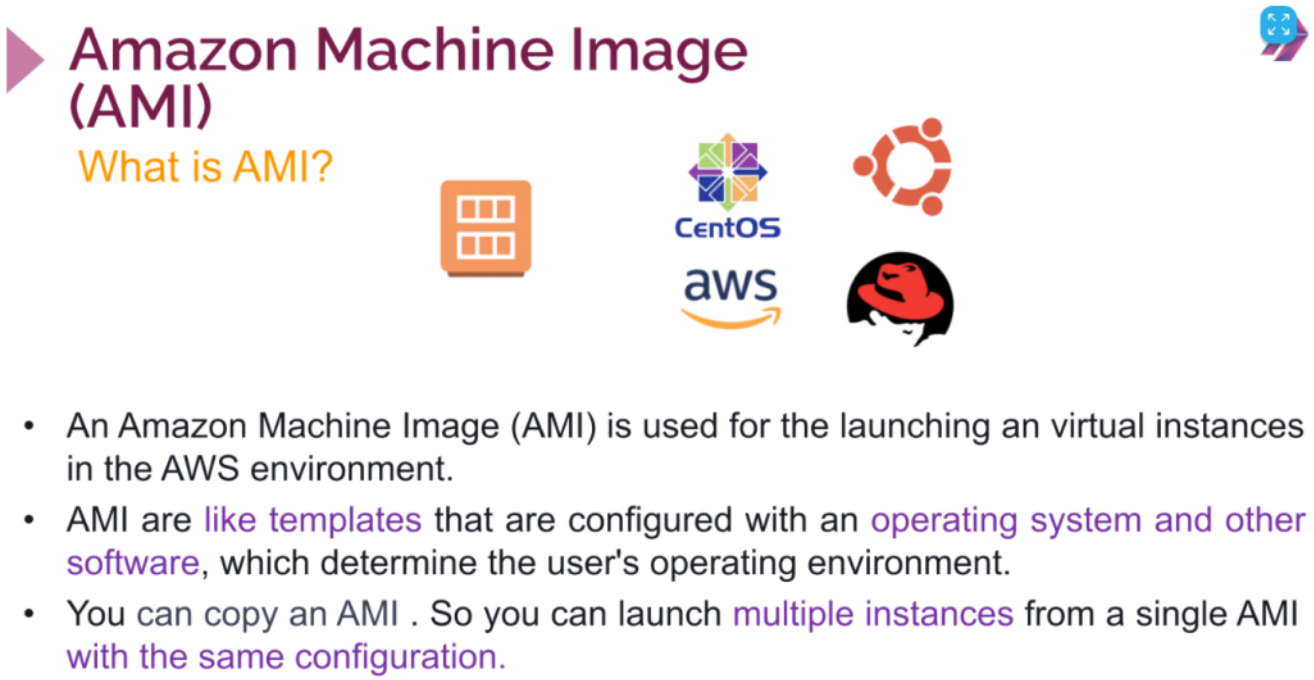
- Amazon Machine Image (AMI)
- Snapshot

## 1 Amazon Machine Image (AMI)

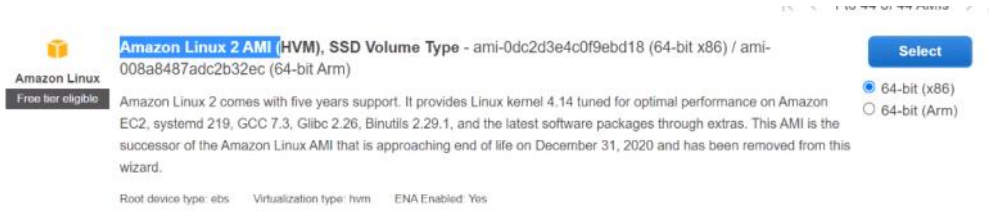
**What is Amazon Machine Image (AMI) ?**

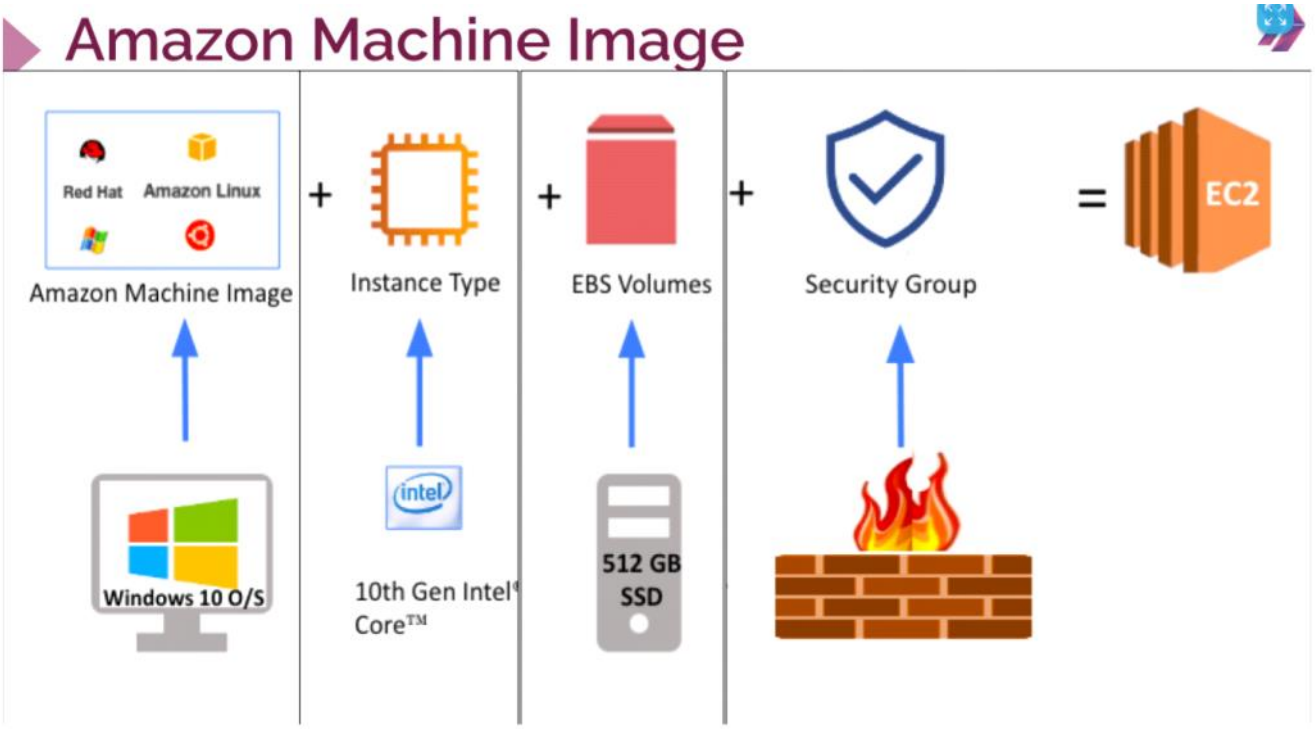
AMI is a virtual image used to create a virtual machine within an EC2 instance. In other words, it is a virtual machine template containing predefined operating system and application files.
When creating a virtual machine, you will get an operating system and application list according to the features presented in that template by choosing which template to create this machine. All AMI provides a template for the root volume of an

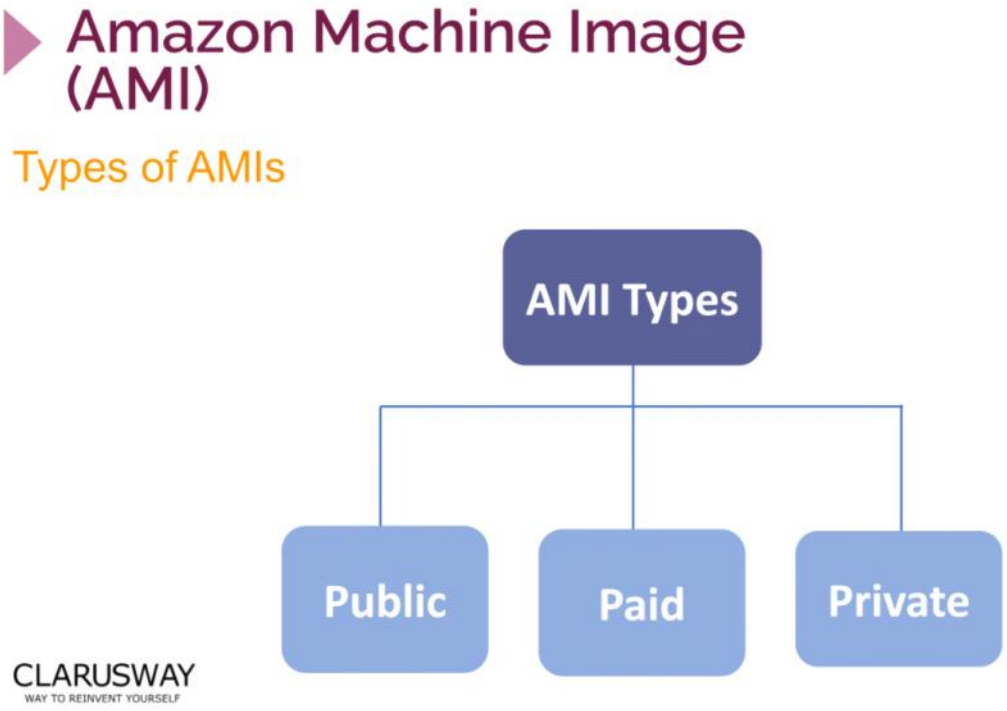
instance. You can copy the AMI and create another instance also.




Evdeki pc lerde os, application var ve virtual instancede de ayni secenekler mevcut. Linux, windows vs secebiliyoruz. AMI template gibi dusunebiliriz. Istersek otomatik snapshot da alabiliriz.






**Public:**

These are the Public Shared Community AMIs and the AMIs managed by the Amazon itself. This package **covers common server features**.
For example, UBUNTU, one of the most famous community-driven distributions in the Linux world, has its own AMIs and you get a server with UBUNTU installed.
And also The Amazon Linux AMI, which Amazon has prepared itself, and includes several AWS applications similar to AWS CLI, as well as the basic

Linux operating system. All of these and more are AMIs available to everyone.

