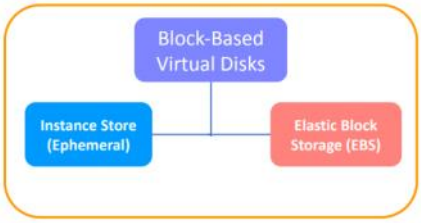
AMI Snapshot-Volumes

Saturday, August 3, 2024 9:46 PM

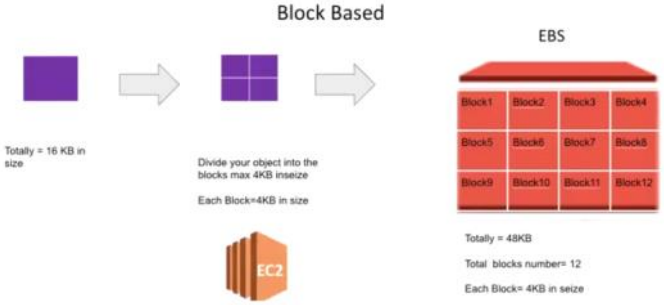
EC2 Volumes

What is Volumes?



- Volumes are durable storage devices (virtual) that can be attached to EC2 instances.
- They are location in which the associated machine stores its data or loads its applications.
- There are two volume types in the block storage category. These are Instance Stores (Ephemeral) and Elastic Block Storage (EBS).

Ec2 fizik bir bilgisayar gibiydi.
Nasil ki bir harddiske ihtiyac varsa burada da volume var.
Volume bir servera baglanmis bir depolama kapasitesidir.
Volume dedigimiz zaman bir servera bagli oldugu akla gelmeli ilk.
2 sekilde inceliyoruz. Instance store ve Elastic Block Storage.

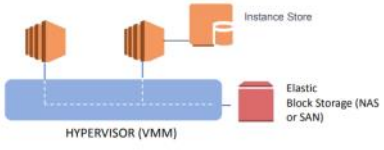


Who can call the data? =Only related EC2

Block based bloklara ayrilmis olarak dusunulebilinir.
16 kb lik bir dosyayi 4-4-4-4 luk parcalara boluyoruz ve EBS de her birini 4 kb llere ayrilmis 48 kblik bir storageta 4 ayri bloga yerlestiriyorum bloga yerlestiriyorum. Datayi cagirdigimiz zaman o tekrar bir butun olarak geliyor.
Bu datayi EC2 ile iliskili oldugu icin bir ec2 icinde sistem ihtiyaclarina cevap vermesi adina burada sakliyoruz.
Normalde bir ec2 icinde foto, video vs tutulmaz.
Bunu bir database olarak da kullanabiliriz.
Normalde ec2 yu puclic yapmiyoruz. Bir yayin yapiyoruz evet ama insanlar buna erisemiyorlar.
Biz bir EC2 uzerinden degil Elastic Load Balancer dedigimiz sey uzerinden yayin yapacagiz normalde.
Suan sadece denemeler icin yapiyoruz.
Ozetle biz volumeleri kendi ec2 icinde ihtiyaclarimizi gidermek adina kullaniyoruz.

EC2 Volumes

Instance Store and Elastic Block Storage



- The storage connected directly to the hypervisor and accessible to each machine associated with the hypervisor is called the Instance Store.
- Instance Storage can be connected to only one instance. And is the closest storage device to your instance.

Elastic block storage virtulazition ile bagli. Ec2 nun ne kadar ihtiyaci varsa o kadar yer aciyor. Kaymak ekmek kadayifi iliskisi. Ama ec2 oldurdugumde istersem EBS i ayakta tutabilirim.
Instance storeda ise ec2 ile tamamiyle butunlesmis. Yani ec2 yu terminate ettigimde bu da olur.

EC2 Volumes

Instance Store (Ephemeral) vs. Elastic Block Storage (EBS).

