Elastic File System (EFS)

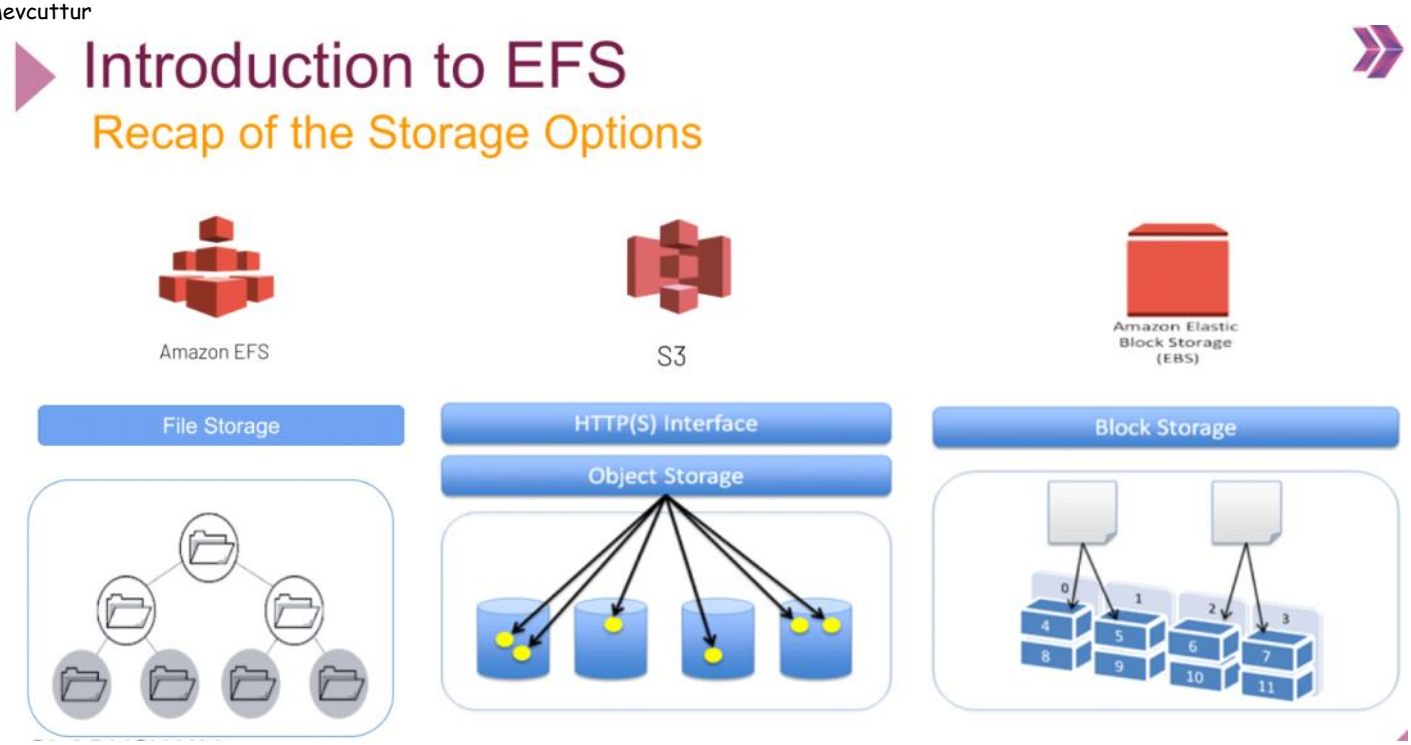
Monday, 16 August 2021 1.18

Table of Contents

- Introduction to EFS

RDS gibi fully managed systemdir. Yetkinin tamanen AWS de oldugunu dusunebiliriz biz sadece baslangictaki parametreleri

olusturuyoruz. Yukaridaki gorselde oldugu gibi dosyalama sistemi mevcuttur



EBS ==> Block Storage. Storage parcalardan olusmaktadir. EBS blok sisteminin kullanim seklidir. EBS ler root volume de olsa EC2 ya baglidir

S3 ==> veriler 3 parcadan olusmaktadir. Veri id si bu unsurlardan biridir. Ulasacagimiz veriye ID vasitasiyla ulasabiliriz. S3 teki veriye her yerden ulasabiliriz. EC2 ya bagli degiliz.

EFS ==> hiyerarsik bir yapi mevuttur.

- Amazon Elastic File System (Amazon EFS) is a **serverless** and **set-and-forget** elastic file system. It can be used with AWS Cloud services and on-premises resources.
- Amazon EFS is designed to **increase and decrease the storage capacity** automatically as you **add** or **remove** files. So, it is a flexible-capacity storage solution.
- Since, Amazon EFS offers a simple web services interface, you can easily establish file systems and avoid the difficulties of deploying, patching, and maintaining complex file system configurations.

- Features of EFS

Features of EFS

Scalability-Cost

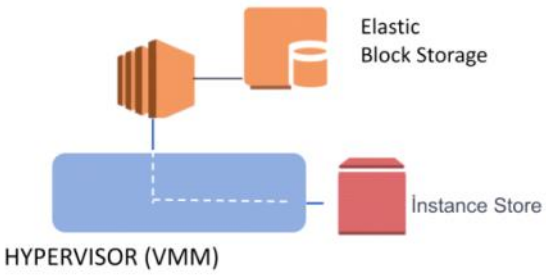
- Since EFS is scalable, it increases and decreases the storage capacity automatically as you add and delete files,
- There is no minimum fee or setup cost.

