

AWS Certified Solutions Architect

AWS VPC (Virtual Private Cloud) Architecture.....	7
VPC Types.....	9
Security Groups.....	10
Network ACL	11
VPC Peering.....	12
Elastic Compute Cloud (EC2).....	13
Instance Types	13
Elastic Block Store (EBS).....	14
Elastic Block Store Types.....	16
Block Device Mapping.....	17
EC2-SR-IOV	18
EC2 Placement Groups.....	18
Cluster Placement Groups	18
Spread Placement Groups	18
EC2 Monitoring	18
EC2 Instance Termination	19
EC2 Instance Metadata and User Data	19
EC2 Migration	19
EC2 ve IAM Role	19
EC2 Bastion Hosts	20
EC2 ENI (Elastic Network Interfaces)	21
EC2 ENI IP Addressing	21
EC2 lifecycle Instance State'leri;	21
TCP/IP Packet Walkthrough.....	22
NAT Instance	23
Errors and Reasons	24
Instance-Store-backed veya EBS-backed impaired statusunda ise;	24
Reserved Instance.....	24
Spot Instances.....	24
On-Demand Instances.....	25
AWS Elastic load Balancer (ELB).....	26
ELB Health Checks.....	27

ELB Cross Zone Load Balancing	27
Amazon Route 53:.....	27
ELB Positioning - Internet Facing vs Internal ELB.....	28
ELB Security.....	28
ELB NACL.....	29
ELB Listeners	30
HTTP/HTTPS listeners;.....	30
ELB Sticky Sessions (Session Stickiness veya Session Affinity).....	31
ELB Security policy for SSL/HTTPS sessions	31
ELB-Connection Draining	31
ELB Monitoring.....	31
ELB Scaling, Prewarming, Testing ve Idle Timeout	32
ELB Scaling - DNS Updates:	32
AWS Auto Scaling.....	35
Scaling Policy	36
AWS Application Auto Scaling.....	36
ASG-EC2 Instance States (Cont.)	37
ASG ve ELB	37
ASG-Health Checks.....	37
ASG ve Spot Instance	37
Auto Scaling Policies	38
Monitoring Auto Scaling Group	38
RDS (Relational Database Service)	39
Multi-AZ RDS Option	40
DB Automated Backups	40
Manual Backups.....	40
RDS DB Security and Encryption	41
RDS Billing	41
Read Replicas	42
RDS Scaling.....	42
Amazon RDS CloudWatch Enhanced Monitoring.....	43
IAM Database Authentication.....	43
AWS Simple Storage Service (S3)	43
Block Storage	43
Object Storage	43
Data Consistency Models.....	44

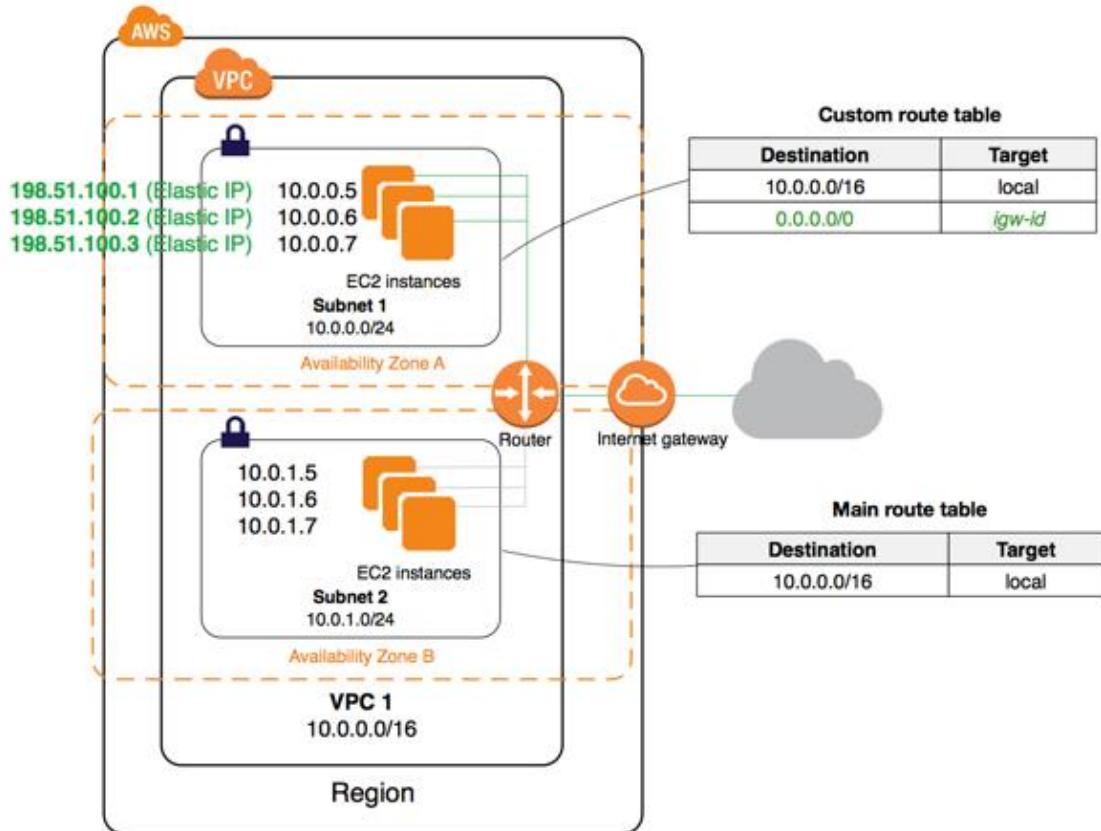
S3 Buckets	44
S3 Objects	45
S3 ACLs and Policys	45
S3 Access Policies.....	46
S3 Pre-defined Groups.....	47
Ne zaman Bucket ve Ne zaman User Policy.....	47
Permission delegation:.....	48
S3 Bucket Versioning.....	48
MFA (Multi Factor Authentication) Delete:	49
S3 Copying/Uploading S3 Objects.....	49
S3 Tiered Storage Classes.....	49
Amazon S3 Select.....	50
Glacier (Archiving Storage)	51
S3 Bucket LifeCycle Policies	52
S3 Encryption	52
S3 Static Website Hosting&Redirection.....	53
Sharing S3 Objects va Pre-Signed URLs.....	54
S3 Cross Region Resource Sharing (CORS)	54
S3 Transfer Acceleration	54
S3 Performans Considerations – Önemi	55
S3 Billing.....	55
S3 Monitoring and Event Notification	55
Hangi Storage?	56
AWS Route 53	57
Route53 - Name Servers	57
Route53 Konfigürasyonu	58
Supported DNS Record Types	58
Alias ve Non-Alias.....	59
Route53 Routing Policies	59
AWS CloudFront (Content Delivery Network)	61
Static ve Dynamic Content.....	61
AWS CloudFront Edge ve Regional Edge Cache	62
Signed URL ve Signed Cookie	62
AWS CloudFront Distributions	62
AWS Services.....	63
ElastiCache	63

ElastiCache for Memcached.....	64
ElastiCache for Redis	64
AWS API Gateway	65
Amazon API Gateway Throttling.....	67
AWS Lambda	67
AWS Lambda Limits.....	69
AWS Lambda Monitoring and Maintenance.....	69
Unified CloudWatch.....	70
Cloud Watch Agent Log.....	70
CloudWatch Logs Insights	71
AWS Lambda@Edge.....	71
AWS Redshift	72
AWS Redshift Backup/Restore ve Monitoring.....	73
AWS Redshift HA, Data Durability, Scaling ve Billing	73
Redshift WLM (Workload Management).....	73
AWS Services Kinesis.....	74
Kinesis Streams	74
Kinesis Data Firehose	75
Kinesis Analytics	76
AWS Services Simple Queue Service (SQS)	77
SQS Polling types and SQS Timers.....	78
SQS Message Lifecycle	78
SQS Limits, Queue Names ve Logging.....	79
AWS Services DynamoDB.....	79
Amazon DynamoDB Accelerator (DAX).....	81
DynamoDB Best Practices:.....	82
DynamoDB Streams	82
AWS EMR (Elastic MapReduce):	82
AWS Services EC2 (Elastic) Contanier Service (ECS).....	82
Kubernetes	83
Amazon Elastic Container Service (ECS);.....	83
ECS Launch Tipleri:	85
ECS Task Tanımı.....	86
ECS ve IAM role;.....	87
AWS Active Directory Services.....	87
AWS Microsoft Active Directory	87

AWS Simple Ad;	88
AD Connector.....	89
AWS CloudFormation.....	90
Amazon Elasticsearch	91
Application Load Balancer (ALB)	91
AWS CLB Listeners.....	91
ECS Service	91
Content-Based Routing,***	92
ALB - Containers and Microservices Support.....	93
ALB vs CLB	94
ALB Monitoring	95
ALB Limits.....	95
CLB'den ALB'ye Migration Avantajlari.....	95
Network Load Balancer (NLB).....	96
Supported Target Types;.....	96
AWS IAM.....	97
IAM Elements.....	98
Identity Federation	99
Amazon Cognito.....	100
AWS Security Token Service (STS).....	101
Web Identity Federation.....	101
AWS STS API Actions and Permissions	102
AIM Use cases;	102
Amazon X-Ray	103
AWS Config.....	103
AWS Glue ve AWS Athena	104
AWS Resource Access Manager (RAM).....	104
AWS Budget:	105
AWS Cost Explorer:	105
AWS Cost Allocation Tags	105
Bring Your Own IP Addresses (BYOIP).....	105
A Route Origin Authorization (ROA)	105
VCP Flow Logs	105
Amazon Aurora	105
Amazon EFS.....	106
AWS Secret Manager ve AWS Systems Manager Parameter	107

Amazon MQ (Managed message broker service for Apache ActiveMQ)	108
AWS Shield	108
AWS Web Application Firewall (WAF)	109
Perfect Forward Secrecy	109
AWS Snowball	109
AWS Snowball Edge	109
AWS Shared Responsibility Model.....	110
Decoupled Architecture	110
AWS Pilot Light.....	111
Hybrid Cloud Architectures with AWS	112
AWS Elastic Beanstalk	112
AWS Storage Gateway	113
AWS Storage Gateway Cached Volumes	113
AWS Step Functions	113
Amazon Resource Names (ARNs) ve AWS Service Namespaces	114
AWS Data Pipeline	114
AWS Certificate Manager.....	115
Origin Access Identity (OIA)	115
Identity Broker Application.....	116
AWS OpsWorks	116
AWS CloudHSM (Hardware Security Module)	116
VPC Endpoints.....	117
Multicast Network Capability	118
AWS Organizations.....	118
AWS CodeDeploy	118
Egress-Only Internet Gateway	118
Blue-Green Deployment	119

AWS VPC (Virtual Private Cloud) Architecture



Yukarıda bir VPC örneği bulunmaktadır. Bir VPC'de iki ayrı availability zone bulunabilir. Bu AZ'lar ayrı subnet'a sahip olacaklardır.

İki zone arasında yukarıdaki gibi router hazır gelecektir ve route table üzerinde yapılacak tanım ile bu iki zone arasında güvenli bağlantı sağlanacaktır.

Route çıkışında bulunan Gateway'de dış dünya ile olan erişim sağlayacaktır.