EC2 INSTANCE STORE

- Hardware attach
- Non-persistent data storage/ Terminated if you stop the instance
- Snapshots are not available
- Used for Buffers, Cache, Temporary data
- No separate cost for storage

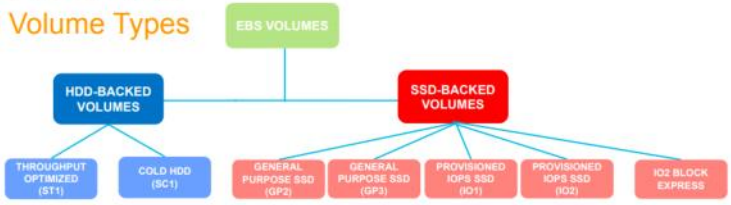
ELASTIC BLOCK STORAGE

- Network attached storage
- Persistent data storage
- Snapshots are available
- Separate cost for EBS even EC2 is stopped

Ebs bir kablo baglantisi gibi bir baglanti oldugu icin burada bir gecikme var.
Dolayisiyla cok hizli olmasini istedigimiz dosyalar oldugu zaman instance store tercih ediyoruz. Ama bu tercihler kullan at seklinde olan dosyalar icin gecerli. Tekrar donup bakmaya ihtiyacimin olmamasi lazim cunku server oldugunde o da olecek.
Bunun snapshot ozelligi yok. Instance storeda bir backup alamiyoruz. Elasticte alabiliyoruz.
Instance
Store ec2 da paket olarak geldigi icin bir cost yok ama ebs te var stop edilmis olsa bile.
Biz daha cok ebs kullanacagiz.

EC2 Volumes

EBS Volume Types



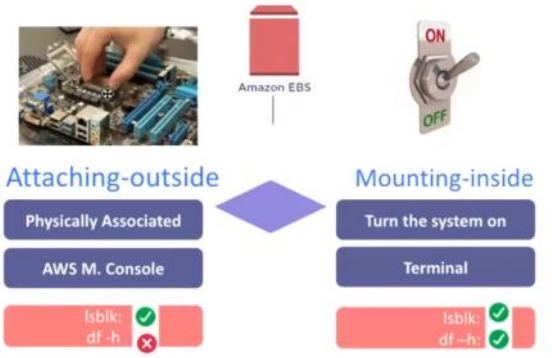
- There are 6 types of volumes in 2 categories for the different use cases.
- HDD-backed volumes are used for large streaming workloads where throughput is a better performance measure than IOPS.
- SSD-backed volumes are used for frequent read/write operations where the dominant performance attribute is IOPS.

Hdd daha eski ve ssd daha yeni donem volumeleri.
HDD daha cok throughputun onemli oldugu durumlarda kullanilir.
SSD ise IOPSun onemli oldugu durumlarda kullanilir.
Coklu GB oldugu durumlarda throughput seciyoruz.
Anlik databaseden sorgu yaptigiiz ya da bunun gibi hizin onemli oldugu durumlarda IOPS kullaniyoruz. Burada surekli olarak data isleniyor.

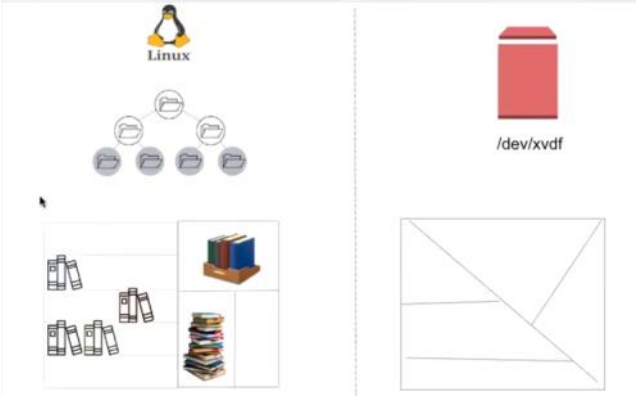
IOPS

Throughput

IOPSta hiz cok onemli. Input output islem hizi. Beklenti hizim fazla
Throughput hem daha kapsamli hem hizi yuksek. GB fazla

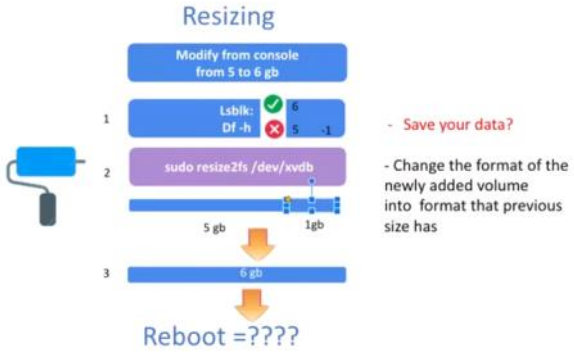


Attach etmek diye bir kavram var. Yani biz yerlestiriyoruz. Mesela buzdolabi aldik ve yerine yerlestirdik. Buzdolabi calismaz. Cunku mounting islemi gerceklesmemistir. Biz ne zaman fise takarsak o zaman caliscak. Iste bu da mounting yontemidir. Df -h ile power baslatmis oluyoruz.