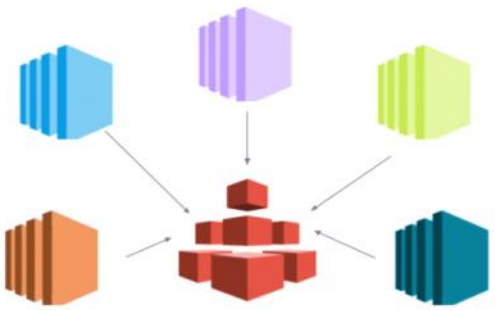
OLCEKLENEBILIR. Bu ozellikle de S3 e benzemektedir. Manuel olarak yapmaya gerek yoktur. Depolama kapasite azalma artma seklinde olabilir ve otomatik olarak olceklenebilmektedir. (EBS de bir sinirlama mevcuttur) yani baslangicta miktar belirtme soz konusu degildir.

Baslangicta bir ucretleme istememektedir.
Database icin EBS kullanmaktadir.

Features of EFS

Attaching





- Unlike *EBS, **multiple Amazon EC2** instances (**Linux only**) even **in different AZ's** can be attached Amazon **EFS** file system **at the same time**.

*Except Nitro-based instances in the same Availability Zone.

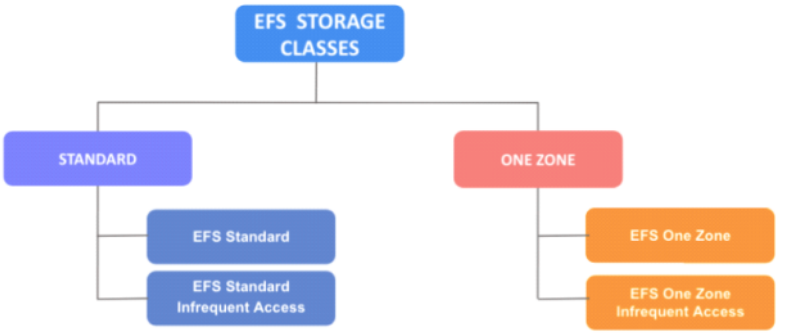
Ilk ec2 actigimizda bir storage olmaktadir ve bu HYPERVISOR idi. Eklenen storage ayni AZ de olmasi gerekmektedir. Ama EFS de bu zorunluluk bulunmamaktadir. Ama istersek harici storage acabilmekteydik. Bir ec2 ya bagli bir ebs den ziyade bir kac ec2 nun storage ulasmasini isteyebiliriz ve bunu da

EFS saglamaktadir.

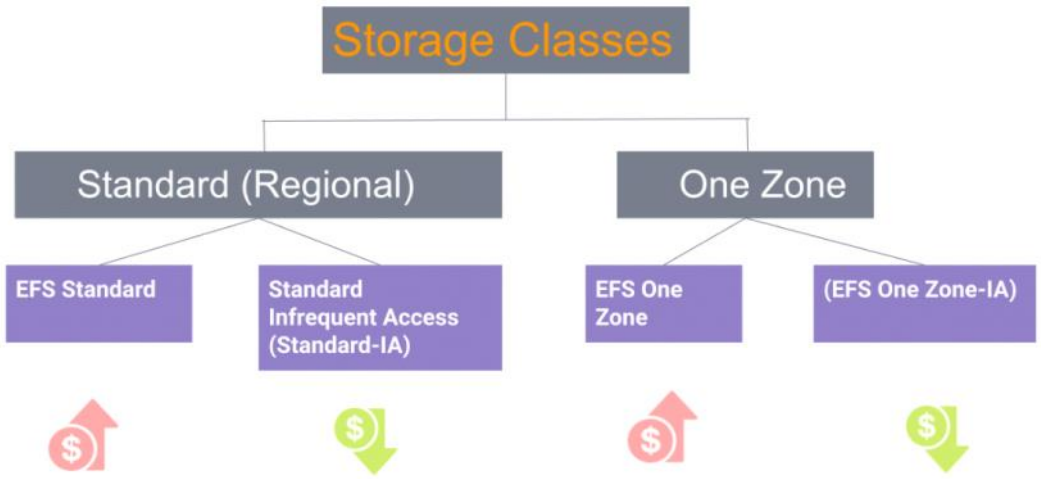
EFS ==> Farkli EC2 larin file sisteme ulasma durumu mevcuttur.
Peering ile farkli vpc ler uzerinden file system e ulasabilmek mumkun

- Amazon EFS file systems can **automatically** scale from gigabytes to petabytes of data without needing to provision storage.
- Compute services including Amazon EC2, Amazon ECS, Amazon Elastic Kubernetes Service (EKS), AWS Fargate, and AWS Lambda can be used compatible with the Amazon EFS file system.
- **Multiple compute instances** (even thousands of) can access an Amazon EFS file system at the same time.
- There is no minimum fee or setup cost and you pay only for the storage used by your file system.
- Amazon EFS is compatible with all Linux-based AMIs for Amazon EC2. It is not supported on Windows instances.

- EFS Storage Classes



EFS Storage Classes



- Amazon EFS offers you the choice of creating file systems using two main types of storage classes.

    - **Standard**
    - **One Zone**

- Under these two types of storage classes AWS offers 4 types of storage classes that are designed for different use cases:
- **EFS Standard**: *A regional storage class* for **frequently** accessed data.
  It offers the highest levels of availability and durability by storing file system data redundantly across *multiple Availability Zones* in an AWS Region.

- **EFS Standard-Infrequent Access (IA)** : *A regional storage class* for **infrequently** accessed data.
  It offers the highest levels of availability and durability by storing file system data redundantly across *multiple Availability Zones* in an AWS Region.

