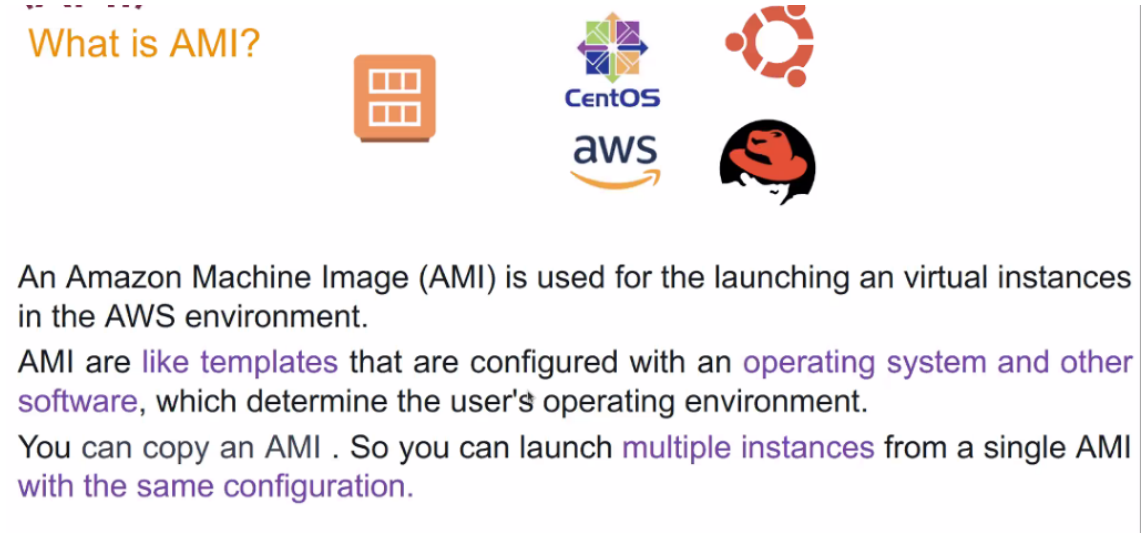
AMI nedir?



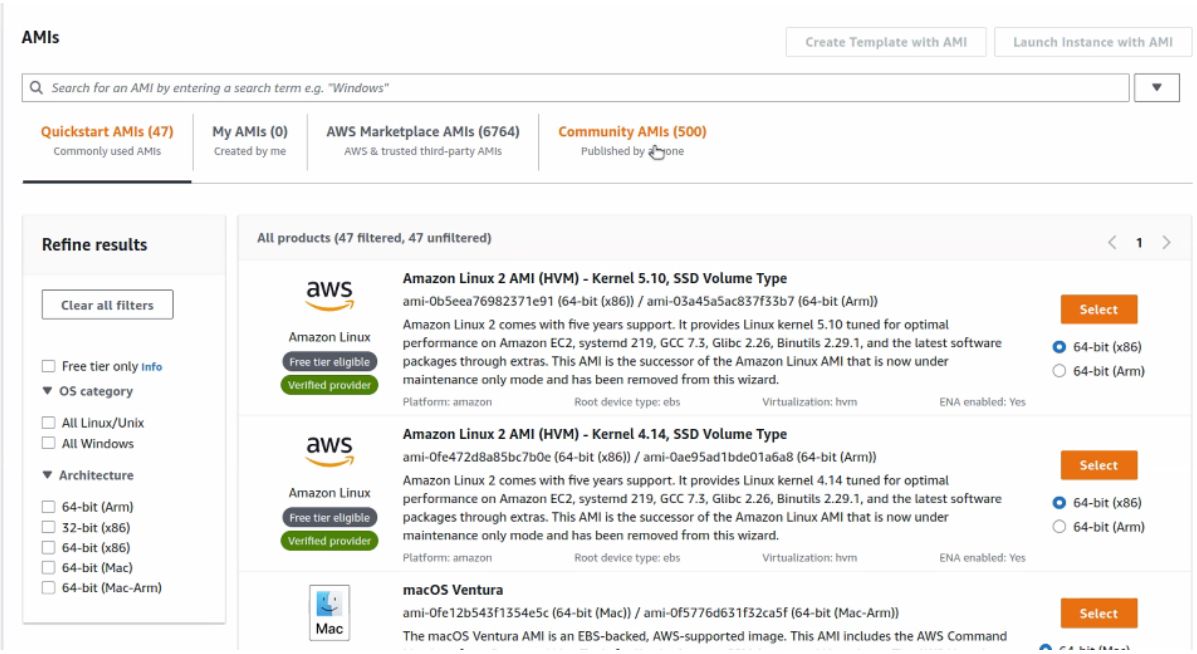
EC2 Launch template tüm makineyi içeren bir EC2 template idi. Ancak AMI sadece EC2 içindeki yazılımsal bir template. İçerisine software gömülmüş bir şablon.

AMI yi sadece işletim sistemini içinde barındırıyor olarak düşünmeyelim. Eğer database ihtiyacımız varsa onu da barındırıyor. AWS bunu kullanıma sunuyor.

Oluşturduğumuz bir instance ın image ini alıp AWS hesabımızda muhafaza ederiz ki daha sonra instance ta bir sorun olduğunda vs. bu image ile bir instance ayağa kaldırmak için kullanırız.

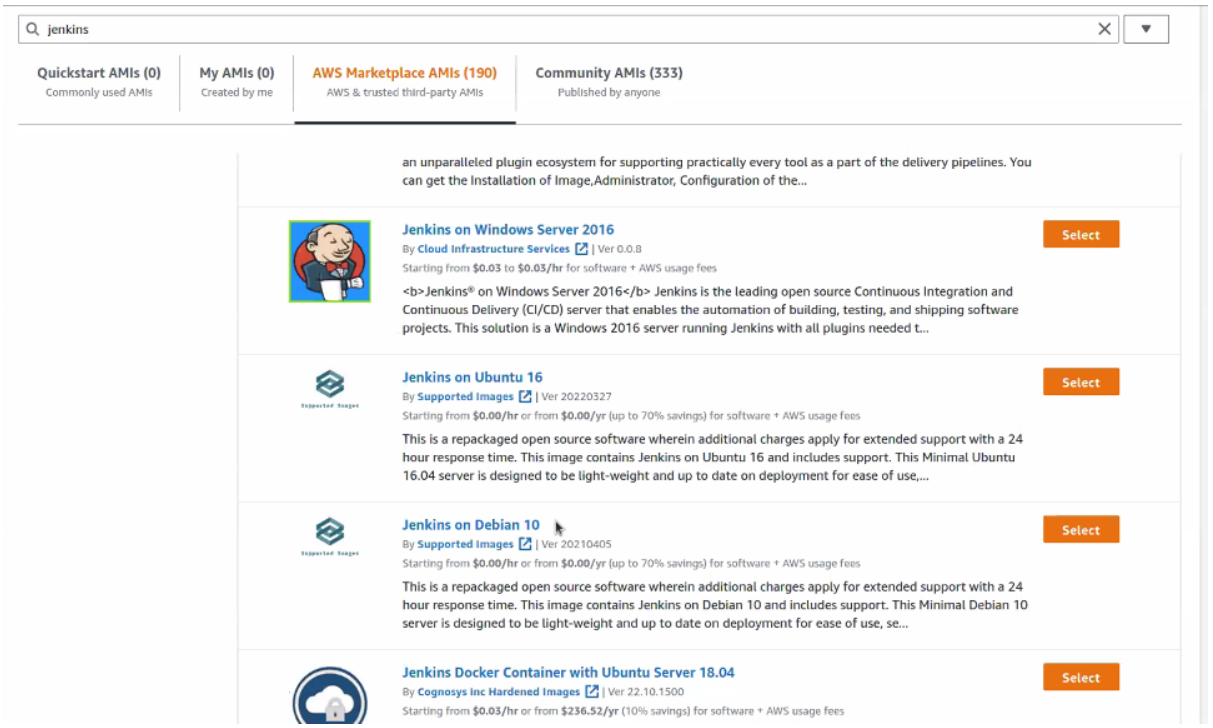
Biz EC2 ayağa kaldırırken yazılımsal olarak işletim sistemi seçip kaldırıyorduk ancak AMI ile yazılımsal başka toollarla birlikte bir instance ayağa kaldırabiliriyoruz. (yazılımlar, aplikasyonlar, depenciesler)

AMI EC2 nun software yedeği denilebilir.

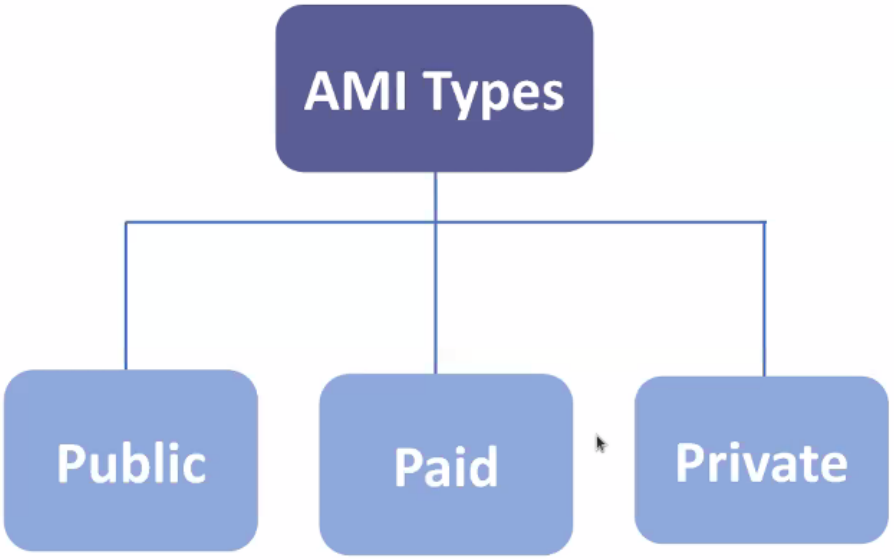


Hızlı başlangıç AMI leri (Marketplace ve Community den kolaylık için çekilmiş AMI ler),
Kendi oluşturduğumuz AMI ler,
AWS Marketplace AMI leri,
Topluluk AMIleri (herkesin kullanımına sunulmuş AMI ler)

Örneğin Jenkis yazılımı ile bir instance ayağa kaldırmak istiyoruz:



Burada ubuntu için, windows için vs. Jenkins yüklü AMI lerden seçip yüklü bir şekilde instance ayağa kaldırabiliriz.