Default VPC'de, main route table private subnet'e aittir. Custom route table'da public subnet'e aittir.

Custom VPC'de ise bu tam tersidir. Main route table public subnet'e aittir ve custom route table private subnet'e aittir.

Default VPC'de, AWS public ve private DNS hostname sağlar.

Non-default VPC'de, sadece private DNS hostname gelir. Bu VPC'ye yeni instance eklenirse, otomatik olarak DNS hostname almaz çünkü DNS resolution ve DNS hostname enable değildir.

Route Table:

- Bir VPC'de 200'e kadar route table oluşturulabilir.
- Bir route table'a, 50'ye kadar route girilebilir.
- Bir subnet'in sadece bir tane route table'ı olabilir.

Subnets:

- Bir subnet sadece bir tane availability zone'a atanabilir.

Soru: Aynı anda bir subnet, birden fazla route table'a atanabilir mi?

Hayır, atanamaz.

Soru: Aynı anda bir route table, birden fazla subnet'e atanabilir mi?

Evet, atanabilir.

Bu soruların anlatmak istediği, bir tane route table birden fazla subnet'e atanabilir ama bir subnetin sadece bir tane route tablosu olabilir.

Bir zone, A route table'a bakarken, B tablosuna switch edilebilir. Eğer hiç bir tabloya atanmaz ise, ana tabloya atanacaktır.

1. Main route table'a, satır eklenebilir, editlenebilir ama bu tablo silinemez.
2. Custom route table, main route table yapılabılır ve bu işleminden sonra bir sonraki main route table silinebilir.

Edit routes

Destination	Target	Status	Propagated	
172.31.0.0/16	local	active	No	
0.0.0.0/0	igw-688c5500	active	No	×
10.0.0.0/24	pcx-07e20766ee7f1fe80	active	No	×

3. Yukarıdaki main route tablosunda bulunan ilk satır, VPC içerisindeki bütün subnet'lerin birbileri ile haberleşmesini sağlar ve bu satırı editleyemeyiz ve silemeyez.

VPC bir kere oluştuktan sonra primary CIDR (**Classes Inter-Domain Routing**) değiştirilemez.

Full block range yani IP address length 32'dir. Bit tanımı max 28, min ise 16 olabilir.

Örnek: 10.0.0.0/28 tanımlarsak, bunun anlamı 28'i subnetler ve available instance'lar tarafından alınır. Geriye kalan 4 bit anlamı da 2^4 yani 16 ip adresin assign olduğudur.

Örneğimizi 28 değil de 16 olarak düşünürsek, bu durumda yarısı network ve diğer yarısı network içinde bulunan instance'lar için olduğu anlamını taşır.

28 yanlış bir seçim olacaktır. Bu seçim sadece account websitesi için kullanılacaksa olabilir. Ama bu da sonradan değiştirilemeyeceğinden, ilerleyen zamanlarda soruna neden olabilir.

Sonradan değiştirmenin tek yolu, yeni bir VPC oluşturmak ve migrate etmektir.

24 bit tanım olduğunu düşünelim. $32-24=8$.

$2^8=256$ ip kalacaktır.

10.0.0.0 incelersek, burada 10.0.0.4'den 10.0.0.254'e kadar toplam $256-5=251$ ip verilebilir. Aşağıdaki ip'ler atanamazlar. Default olarak aşağıdaki şekilde tahsis edileceklerdir.

10.0.0.0 Base Network

10.0.0.1 VPC router

10.0.0.2 DNS related

10.0.0.3 Reserved for future use

10.0.0.255 Last IP

VPC içerisindeki subnet'ler basic TCP/IP kuralı gereği, overlap (çakışma durumu) olamazlar.

VPC Types

Default VPC: AWS account'u oluştugu zaman, her AWS region'da default olarak olur.

CIDR, Security Group, N.ACL, Route Table ve Internet Gateway default olarak olur.

Custom VPC: Kullanıcı kendi oluşturur ve CIDR'a kendi karar verebilir.

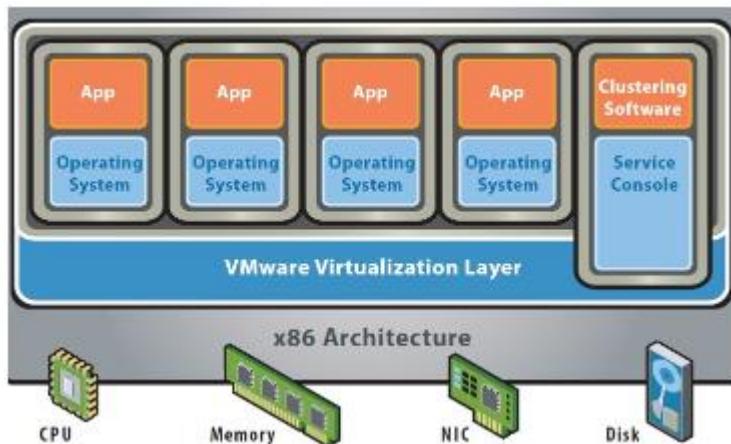
Security Group, N.ACL ve Route Table default olarak gelir.

Internet Gateway default olarak gelmez.

Hypervisor

AWS mantığında arka planda çalışan fiziksel sunucu, CPU, memory, NIC (Network Card-AWS'de ENI denir. Elastic Network Interface) ve Disk vardır.

Bare-Metal (Hypervisor) Architecture

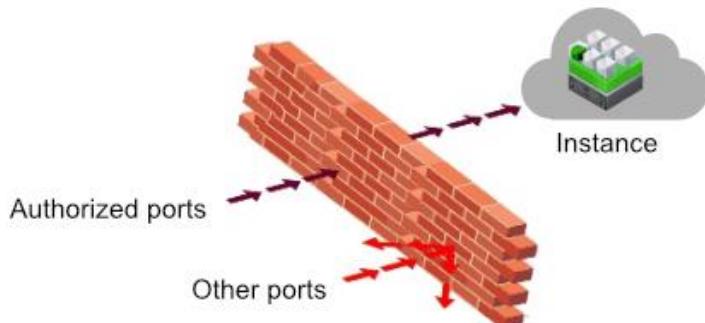


Bunun üzerinde yukarıdaki gibi virtualization layer veya hypervisor vardır. Bu alan fiziksel kaynakları sanallaştırarak bir üst katmanda oluşan işletim sistemine atar.

Hypervisor çoklu işletim sistemlerinin aynı domain üzerinde çalışmasını sağlayan bir kod parçasıdır.

Sanallaştırma sayısı, fiziksel kaynağın sayısı ve XEN limitation'a bağlıdır.

Security Groups



- Security Group virtual firewall'dur.
- Virtual server seviyesinde, network trafiğini kontrol eder.
- Stateful'dur. Yani durum bilgisi var demektir.
- Her EC2 instance için en fazla 5 tane security group tanımı yapılabilir.
- Sadece allow rule vardır. Deny rule olmaz.
- Security groups ends with an implicit deny all
- Network interface level'da çalışır.
- Security group'da yapılan değişiklikler anında etki eder.
- Bütün outbound trafic izin verilmiş durumdadır. Inbound trafiği kurallara bağlı olarak çalışır.
- Security Group'da yapılacak 110.238.98.71/32 tanımı sadece belirtilen IP için tanım anlamına gelir.
 - Tanım 110.238.98.71/0 şeklinde yapılacak olursa, bu bütün network'u refer eder.

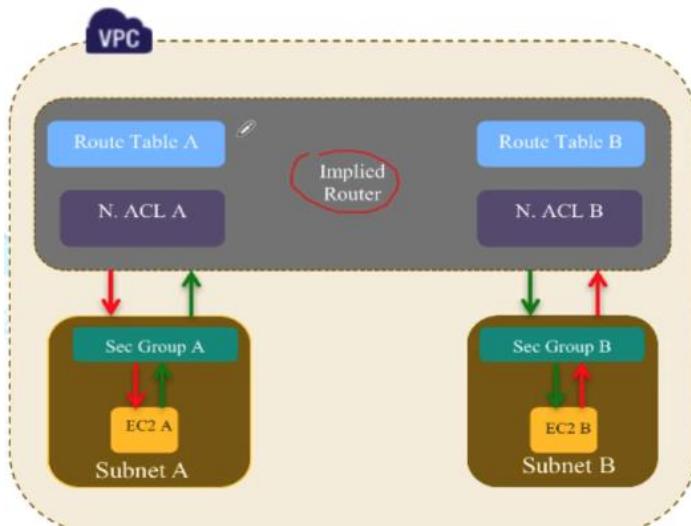
Bir tane Web ve DB sunucusu olduğunu düşünelim. Web sunucusundan çıkan talep, Web sunucusu için outbound'dur ve bu talep DB sunucusu için inbound'dur.

Inbound ve Outbound security group'ları directional'dır ve direction'a bağlı olarak etki sağlar (have an impact). Bu yüzden Stateful'dur.

Aynı VPC içerisindeki iki AZ'dan birisi birine ping atıyor ve diğeri atamıyorsa,

Unutma, outbound traffic her zaman açıktır. B, A'ya ping atamıyorsa; Sorun B'de değildir. Sorun ya A'nın inbound kurallarında yada N.ACL'de dir.

Network ACL



Security Group güvenliğin ilk kapısıdır. ACL ise ikinci kapısıdır.

- ACL subnet leveldir.
- Her subnet'de ACL tanımlıdır ve custom ACL'de tanımlanabilir.
- Custom ACL'de inbound ve outbound bütün trafik default kapalıdır.
- ACL stateless'dır. Implicit değildir. Bir trafiğin açık olması içim tanımlanması gerekmektedir.
- Permit ve deny rule tanımlanabilir.
- Belirli bir ip bloğunu engellemeye yardımcı olabilir.

Bir VPC içerisinde yer alan bir instance aynı VPC içerisinde yer alan başka bir instance'a bağlanamıyor ise,

- Source instance'in, ACL'in de olabilir.
- Destination instance'in, security group veya ACL'in de olabilir.
- Problem routing table konfigürasyonunda olmaz.
- **NACL kuralları uygularken, kural sayısına göre, en düşükten en yükseğe doğru değerlendirir ve eşleşen allow/deny durumunu görürse, daha sonra yer alan kurallara baktırmaya gerek duymadan uygular.**

Unutmayın, ACL stateless'dır. Belirli bir akış için, gelen ve giden trafiğe izin verilmesi gerektiğini unutmayın.

Gelen trafik demek, dışarıdan subnet'e gelen trafiktir. Giden trafik ise subnet'den çıkan trafiktir.

N.ACL vs Security Groups

Security Group	Network ACL
Instance leveldir.	Subnet leveldir.
Sadece allow rule vardır.	Allow ve deny rule vardır.
Stateful: Dönüş trafiği bir tanıma ihtiyaç duymadan allowed'dur.	Stateless: Dönüş trafiği kural tanımlanarak izin verilmelidir.
We evaluate all rules before deciding whether to allow traffic. Trafikte izin verilip verilmeyeceğine karar vermeden önce tüm kuralları değerlendiriyoruz.	We process rules in number order when deciding whether to allow traffic. Trafikte izin verilip verilmeyeceğine karar verirken kuralları sırayla işleriz.
Bütün outbound traffic izin verilmiş durumdadır.	

Soru: EC2A instance'ı, EC2B instance'ına ping atabiliyor ancak EC2B, EC2A instance'ına ping atamıyor.