**Paid:**

Another type of AMI is paid versions that we call Paid. These are ready-made packages **created by various companies or independent developers**, including various applications as well as the operating system.
For example, an application creator creates a Linux server image with its own application installed in the AWS store called Amazon Market Place and sets a price. You can buy it by accepting this price

**Private:**

There are also AMIs that we can create and manage with AWS Marketplace and Private Image BuildService. It's now in public beta and enables AWS customers to purchase your installable software products through AWS Marketplace. Then you can install those products and specify them with **Private Image Build Service** for your IT needs as you see in the picture above.


Instancelerin EBS lerin anlik kopyasini cikariyoruz ve S3 e atiyoruz.

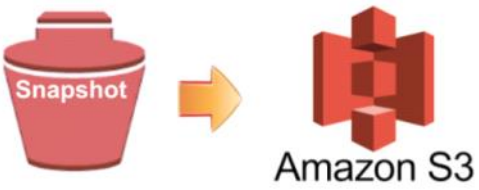
# Snapshot

## What is Snapshot?


- It is a point-in-time copy of your Amazon EBS Volume/Instance
- Snapshots are used for the purpose of
  - Backup
  - Copying AMI for creating multiple instances with the same features.
  - Creating a new Volume

# Snapshot

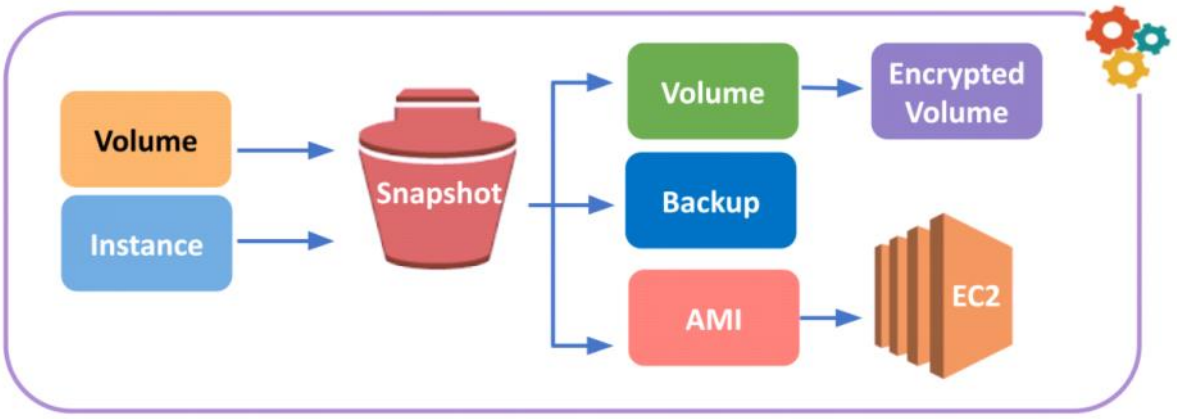
## Features of the Snapshot




- Source from Volume or Instance
- Stored in Amazon S3
- Incremental storage
- Data Lifecycle Manager (DLM)

# Snapshot

## Lifecycle of Snapshot

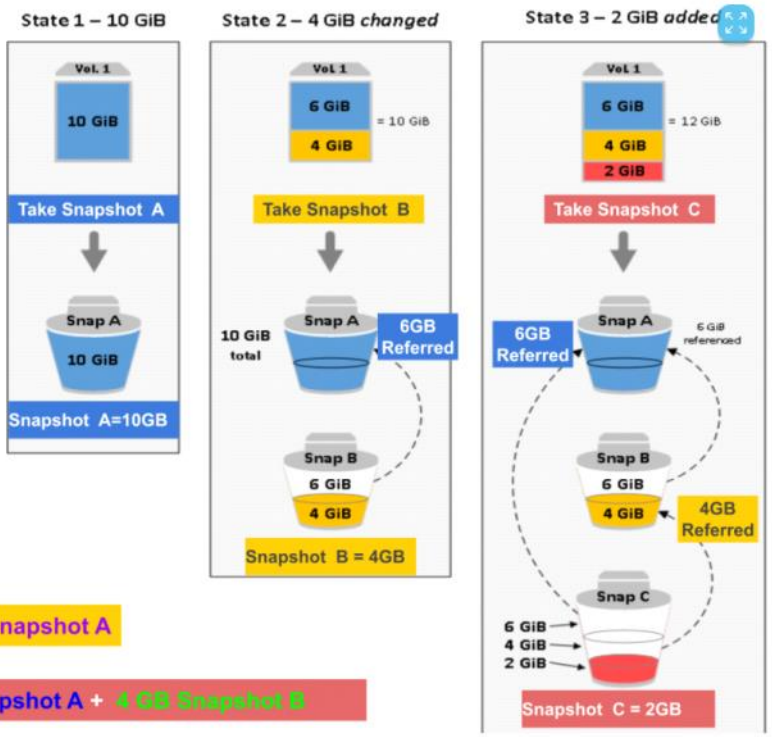



https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSSnapshots.html

# Snapshot

## Incremental Backups

**32 GIB vs. 16 GIB**




Snapshot A = 10 GB

Snapshot B = 4 GB Changed + Referred 6 GB Snapshot A

Snapshot C = 2 GB Added + Referred 6 GB Snapshot A + 4 GB Snapshot B

# Snapshot

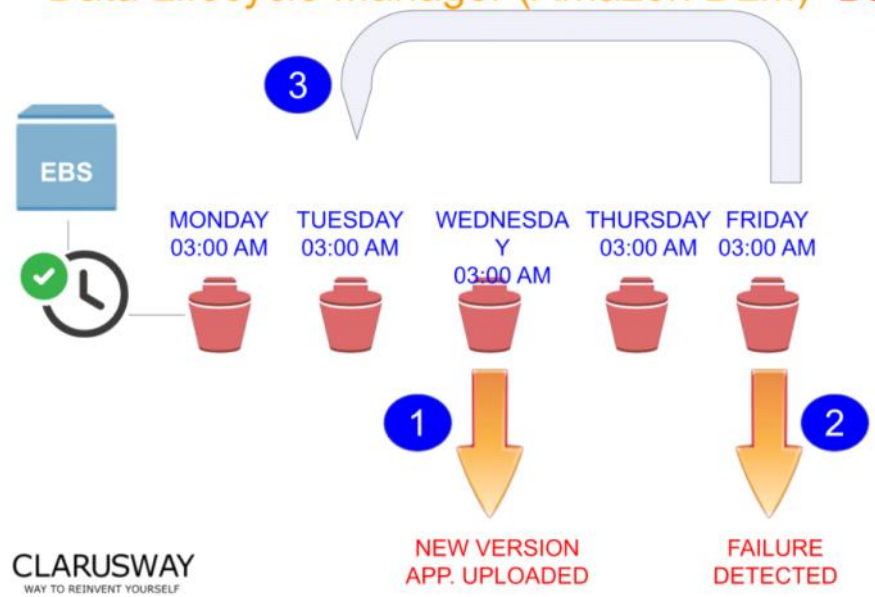
## Data Lifecycle Manager (Amazon DLM)


# Snapshot

## Data Lifecycle Manager (Amazon DLM)


# Snapshot

## Data Lifecycle Manager (Amazon DLM)- Backup and Restore



# Snapshot

## Encryption of Root Device via Snapshot

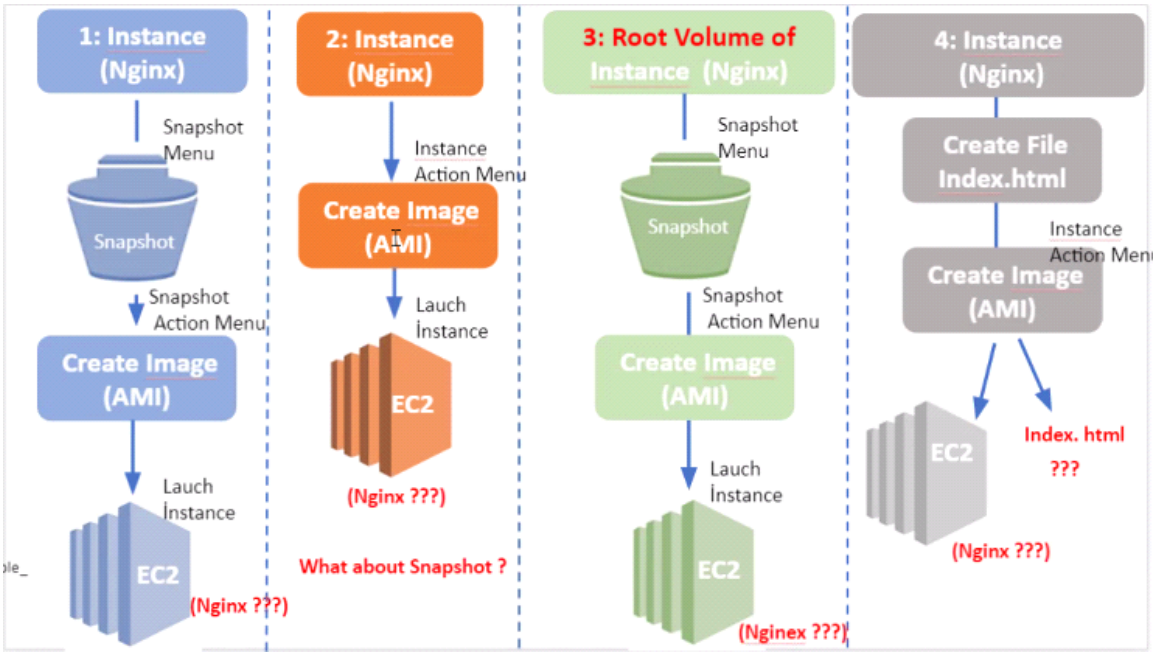

- Root device (volume) cannot be encrypted after creation. "How to encrypt unencrypted volume after creation" is a common question that can be asked in certification exams!
  - Take snapshot of unencrypted volume.
  - Copying the unencrypted Snapshot,
  - You are able to encrypt this Snapshot while coping
  - Create an encrypted volume from this copied Snapshot.