Public olması herkesin ulaşılabildiği anlamına gelir. Bunlar ücretsiz demek değildir, Paid de olabilir.

Private ise bizim sadece kendimiz için oluşturduğumuz AMI lerdir.

Lauch Template=EC2=donanım + yazılım + diğer bileşenler
AMI= yazılımlar(işletim sistemi+diğer yazılımlar)

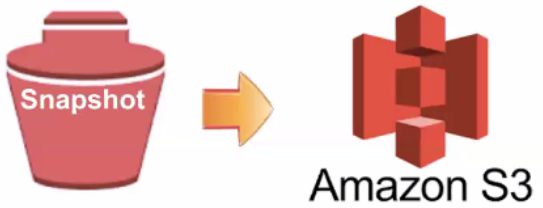
SNAPSHOT:

Bir EBS volume un ya da instance ların point-in-time kopyasıdır. Bizim volume, AMI ya da isntance larımızın çekirdeğini, hamurunu temsil ediyor.

- 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

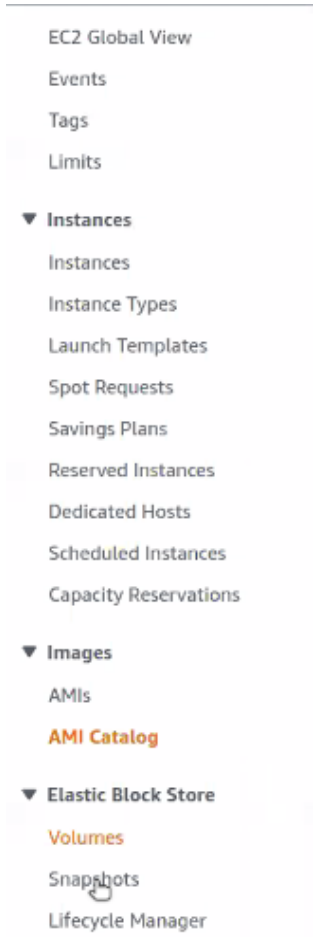
Snapshot ları manuel olarak ya da otomasyona tabi tutarak alabiliriz. Sayı ve zaman olarak otomasyon oluşturabiliriz
Snapshot amaçları,
Volume/Instance ın kopyasını oluşturmak. Recovery ihtiyacımız olduğunda kullanmak adına Backup olması için.
Aynı özelliklerde birçok instance oluşturmak üzere AMI ler oluşturmak.
Yeni bir volume oluşturmak

Features of the Snapshot

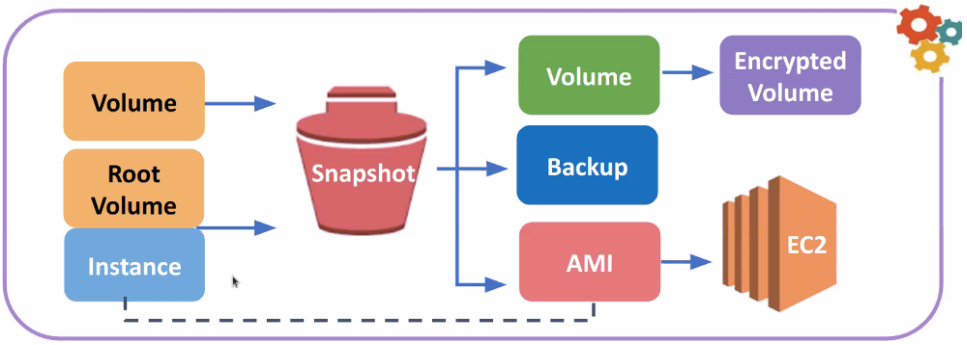


- Source from Volume or Instance
- Stored in Amazon S3
- Incremental storage
- Data Lifecycle Manager (DLM)

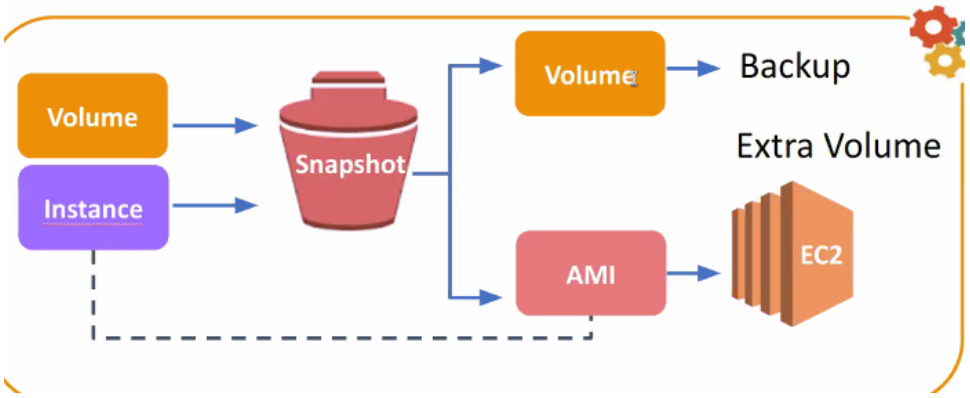
Snapshot un kaynağı Volume ve Instance tır ve S3 servisinde depolanır. Ancak biz bunları EC2 menüsünde görebiliyoruz:



Snapshot'un yaşam döngüsü:

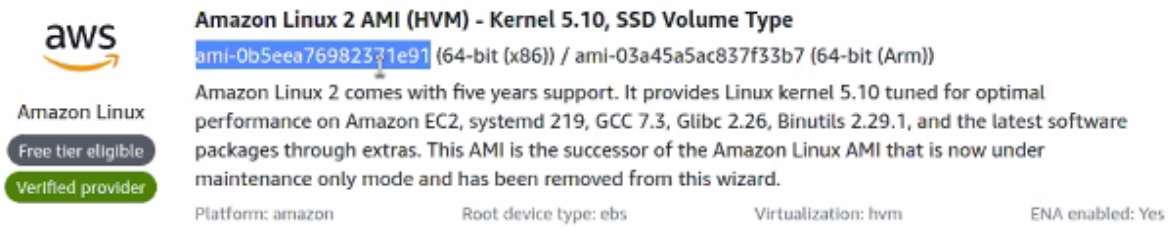


Root volume (PC deki C)
AMI nin ve Volume'un kaynağı snaphot lardır. Biz instance oluştururken seçtiğimiz AMI ler i de AWS kendi tuttuğu snapshotlardan oluşturuyor.
Volumei Root volume ve instance lardan snapshat oluşturuluyor, sonra bu snapshotlardan volume, backup ve AMI ler oluşturuluyor. AMI lerden de EC2 ayağa kaldırılıyor.

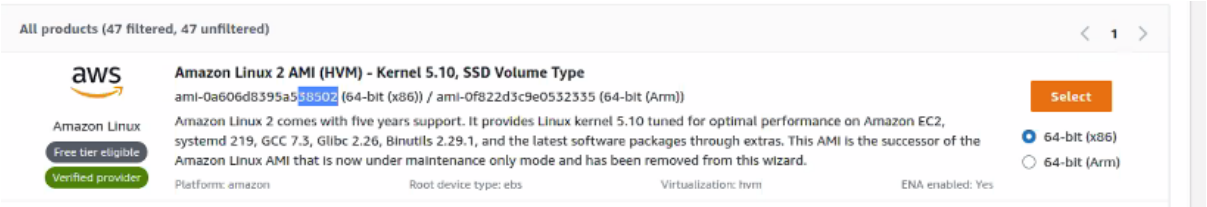


AMI lerin id si vardır. Bu AMI ler regionaldır:

N.Virgina:

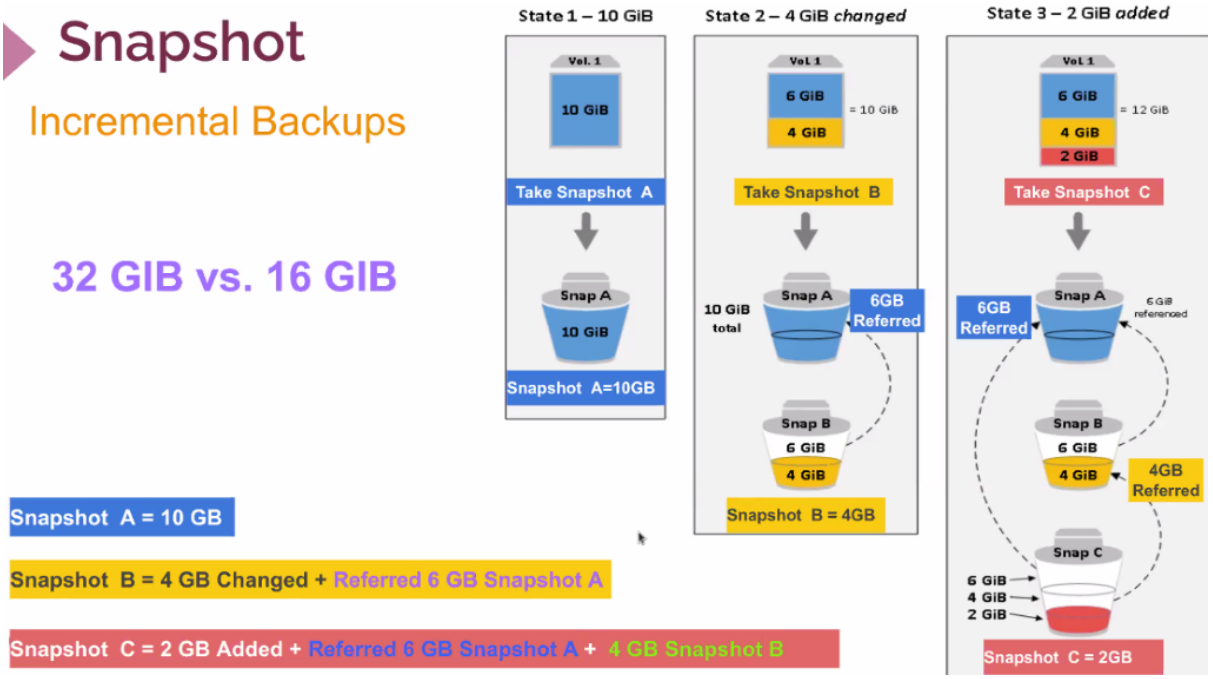


Ohio:

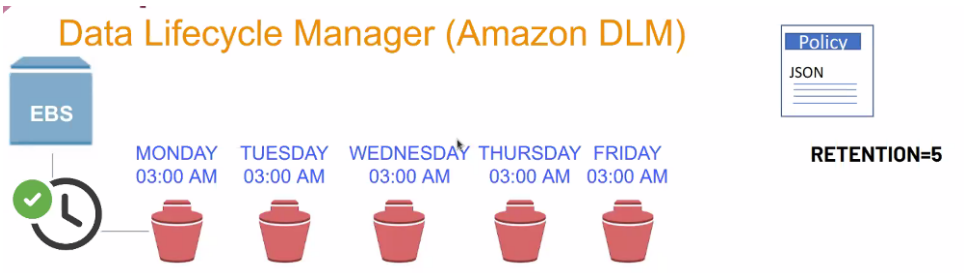


Örneğin bir Linux e güncelleme geldi, bu güncelleme üzerine bu AMI ler de yeni bir id ile oluşturulur. Burası önemli. Güncelleme öncesinde oluşturduğumuz bir template i kullanamama durumu oluşur.

Incremental Backups:



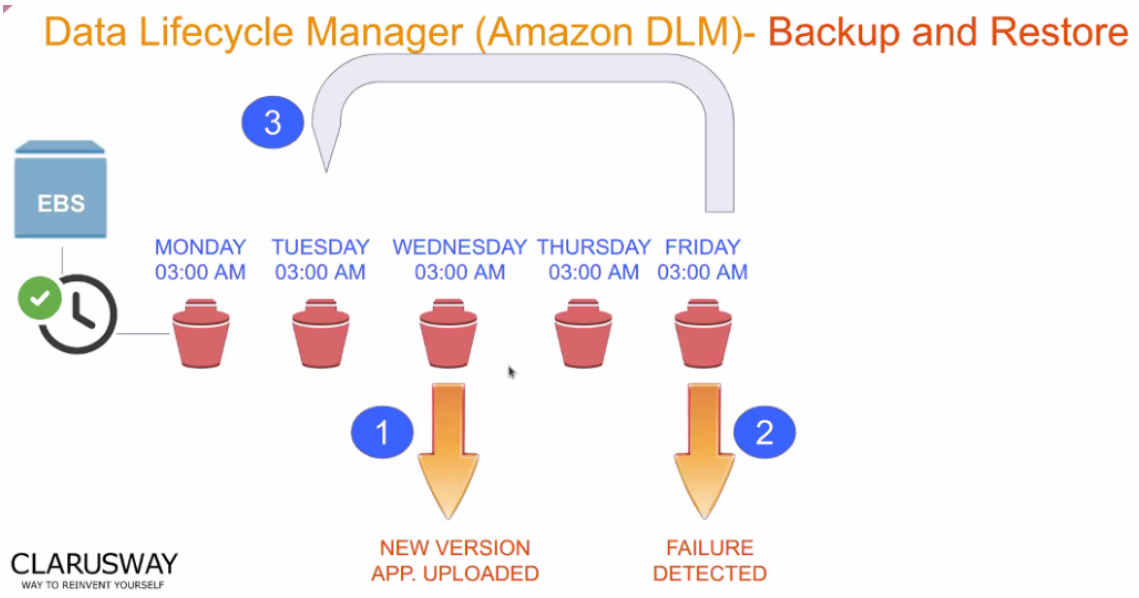
Elimizdeki bir volume un snapshot unu almak istiyoruz. İlk gün bunu aldığımızda boyutu 10 GB olur. İkinci gün de bu 10 GB ın 4 Gb lık kısmında değişiklik var. Şirket politikamız gereği ikinci gün de snapshot almamız gerekiyor.
AWS ne yapıyor? Değişmeyen 6 GB lık kısmı referans göstererek, sadece değişen 4 GB lık kısım için snapshot alıyor. Tabi biz tamamını tek snapshot olarak görüyoruz.
Üçüncü gün ise 2 GB bir değişiklik oldu. Bu kez de ilk gün için 10 GB, ikinci gün için değişen 4 GB, üçüncü gün için ise değişen 2 gb ın snapshot ını almış oluyor.



Her gün şu saatte snapshot ını al, şu güne kadar şu kadarını tut gibi otomatik hale getirebiliriz:

Günlük bir kere 5 günlük bir retention belirledik, 6.gün 1.gün ün snapshot ını silecek, bu şekilde güncel snapshot ları tutmaya devam edecek.

En fazla 35 gün gibi bir retentiona seçeneği veriyor. İlerleyen süreçte bunu manuel olarak ya da yazığımız bir script ile bunu kendimiz manuel olarak otomatize edebiliriz.



Bir yazılım geliştirme ekibi bir aplikasyon üzerinde çalışıyor. 1 numralı kısımda yeni bir version yayınladık, sonra 2 numarada bu güncel versionda hatalar meydana geldiğini gördük. Snapshot sayesinde 3 numaralı gündeki snapshot ile o versiona geri döneriz ve hataları gidermeye çalışırız.

Sınavda bu şekilde bir senaryo ile karşımıza soru çıkıyor:

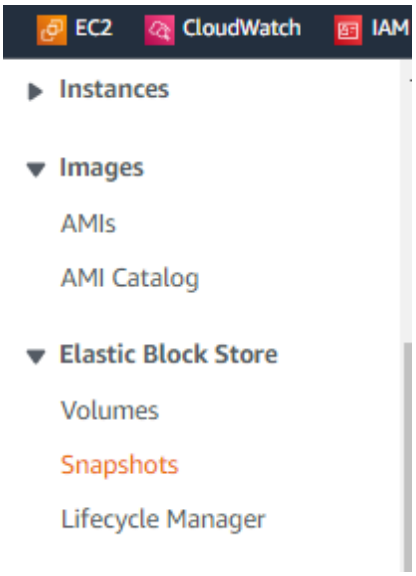
Snapshot

Encryption of Root Device via Snapshot

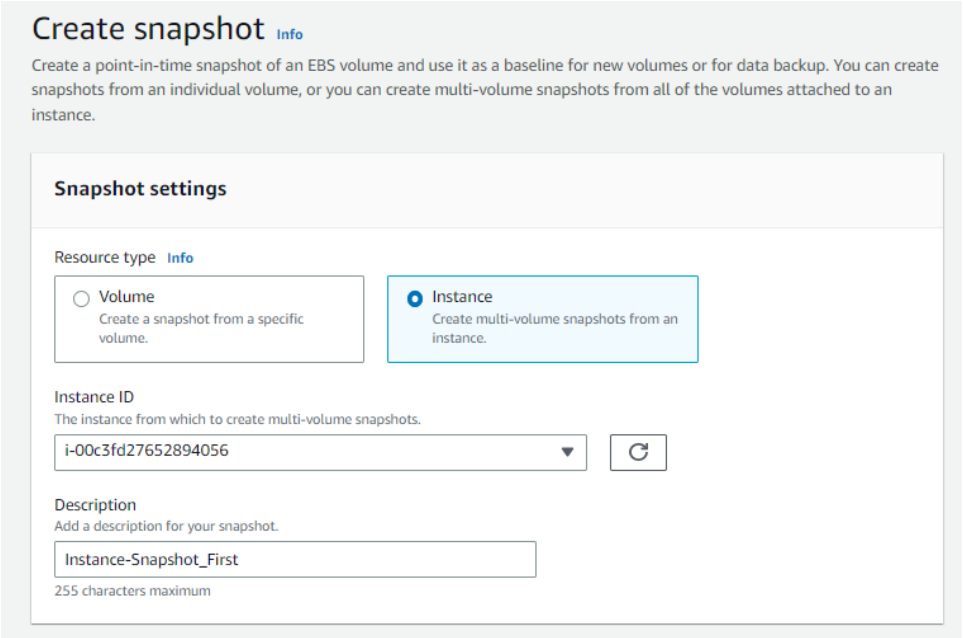
- Root device (volume) cannot be encrypted after creation. "How to encrypt unencrypted volume after after creation" is a common question that can be asked in certification exams!
  - Take snapshot of unencrypted volume.
  - Copying the unencrypted Snapshot,
  - You are able to encrypt this Snapshot while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