Sorun hangi katmanda olabilir?



Ping geri dönüşü olması gereken bir işlemdir. B'den A'ya olan 2 Sec Group ve 2 ACL engel olabilir ve ayrıca A'dan B'ye 2 ACL'de engelliyor olabilir. Toplam 6 olasılık vardır.

Security groups stateful'dur bunun anlamı, gelen request için izin var ise yani inbound rule var ise, dışarı çıkışacak trafik de otomatik olarak izin verilir.

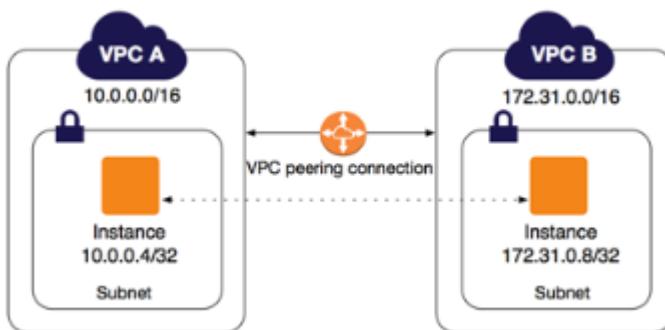
Evdən ssh ile bir instance'a bağlanmak isteyelim ve hem Security Group hem de NACL tarafında inbound trafığın giriş izni olsun yani "allow all inbound traffic" seçilmiş olsun.

NACL için outbound trafik girişi olmasın yani deny all olsun ve instance Security Group'da outbound için bir tanım girilmemiş olsun.

Bu gibi bir senaryoda, bağlantı başarılı olmayacağından emin olmak için traffic'ı içeri girebilir ama NACL nedeni ile SSH talebi geri çekilecektir ve bağlantı başarısız olur.

VPC Peering

VPC peering, private IPv4 veya private IPv6 adresini kullanarak, iki VPC arasında trafik yönlendirmesini sağlayan bir network bağlantısıdır.

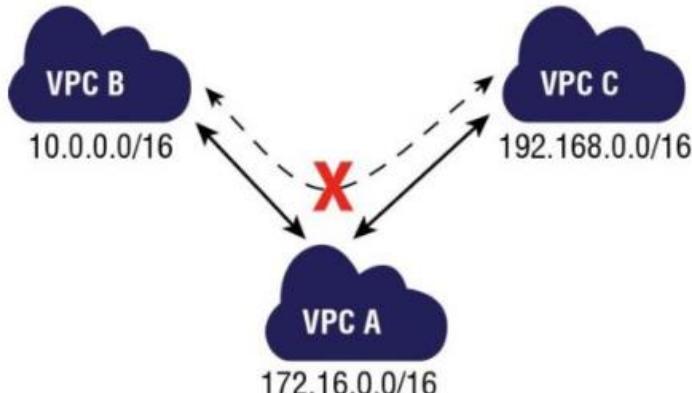


Her iki VPC'de bulunan instance'lar, aynı network'de gibi birbirleri ile haberleşebilirler. Peering connection aynı veya başka bir AWS account'ına ait VPC için yapılmabilir. Communication için single point of failure veya herhangi bir bandwidth bottleneck yoktur.

Peering farklı region'larda olan VPC'ler arasında da kurulabilir. Buna Inter-Region VPC peering denir. Inter-Region VPC peering şu an için Çin'de bulunan region için desteklenmemektedir.

VPC Peering Connection aşağıdaki durumlar için geçerli değildir.

- Overlapping CIDR blokları için VPC peering yapılamaz.
- Gateway veya private bağlantı yoluyla, edge to edge routing desteklenmez.
- Transitive Peering yani geçişli değildir.



VPC-A ile VPC-B arası bir peering olsun ve bir de VPC-A ile VPC-C arasında bir peering olsun.

Bu iki peering üzerinden VPC-B ile VPC-C haberleşemez (**Transitive Peering**). Bu VPC'ler arasında da bağlantı isteniyorsa, ayrıca kurulmalıdır.

Elastic Compute Cloud (EC2)

- EC2 servisleri cloud üzerinde compute kapasitesini ayarlamayı sağlar.
- EC2 availability SLA %99.95'dir ve bu ayda 22 dakikaya denk gelmektedir.
- Arkada çalışan fiziksel sunucuyu shared veya dedicated olarak seçmek mümkündür.
- Instance'a erişebilmek için key ve key pair adı gerekmektedir.
- Her account 20 tane EC2 instance oluşturabilir ama bu limit soft limittir. AWS ile irtibata geçilerek arttırılabilir.
- EC2 instance arkasında fiziksel bir host vardır. Instance durdurulursa, AWS genellikle instance'ı yeni bir host'a taşıır.
- **AWS Console'dan kullanılabilecek RUN Command ile EC2 instance'larına RDP ve SSH yapmaya gerek kalmadan, yapılandırmak için kullanabiliriz.**

Classic EC2, EC2 instance'i her başlatıldığında EC2-Classic serisinden private bir IPv4 adresi alır.

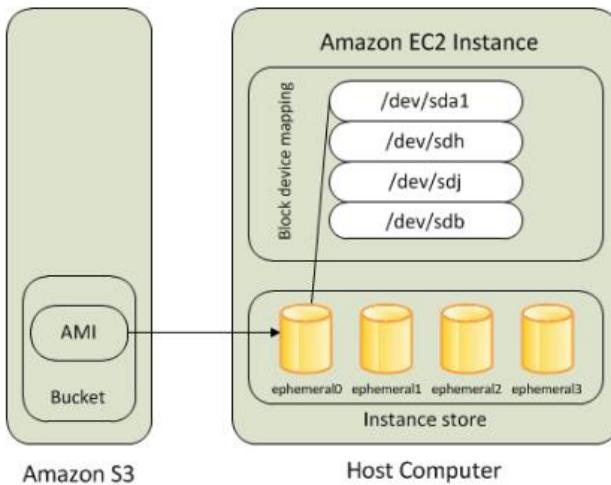
VPC içerisinde yer alan EC2 instance ise, default VPC adres range'inden **statik bir private IPv4 adresi alır** ve EC2 instance restart olduğu zaman bu IP adresi değişmez. Bu nedenle IP değişimi istenmiyorsa, VPC seçilmelidir.

Nondefault VPC'de static private IPv4 adresi alır.

EC2'de 2 çeşit block store device support etmektedir.

Instance Types

- **General Purpose Instances**
 - Compute, Memory ve Network kaynaklarının dengeli şekilde konfigüre edilmesini sağlar ve genel kullanım için uygundur.
- **Compute Optimized Instances**
 - Batch processing ve media trascoding gibi yüksek Compute ihtiyacı olan instance'lar için uygundur.
- **Memory Optimized Instances**
 - Memory'de ki çok büyük data setlerinin işlenmesi ihtiyacı olan instance'lar için uygundur.
- **Storage Optimized Instances**
 - Local storage'da, çok büyük data setlerine yapılacak yüksek oranda sequential read ve write işlemleri için uygundur.



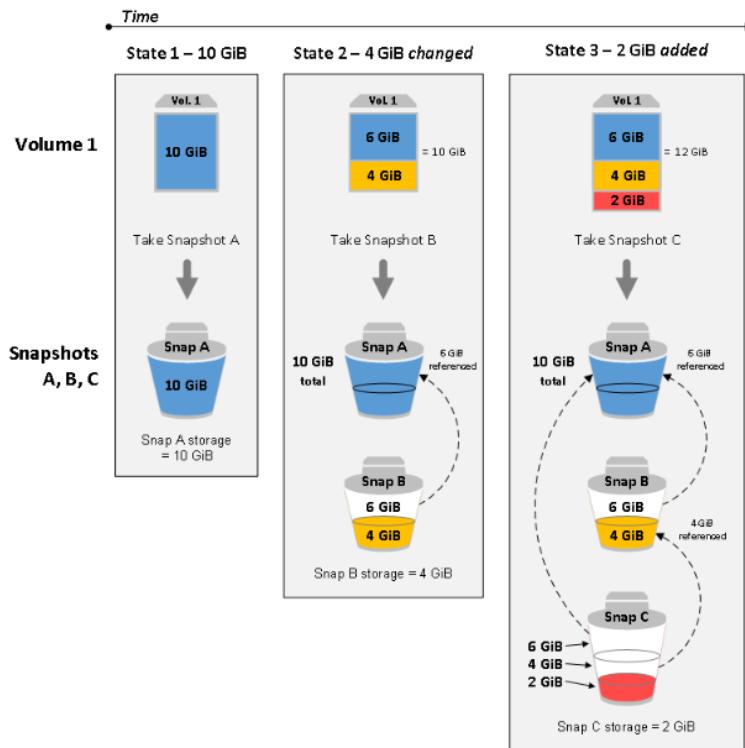
Elastic Block Store (EBS)

- Persistent yani kalıcıdır.
- Network attached virtual drive destekler.
- %99.999 availability
- Manual veya automatic point time recovery yapılabilir
- **Bir AZ'da EBS oluşturulduğunda, donanım bileşenlerinin arızalanmasına önlem almak amacıyla, aynı AZ içerisinde otomatik replicate edilir.**
- Aynı anda sadece bir tane EC2 instance'ına attach olabilir. **Paylaşımılı olarak kullanılmaz.**
- EC2 terminate edilirken, EBS terminate edilmemesi tercih edilebilir.
- Volume type, volume size, IOPS kapasitesi herhangi bir servis kesintisine gerek olmadan değiştirilebilir.
- AES-256 encryption destekler.
- EC2 monitor edilmek isteniyor ve CloudWatch dışında bir alternatif aranıyorsa, Simple Email Service (SES) yerine Simple Notification Service (SNS) tercih edilmelidir.
- **Amazon EBS snapshot; oluşturma, retention belirleme, silme gibi işler için, Amazon DLM (Data Lifecycle Manager) kullanılabilir.**

EC2 instance'ına internetten erişilmek isteniyorsa üç koşul sağlanmalıdır.

- VPC'ye attach olmuş bir Internet Gateway (IGW)
- VPC Route Table'a, Internet Gateway için satır giriş'i
- EC2 instance'a attach olmuş Public IP

EBS'de Incremental snapshot alınabilir ve bu sırada volume normal şekilde kullanılmaya devam edilebilir. Bunu 3 aşamalı olarak düşünebiliriz.



- 10 GB volume düşünelim ve ilk snapshot alınırken 10gb verinin tamamı kopyalanacak
- 2. aşamada, volume hala 10gb veri barındırıyor ancak bunun 4GB değişmiş olsun. Snap B bu 4GB alacak.
- 3. aşamada 2GB data daha volume'a eklenmiş olsun ve alan toplam 12GB olmuş olsun. Snap C, Snap B alındıktan sonra eklenen 2 GB'ı kopyalayacaktır.
- Snap C,Snap B'de depolanan 4 GB veriyi ve Snap A'da depolanan 6 GB veriyi barındırır.