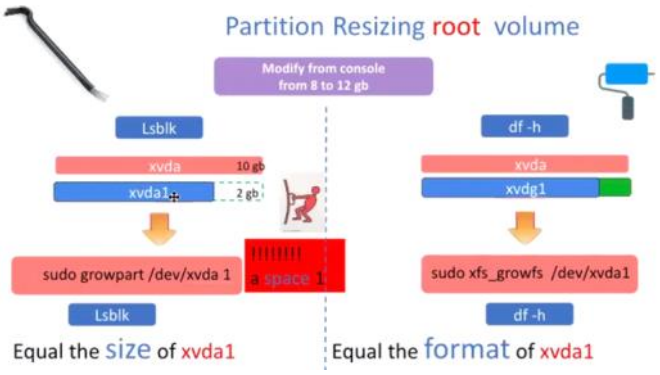
Formatlama bir raf duzeni gibi dusunelim. Kendi bir kitapligimizin iskeleti var ama raf sistemi yok. Iste formatlama ile bir raf sistemi kurmus oluyoruz. Yani dosyalarin nasil dagilacagini nasil calisacagini duzenlemis oluyoruz formatlama ile.

Burada da mounting etme yontemi anlatiliyor. Burada makinede iki ayri dosya grubu var. Gorunurde iki dosya olusturmus olsamda o henuz bu makineye monte edilmemis. EBSde formatlanmis ama makineye atilmamis.
Bunun icin once sudo mkdir /mnt/mp1 ile olusturuyoruz rootun altinda. Daha sonra mount yapiyoruz su sekilde: sudo mount /dev/xvdf /mnt/mp1
Burada rootun altinda /mnt/mp1 yarattik ve /dev/xvdf yi bu roottaki dosyanin icine aldik ve root icinde olusturudugumuz mnt dosyasinin altinda makineye bagladik.



Görsel, bir disk bölümünün boyutunun 5 GB'den 6 GB'ye çıkarılması sürecini açıklayan bir diyagram gibi görünüyor. İşte adımların Türkçe açıklaması:

1. **Konsoldan Değişiklik Yapma (5'ten 6 GB'ye)**:
   - İşlem, konsol üzerinden disk bölümünün boyutunun 5 GB'den 6 GB'ye çıkarılmasıyla başlıyor.
2. **lsblk ve df -h ile Kontrol Etme**:
   - lsblk komutu, yeni boyutu 6 GB olarak gösteriyor.
   - df -h komutu ise hala eski boyut olan 5 GB'yi gösteriyor, bu da dosya sisteminin henüz genişletilmediği anlamına geliyor.
3. **Dosya Sistemini resize2fs ile Genişletme**:
   - sudo resize2fs /dev/xvdb komutu, dosya sistemini geni şleterek yeni eklenen 1 GB'yi de kapsamasını sağlar ve toplam kullanılabilir alanı 6 GB yapar.
4. **Genişletme Sonrası Dikkat Edilmesi Gerekenler**:
   - Diyagramda verilerin kaydedilmesi ve yeni eklenen bölümün, önceki boyutun format ına uygun şekilde formatlanması gerektiğine dair bir not bulunuyor.
5. **Yeniden Başlatma**:
   - Değişikliklerin geçerli olması için yeniden başlatmanın gerekli olup olmadığını hatırlatan veya belirten bir soru işareti ile gösteriliyor.