Hands-on a geçiyoruz:

User dataya yazılı kısmı yapıştırarak EC2 ayağa kaldırdık.
SNAPSHOT sayfasına gidiyoruz:



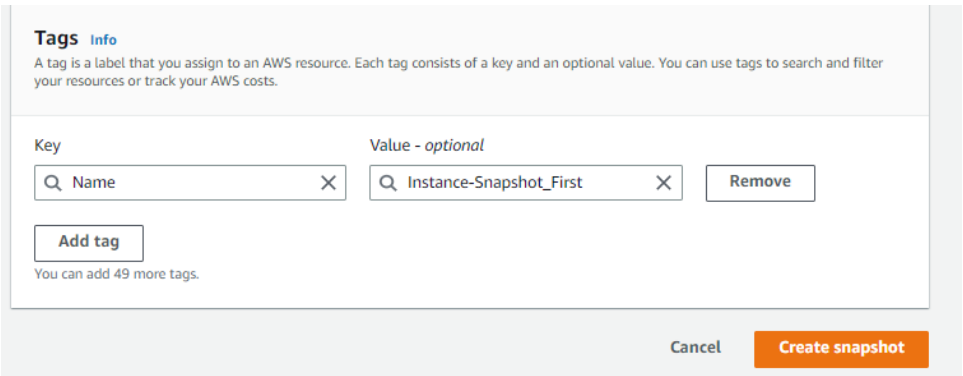
İlk seçenek olarak instance dan snapshot alacaz. Create Snapshot diyoruz:



Isntance ı seçiyoruz ve kaldırdığımız instance ın id sini seçiyoruz. Description yazıyoruz.

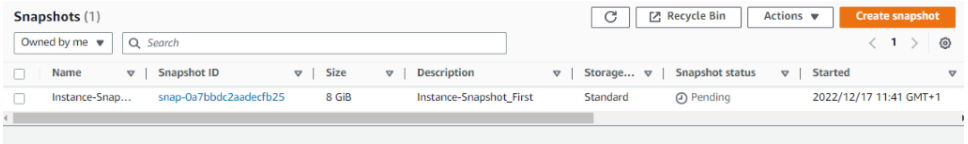
Exclude kısımları boş kalıyor.

Snapshotlar otomatize verilirken Datalife cycle manager tag lere bakarak bunu ypaıyor. Tag ı seçmemiz önemli:

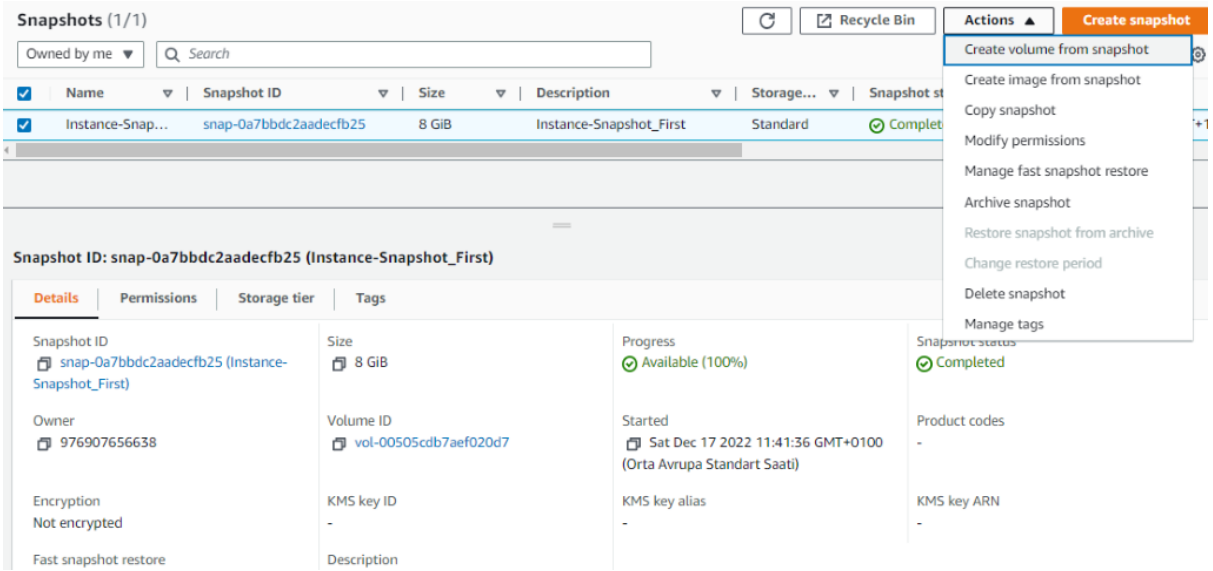


Snapshot lar ücretli ancak, bu bir depolama seçeneği var ve bizim free tier ımız var. Ancak biz instance ı stop etsek bile snapshot bizim free tier ımızdan kullanmaya devam eder.

Bir süre bekledikten sonra snapshot ımız hazır:

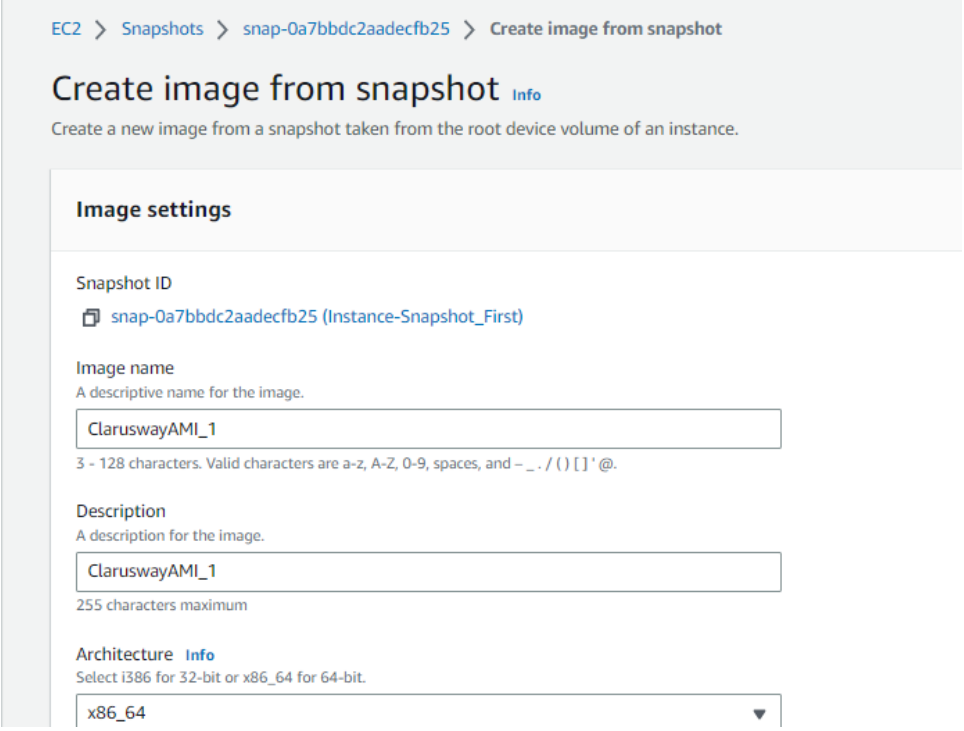


Snapshot ı seçip Actions a geldiğimde snapshot tan volume veya AMI oluşturabileceğimi görebiliyorum:



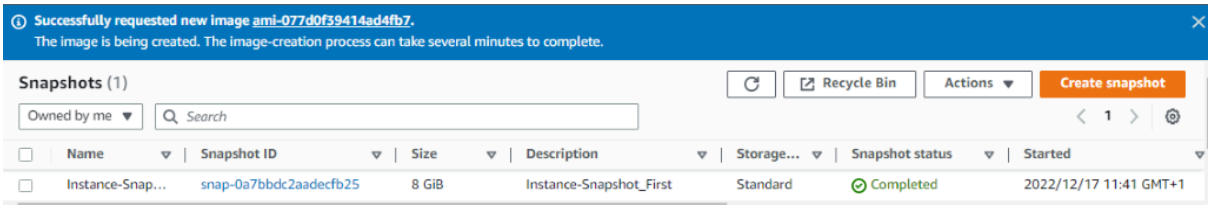
Create image from snapshot diyerek AMI oluşturulaım:

Üstte snapshot ımızın id si görünüyor. Name ve description veriyoruz:

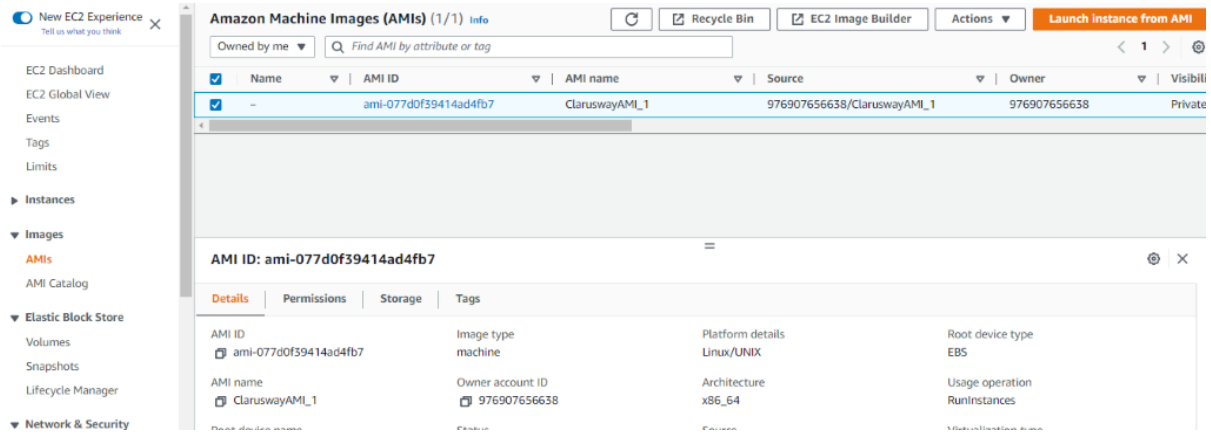


Diğer detaylara şu an için girmiyoruz. Create Image diyoruz.