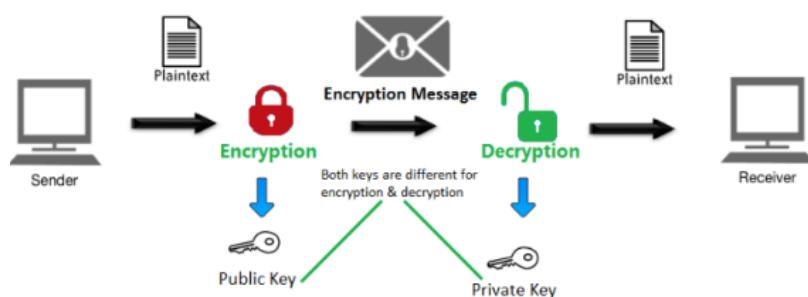
Instance RAID konfigürasyonu kullanıyor ve snapshot alınıyorsa, süreç biraz daha farklı işler. Snapshot'ın consistent olması için, alınmadan önce **bütün I/O aktiviteleri durdurulmalı ve diskteki bütün cache flush edilmelidir**.

Key Pair: EC2 giriş bilgilerini şifrelemek ve şifresini çözmek için public-key şifrelemesi kullanılır. Public-key şifrelemesi, password gibi bir veri parçasını şifrelemek için bir public-key kullanır ve ardından alıcı verilerin şifresini çözmek için privaye-key kullanır.

Public-key ve private-key, **key pair** olarak bilinir.

Linux'da, public key `~/.ssh/authorized_keys` altında bulunmaktadır.

Bir kullanıcıya şifre yerine privaye key ile bağlanması sağlanarak, güvenli şekilde sisteme bağlanması sağlanmış olur.



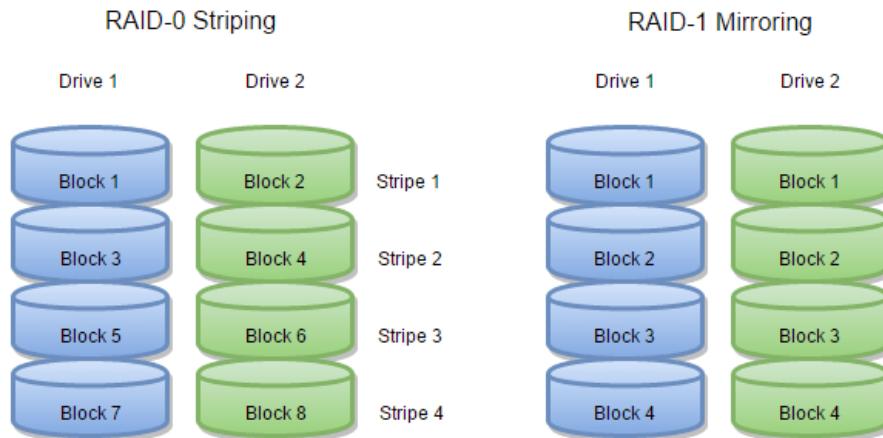
• Instance-Store

- Virtual hard drive, EC2 instance için allocate edilmiş oluyor.
- Device başına 10Gbit limit vardır.

Root ve boot volume her ikisi de olabilir. EBS-Backed yani EBS destekli root volume var ise root volume EBS demektir. Instance-Store backed yani instance-store destekli EC2 var ise root volume Instance-Store demektir.

Instance-Store ephemeral storage'dır yani geçici storage'dır. Instance durdurulur veya terminate edilirse, bu storage'da tutulan veriler kaybedilir. EBS'de ise, instance terminate edilse dahi veri saklanır.

Amazon EBS'de, işletim sistemi support verdiği sürece, herhangi bir standart RAID konfigürasyonu kullanılabilir.



RAID-0'da mirroring yoktur. RAID-1 kullanılırsa, her block için mirroring sağlanmış olur. Daha fazla throughput isteniyorsa, EBS volume'u artırmak bir seçenek olabilir.

Elastic Block Store Types

Block store device'a örnek; CD driver, hard disk, SSD gibidir.

- SSD-Backed'dir.
- Küçük veritabanları, Dev/Test ortamları, low latency apps, IOPs gerekli yani transactional workload'lar için daha iyi performans gösterir.
- 1 Tib-16 Tib (Tebibyte) arası volume size destekler ama bu değişebilir.

1 Tetabyte= 1.09951 Terabyte

- Max IOPS, 1000 IOPS'dur.

Provisioned IOPS

- SSD-Backed'dir
- Mission critial uygulamalar ve I/O bağımlı SQL/NoSQL veritabanları içindir.
- 4 Tib-16 Tib volume size destekler ama bu değişebilir.
- GB başına 50 IOPS sağlamaktadır. 10GB volume için, 500 IOPS sağlar.
 - 640GB ve üstü volume'lar için en fazla 32.000 IOPS sağlamaktadır.

Throughput Optimized HDD (not SSD)

- Streaming, Big Data, log processing and DWH sistemler için uygundur.
- Boot olarak kullanılmaz.
- Size 500GB ve 16Tib arasında değişebilir.

Cold HDD

- Daha az ulaşılan veriler içindir.
- Boot olarak kullanılmaz
- Size 500GB ve 16Tib arasında değişebilir.

Detay icin: https://aws.amazon.com/ebs/features/?nc1=h_ls

FEATURES	SSD Solid State Drive	HDD Hard Disk Drive
Best for workloads with:	small, random I/O operations	large, sequential I/O operations
Can be used as a bootable volume?	Yes	No
Suitable Use Cases	<ul style="list-style-type: none"> - Best for transactional workloads - Critical business applications that require sustained IOPS performance - Large database workloads such as MongoDB, Oracle, Microsoft SQL Server and many others... 	<ul style="list-style-type: none"> - Best for large streaming workloads requiring consistent, fast throughput at a low price - Big data, Data warehouses, Log processing - Throughput-oriented storage for large volumes of data that is infrequently accessed
Cost	moderate / high 	low 
Dominant Performance Attribute	IOPS	Throughput (MiB/s)

Block Device Mapping

Block device mapping, AMI'da (Amazon Machine Images) kaç tane volume, bu volume'ların tipleri ve bunların EBS veya Instance-Store olduğu bilgisini tutar.

- Hem EBS hem de Instance-Store volume bilgisini tutar.
- AMI ile başlatılan bir instance'da, hangi block storage volume (root ve data) ekleneceğini/oluşturulacağını tanımlamak için kullanılır.
- AWS console'da, EC2 instance block device mapping sadece instance'ın EBS volume'larını gösterir. Instance-Store volume görmek için curl komutu ile bakılmalı.

curl <http://169.254.169.254/latest/meta-data/block-device-mapping>

- Instance başlarken veya başladıkten sonra block device mapping'de değişiklik yapılabilir. Ama bu değişiklik EBS için geçerlidir. EBS volume ekleyebilir ve çıkarabiliriz. Ancak Instance-Store volume ekleyemeyiz. Instance-Store volume sadece instance oluşturulurken oluşturulabilir.

Sınırlandırmaları

- Root volume size sadece artırılabilir ve type modify edilebilir.
- Volume size, belirtilmiş snapshot boyutu ile ya aynı yada daha fazla ayarlanabilir.

EC2-SR-I/O

SR-I/OV, "Single Root I/O Virtualization" baş harflerinden oluşur. Tek bir PCI device'ın birden fazla virtual machine tarafından paylaşılmasına izin verir.

- Saniye başına data fazla package transferi sağlar.
- Düşük latency sağlar.
- Çok düşük network delay sağlar.

EC2 Placement Groups

Düşük network latency gerektiren bir compute cluster kümesi için performansı optimize etmek için Placement Group tavsiye edilir. Placement group içerisinde yer alan EC2 instance'ları arası high network throughput sağlanmaktadır.

- Aynı AZ veya farklı AZ'da bulunan logical group yani cluster üyelerinin; low latency, high network throughput yaparak haberleşmesini sağlar.
- İki farklı şekilde cluster oluşturulabilir.
 - **Cluster:** Single AZ içerisinde low latency
 - **Spread(yayılmış):** Multiple AZ işlem görür.
- Placement group oluşturmak ücretli değildir.

Placement Group'da ihtiyaç duyulan instance sayısını tek bir launch işlemi ile yapılması ve tüm instance'lar için aynı tür seçilmesi tavsiye edilir.

İlk Placement Group oluştuktan sonra daha fazla instance eklemek istediği zaman "**insufficient capacity error**" hatası alınır. Placement Group'da yer alan instance'ları durdurup yeniden başlatmak, istenilen tüm instance'lar için o kapasiteye sahip donanıma geçmesine sağlayabilir.

Cluster Placement Groups

- Tek bir AZ'da yer alır.
- Uygulama low latency, high network throughput veya her ikiside ihtiyaç halinde kullanılır.
- SR-I/OV network destekleyen instance type'ları için low latency, saniye başına en yüksek paket network performansı için idealdir.

Spread Placement Groups

- Her biri farklı donanımlara sahip instance grubudur.
- Çok kritik olmayan uygulamaların birbirlerinden ayrı tutulması için tavsiye edilir
- Eş zamanlı olabilecek riskleri düşürür.
- Birden fazla AZ yer alabilir.
- Bir placement group'a ait bir AZ'da en fazla 7 çalışan instance olabilir.

EC2 Monitoring

- EC2 metric datalarını default olarak her 5 dakikada bir AWS CloudWatch'a gönderir.
- Detailed monitoring enable edilebilir ve veri gönderimi her 1 dk indirilebilir ama bu ücretlidir.
- CloudWatch ile, stop, restart, terminate ve recover olarak alarm action tanımlanabilir.
- Maaliyeti azaltmak için sadece stop ve terminate kullanılabilir.
- Örneğin bir iş başladı ve CPU kullanımı çok arttı. İş bittiği zaman CPU kullanımımız azalacaktır. CloudWatch'u bu şekilde ayarlarak, CPU kullanımız azaldığı zaman (%5 düştüğünde) tetikleyebiliriz. Bundan sonra da instance termimate edilebilir ve maaliyet azaltılmış olur.

- CloudWatch tarafından uygulamalarınızı izleyebilmemiz, sistem geneli performans değişikliklerini anlayıp bunlara yanıt verebilmemiz, kaynak kullanımını optimize edebilmeniz ve çalışma durumunun birleşik bir görünümüne sahip olabilmeniz için veriler ve eyleme dönüştürülebilir öngörüler sağlanır.
- EC2 instance'i stop konumuna alınır ve ona bağlı olan EBS data veya root dizini var ise, EC2 üzerinden bir faturalandırma olmaz ama EBS volume üzerinden taksimetre devam eder. Tabii burada data transferi olmadığından, ücret daha düşük olacaktır. EC2 kapalı olsa bile EBS üzerinden detach, attach, modify gibi işler yapılabilir.
- EBS-Backed (root volume EBS) olan instance kapanırsa; private Ipv4 korunur, public Ipv4 release olur, elastic IP adres var ise korunur. Kullanılmayan elastic IP üzerinden faturalandırılmaya başlanır.

EC2 Instance Termination

- Default olarak EBS instance terminate edilirse, root volume'da gider. By default yes ama "**DeleteOnTermination**" bunun için değiştirilebilir.
- Termination'dan sonra, sonradan eklenmiş volume'lar, persist (devam eder).
- EBS volume'ların "DeleteOnTermination" seçeneği launch veya çalışır durumda değiştirilebilir.