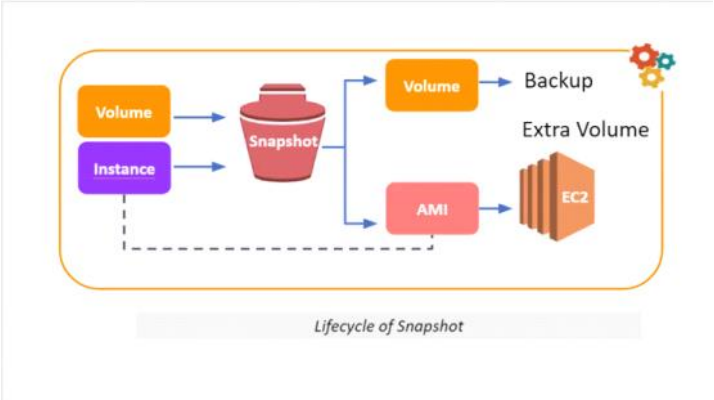
# Snapshot

## Let's get our hands dirty!

- Create Snapshots
- Make Public The Snapshot
- Data Life Cycle Manager
- Creating AMI from the Snapshot
- Creating Volume from the Snapshot
- Creating an Image from Instance

- Bu gun 4 ayri seneryo gerceklestirecegiz
- AWS/hands-on icerisindeki AMI.md dosyasina gore ilerleyecegiz 'Hands-on EC2-05 : Working with EC2 Snapshots'



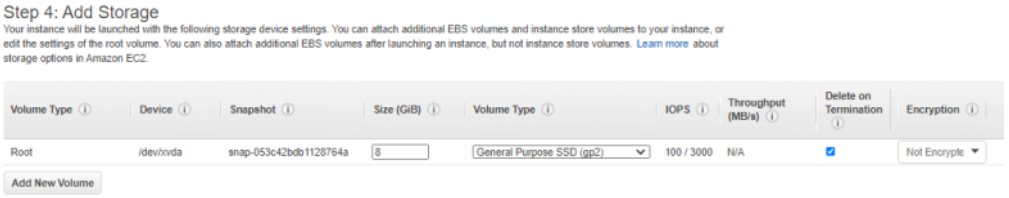




- Oncelikle sample instanceyi olusturalim

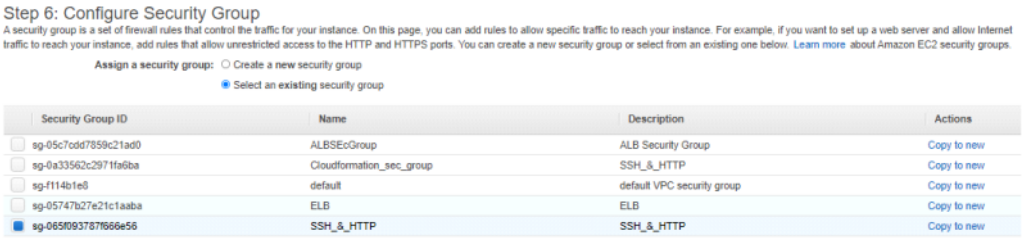


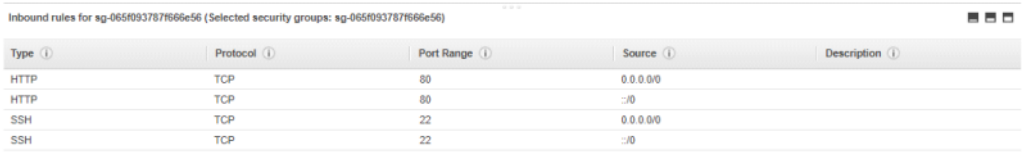
- User data ekleyecegiz.

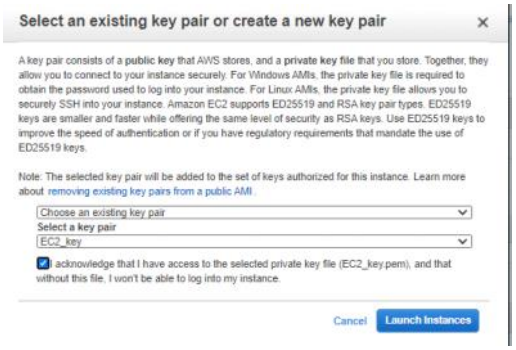
```
#!/bin/bash

  yum update -y
  amazon-linux-extras install nginx1.12
  yum install wget -y # amac github dan cekmek
  cd /usr/share/nginx/html
  chmod o+w /usr/share/nginx/html
  rm index.html
  wget https://raw.githubusercontent.com/awsdevopsteam/route-53/master/index.h
tml
  wget https://raw.githubusercontent.com/awsdevopsteam/route-53/master/ken.jpg
  systemctl start nginx
  systemctl enable nginx
```



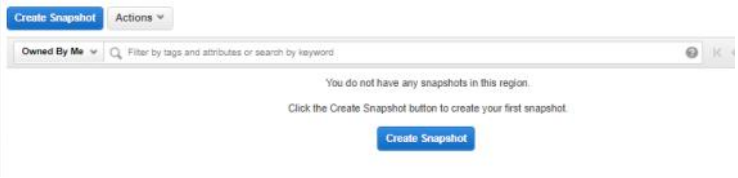


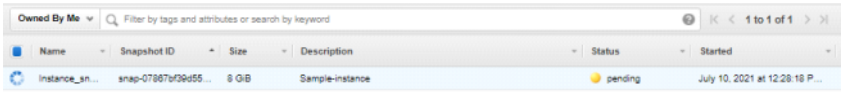


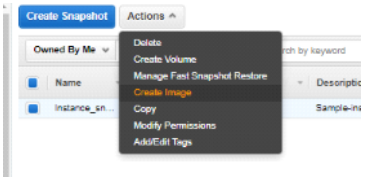



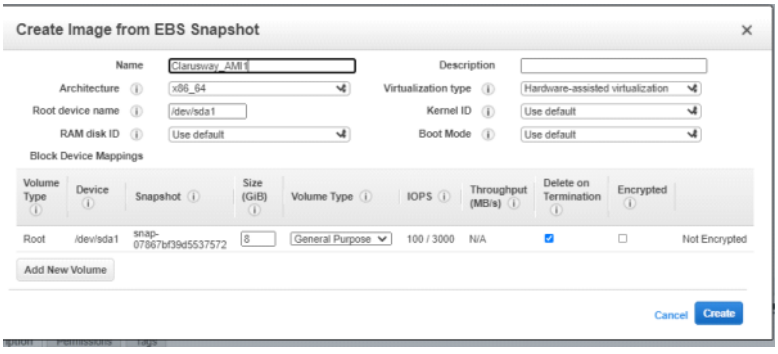

- DNS i actigimizda asagidaki gorseli alacagiz


- AMI ve snapshots boardlarini iki ayri pencerede acalim.
- Kaynak olarak instance secip snapshot alacagiz.
- Create
- 