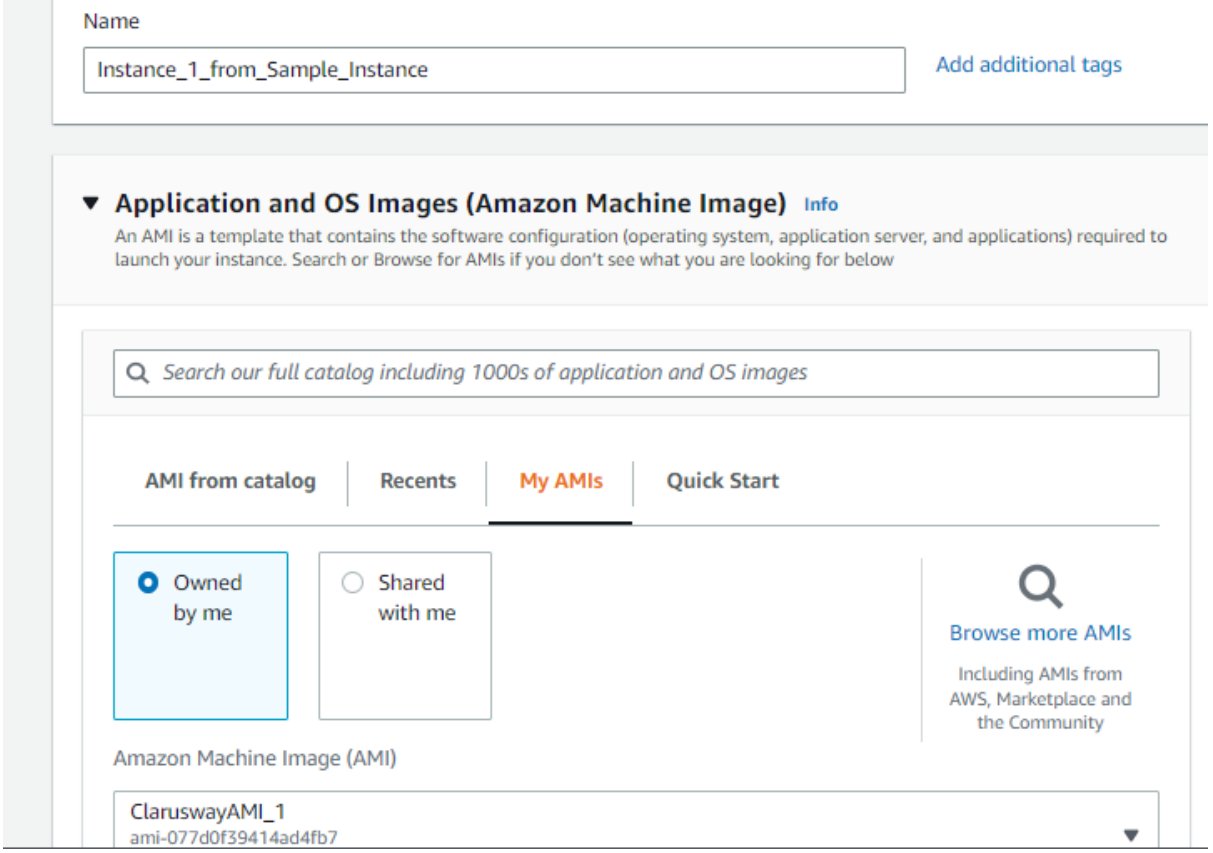
Bi id ile AMI miz oluşturuldu:



Sol taraftan AMIs e tıklayarak AMI menüsüne gidiyoruz ve AMI mizi görüyoruz:



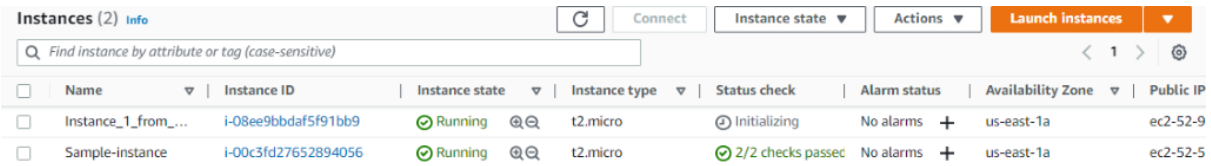
Launch instance diyerek bu AMI ile bir instance ayağa kaldırıyoruz:



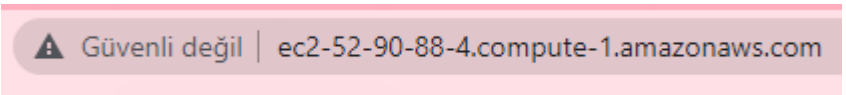
İsim veriyoruz, MY AMIs kısmında Owned by me seçeneğini tıklayarak oluşturduğumuz AMI yi seçiyoruz.

Burada aynı user data yı tekrar yzmamıza gerk yok çünkü biz daha önceki user datayı image içine gömdük.

LAunch instance diyerek instance ımızı ayağa kaldırıyoruz.

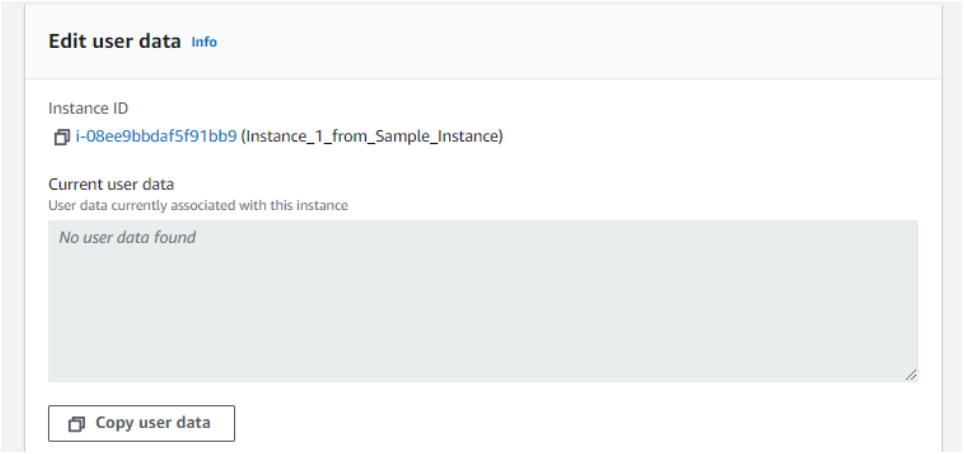


Web aplikasyonumuz çalıştı:

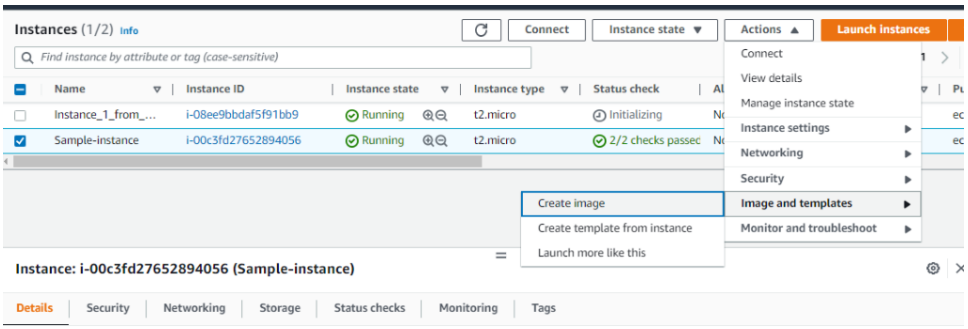


Street Fighter - Winner

İkinci instance ımızın user datasını kontrol ettiğimizde boş olduğunu görüyoruz. Aplikasyonun çalışma nedeni önceki instance ın image ını alıp bu instance da kullanmış olmamız.