EC2 Instance Metadata and User Data

- IPv4, IPv6, DNS hostname, AMI-ID, Instnace ID, Instance type, local-hostname, public keys, security group gibi bilgilerdir.
- Sadece görülebilir ve encrypt değildir.
- http://169.254.169.254/latest/meta-data** adresinden görülebilir. EC2 üzerinden curl ve get komutu ile bakılabilir.
- User data 16kb limit vardır.
- Instance launch olurken, bash script gibi script çalıştırılarak sunucuya önceden paketler yüklenmesi sağlanabilir. Bu user data'dır.
- Sadece instance üzerine login olduktan sonra görülebilir.
- Instance stop olduktan sonra, user data değiştirilebilir. Instance/actions/Instance-Settings/view change user data
- Encrypted değildir.

EC2 Migration

- VM Import; VMWare, Microsoft, XEN VMs Cloud'a migrate edilmek için kullanılır.
- VM Export; EC2 instance'i, Microsoft, XEN VMs Cloud'a migrate edilmek için kullanılır.
 - EC2 için kullanılan export özelliği, VM import'tan gelmektedir. Native AWS EC2 instance değildir.
 - Bu özellik API veya CLI support eder. AWS console üzerinden değildir.
- VMDK veya VHD image generate edilmeden önce, VM stop statüsünde olmalı. Suspend veya paused olmaz.
- VMWare için; AWS'de VM connector adında vmware vCenter (control of vmware virtulaize enviroment) için plugin vardır.
 - Bu plugin VMs to AWS S3 migration'ı sağlar
 - EC2 AMI covertion'u sağlar.
- AMI başka region'lardan erişilebilir değildir. Başka bir region'da kullanılmak isteniyorsa, öncelikle kopyalanmalıdır.**

EC2 ve IAM Role

Örneğin bir uygulama var ve bu uygulama S3 üzerinde read ve write işlemleri yapması gerekiyor. Bunun iki çözümü vardır. Birincisi EC2 üzerine bağlantı bilgilerinin (username/password, key,...) barındırılması ki bu güvenlik için risktir.

İkinci çözüm ise, **EC2 instance'a IAM role atanır. IAM role'e policy, privileges atanır ve bu şekilde app S3 üzerinden işlem yapabilir.**

IAM rolleri, uygulamaların kullandığı güvenlik credential'larına ihtiyaç duyulmadan, güvenli olarak API request'leri yapabilmeleri için tasarlanmıştır.

Çeşitli AWS servislerinin güvenli şekilde EC2 instance'a bağlanması isteniyorsa, IAM role oluşturulmalı ve bu EC2 instance'a assign edilmelidir.

IAM Role, **global bir servistir** ve başka bir region'da var olan bir IAM role kullanılmak istenirse, o region'da yenisini oluşturmaya gerek yoktur. Mevcut IAM role'ü yeni instance'a assign etmek, yetkilerin aktarılması için yeterli olacaktır.

AMI kullanarak oluşturulan EC2 instance'ları bir DynamoDB tablosuna bağlanmak için Stored Access key kullanıyor olsun ve bu mimariyi daha güvenli hale getirmek isteyelim.

EC2 instance'larda çalışan uygulamalar için temporary credential'ları yönetmek için IAM role kullanılmalıdır. IAM role kullanıldığında, kullanıcı adı ve parola gibi uzun süreli credential'ların verilmesine gerek kalmaz. Bunun yerine IAM role geçici izinleri sağlayarak, uygulamaların diğer AWS servislerini kullanması sağlanabilir.

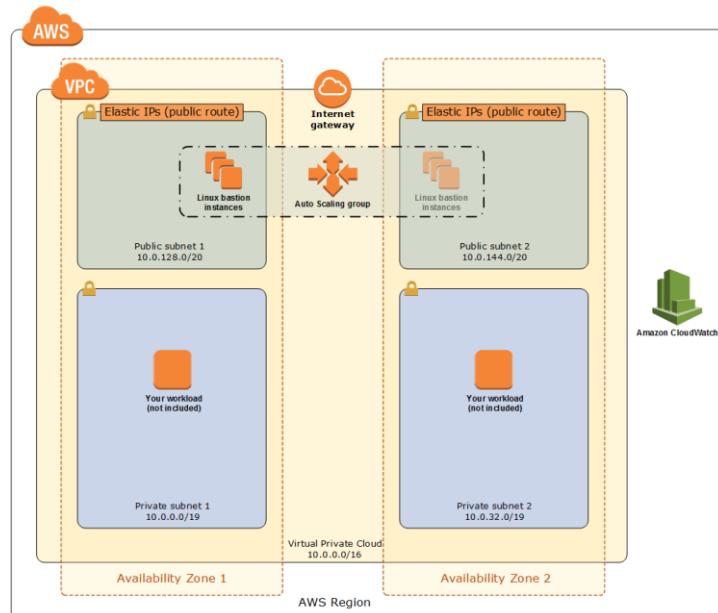
EC2 instance'a, başladığında instance ile associate edilecek bir IAM role tanımlanır. Instance'da çalışan uygulamalar, API isteklerini imzalamak için bu geçici credential'ları kullanabilirler.

Buradaki en iyi seçenek AMI'dan access key'leri kaldırırmak ve sonrasında da DynamoDB tablosuna erişme izni olan yeni bir IAM role'ü oluşturup bunu EC2 instance'a atamaktır.

EC2 Bastion Hosts

- Bastion hosts linux için ssh, Windows için RDP'dir.
- Inbound; VPC'de bulunan public ve private EC2 instance'ları secure bağlantı yapılmasını sağlar.
- Bastion hostlar'da, auto-assign public IP veya elastic IP adresi vardır. Security için elastic ip tavsiye edilir.
- Security group kullanarak hangi IP CIDR ların bastion host'a erişebileceği belirlenebilir.
- Bastion host high availability'dir ve auto scaling group'da kullanılabilir.
- 2 kapasiteli ASG (auto scaling group) tanımlanıp, multiple AZ seçip, her biri için birer elastic IP kullanılmalıdır.

Best practice bu şekilde bir bastion host giderse, hiç bir şey değişmeden işlem devam eder. Bu sırada yeni bir bastion ayağa kaldırılması 3-5 dk sonra tamamlanacaktır. Örnek model aşağıdaki gibidir.



- ASG kaç tanımlanırsa, bu sayıya bağlı kalmaya çalışacaktır.
- Bastion host'a tanımlanan Elastic IP addresses ile HQ firewall üzerinden tanım yapmak çok daha kolaydır. Değişken değildir.

- ASG'de tanımlanan instance'lardan birisi giderse, Auto Scaling Group yeni bir instance başlatır ve Elastic ip adresi yeni instance'a attach olur.

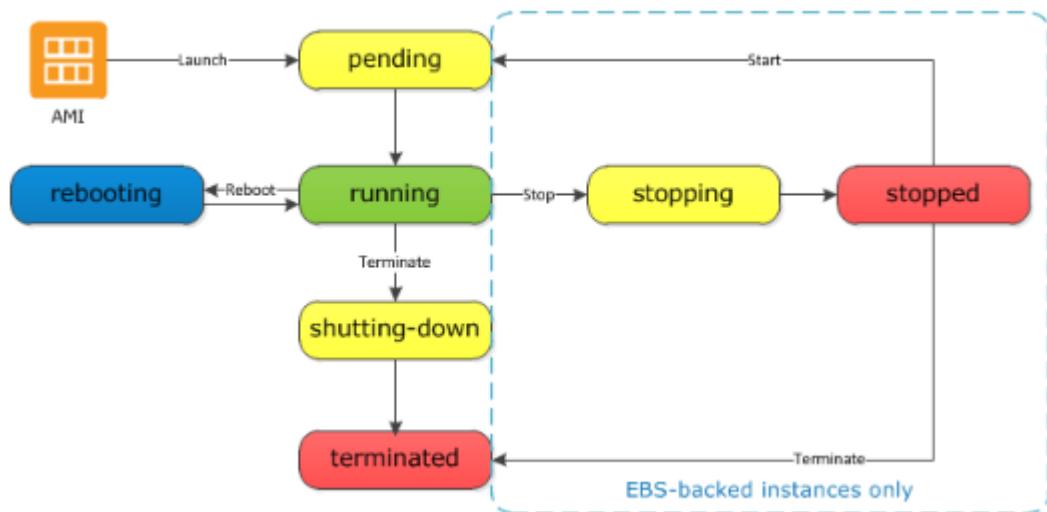
EC2 ENI (Elastic Network Interfaces)

- Eth0 primary network interface'dir Move veya detach edilemez.
- Instance family tipine göre daha fazla network interface eklenebilir.
- Bir ENI bir AZ bağlıdır.
- Instance çalışırken attach olan ENI hot attach denir.
- Instance durmuşken attach olan ENI warm attach denir.
- Instance launched statüsünde ise cold attach denir.

EC2 ENI IP Addressing

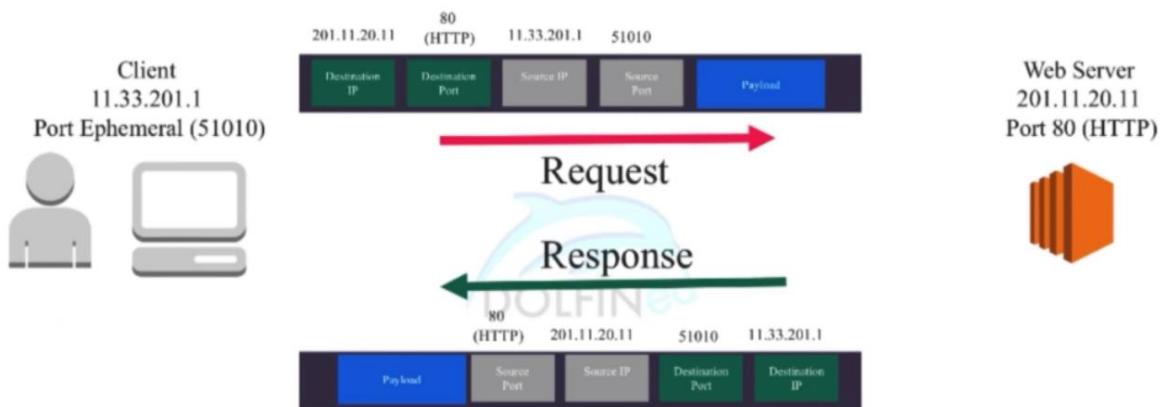
- Her private IPv4 ip adresi sadece bir tane Elastic IP'ye associate olabilir.
- To attach a network interface in a subnet to EC2 instance in another subnet, they both must be in the same AWS region and same AZ

EC2 lifecycle Instance State'leri;



- Pending,**
 - Instance running duruma geçmek için bekliyor.
- Running,**
 - Instance çalışıyor.
- Stopping,**
 - Instance stop durumuna geçmek için bekliyor.
- Stopped,**
 - Instance durmuştur, restart edilebilir.
- Terminated,**
 - Instance kalıcı olarak silinmiştir ve restart edilemez.
 - Reserved-Instance'lar bu durumda halen faturalandırılır. Faturalandırılma, görev süresi sonuna kadar devam eder.

TCP/IP Packet Walkthrough



1. Bir kullanıcı cnn.com'a bağlanmak istesin.
 2. Bunun için öncelikle web adresini yazacak ve bu web adresi dns server'a giderek, ona karşılık gelen IP adresini getirecek.
 3. IP, OS veya kullanıcı bilgisayarında cache'lenecek. Kullanıcı adresi yazıp enter'a basınca, ilk paket gidecek
 4. http (80), https(443) veya 8080 dinliyorsa, kullanıcı yazdığı adres ile doğru ip:port ikilisine gidecek
- Bu denklende request ve response kısımları vardır.