Bu diyagram, kök bölümünün (root volume) boyutunu artırma işlemini açıklıyor. İşlem, iki ana adımda gerçekleştiriliyor:

1. **Bölüm Boyutunu Konsoldan 8 GB'den 12 GB'ye Çıkarma**:
   - lsblk komutuyla mevcut disk bölümleri kontrol ediliyor. xvda adlı ana disk 10 GB, xvda1 ise 2 GB olarak görünüyor.
   - sudo growpart /dev/xvda 1 komutuyla xvda1 bölümünün boyutu, ana diskle eşitlenecek şekilde genişletiliyor.
   - Bu işlem sonrasında, lsblk komutu ile kontrol edilerek bölüm boyutunun eşitlendiği doğrulanıyor.
2. **Dosya Sistemi Boyutunu Eşitleme**:
   - df -h komutu kullanılarak disk kullanımı görüntüleniyor.
   - sudo xfs_growfs /dev/xvda1 komutuyla dosya sistemi genişletiliyor, bu da yeni alanın kullanılabilir hale gelmesini sağlıyor.
   - Son olarak, df -h komutu ile dosya sisteminin boyutunun eşitlendiği ve kullanılabilir hale geldiği doğrulanıyor.

Diyagram, kök bölümünün büyütülmesi işleminin hem bölüm boyutunun artırılması hem de dosya sisteminin bu yeni boyutla uyumlu hale getirilmesi gerektiğini vurguluyor. Her iki adım da mevcut verilerin korunması ve yeni alanın düzgün kullanılabilmesi için önemlidir.

1 Amazon Machine Image (AMI)

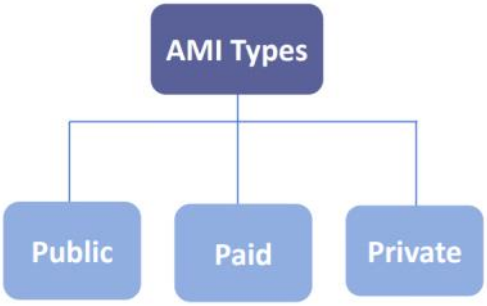
Amazon Machine Image (AMI)

What is AMI?

Instancemizin icinde bulunmasini istedigimiz operating sistemler, programlara AMI diyoruz.

- An Amazon Machine Image (AMI) is used for the launching an virtual instances in the AWS environment.
- AMI are like templates that are configured with an operating system and other software, which determine the user's operating environment.
- You can copy an AMI . So you can launch multiple instances from a single AMI with the same configuration.

Types of AMIs



3 tip image var.
Public herkese acik.
Paid firmalar kendine lazim olani yukluyor mesela SQL server gibi. CISCO gibi.
Bir image sadece kendinize ya da istediklerinize sunabilceginiz sekilde tasarlamissaniz bu private.

CLARUSWAY

2 Snapshot

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

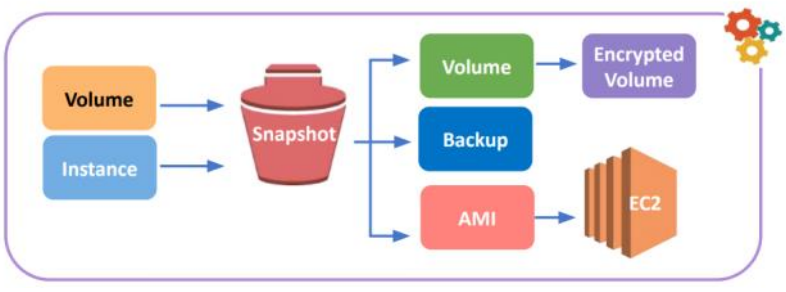
Snapshotlar imagein ya da volumeun bir onceki hali.
Instancelarin uzerinde bir sikistirma ile tutuldugu haliyle kullanilan bir backup.

Snapshot
Features of the Snapshot

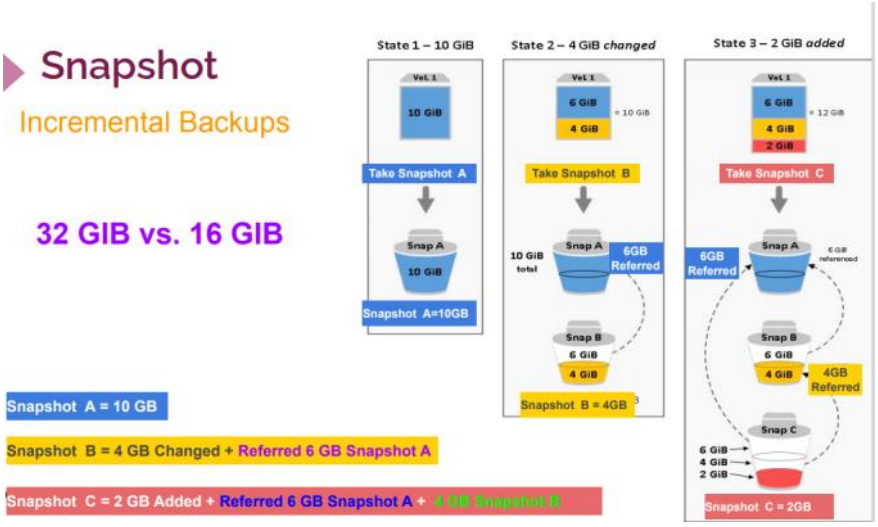
- Source from Volume or Instance
- Stored in Amazon S3
- Incremental storage
- Data Lifecycle Manager (DLM)