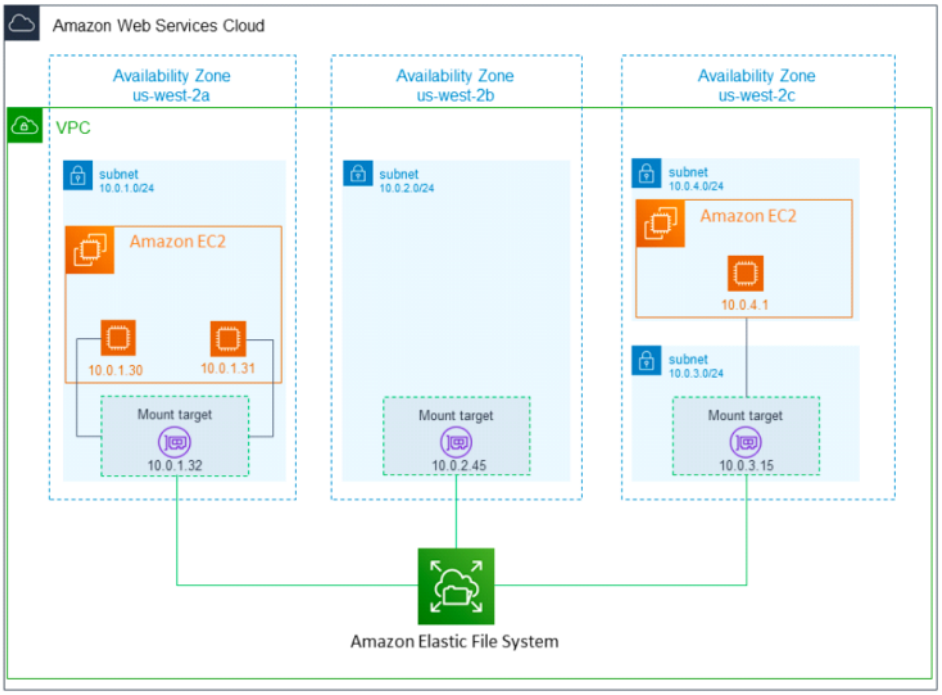
- **EFS One Zone** : For **frequently** accessed files stored redundantly within a *single Availability Zone* in an AWS Region.

- **EFS One Zone-IA** : A lower-cost storage class for **infrequently** accessed files stored redundantly within a *single Availability Zone* in an AWS Region.

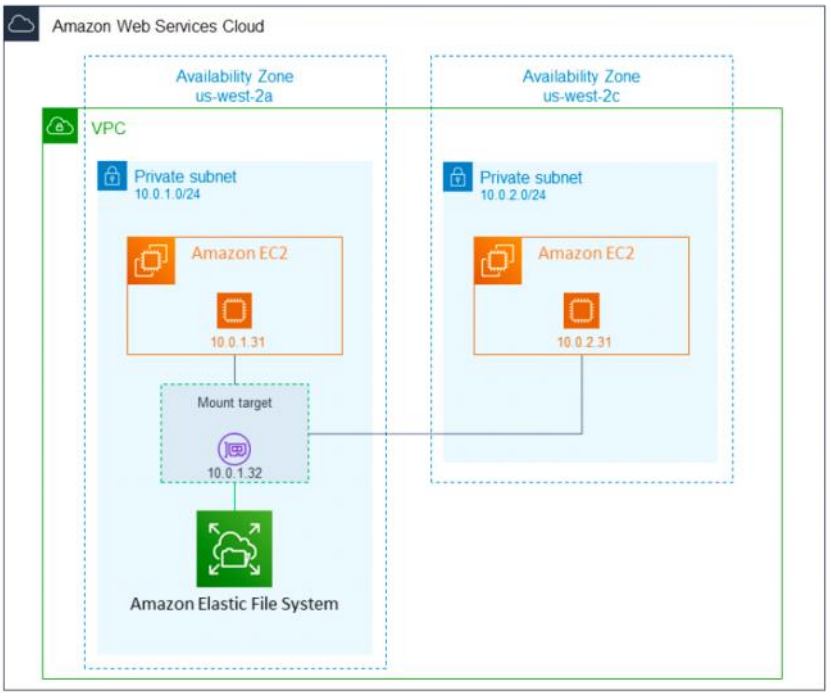
- Mount Target

- To **access** your Amazon EFS file system in a VPC, you create **one** or **more mount targets** in the VPC.

- You can create **multiple mount targets** in each availability Zone in the AWS Region for the file systems using **Standard** storage classes, like the picture seen below.



- As for file systems using **One Zone** storage classes, you can create only a **single mount target** that is in the same Availability Zone as the file system.



3 Comparison of Storage System

- [Comparing EFS with EBS and S3]

- Cost Optimized : S3 > EBS > EFS
- Speed : EBS , EFS >S3
- EC2 mount : S3 : No
  EBS : Single*
  EFS : Multiple
- Storage Capacity : S3, EFS = ∞ vs. EBS =16 TB

Amazon EFS

- Large quantities of data,
- Large analytic workloads.
- Global content management

S3

- Website images and videos,
- Data analytics of mobile/web applications.
- Data which is needed to be accessed from anywhere.

EBS

- High IOPS required data ,
- Database management.

https://aws.amazon.com/efs/enterprise-applications/

		File Amazon EFS	Object Amazon S3	Block Amazon EBS
Performance	Per-operation latency	Low, consistent	Low, for mixed request types, and integration with CloudFront	Lowest, consistent
	Throughput scale	Multiple GBs per second	Multiple GBs per second	Single GB per second
Characteristics	Data Availability/Durability	Stored redundantly across multiple AZs	Stored redundantly across multiple AZs	Stored redundantly in a single AZ
	Access	One to thousands of EC2 instances or on-premises servers, from multiple AZs, concurrently	One to millions of connections over the web	Single EC2 instance in a single AZ
	Use Cases	Web serving and content management, enterprise applications, media and entertainment, home directories, database backups, developer tools, container storage, big data analytics	Web serving and content management, media and entertainment, backups, big data analytics, data lake	Boot volumes, transactional and NoSQL databases, data warehousing & ETL

- **How is Amazon EFS different than Amazon S3?**

  Amazon **EFS** provides a traditional file-sharing permissions model, **file-based** storage, via the NFSv4 protocol.