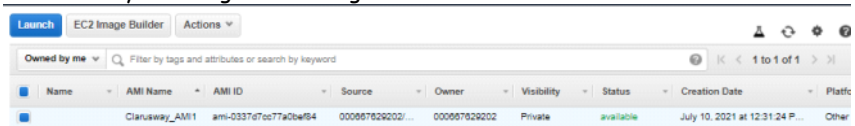




- snapshot dan AMI olusturacagiz

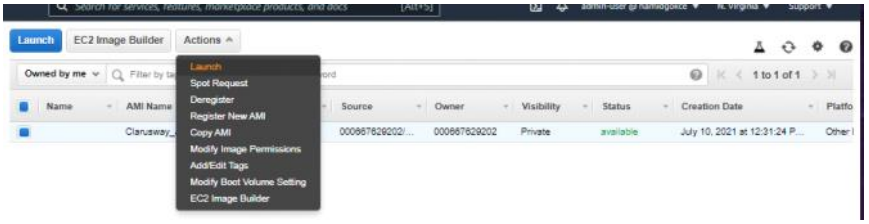


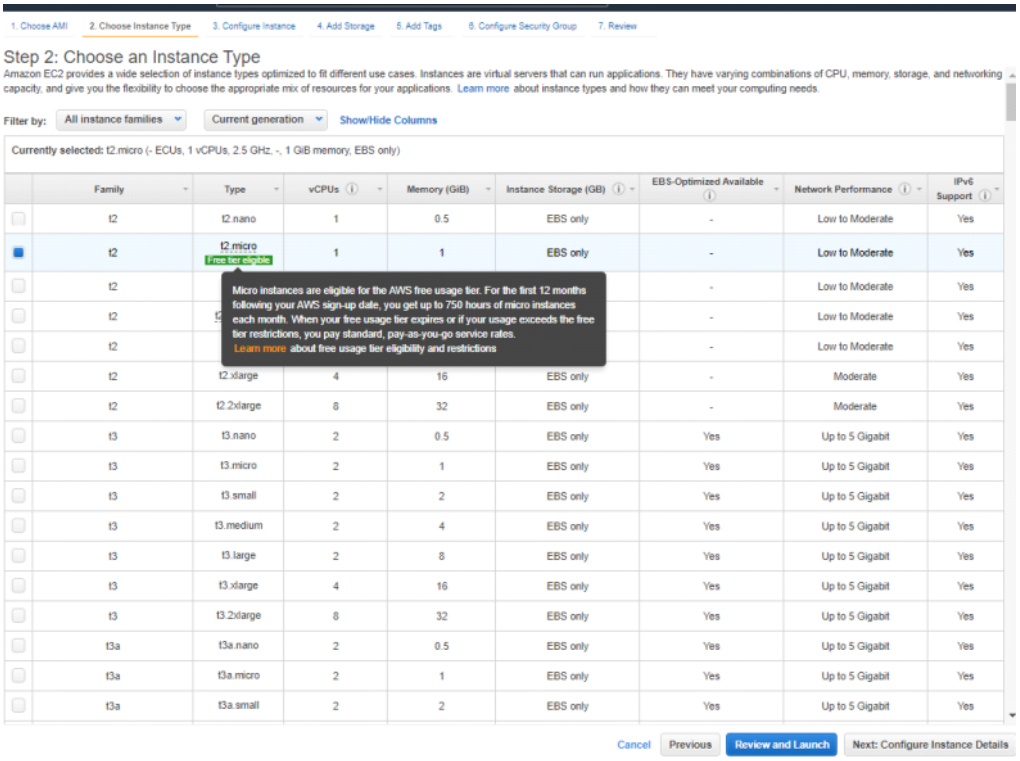

- AMI sayfasinda gorseli alacagiz



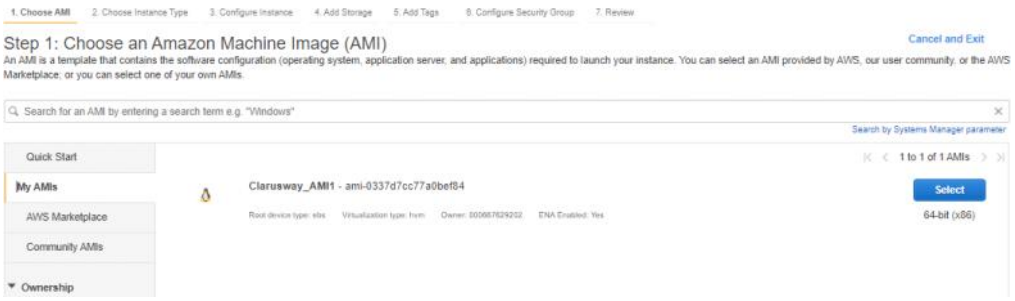

- Simdi EC2 olusturacagiz AMI den



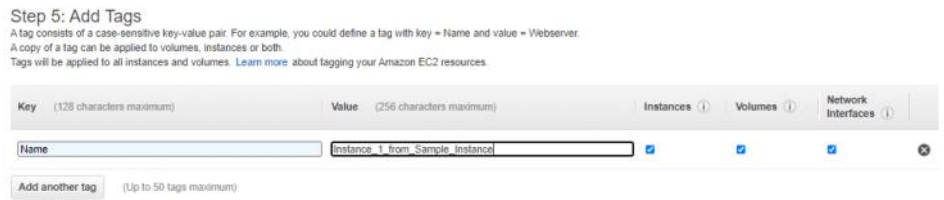

- Otomatik olarak ikinci sayfadan baslatacak



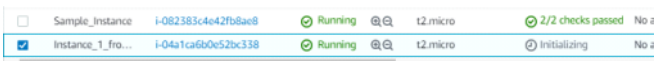

- Launch instance den direk bu sayfaya da gelebiliriz

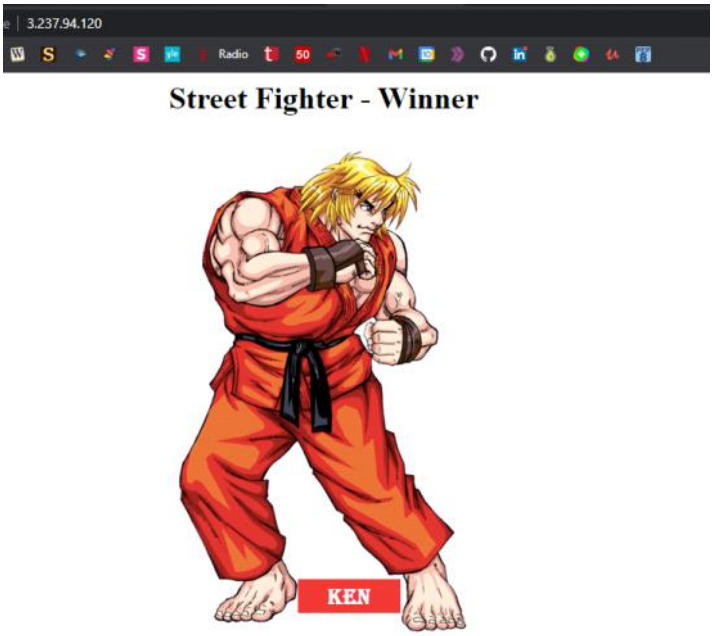



- User data kismi olmayacak ama arka planda hazir oldugunu bilelim
- Tag atayalim ve 22/80 portlari olacak

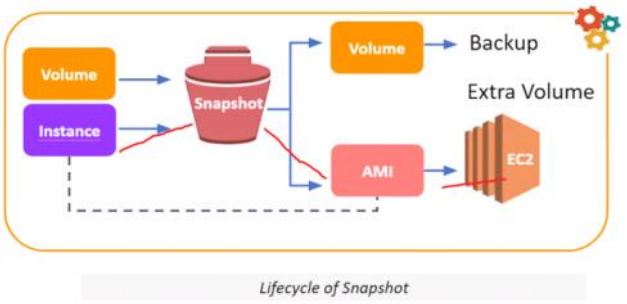



- Olusturdugumuz diger instance de calisir hale geldi ve dns i kopyalarip goruntuyu alacagiz



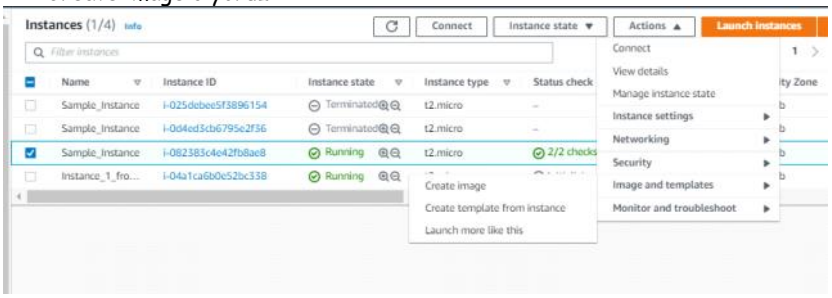



Gorselde kirmizi cizgili senaryoyu tamamladik



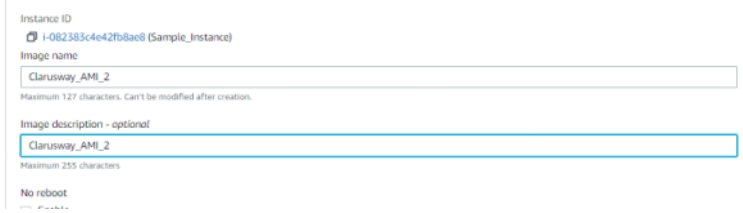

*Lifecycle of Snapshot*

- Create image diyoruz

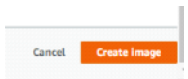



- Image name ve description verecegiz.