Format olarak ec2 icersinde olmasina ragmen, awsin depolama kaynagi olan s3 icersinde tutulur.
Snapshotin iki kaynagi var: volume ve instance
Instance ya da root volume snapshotini almak arasinda bir fark yok.

Snapshot
Lifecycle of Snapshot



Snapshotin iki kaynagi var: volume ve instance
Root volumeunde, additional volumun de snapshotini alabiliriz ya da instancein kendisinin de snapshotini alabiliriz.
Instance ve additional volumeun snapshotini almak ayni kapiya cikar fakat root volume snapshoti farkli.
Olusan snapshotta yani zipten 3 sey uretebiliyoruz: volume, backup ya da AMI.
AMI uretirsem buradan bir ec2 olusturabiliyorum.
Snapshot bu isin DNAsi gibi.



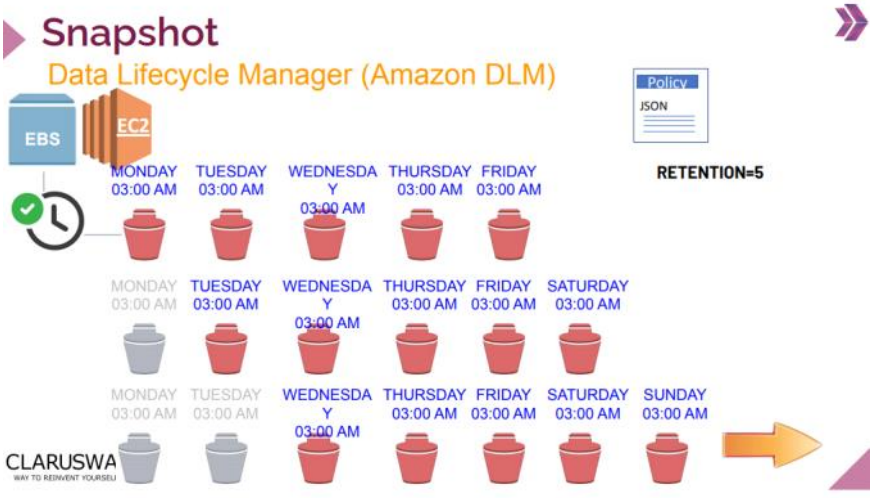
Elimde 10 GB bir snapshot var:

1. Gun snapshotini aldigimda elimde 10 gb lik bir data oluyor.
2. Gun 6 gb elde kaliyo 4 gbin snapshotini aliyoruz. Yani 6 gb lik kismi dunku snapshotta zaten mevcut, oraya refere edildi. Incremental backup hic olmayan veri icin backup almak demek. Boylece 2. gun sonunda elde 4 gb lik bir data olmus oluyor.
3. Gun ilk gunun 6 gb ve ikinci gunun 4gb backupi refere edilip 2 gb kisim icin yeni bir backup kismi aciyor.

Gunun sonunda elde olan 16 gb kismi 10unu ilk gun 4unu ikinci gun ve 2 sini de son gun kullanmis oldu. Incremental backup yapmasaydik ilk gun 10, ikinci gun 10 e 3. gun 12 gb tan toplam 32 gb lik alan harcamis olacakti.

Dezavantaji,biraz yavaslatiyor.
Avantaji, yer tasarrufu sagliyor.
Bunu life cycle management yapiyorsak yapiyoruz yani aws yonetiyprsa.