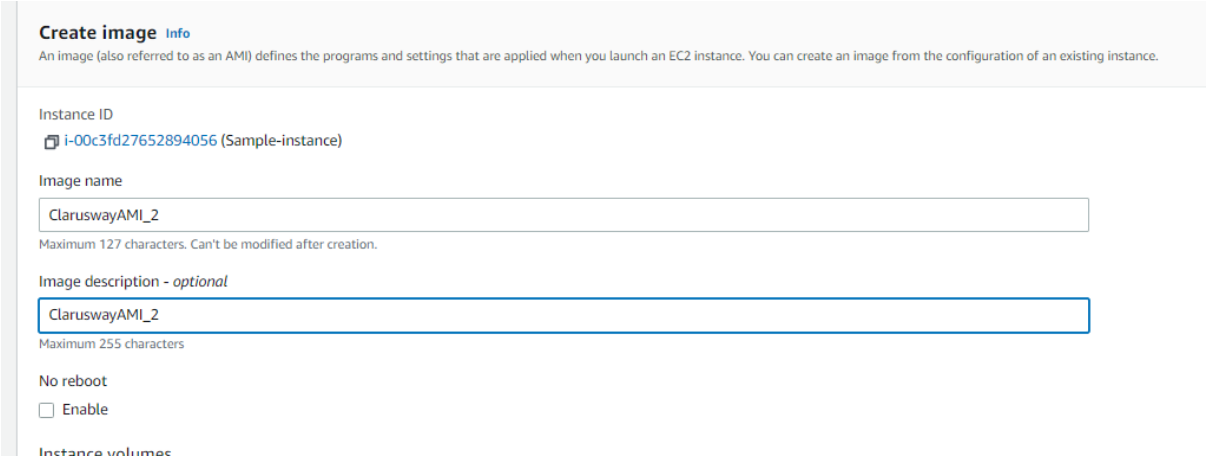


Şimdi instance dan direk AMI oluşturalım:

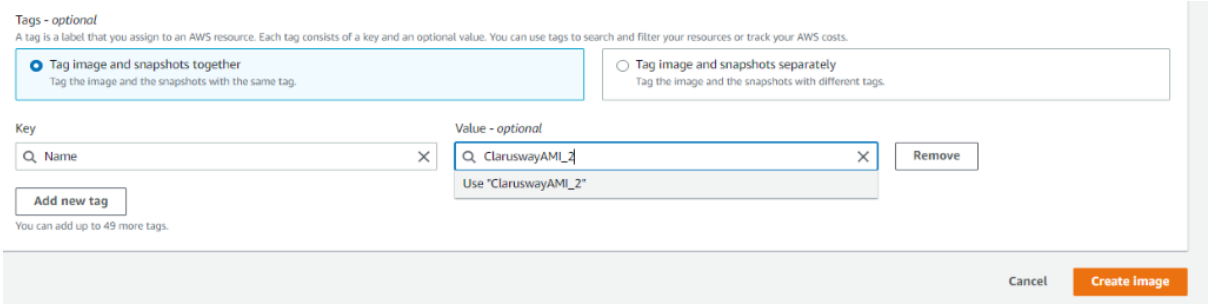


Instance seçiliyken Actions/Images and Templates/Create image

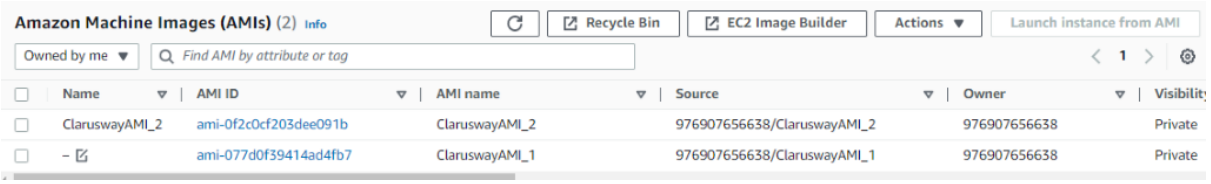
Name ve description ekliyoruz:



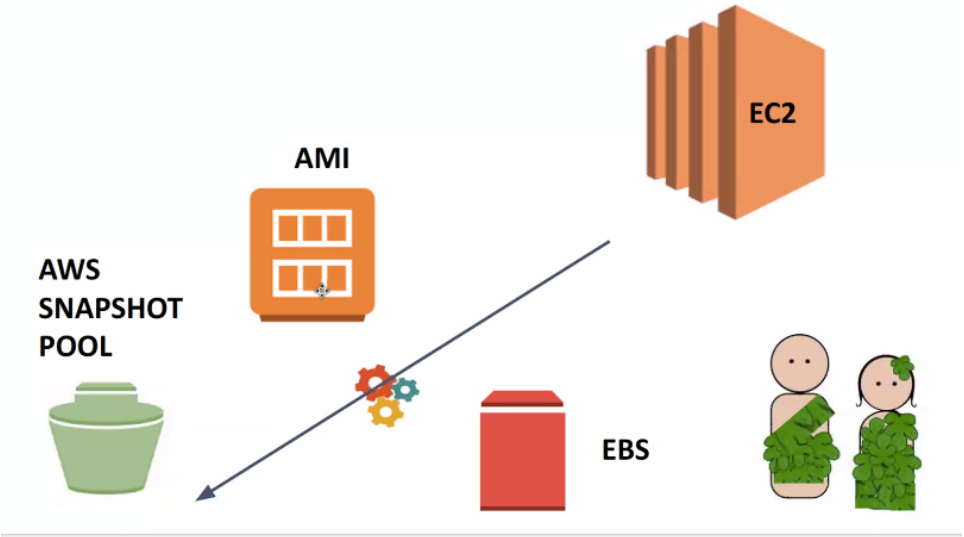
Tag ekleyelim ve create image diyelim:



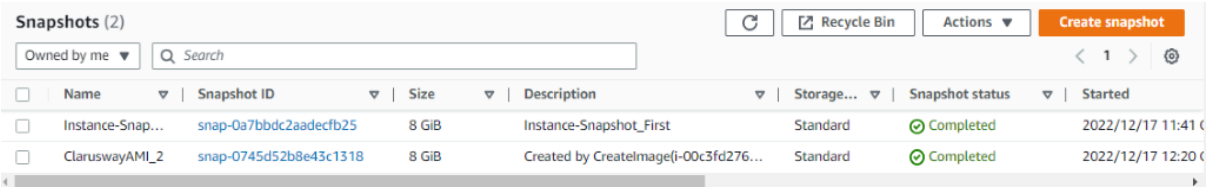
AMIs menüsüne gidiyorum ve ikinci AMI nin oluşturulduğunu görüyorum:



Biz EC2 dan image oluşturduk ama herşeyin ana kaynağı snapshot demiştik:

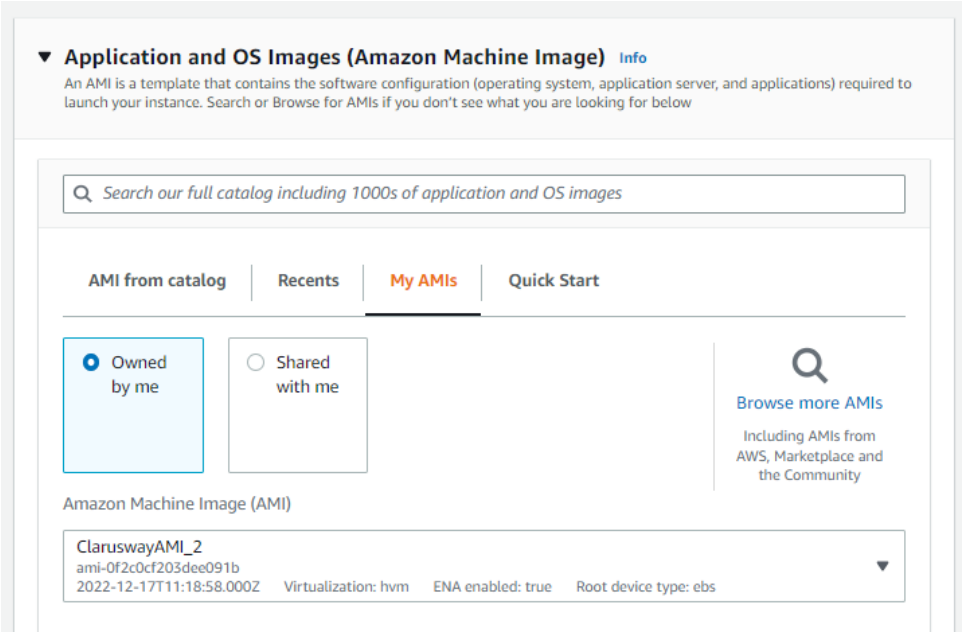


AWS acaba bunu nasıl yaptı?:
Snapshot menüsüne gidiyoruz ve ikinci snapshot un AWS tarafından arka tarafta oluşturulduğunu görüyoruz:



Descriptionunda da "Created by CreateImage(i-00c3fd2765" yazıyor.
Bu snapshot ları unutmayacağız, bize ücret olarak geri döner.