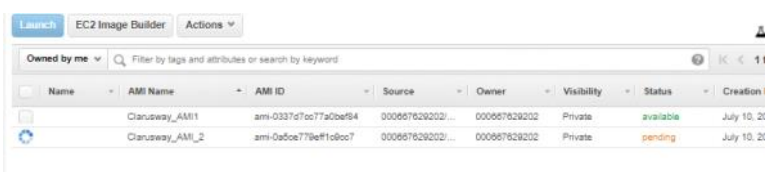
-

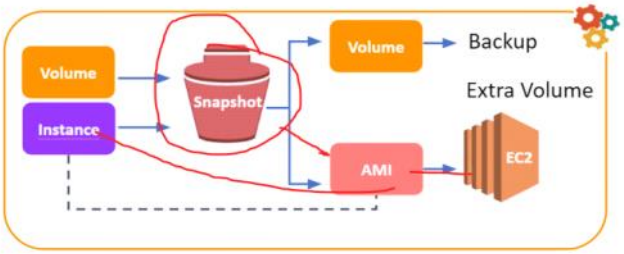



-



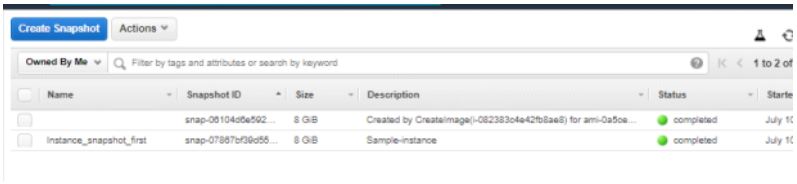

- Olusturdugumuz AMI_2 pending asamasinda AMIs sayfasinda



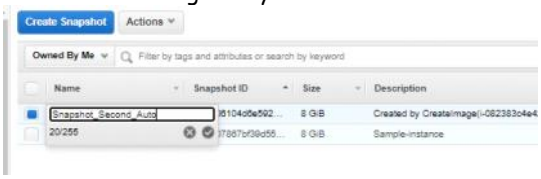


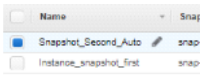

*Lifecycle of Snapshot*

- Gorseline ulasacagiz

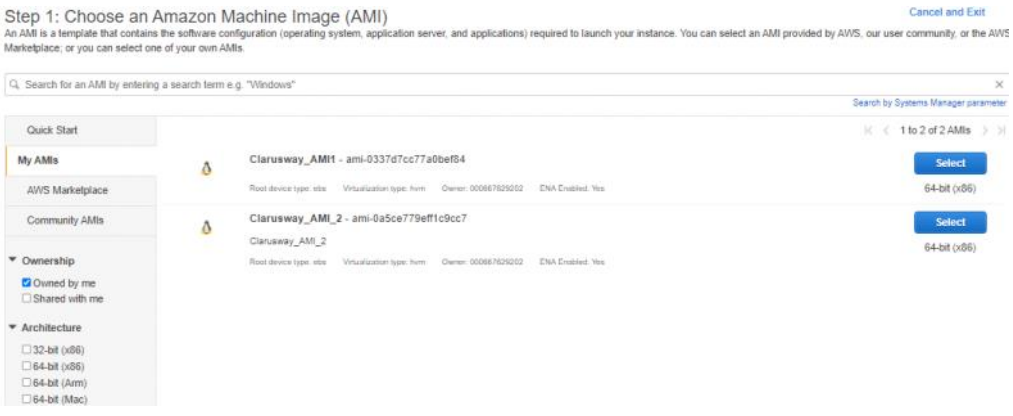


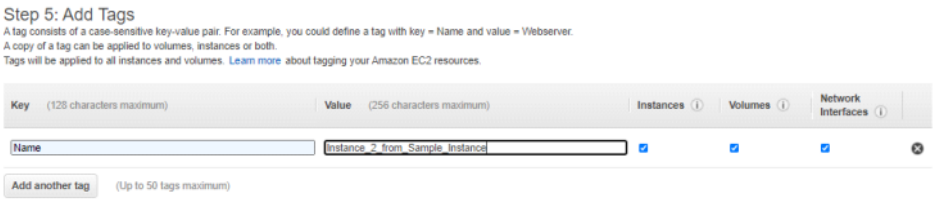

- Gorseldeki tag i veriyoruz

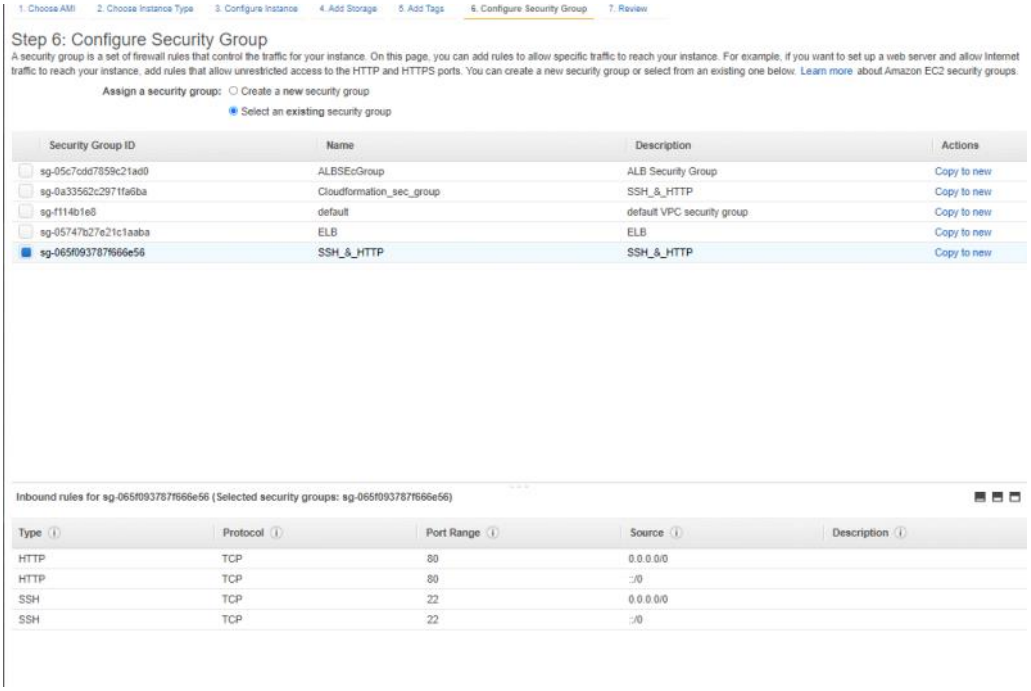


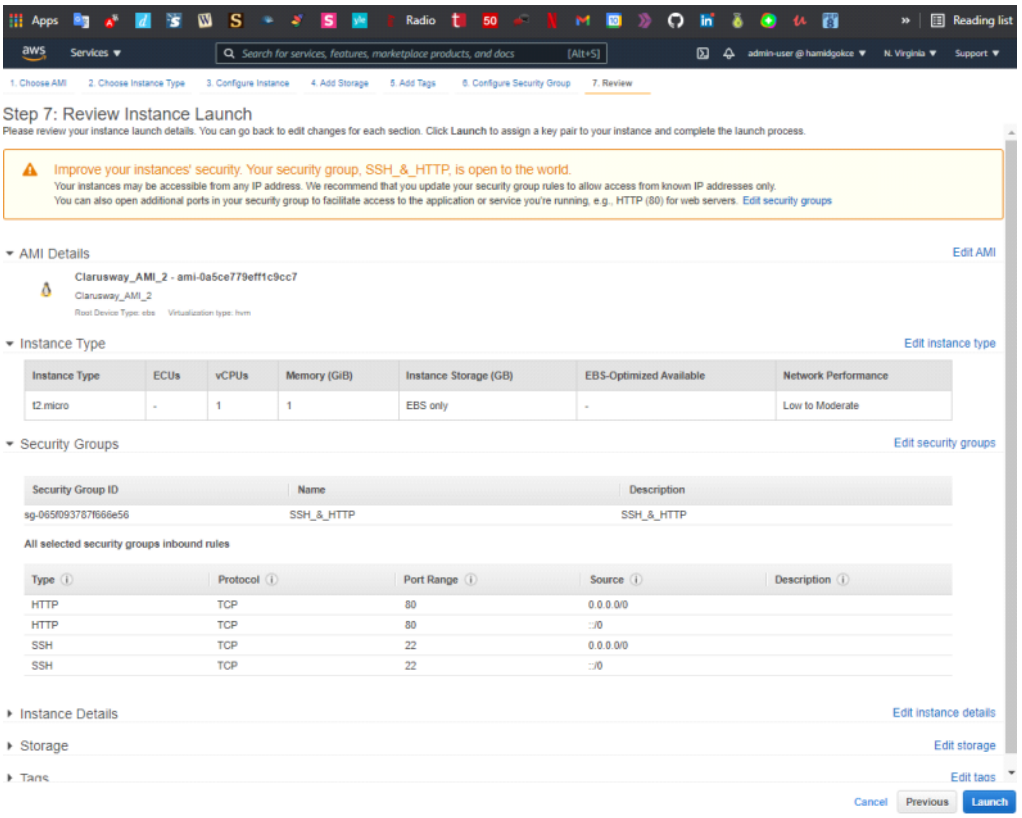




2. Yi sececegiz

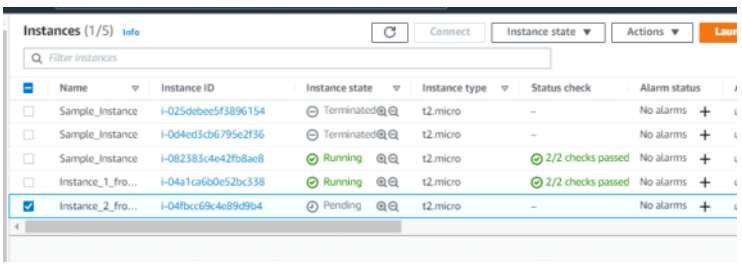


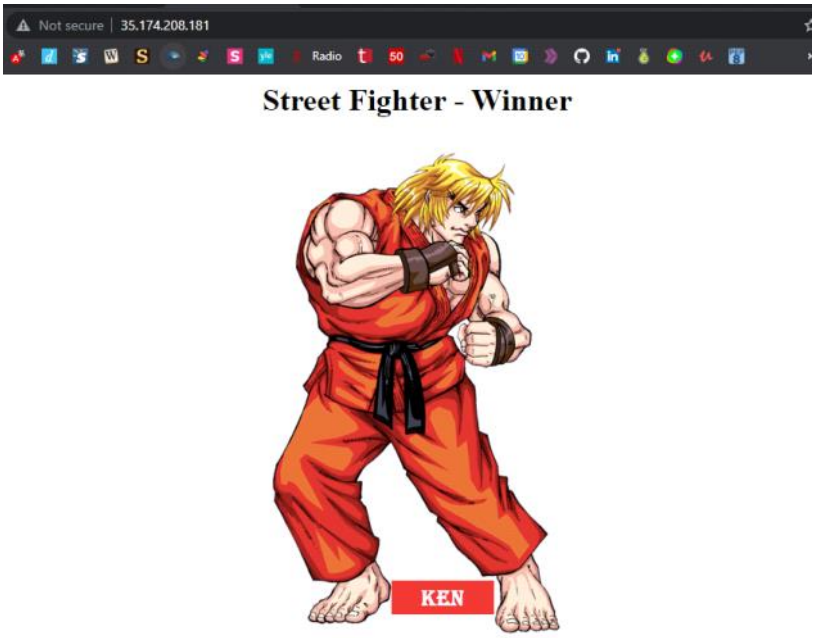