  Amazon **S3** is an **object-based** storage platform that uses a simple API for storing and accessing data.

- **How is Amazon EFS different than Amazon EBS?**

  Amazon **EFS** provides shared file storage for use with compute instances both in the AWS Cloud and On-premises servers. But **EBS** is designed for cloud purposes.

  While **EFS** can be mounted multiple EC2 instances, **EBS** can only be mounted **single** Amazon EC2 instance*.

  ***Exception**: Multiple EC2 Nitro System Based Instances can be mounted single EBS

- **Amazon EFS'nin Amazon S3'ten farkı nedir?**

Amazon EFS , NFSv4 protokolü aracılığıyla geleneksel bir dosya paylaşım izinleri modeli, **dosya tabanlı** depolama sağlar.

Amazon S3, verileri depolamak ve verilere erişmek için basit bir API kullanan **nesne tabanlı bir** depolama platformudur.

- **Amazon EFS'nin Amazon EBS'den farkı nedir?**

Amazon EFS, hem AWS Cloud hem de Şirket içi sunucularda bilgi işlem örnekleriyle kullanım için paylaşılan dosya depolama alanı sağlar. Ancak EBS, bulut amaçları için tasarlanmıştır.

EFS, birden çok EC2 bulut sunucusuna monte edilebilirken, EBS yalnızca **tek bir** Amazon EC2 bulut sunucusuna bağlanabilir *.

* **İstisna** : Birden fazla EC2 Nitro Sistem Tabanlı Eşgörünüm tek bir EBS'ye monte edilebilir

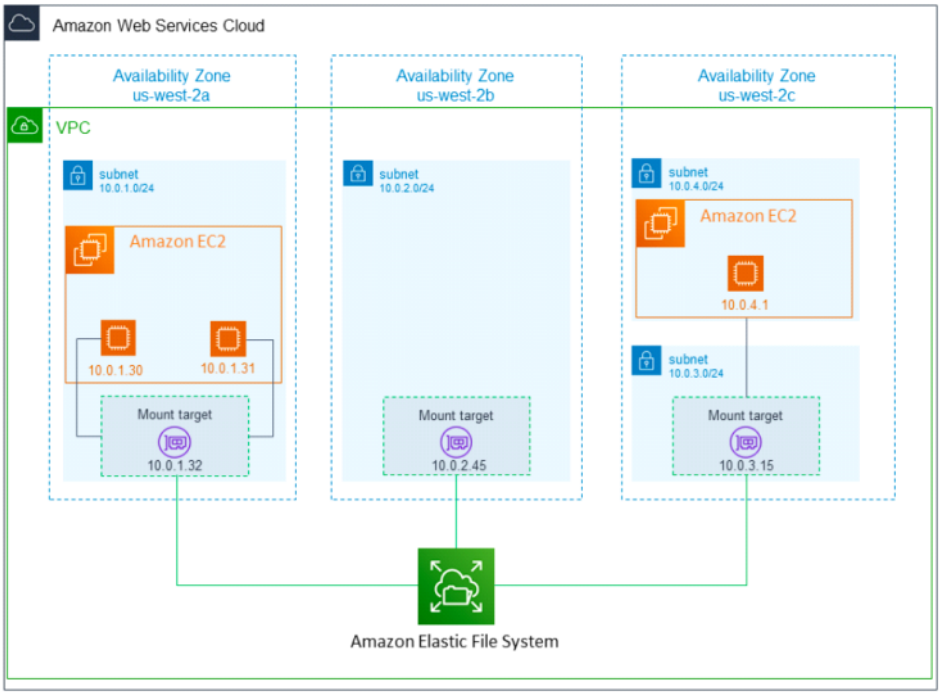
Amazon **EFS**, Amazon **EC2** ile birlikte kullanılmak üzere tasarlanmış olan bir *dosya depolama hizmetidir*. Amazon EFS, bir dosya sistemi arabirimi ve dosya sistemi erişim semantiği (yüksek tutarlılık ve dosya kilitleme gibi) sunmasının yanı sıra bu depolama hizmetine binlerce Amazon EC2 bulut sunucusu tarafından aynı anda erişim sağlanabilir.

Amazon EBS, Amazon **EC2** ile birlikte kullanılmak üzere tasarlanmış olan bir *blok düzeyinde depolama hizmetidir.*

Amazon **EBS**, tek bir EC2 bulut sunucusundan gelen ve veri erişimi için en düşük gecikme süresine ihtiyaç duyan iş yükleri için yüksek performans sunabilir.

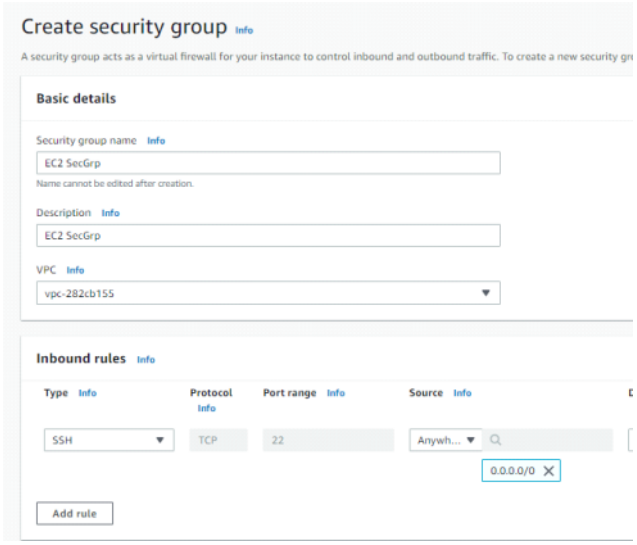
Amazon **S3** bir *nesne depolama hizmetidir.* Amazon S3, verileri her yerden erişilebilen bir İnternet API'si aracılığıyla kullanıma sunar.