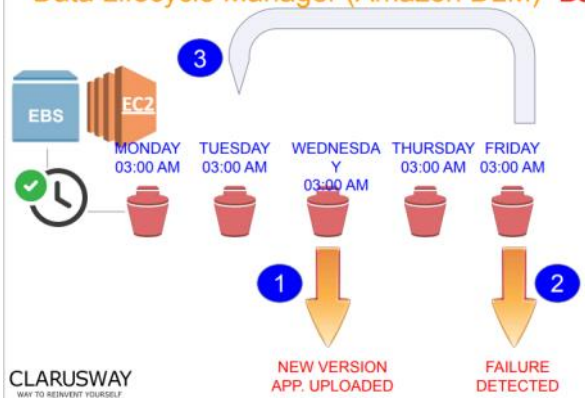
Backupi sisteme oturtma isini lif cycle manager yapiyor.
Retention giriyosun mesela 5 gun. Hergun saat 03.00 da.
Pazartesiden cumaya 5 gun aliyor backup.



Sonra pazartesiyi silip Cumartesi aliyor. Sonra saliyi silip pazari aliyor ve boyle devam ediyor.

Snapshot

Data Lifecycle Manager (Amazon DLM)- Backup and Restore



Bunu su sebeple yapiyoruz:
Mesela carsamba yeni bir versiyon alindi ama Cuma bir hata yapildi. Tekrar carsamba gunu alina backuptan o versiyona ulasilmis olunabiliyor.

Snapshot

AWS Backup

AWS Backup is fully managed service that centralizes the all backup process in your project including.

Backup alma isini tum AWS capinda yurutuldugu durum.
Data life bu isin sadece ec2 bazinda yurutuldugu hali ama aws backupta tum compute ve storage unsurlarinin tekardan backup edilmesi durumudur. Tek merkezden taglere sahip tum unsurlarin backupini almis oluorsun.

Snapshot

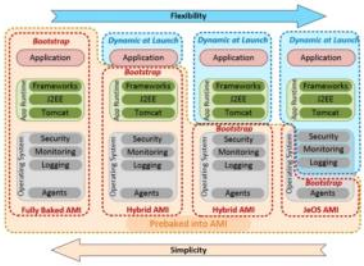
Encryption of Root Device via Snapshot

- 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.

Bir instance veya databasei encrypt etmemissek volumu, snapshot esnasinda backupini alip tekrar bunu (iinstance ya da database) encrypt edebiliyoruz.
Normalde bir instanceta volumu encrypt etmemissek calistirdiktan sonra onu encrypt edemeyiz. Ama snapshottan backupi alindiginda bu mumkun.

Golden AMI

- A golden AMI is an AMI that contains security patches, configuration, and agents required to by an organization. A “just enough OS” (jeOS) is the most basic golden AMI.
- It may also contain specific software components that make it easier and faster to start-up an instance .



CLARUSWAY
WAY TO REINVENT YOURSELF

Flexibility ve simplicity unsurlari var burada.

- **Golden AMI Tanımı**:
  - Güvenlik yamaları, yapılandırma ve ajanlar gibi bileşenleri içeren özel bir AMI'dir (Amazon Machine Image).
  - "JeOS" (Just enough OS), sadece gerekli olan işletim sistemi bileşenlerini içeren en temel Golden AMI'dir.
- **İçerik**:
  - Güvenlik yamaları
  - Yapılandırma ayarları
  - Ajanlar (Agents)
  - Diğer yazılım bileşenleri
- **Amaç**:
  - Bir örneği (instance) daha hızlı ve kolay bir şekilde başlatmak için gereken yazılım bileşenlerini sağlamak.
- **Diyagram Açıklaması**:
  - **Tamamen Hazır AMI (Fully Baked AMI)**: Tüm yazılım bileşenleri önceden yüklenmiş ve yapılandırılmıştır.
  - **Hibrit AMI (Hybrid AMI)**: Bazı bileşenler önceden yüklenmiş, bazıları ise dinamik olarak başlatıldığında yüklenir.
  - **JeOS AMI**: Minimum gereksinimleri karşılayan, sadece gerekli işletim sistemi bileşenlerini içeren AMI'dir.
- **Esneklik ve Basitlik Dengesi**:
  - Tamamen hazır AMI'ler basitliği, JeOS AMI'ler ise esnekliği temsil eder.
  - Hibrit AMI'ler bu iki uç arasında bir denge sunar.