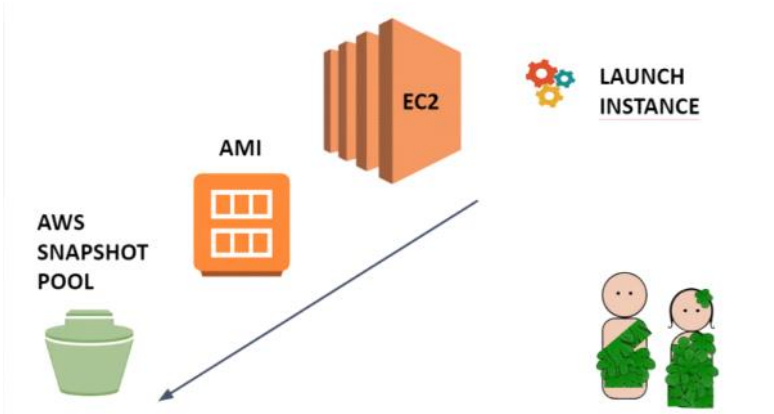



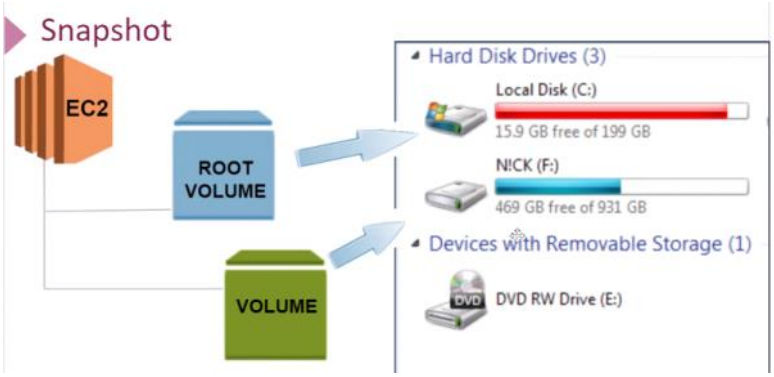

- AMI den olusturdugumuz instancenin acilmasini bekliyoruz




- Ucuncu instanceden de goruntuyu alacagiz DNS yardimiyla

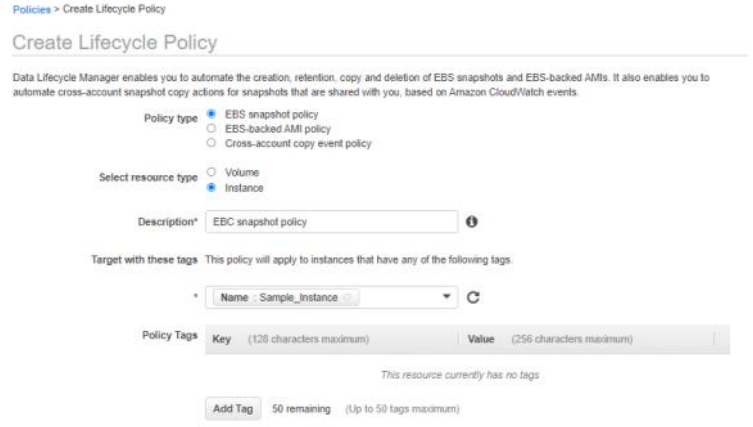


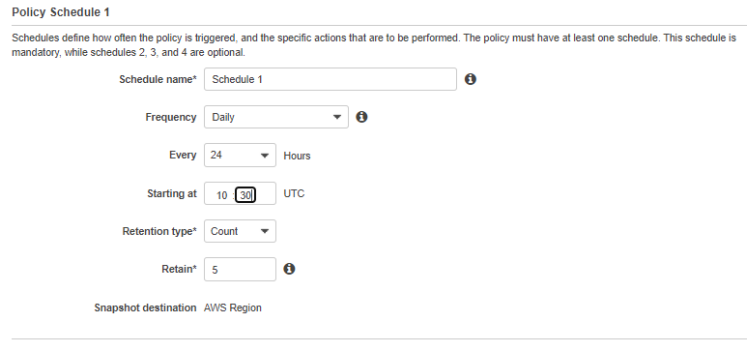

- Aciyoruz life sycle

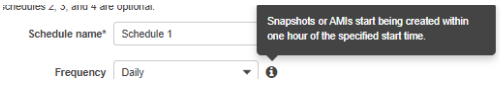

Asagidaki gorseli volume olarak yaptik :(

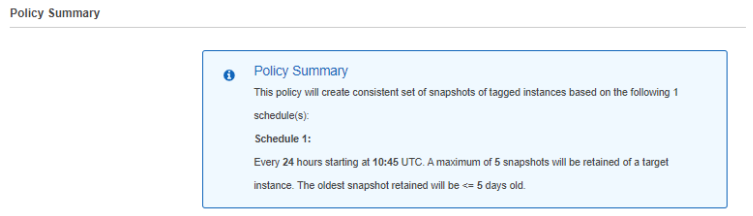



- 3 saat simdiki saatten geri yaz instance olusum saatinden

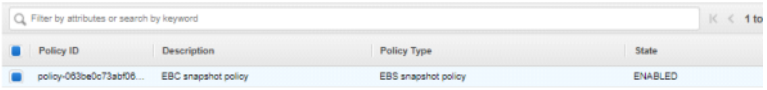
Vermis oldugumuz verilere gore her gun snapshot al diyoruz