Biz bu AMI ile de instance ayağa kaldıralım. Burada da aplikasyonumuzu göreceğiz.
AMI menüsüne gidip ikinci AMI yi seçip Create Instance from AMI diyoruz. AMI olarak 2. AMI yi seçiyoruz:

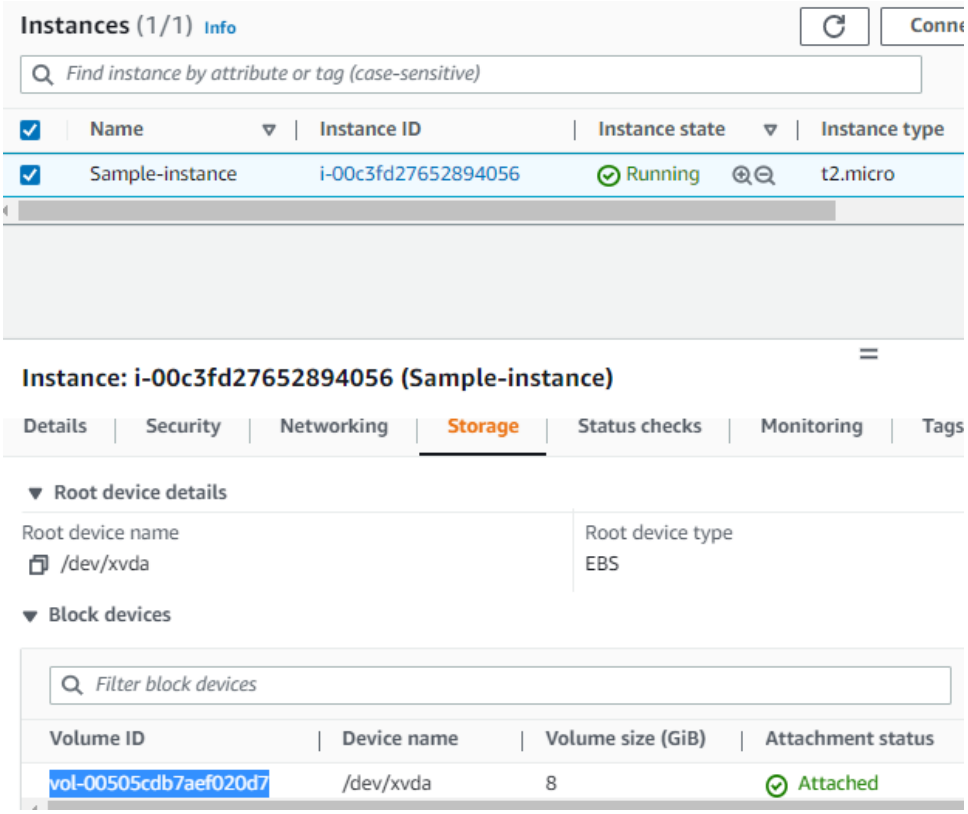


Instance ın public DNS ini browsere kopyaladığımızda aplikasyonumuz çalışacaktır.

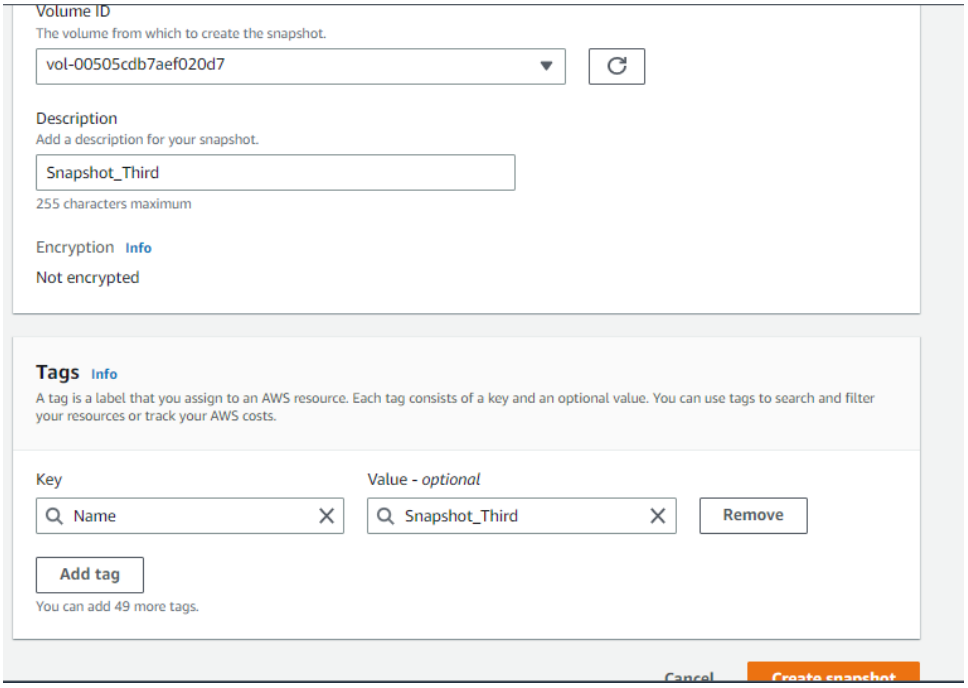
AMI sayımız kadar snapshot oluşacak bu ikinci kısımda bunu görmüş olduk.

Şimdi üçüncü kısıma geçelim:
Bu kez volume dan snapshot alacaz. Ne farkı var? Instance ın sahip olduğu volume lar kadar snapshot labiliyorken volume dan snapshot alacakken tek bir volume dan snapshot alabiliriz. Ayrıca volume dan AMI oluşturmak için root volume olması gerekir. Biz sample instance ımızın volume unu root volume olarak alacağımız için AMI oluşturabileceğiz.

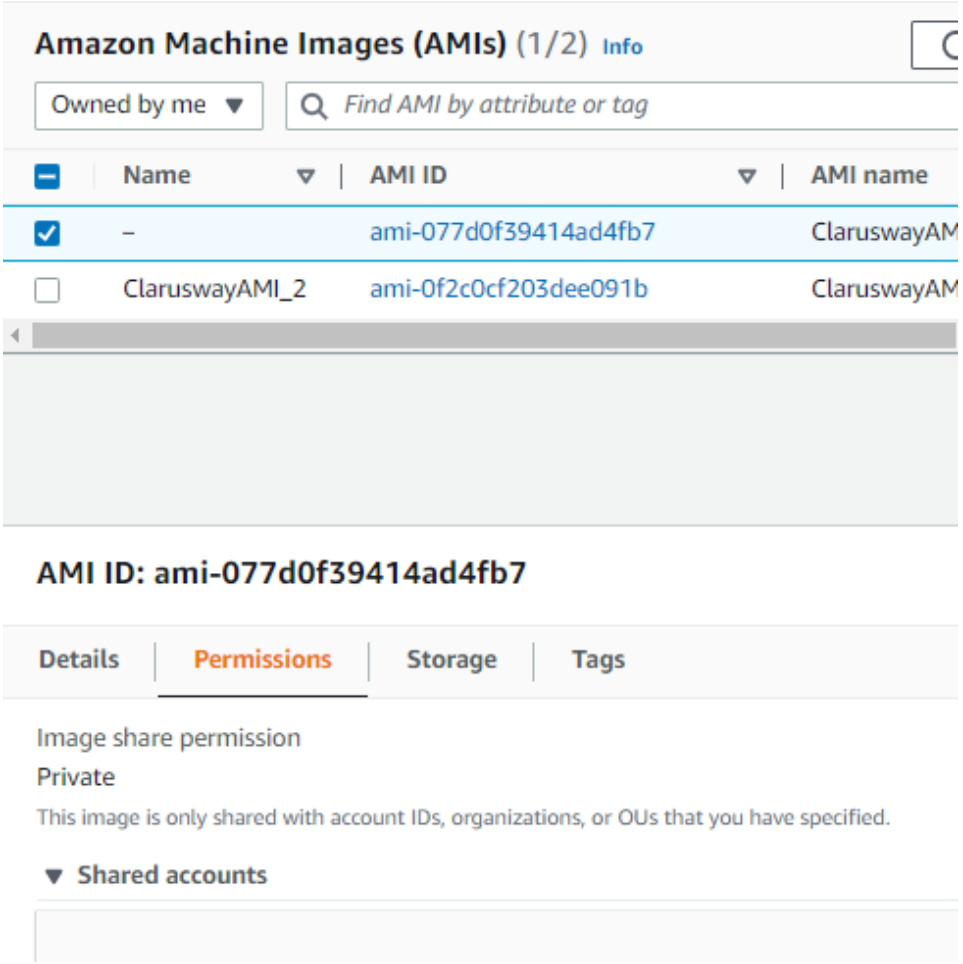
Snapshot menüsünden Create Snapshot diyoruz ve volume u seçiyoruz:
Volume id için instance a gidip instance ı seçip aşağıdaki sotrage kısmından volume id kontrol edip snapshot oluştururken onu seçiyoruz:



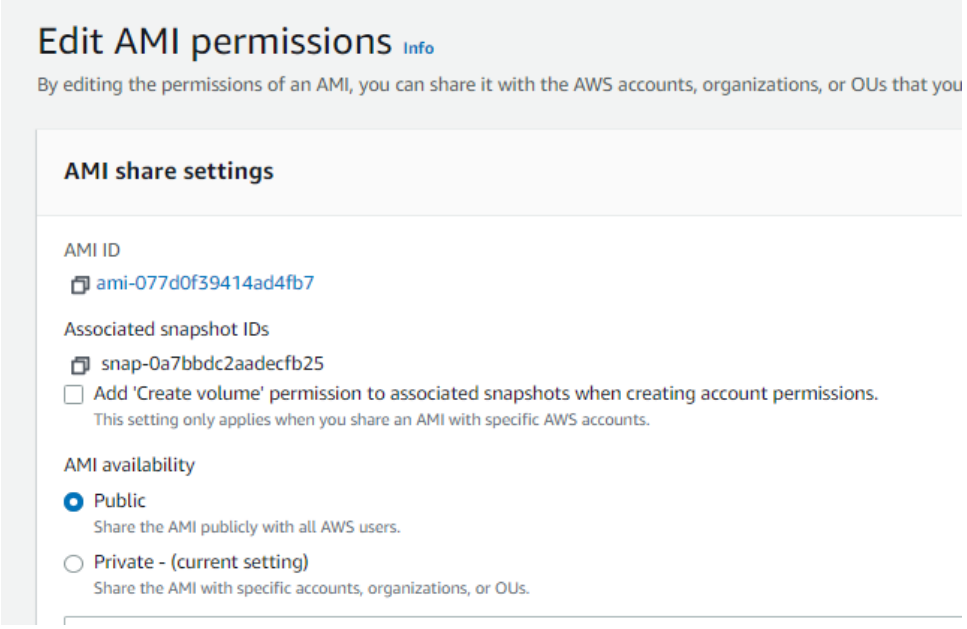
Description ve tag ekliyoruz ve create snapshot diyoruz:



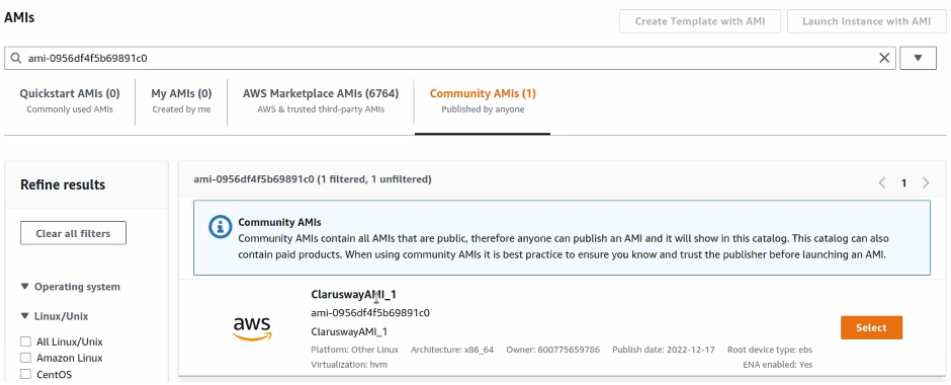
Bu oluşturulurken ilk AMI mizi public hale getirelim:
AMIs menüsüne gidiyoruz.
AMI yi seçip aşağıdaki permission ayarına geliyoruz.



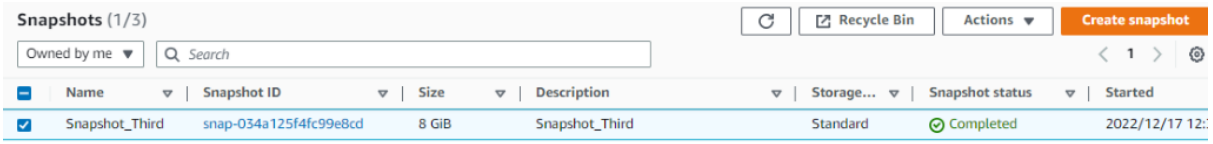
Sol tarafta edit AMI persmissions a basıyoruz. Burada public i seçiyoruz:



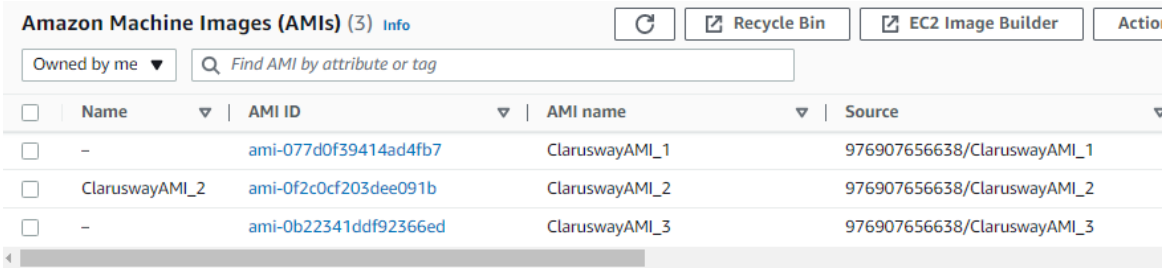
Public olması biraz zaman alır. Hocanın oluşturduğu public AMı yi görebiliyoruz:



Oluşan 3. snapshottan 3. AMI yi oluşturalım:



3. AMI oluştu:

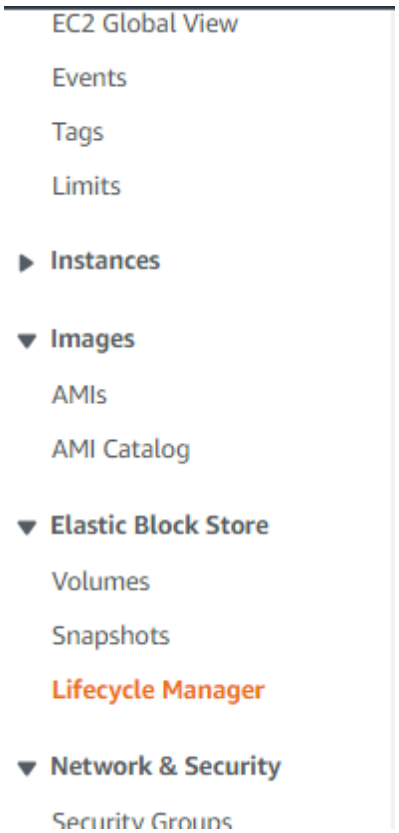


BU AMI den de bir instance ayağa kaldırabiliriz. Yukarıdaki aynı yöntemi uyguluyoruz. Aplikasyonumuzun bu instance ile de çalışacağını görebiliriz.

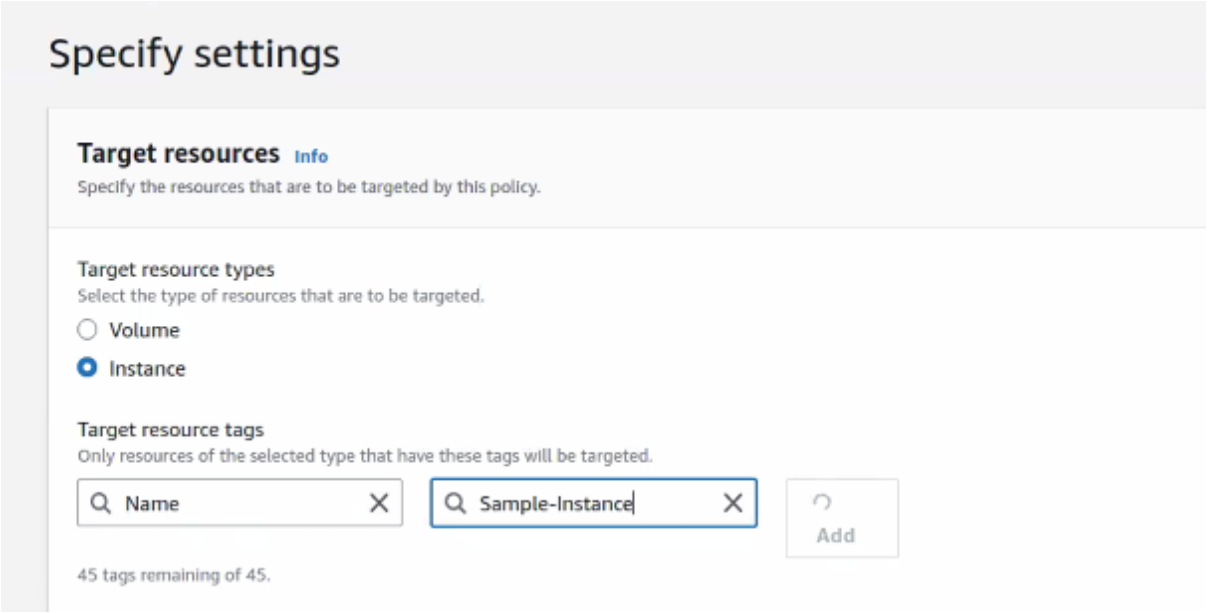
Şunu görmüş olduk aynı zamanda, AMI ile yazılımsal toolları hazır kullanarak instance ayağa kaldırırken donanımsal bileşenleri kendimiz ayarlamamız gerekiyor. Yani söylediğimiz gibi AMI ler software tutan templatelerdir.

Datalife Cycle Manager:

Soldaki menüden girebiliyoruz:



Next Step deyip devam ediyoruz. Bize kaynak soruyor, birden fazla tag de ekleyerek resource ekleyebiliriz:
Burası önemli çünkü tagler üzerinden çalışıyor.



Description ekliyruz ve burada Life cycle manager a IAM role veriyoruz çünkü o bizim adımıza bu işlemleri yapacak.:

Burada zaman ve retention type ları set edip policy oluşturuyoruz:

Instanceları terminate ediyoruz. Snapshot ları önce silemiyoruz çünkü bunlar birer AMI ile ilişkili.
Önce AMI lerle ilgili deregister işlemi yapıyoruz:
AMI ler deregester olduğu için snapshotlar otomatik silinmeyecektir diye uyarıyor:
Snapshot menüsüne gidip onları da silebiliriz artık:

Wordpress ile bir instance ayağa kaldırmak için Launch Instance diyorum:

Browse more AMIs diyoruz.
WordPress Certified by Bitnami and Automattic
Aratıyoruz ve
Free tier olarak ortadakini seçiyoruz.
Continue diyoruz.

Instance type ı kendisi oluşturdu.
Security group kendisi oluşturdu.

Key pair i seçiyoruz Launch instance diyoruz.

Instance ımızın IP adresini browser a yapıştırıyoruz.

Konfigure etmek için kullanıcı adı ve parola gerekirse bunun için instance ımızı seçip Actions kısmından Monitor and troubleshoot seçiyoru. Oradan da Get system log diyoruz. Buradan kullanıcı adı ve password alınması gerekiyor.