Asagidaki mimariyi Hands-on da yapacagiz



Öncelikle security grouplari olusturacagiz (EC2 lar ve EFS ler icin)

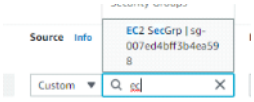
Ec2 icin;

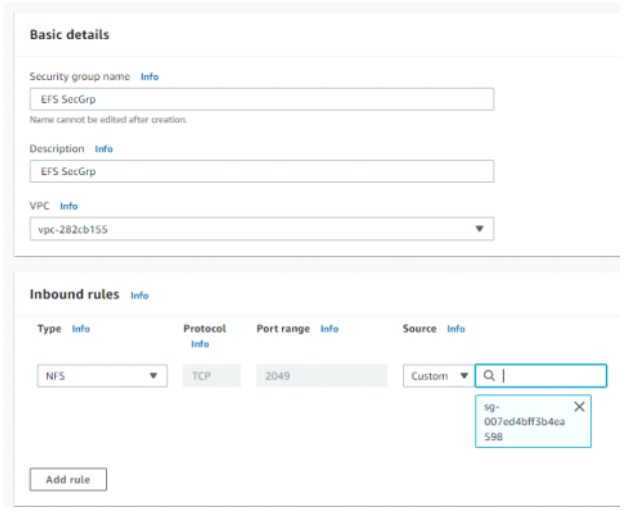


Tekrar

EFS icin;

Source yi ec2 ya gore custom etmek gerekmektedir.
Olusturdugumuz sec group a ait Ec2 lar EFS e ulasabilecek anlamindadir.





Kaynak kismini yani EC2 lari olusturalim

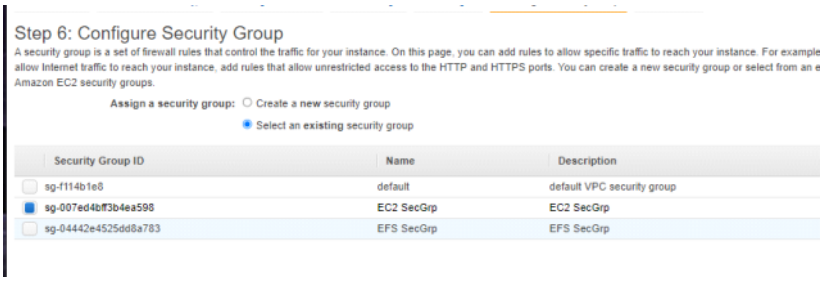
```
### Step 3 - Create EC2 :

- Configure First Instance in N.Virginia

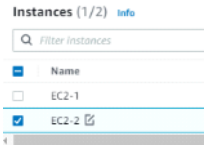
```text
AMI             : Amazon Linux 2
Instance Type   : t2.micro
Network         : default
Subnet          : default
Security Group  : EC2 SecGrp
    Sec.Group Name : EC2 SecGrp
Tag             :
    Key         : Name
    Value       : EC2-1
```
```

```
```text
AMI             : Amazon Linux 2
Instance Type   : t2.micro
Network         : default
Subnet          : default
Security Group  : EC2 SecGrp
    Sec.Group Name : EC2 SecGrp
Tag             :
    Key         : Name
    Value       : EC2-2
```
```

Ayni AZ da calismasi icin number of instance yi 2 diyelim



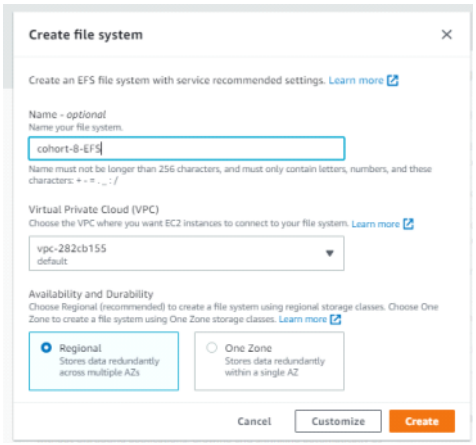
Isim verelim EC2 lara;



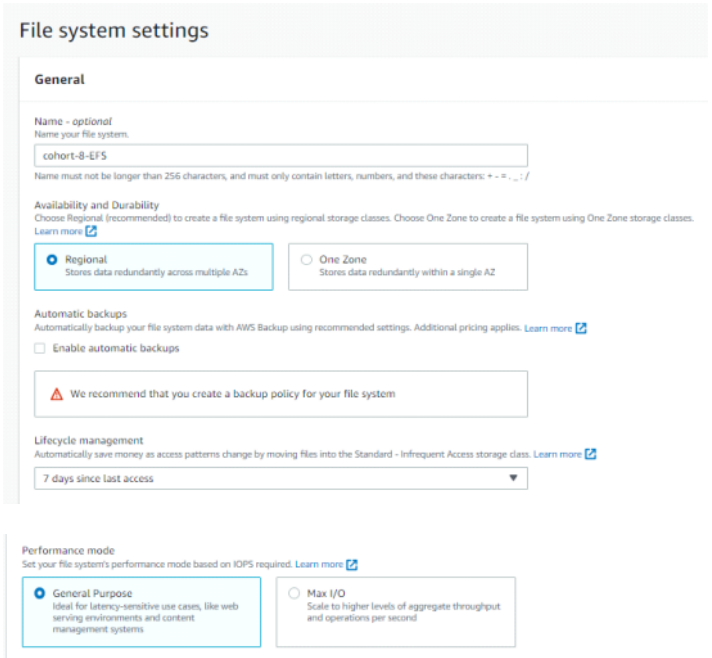

```
## Part 2 - Creating EFS
```

Her Region icin farkli bir fiyatlandirma olabilmektedir.


```
Name                            : FirstEFS
Virtual Private Cloud (VPC)     : Default VPC (Keep default)
Availability and Durability     : Regional (Keep default)
```