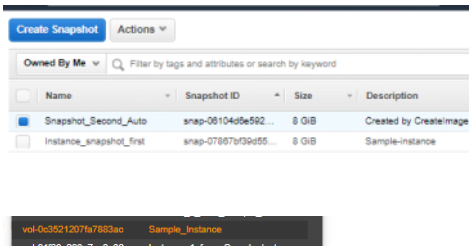



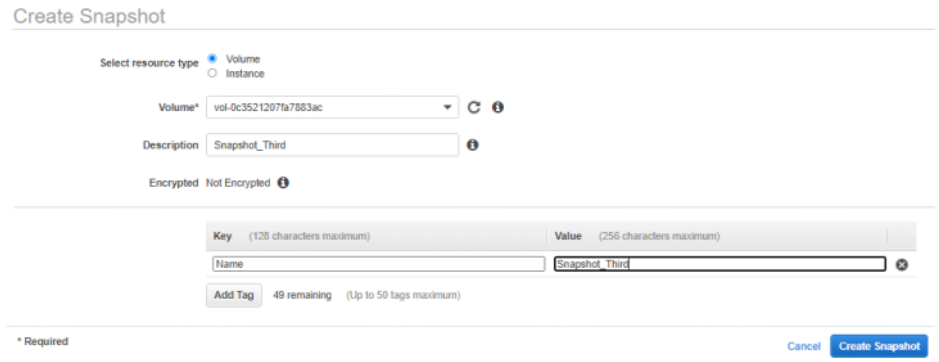



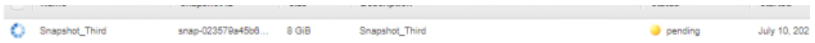




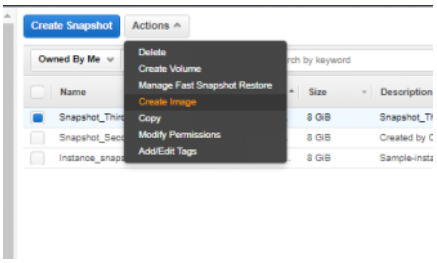

- Sanpshot alani

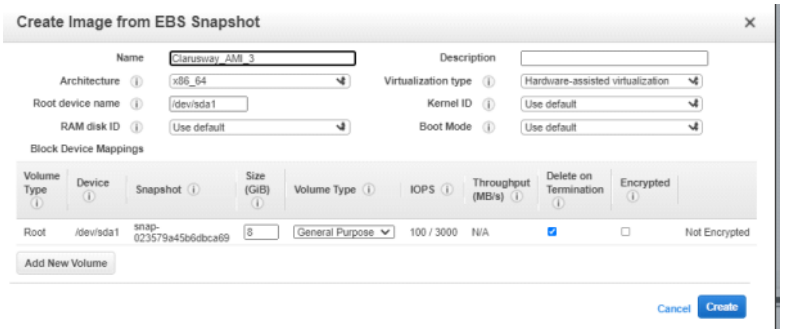


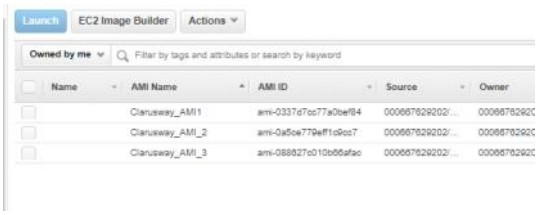




- Snapshot dan AMI yapacagiz

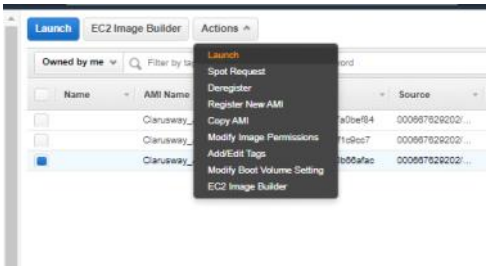


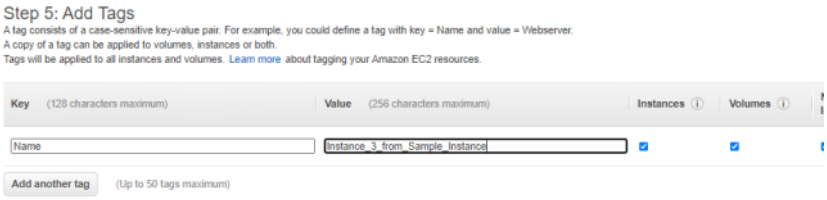

- 3. yu gorecegiz

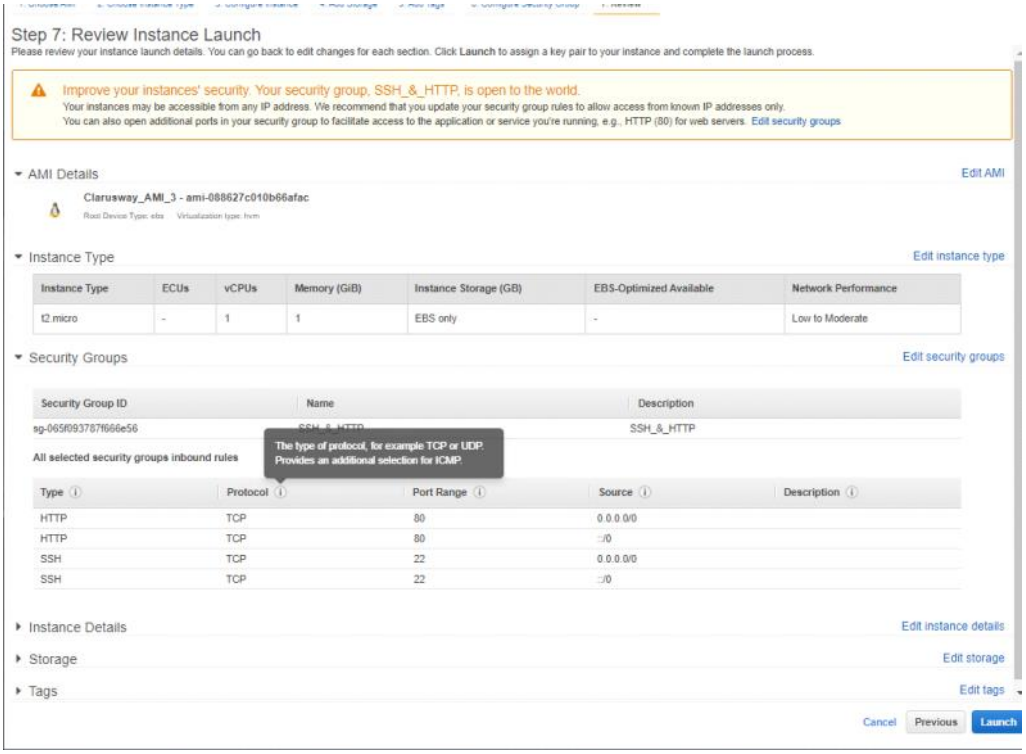


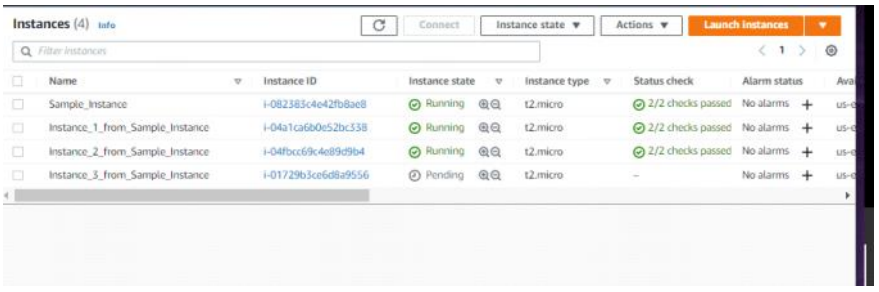

- Yeni instance olusturacagiz AMI_3 den



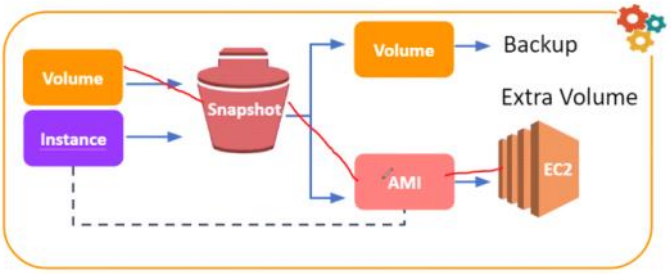

- 22 ve 80 de aciyoruz



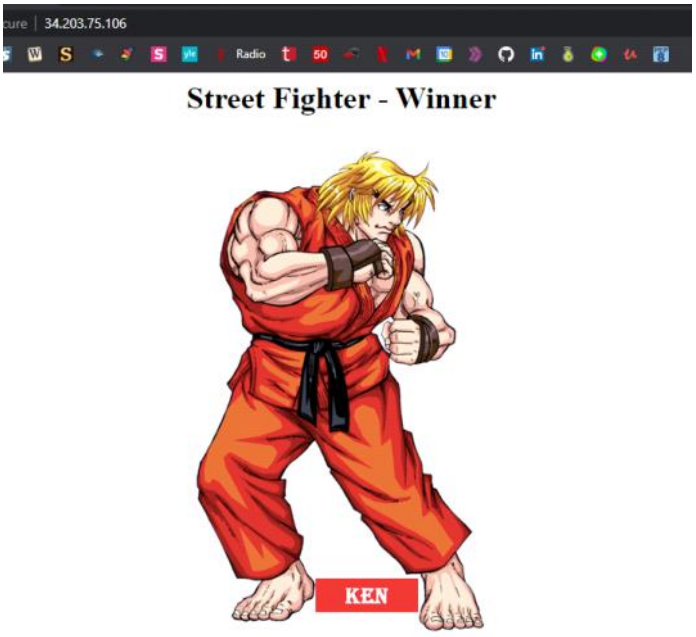




3. Instance root volume den geldik

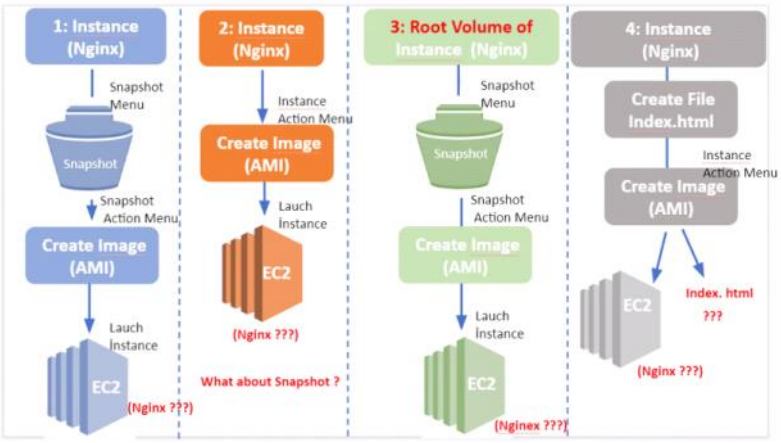



*Lifecycle of Snapshot*

- DNS KEN i gorelim. Bir AMI ile birden fazla instance olusturabiliyoruz




-



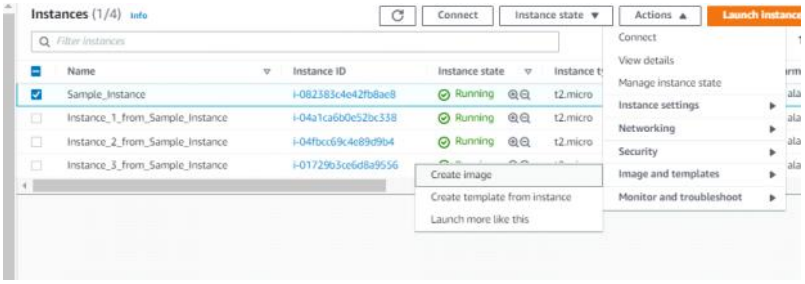

- VS cod ile instance ye baglanalim
- Sample instance den baglanalim

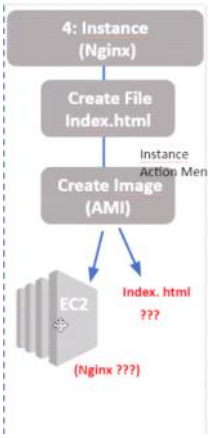
Ssh root user ile baglanmaya calisacak ve basarisiz olacak

- VS codda bir gorseli olusturalim

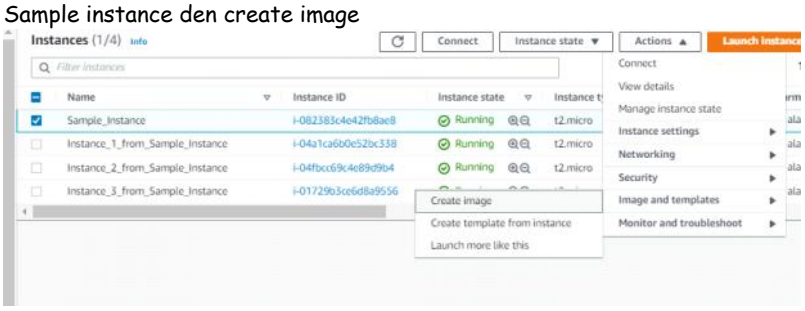
```
[ec2-user@ip-172-31-10-4 ~]$ mkdir Hamid
[ec2-user@ip-172-31-10-4 ~]$ touch gokce.txt
[ec2-user@ip-172-31-10-4 ~]$ nano gokce.txt
[ec2-user@ip-172-31-10-4 ~]$ cat gokce.txt
hello world
[ec2-user@ip-172-31-10-4 ~]$ ls
gokce.txt  Hamid
[ec2-user@ip-172-31-10-4 ~]$
```