Request →
Response ←

Request Destination IP: CNN IP'si

Request Destination Port: CNN port(http veya https)

Request Source IP: Kullanıcı uygulamasına bağlı IP

Request Source Port: Kullanıcı uygulamasına bağlı port

Response Destination IP: Kullanıcı uygulamasına bağlı IP

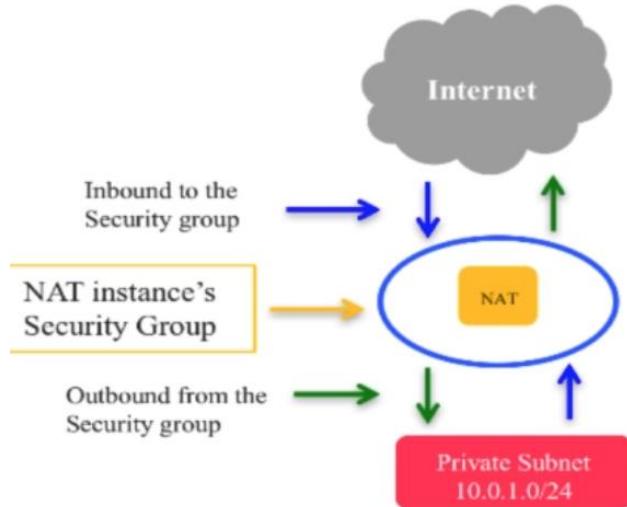
Response Destination Port: Kullanıcı uygulamasına bağlı port

Response Source IP: CNN IP'si

Response Source Port: CNN port(http veya https)

NAT Instance

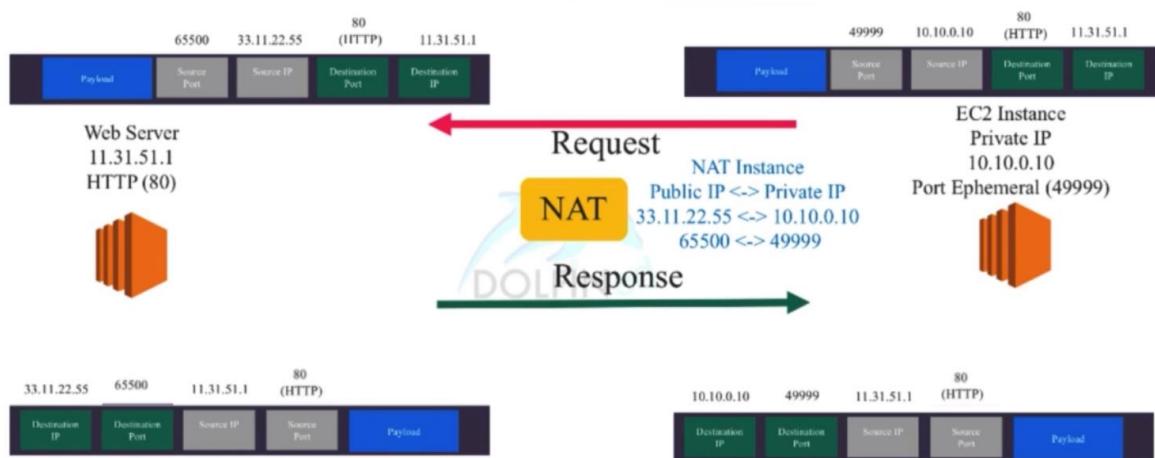
Private subnet'e sahip olan instance'ların, internet'e veya diğer AWS servislerine outbound IPv4 trafiğinin sağlanması için, public subnet'e sahip olan bir NAT instance kullanılabilir.



Aşağıdaki akışta, NAT instance kullanıldığı zaman, trafiğin nasıl olacağı anlatılıyor.

NAT source olarak EC2 instance'ı biliyor ve target web sunucusu ise source olarak NAT'ın bilgilerini biliyor. Paket EC2'den NAT instance'a geliyor ve NAT instance bu paketi kendi IP ve Port bilgisi ile gönderiyor.

Paket karşılığını alınca, web server bunu NAT instance'â gönderiyor ve NAT kendi içinde asıl kaynağın IP:Port ikilisini bildiğinden, paketi sahibine iletiyor.



Not:

```
# ssh-add -K Ec2Testing.pem
```

komutu ile pem dosyasını ssh chain'e eklemiş oluyoruz. Bu şekilde bir sunucudan başka sunucuya ssh yaparken, chain'e aldığımız dosyayı tekrar kullanabiliriz.

Chain durumu aşağıdaki komut ile görebiliriz.

```
# ssh-all -l
```

Bağlantı kurarken de aşağıdaki şekilde bağlanabiliriz.

```
# ssh -A ec2-user@ip
```

Errors and Reasons

→ Error	✗ Reason(s)	/ What to check/fix
<u>Network error: Connection timed out OR Error connecting to [instance], reason: -> Connection timed out: connect</u>	Your connection request might not be reaching the Instance, or response might not be making it back to you (basically, connectivity is broken either by networking, security, or CPU load)	<ul style="list-style-type: none">Instance Security group,subnet route table,subnet N ACL,Instance has a public or Elastic IP, Corporate network FW rules,High CPU load on your instance
Host key not found in [directory], Permission denied (publickey), or Authentication failed, permission denied	Your authentication had failed. Due to either : <ul style="list-style-type: none">Wrong username for the AMI orWrong private (.pem) key	<ul style="list-style-type: none">Verify you are using the correct user name for the AMIVerify that you are using the correct private key file (.pem) which you created or used when you created the instance
Unprotected private key file	Your private key file must be protected from read and write operations from any other users.	<ul style="list-style-type: none">Use the chmod linux command to change the permissions on the file

NOT: Aşağıdaki AWS servlerine root ile erişim vardır.

- EC2
- OpsWorks
- Elastic Beanstalk
- Elastic MapReduce

Instance-Store-backed veya EBS-backed impaired statusunda ise;

Instance-Store-backed

- Terminate the instance and launch a new one; Stop edilemez, AMI Snapshot alınarak, yenisine taşınabilir.

EBS-backed

- Stop and Restart the instance; Restart aşamasında yeni bir sunucuya impaired olur.

Not: AZ seçiminde best practice bu işi AWS'e bırakmaktadır. AWS en müsait AZ hangisi ise onu seçecektir. Aksi halde bizim seçeceğimiz AZ çok yoğun olabilir ve Insufficient Capacity Error hatası alma ihtimalimiz olabilir.

Reserved Instance

- **Reserved Instance region scope'dur ama AZ modify edilebilir.**
- Instance family spesific'dir ama bir aileden herhangi bir instance olabilir.
- Standalone instance veya auto scale olarak kullanılabilir.
- **Long term için daha iyi bir seçenekdir.**
- Satın alma süresi dolmadan geri iade olmaz ama AWS RIs marketinde teklif edilebilir.
- Ucuz değildir.
- 1 ila 3 yıl arası taahhüt verilmeldir.

Spot Instances

- **Availability garanti etmez** ama çok ucuzdur.
- Çalışacak olan uygulamanın kazara olacak olan termination'i tölgere etmezi lazımdır.

- Kullanıma başlanılması çok hızlı olmaz. Zaman var ise, spot instance iyi bir seçimdir.
- **Instance'lar, 2 dakikalık bildirimle kapasite gereksinimleri nedeniyle, Amazon EC2 tarafından kesintiye uğrayabilir ve terminate edilebilir.**

Eğer Amazon EC2 tarafından ilk 1 saat içerisinde terminate edilirse, kullanımdan herhangi bir ödeme talep edilmez. Bu termination kullanıcı tarafından yapılrsa, en yakın saniyeye kadar faturalandırılır.

Termination ilk saat geçtikten sonra Amazon EC2 tarafından yapılrsa, kullanım en yakın saniyeye kadar ücretlendirilir.

Eğer Windows üzerinde çalışiliyorsa ve kullanıcı kendi terminate ederse, bütün saat için ücretlendirilir. Spot fiyatı, belirlenen maksimum fiyatın üzerine çıkarsa ve Amazon EC2 Spot instance'ı terminate ederse, kısmi kullanım saatı için ücret alınmaz.

On-Demand Instances

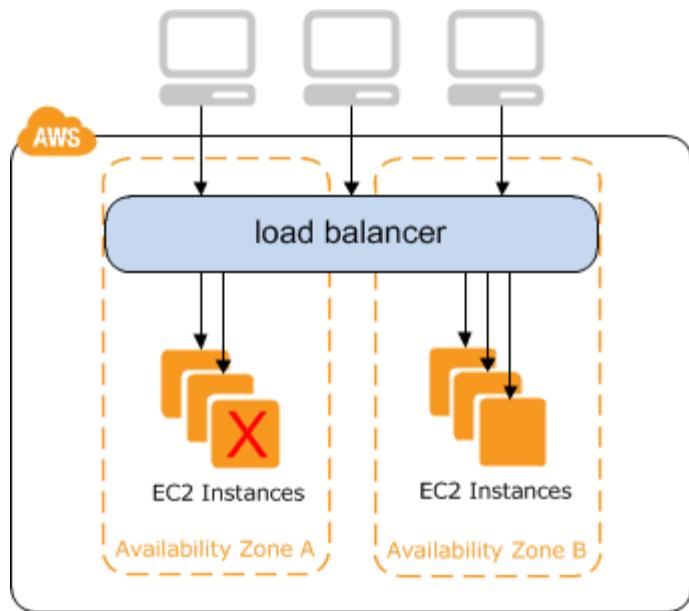
- En pahalı olandır ama kısa dönem için iyidir ve hızlı alınabilir.
- Hangi instance seçildiğine bağlı olarak, saatlik veya saniyelik fatıralandırılır ve compute kapasitesi esas alınır.
- Ön ödeme yapmamak veya uzun taahhüt olmadan kullanmayı tercih edenler için uygundur.
- Büyümesi öngöremeyen ve kesinti kabulu olmayan uygulamalar için uygundur.

Amazon EC2 Spot instance, On-Demand instance'a kıyasla indirimlerle elde edilebilecek, yedek compute kapasitesi için uygundur.

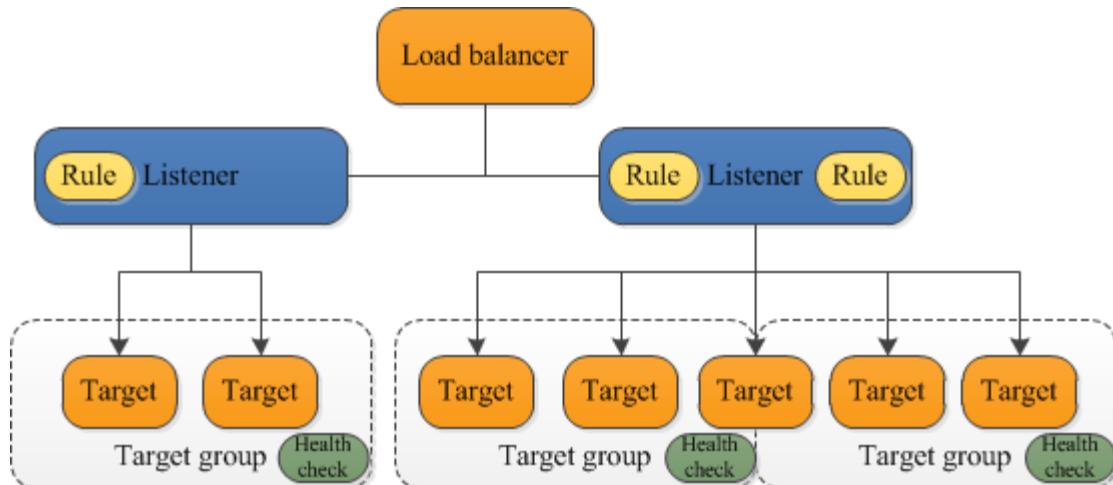
On-Demand ile Spot instance arasındaki fiyat farkı 10 katına kadar çıkabilemektedir.