Customize



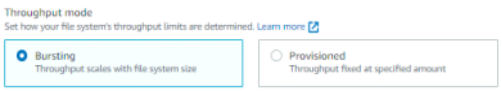
Performance mode : EFS yi ne amacla kullandigimizi gore secenekler secilebilir. Sirket icin ise performans onemli olmamakla birlikte data analitik durumlarinda performans beklentimiz olabilir. (saniye basi read write sayisi secenekler konusunda yonlendirebilir)

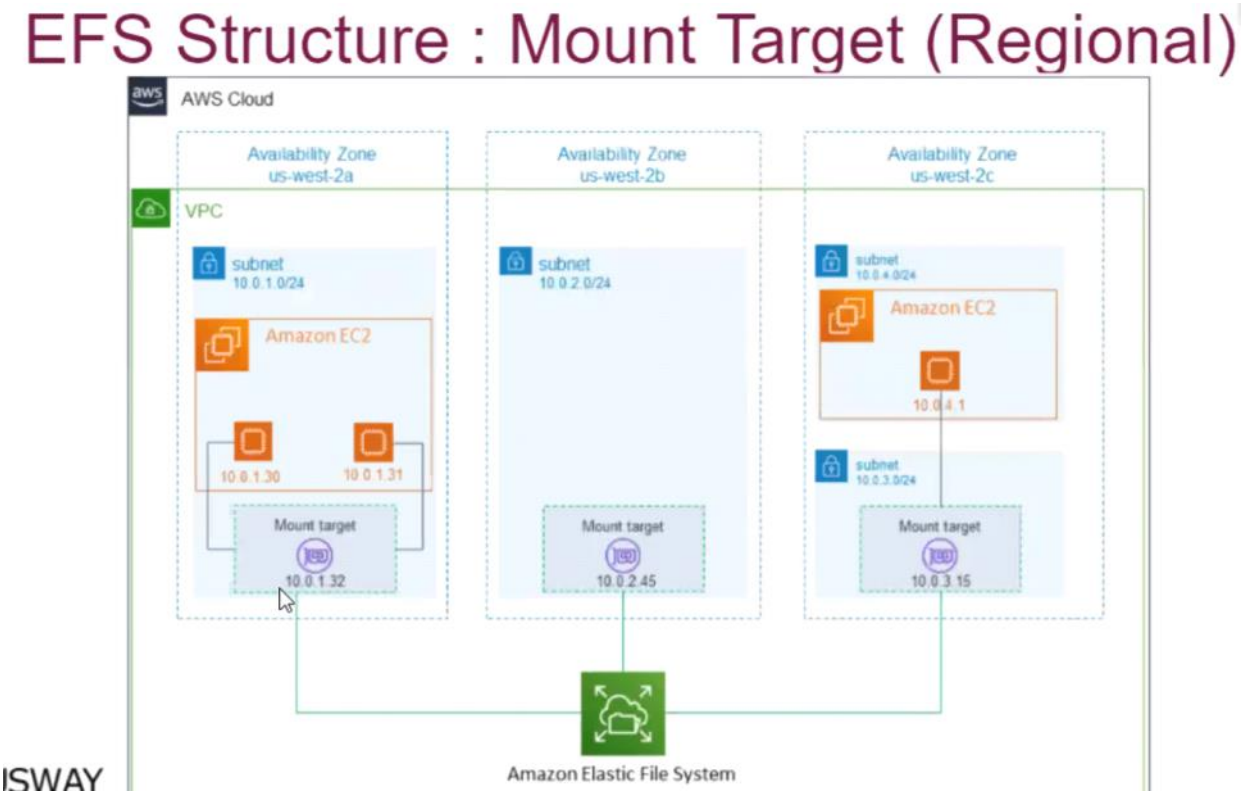
Throughtput mode: veri/saniye yani debi de diyebiliriz