Amac ami olusturdugumuzda yukarida kileri de gorecegiz

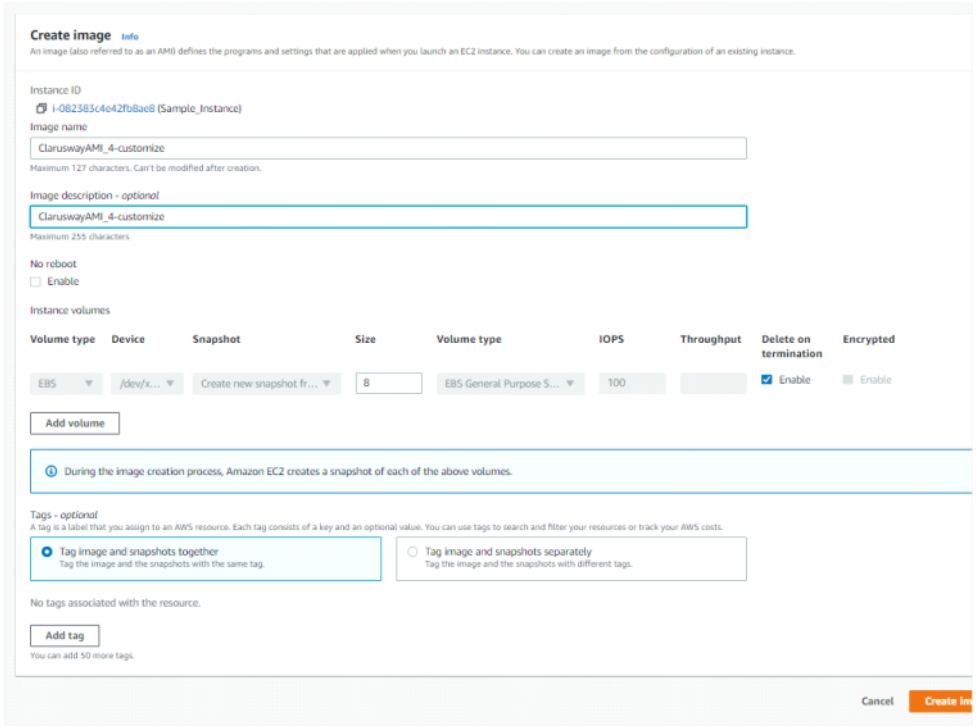



Sample instance den create image

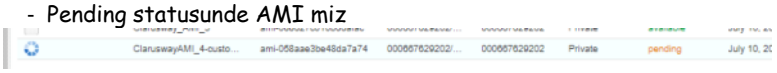



- Name ve aciklama yaziyoruz. Amac instance nin AMI sini alma



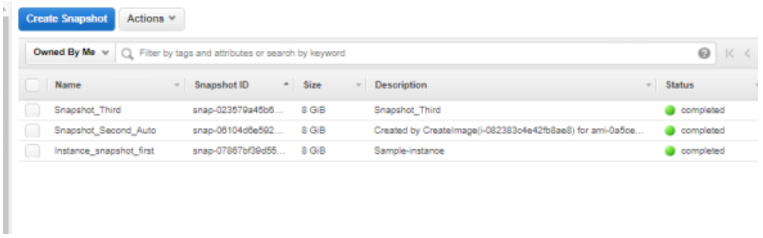

- AMI olusmaya calistigini gorecegiz


- Pending statusunde AMI miz

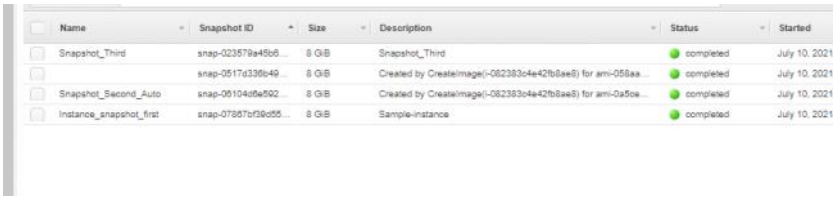



- Ve geldi AMI


- Otomatik snapshot gelecek asagidaki gorsele



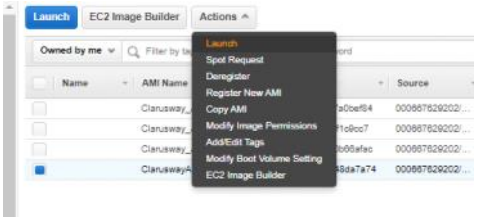

AMI olustuktan sonra snapshot gelecek- Tag siz olan



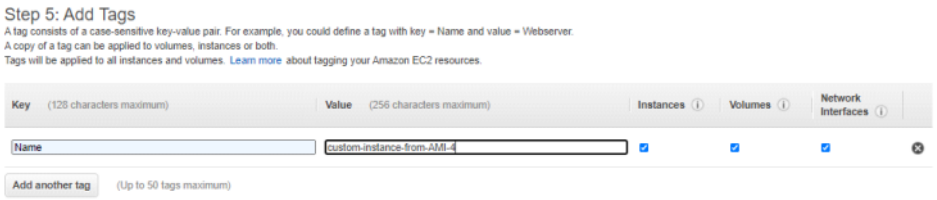

- Baglandigimiz instance den dusecegiz ve tekrar girelim

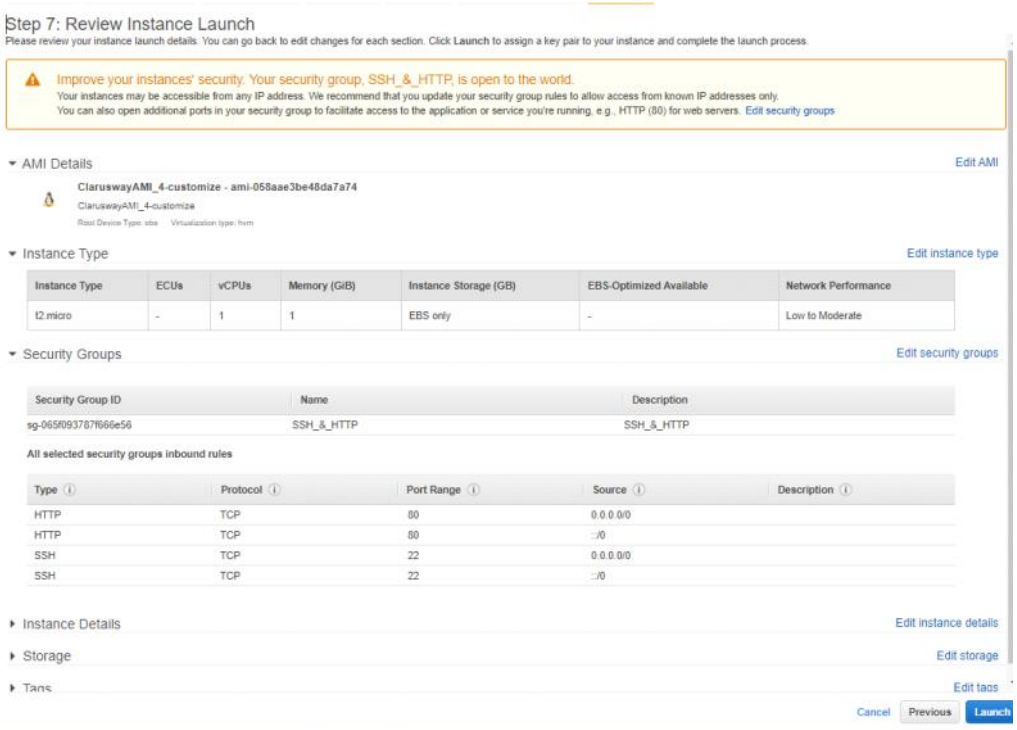
```
[ec2-user@ip-172-31-10-4 ~]$ mkdir Hamid
[ec2-user@ip-172-31-10-4 ~]$ touch gokce.txt
[ec2-user@ip-172-31-10-4 ~]$ nano gokce.txt
[ec2-user@ip-172-31-10-4 ~]$ cat gokce.txt
hello world
[ec2-user@ip-172-31-10-4 ~]$ ls
gokce.txt  Hamid
[ec2-user@ip-172-31-10-4 ~]$ ^C
[ec2-user@ip-172-31-10-4 ~]$ Connection to ec2-44-193-219-157.compute-1.amazonaws.com closed by remote host.
Connection to ec2-44-193-219-157.compute-1.amazonaws.com closed.

hamid@LAPTOP-U8P05O4G MINGW64 ~/.ssh
$ ssh -i "EC2_key.pem" ec2-user@ec2-44-193-219-157.compute-1.amazonaws.com
Last login: Sat Jul 10 10:24:46 2021 from 87-95-119-66.bb.dnainternet.fi

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-10-4 ~]$
```

- Ami den EC2 olusturalim

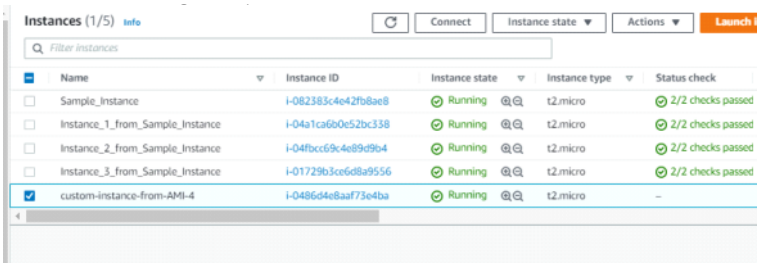



- Tag verelim / 80 ve 22 den acalim




- Ken i tekrar gormeye calisalim son instance den




Olusan instance nin icerisne girelim  SSH ile

Root olarak kaldigi icin giremeyecegiz

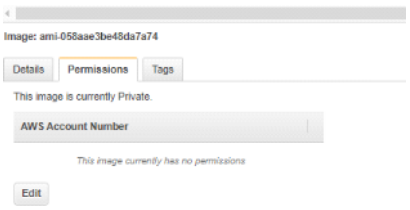
```
hamid@LAPTOP-U8P05O4G MINGW64 ~/.ssh
$ ssh -i "EC2_key.pem" root@ec2-44-193-79-112.compute-1.amazonaws.com
The authenticity of host 'ec2-44-193-79-112.compute-1.amazonaws.com (44.193.79.112)' can't be established.
ED25519 key fingerprint is SHA256:HgpTbbbGJkUZO3jnMGgouXIKUNqteoogaFWTKTJ9EnE.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-193-79-112.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Please login as the user "ec2-user" rather than the user "root".
```

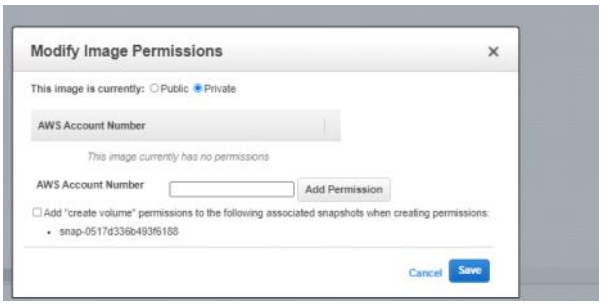
Root kismina ec2-user yaz
Yukarida olusturdugumuz dosyalarida bu instance de de gorecegiz

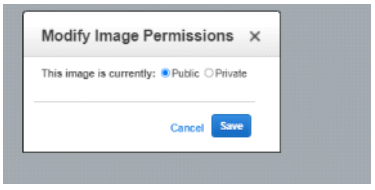
```
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-6-200 ~]$ ls
gokce.txt  Hamid
[ec2-user@ip-172-31-6-200 ~]$
```

- Baskasinin AMI sini gormeye  calisacagiz. Ama AMI private oldugu icin goremiyoruz edit kismindan public yapalim. Ve ami yi baskasina gonderim onun AMI sina giremeye calisalim

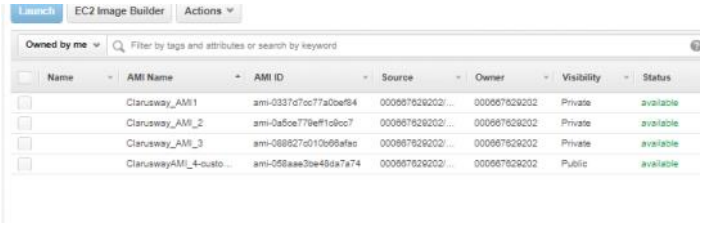
==> permissions / edit




- Ami public olarak gorunecektir




- Marcus hocanin gonderdi ami den asagidaki clarusway in AMI sine ulasabiliyoruz