Kapasite için ek olarak Spot instance tercih edilmiş ve sonrasında tekrar kapasite azaltılacak ve Spot instance cluster'dan çıkarılacak ise, EC2 instance 2 dakika kesintiye uğrayacaktır.

AWS Elastic load Balancer (ELB)



- Application load balancer, Network load balancer ve classic load balancer olmak üzere 3 çeşit load balancer vardır. Bu bölümde sadece classic load balancer üzerinde durulacaktır.
- Classic load balancer HTTP, HTTPS, TLC, SSL (TCP based) destekler, HTTP/2 desteklemez.
- ssh veya ping ile yapılacak erişimler load balancer üzerinden gelmez.
- Dışarıdan gelen HTTP, HTTPS, TLC veya SSL talepleri load balancer üzerinden gelir. İçeriden dışarıya giden aynı tür talepler yine load balancer üzerinden çıkar.
- ELB üzerinden gelen ve giden trafiği dinleyen listener'lar bulunmaktadır.



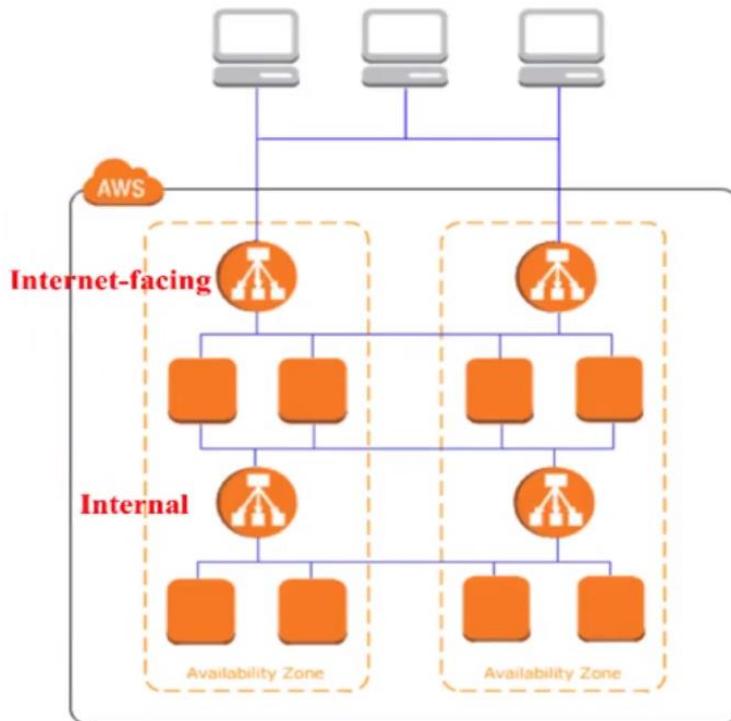
- Hangi ELB listener'in connection request'leri yani hangi protocol/port'u (TCP) dinleyeceğini konfigüre edebiliriz.
- Fronted ve backend olmak üzere iki tip listener vardır.
 - Frontend listener'lar, client'dan ELB'ye olan trafiği kontrol ederler.
 - Backend listener'lar, ELB'den EC2 instance'a giden trafiği kontrol eder.
- ELB kullanımı saatlik ücretlendirirler.
- ELB silmek, EC2 instance'ı etkilemez.
- EC2 eth0'da birden fazla ip adresi var ise, ELB trafiği primary ip adresine gönderecektir.
- ELB tag eklenebilir.
- VPC'de, ELB sadece Ipv4 destekler.

- ELB servislerinin scale olabilmeleri için, her AZ'da load balancer için tanımlanan subnet en az /27 olmalıdır. ELB için en az 8 available ip olması gerekmektedir.
 - ELB, EC2 instance'ları ile haberleşmek için bu ip adreslerini kullanır. Bu ELB security group ve NACL içindir.

ELB Health Checks

- Load Balancer aynı zamanda kendisine register olmuş olan instance'ların durumlarını da kontrol eder ve eğer bunlardan birisi sağıksız durumda ise ona trafik göndermeyi durdurur. Tekrar sağııklı olursa, tekrar trafik göndermeye devam eder.
- Sağııklı instance "In-Service" statüsünde, sağıksız instance "Out-of-Service" statüsündedir.
- AWS console, health check için HTTP yani 80 portundan request göndererek kontrolunu yapar.
- AWS API, 80 portu üzerinden TCP ping kullanır.
- Registered instance'lar, timeout süresi dolmadan, ttp "200 OK" mesajı göndermeliler.
- Timeout default 5sn ve 2-60 saniye arası ayarlanabilir.
- Healthy check interval default 30 saniyedir ve 5-300 saniye arası ayarlanabilir.
- Unhealthy threshold default 2 sn dir ve 2-10 arası ayarlanabilir. 2'den sonra instance hala unhealthy statüsünde olur ise, trafik gönderirmi durur.
- Healthy threshold default 10 sn dir ve 2-10 arası ayarlanabilir. 10'dan sonra instance hala health statüsünde ise, TCP ttafiği gönderilmesi devam eder.

ELB Cross Zone Load Balancing



- ELB farklı region'lar arası işlem yapamaz. Baska region'lar arası işlem yapmak için Route53 kullanılmalıdır.

Amazon Route 53: Yüksek oranda erişilebilir ve ölçeklenebilir bir Cloud DNS web hizmetidir.

- Default olarak disable olan bu servis, yükün instance'lar arası dağılmasını sağlar. Örnek olarak 7 instance bir AZ ve 3 instance başka bir AZ'da olan bir ELB olsun. **Cross Zone Load Balancing** enable olmaz ise, trafik AZ arasında dağılacaktır. Enable olur ise, yük instance'lar arasında dağılacaktır. Bunun anlamı 7 instance olan AZ'a trafikin %70 gönderilecektir.

- ELB servisi AWS console, command line, SDKs(Software Development Kit) ve API query ile konfigüre edilebilir.
- ELB adı unique olmalı ve 32 karaktere kadar olabilir.

ELB Positioning - Internet Facing vs Internal ELB

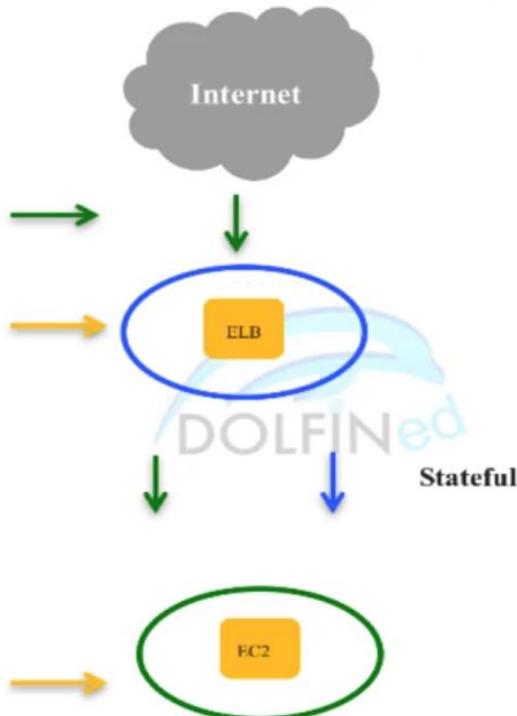
- İki tür ELB vardır. Bunlardan birisi genel olarak bilinen Internet Facing ve digeri Internal ELB'dir.
- Internet Facing
 - Public ip adresi vardır ve DNS bu ip adresini cozebilir.
 - Trafigi, instance'ların private ip adreslerini yönlendirir.
 - Her AZ için birer public ip adres bulunması gerekmektedir.
 - ELB DNS name, elb.amazonaws.com uzantısında olmalıdır ve AZ içerisindeki bütün public IP'leri barındırır. örneğin: name-1234567890.region.elb.amazonaws.com
- Internal ELB
 - Private ip adresi vardır ve DNS, ELB DNS adını cozebilir.
 - Trafigi, instance'ların private ip adreslerini yönlendirir.
 - ELB DNS name, internal ile başlamalı ve elb.amazonaws.com uzantısında olmalıdır ve AZ içerisindeki bütün private ip'leri barındırır. örneğin: internal-name-1234567890.region.elb.amazonaws.com (**0 yok**)

ELB Security

ELB – ELB Security Group Settings

- **Visualization...**

Inbound
Source: 0.0.0.0/0
 (VPC CIDR for Internal)
Protocol: TCP
Port: ELB Listeners
ELB Sec Group
Outbound
Destination: EC2 Sec Group
Protocol: TCP **Port:** Health Check
Protocol: TCP **Port:** Listener
Inbound
Source: ELB Sec Group
Protocol: TCP
Port: EC2 Instance Listeners &
EC2 Health Check
EC2 Registered Instances Sec Group
Outbound
Destination: ELB Sec Group
Protocol: TCP **Port:** Ephemeral



- Farklı protocol/port tanımı ile birden fazla front ve backend listener oluşturabiliriz ve connection request'leri hangi listener'in dinleyeceğini protocol/port ikilisi ile tanımlayabiliriz.
- Security Group tanımlamak zorunludur.
- ELB default VPC'de oluşturulursa, bu ELB için security group oluşturamayız. AWS bunu gerekli rules/port ikilisine göre oluşturur.

- Non-default VPC'de, ELB icin security group secebiliriz. Biz secmezsek, AWS bunu gerekli rules/port ikilisine gore olusturur.
- Talep internetten gelecek ve ELB frontend listener'a ulasacak. ELB backend listener'dan EC2 gidecek.