Provisioned: secersek ayrica ucretlendiriliz

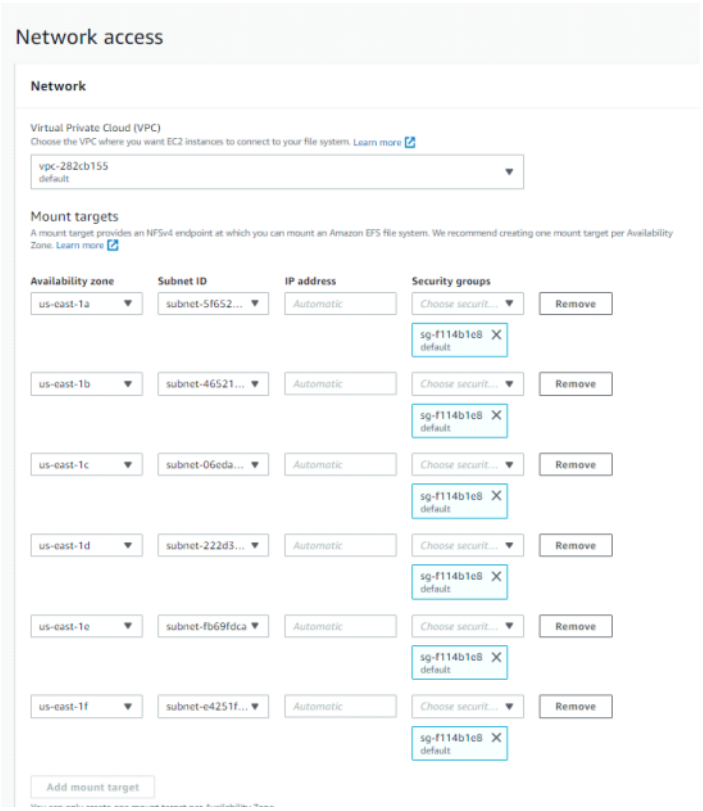
IOPS

Throughput

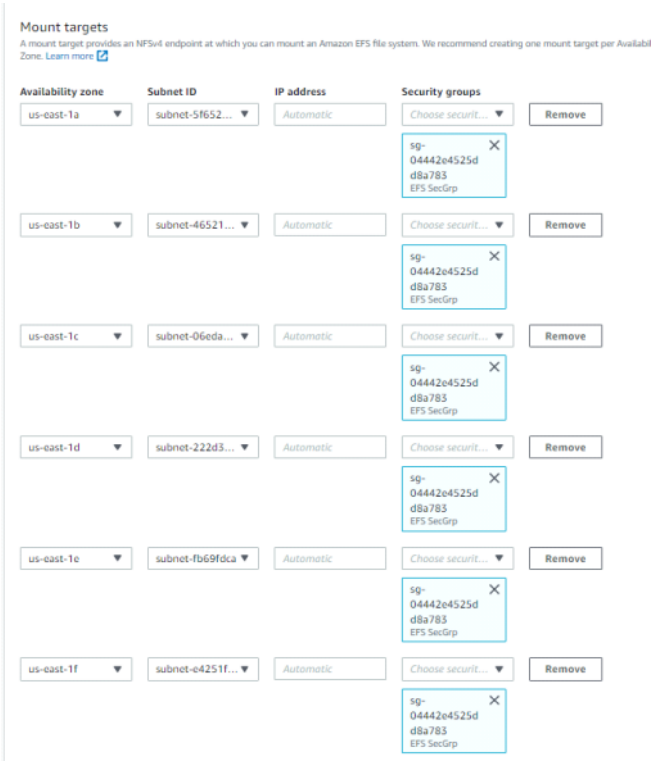




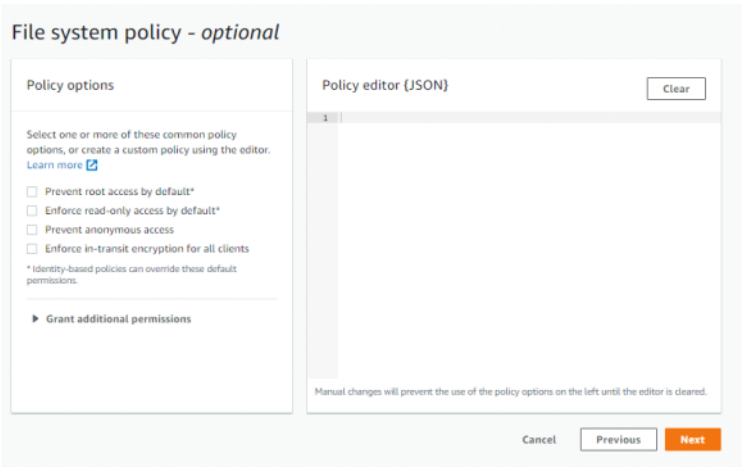
Amazonun verdigi sec grouplari kullanmak istemiyoruz ve defaultlari silelim.



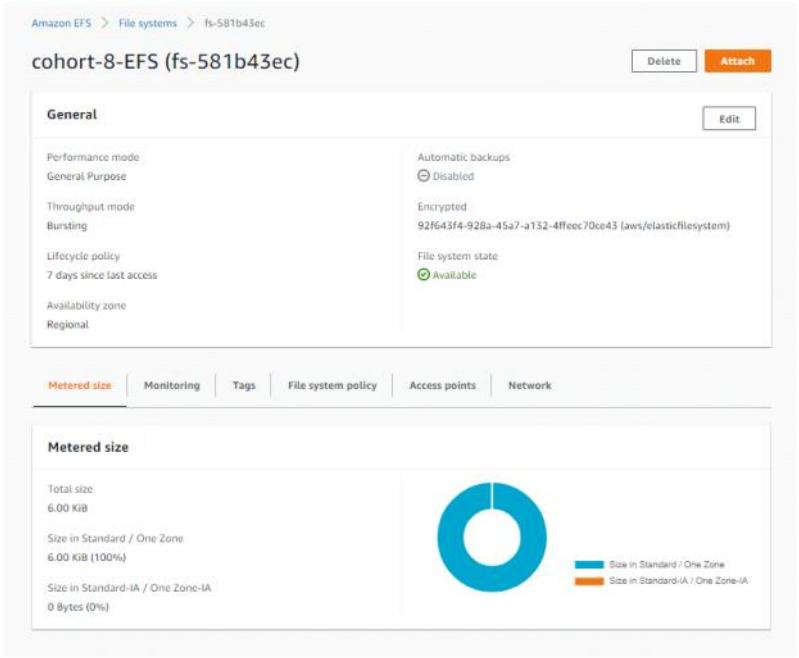
Ve ilk olusturdugumuz EFS sec grouplari secelim



File system policy : erisim yetkilerini set edebiliriz / IAM servisi ile iliskili



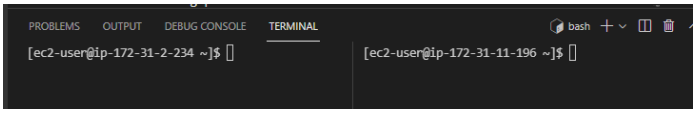
GORSEL :




```
## Part 3 - Attach the EFS to the multiple EC2 Linux 2 instances

### STEP-1: Configure the EC2-1 instance
```

VS cod vasitasiyla EC2 lara baglanalim (EFS sadece Linux tabanli OS leri destekliyor)



Her iki ec2 yu da update edelim

```
sudo yum update -y
```

Install the "Amazon-efs-utils Package" on Amazon Linux (her iki instanceye)

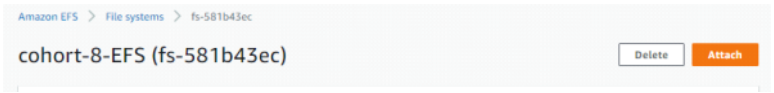
```
sudo yum install -y amazon-efs-utils
```

Create Mounting point;
Nereye mount edecegimizin onemi yok ve klasor olusturmamiz gerekiyor

Asagidaki gorseli ilk ec2 ya uyguluyoruz

```
[ec2-user@ip-172-31-2-234 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-2-234 ~]$ sudo mkdir efs-1
[ec2-user@ip-172-31-2-234 ~]$ ls
efs-1
[ec2-user@ip-172-31-2-234 ~]$
```

Attach islemini konsol uzerinden yapiyoruz



Gorseldeki komutu kopyalayalim ilk ec2 ya yapistiralim

(dosya ismine dikkat edelim)

```
[ec2-user@ip-172-31-2-234 ~]$ sudo mount -t efs
 -o tls fs-581b43ec:/ efs-1
```

Vs coddan efs-1 icerisine girelim bir dosya olusturalim ve kaydedelim

```
[ec2-user@ip-172-31-2-234 efs-1]$ cat test.txt
hello from first ec2
```

Diger EC2 da klasor olusturalim, klasor isminin bir onemi yok

```
[ec2-user@ip-172-31-11-196 ~]$ sudo mkdir efs-2
[ec2-user@ip-172-31-11-196 ~]$ ls
efs-2
```

Attach islemini tekrar edelim.

```
[ec2-user@ip-172-31-11-196 ~]$ sudo mount -t ef
s -o tls fs-581b43ec:/ efs-2
```

Ls komutu ile ilk ec2 da olusturdugumuz dosyayi ikinci ec2 da gormus olduk

```
[ec2-user@ip-172-31-11-196 efs-2]$ ls
test.txt
```

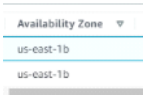
```
[ec2-user@ip-172-31-11-196 efs-2]$ cat test.txt

hello from first ec2
```

2. EC2 ta ayni dosya uzerinde yapacagimiz degisikligi de ilk dosyada gorebiliriz. Bu islem tam ters olarak da gerceklesebilir

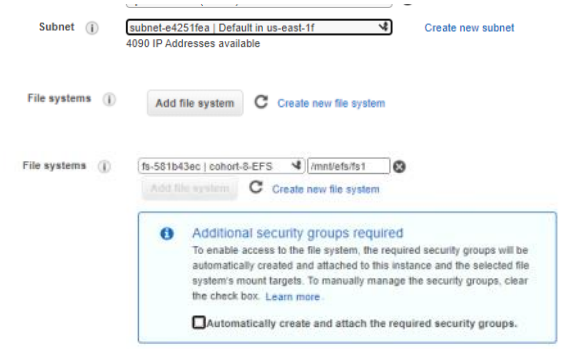
EC2 konsolundan yeni instance olusturalim

Acik iki EC2 nun AZ sini kontrol edelim



Yeni olusturacagimiz ec2 ya sayfasina gidelim

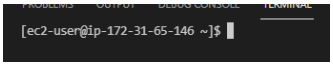
Farkli AZ secelim



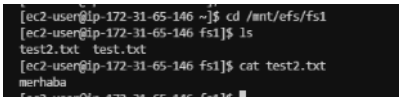
Specify Amazon EFS file systems to mount to your instance. To enable traffic between the instance and the file systems, the file systems must have a security group that allows inbound access for the TCP protocol on the NFS port from the instance, and the instance must have a security group that allows outbound access to the mount targets on the NFS port

Bu islem sonrasinda user data gorunmektedir

EC2-3 nolu EC2 ya VS cod dan baglanalim

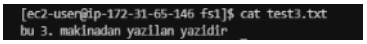


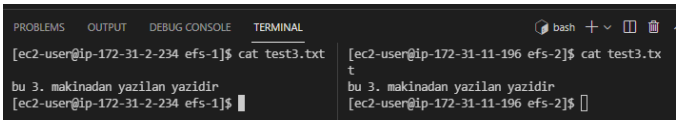
update islemi yapalim
Makinayi kurarken mounting islemlerini yapmis olduk ve file sistemine erismeyi bekliyoruz.



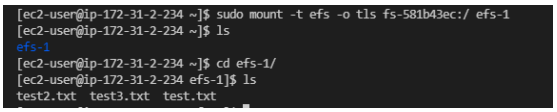
Ilgili klasore gittigimizde onceki dosyalari gorebiliriz

Dosya olusturalim ve ilk iki makinada kontrol edelim



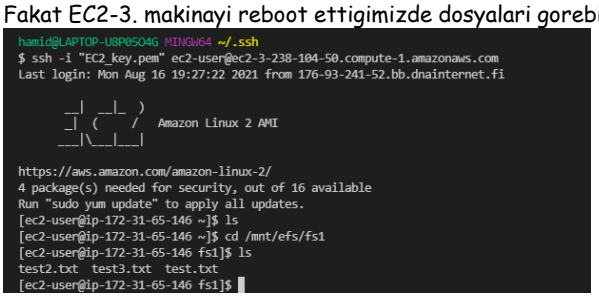


EC2-1 makinasini reboot ettigimizde klasoru gorecegiz ama daha onceki icerikleri goremeyecegiz, bu sebeple tekrar mount etmek gerekmektedir.



Yukarida gorundugu uzere mount islemini tekrar edince dosyalara ulasabiliyoruz

Fakat EC2-3. makinayi reboot ettigimizde dosyalari gorebilecegiz



EC2 lari terminate edelim ve olusturdugumuz EFS yi silelim