ELB icin: **Inbound Source:** 0.0.0.0/0 (internet), Protocol: TCP, Port: ELB Frontend listener

Outbound Destination: EC2 Security Group, Protocol: TCP, Port: Health Check, Protocol: TCP, Port: Backend Listener

EC2 icin: **Inbound Source:** ELB Security Group Protocol: TCP, Port: EC2 Instance Listener & EC2 Health Check

Outbound Destination: ELB Security Group Protocol: TCP Port:Ephemeral (1-65535)

- ELB Security Group:

Internet facing:

- Source: 0.0.0.0/0, Protocol TCP, Port: ELB Listener (http, https, TCP, SSL)

Internal ELB:

- Source: VPC CIDR, Protocol TCP, Port: ELB Listener (http, https, TCP, SSL)

Allow Outbound for both:

- Destination: EC2 registered instance security group, Protocol TCP, Port: Health Check
- Destination: EC2 registered instance security group, Protocol TCP, Port: Listener

- Registered EC2 Instances Security Group:

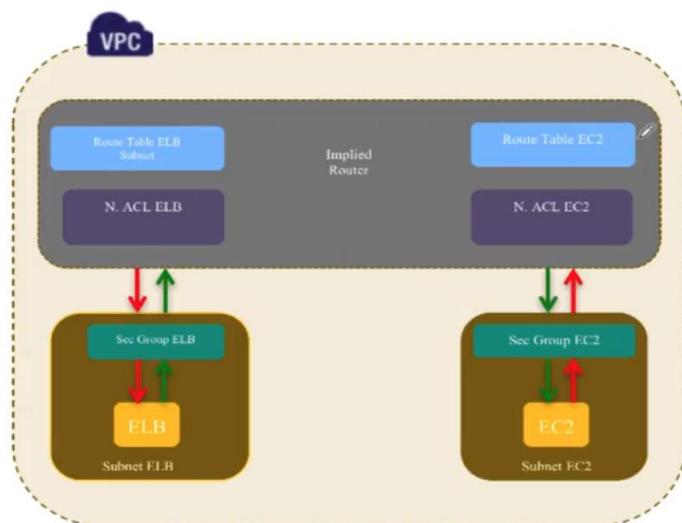
Allow Inbound:

- Both Internet facing & Internal ELB
 - Source: ELB Security Group, Protocol: TCP, Port: Health Check
 - Source: ELB Security Group, Protocol: TCP, Port: Listener

Allow Outbound:

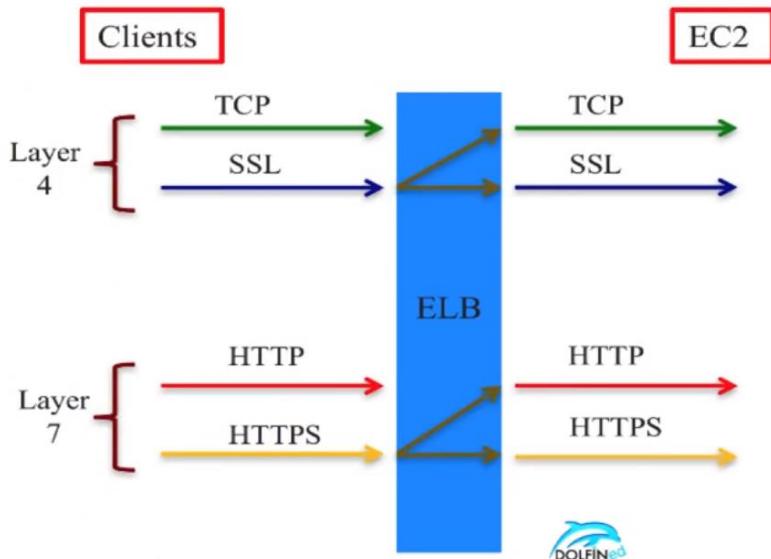
- Both Internet facing & Internal ELB
 - Destination: ELB Sec Group, Protocol: TCP, Port: Ephemeral (1-65535)

ELB NACL



- Ayarlama sadece non-default NACL icin gereklidir. Default NACL inbound ve outbound icin her sey aciktir.
- NACL ayarlari primary olarak ELB ayarlarini ve response trafigi odak alir.

ELB Listeners



- **Layer 4:** Inbound TCP veya SSL (encrypted) olabilir. TCP, TCP olarak devam eder. SSL, SSL veya TCP olarak devam edebilir.
- **Layer 7:** Inbound http veya https (encrypted) olabilir. http, http olarak devam eder. https, https veya http olarak devam edebilir.

HTTP/HTTPS listeners;

- https listener kullanılabilmesi için, ELB'nin X.509 SSL/TLS sertifikasına ihtiyacı vardır.
- Bu sertifika ile ELB backend EC2 request'i göndermeden, client session'ı decrypt eder.
- Bu sertifika AWS ACM (Amazon Certificate Manager) ile oluşturulur veya kendi IAM ile oluşturulabiliriz.

Akış Client-ELB-EC2 şeklindedir ve EC2, ELB bilgilerini bilir. Ancak proxy protocol enable edilirse, source bilgisi bilinemez.

- Bu şekilde EC2 http header'ını okuyabilir.

Not:

- Sunucularda httpd kurulu olmaz veya çalışmaz ise Load Balancer EC2'leri unhealthy görecektir.
- Gelen request'ler hakkında bilgi isteniyorsa, Access Logs enable edilir ve bu logların S3'de tutulması sağlanabilir.
- /var/www/html/ dizininde nano index.html yazarak açılan sayfaya yazı yazabiliriz. Bu sayfada yazılanlar, name-1234567890.region.elb.amazonaws.com sayfasında olacaktır. Load Balancer enable edilir ve her iki instance'da da ayrı yazılmırsa, Load Balancer hangisine yönlenirse, o görülecektir.

ELB Sticky Sessions (Session Stickiness veya Session Affinity)

Bir bakima cache mekanizmasi ile user ve instance arasında hızlı bir şekilde haberlesme mekanizmasıdır. Haberlesme cookie'ler üzerinden gerçekleşir ve hem EC2 tarafında hem de user tarafında tutulur. Bu cookie'ler sayesinde client bilgilerine ulaşılabilir.

- Load Balancer'in, session taleplerinin belirli bir instance'a gönderilmesini sağlar ve diğer instance'lar aynı cookie'ye sahip değildir.
- Çalışan uygulamanın kendi session cookie'si var ise, Load Balancer'in bu cookie'leri kullanması sağlanabilir ve burada bulunan expiration süresi dikkate alınır.

Bu cookie'ler PHP veya Java cookie'leri olabilir.

- Çalışan uygulamanın kendi session cookie'si yok ise, Load Balancer'in session cookie'si oluşturulması ve bunun kullanılması sağlanabilir ve burada bulunan expiration süresini ELB belirler.
- Cookie expire olursa veya remove edilirse, session artık sticky session olmaktan çıkar ve yeni cookie insert edilene kadar normal bir session gibi davranışır.
- Cookie expire olmadı ama instance unhealthy statusunda olursa, session başka bir instance'a yönlendirtilir ve eğer arka planda çalışan paylaşımlı database veya application var ise session sticky session olarak çalışmaya devam eder. Paylaşımlı değil ise session artık sticky session olmayacağıdır.

ELB Security policy for SSL/HTTPS sessions

- TCP veya HTTP olsun, haberlesme sırasında iki tür akış vardır. Authentication ve Encryption
- Bir internet sitesine girmeye çalıştığımız zaman, **this certificate is not verified or validated** gibi bir mesaj authentication kısmıdır ve server side certification vardır.
- Hem client side hem de server side'da sertifika kontrolü olursa, bu two way authentication olduğu anlamına gelir.
- ELB tek bir X.509 sertifikasını destekler. Birden fazla sertifika için, birden fazla ELB gerekmektedir.
- **ELB https ile client side sertifikayı desteklememektedir.** Bu demek ki ELB two way authentication desteklememektedir.
- Two way authentication isteniyorsa, http veya https'den ziyade TCP tercih edilmelidir.
 - ELB için proxy protocol oluşturmak da bunun için workaround olabilir.
- **Client side certificate hic bir şekilde Sticky Session özelliğini desteklememektedir.**

ELB-Connection Draining

- Default olarak disable'dır.
- **Connection Draining enable edilirse ve sistemde unhealthy durumunda olan EC2 instance var ise, default olan 300 saniye boyunca o instance'a gönderilmiş olan işleri bekletecektir** ve instance tekrar sağlıklı duruma gerilirse, işleri tamamlayacaktır. Bu süre zarfında instance tekrar sağlıklı duruma gelmez ise, session'lar terminate edilecektir.
- Default 300 saniye olan süre, 1-3600 saniye arası ayarlanabilir.
- ELB unhealthy olan instance'a yeni iş göndermeyecektir.

DNS Failover for ELB: Route53'ü, farklı region'larda olan iki ELB arasında DNS failover için kullanabiliriz.

ELB Monitoring

- AWS Cloud Watch:
 - Her dakikada bir metric'leri cloud watch'a gönderir.
 - AWS Cloud Watch notification gönderilmek üzere ayarlanabilir.
- Access Logs
 - Default disable'dır.
 - **Requester kim olduğunu, request time'i, request IP'si, request type gibi bilgileri barındırır.**
 - Store edilecek logları S3 üzerinden belirli bir şekilde tutulabilir.
 - Access Log için değil ama S3 kullanımı için ücretlendirilir.

- AWS Cloud Trial:
 - API call'larını yakalamak için kullanabiliriz.
 - S3 üzerinde tutulabilir.

ELB Scaling, Prewarming, Testing ve Idle Timeout

- Prewarming Pre-Warming - ELB Scaling:
 - ELB 100k talep karşılayabiliyor olsun ve 100k+1 kadar talep gelsin. Bu durumda ELB gelen http request'i que alma gibi bir yapısı yoktur. Gelen talebi http error mesajı olan Error 503 olarak döndürür. ELB Scaling ile threshold aşılırsa, 1-7 dk arasında yeni bir node eklenir ve talebin karşılanması sağlanır.
 - Bu artış bilgisini de Cloud Watch tutarak, uyarı oluşturur.
 - Artış exponential olursa, hata alacak talepler olacaktır.

ELB Scaling - DNS Updates:

- ELB büyündüğü zaman yani yeni bir node geldiği zaman, yeni bir Public IP adresi gelmiş olacak. DNS tanımında önceden 3 IP olduğunu ve yeni gelen node ile bunun 4'e çıktığını düşünelim. Bu durumda DNS tanımına eklenmesi gerekecek ve cache'den okumalar buna engel teşkil edecek.

Bu nedenden dolayı, ELB her 60 saniye DNS record TTL (Time To Live) kullanıyor ve bu süre sonunda cache'den okumayı bırakır ve cache tekrar oluştur.

ELB Connection Timeout: Default 60 saniyedir ve daha az olursa, ELB Instance'ın unhealthy olduğunu düşününebilir.

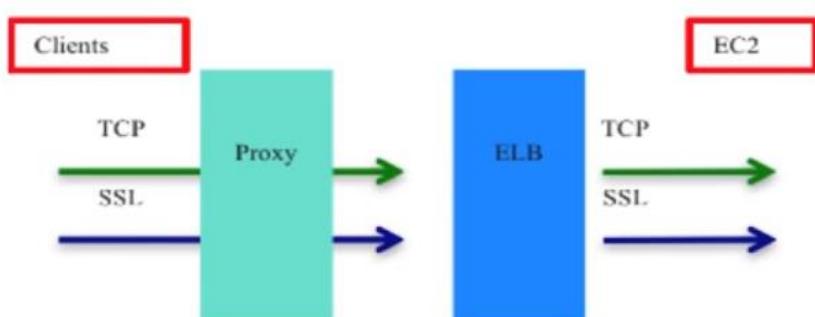
Notlar:

- **Three-Tier Web App:** Web + App ve Database katmanı var demektir.
- Özel olarak tanımlamazsa, AWS ELB servisi en son pre-defined security policy kullanacaktır.
- Backend encrypted connection için, pre-defined security policy her zaman kullanılır.
- SSL connection'larda, server order preference by default enable'dır.
- Server Order Preference enable ise, ELB cipher client'ın kullandığı işe eşleşecektir.

Proxy Server: Client ile dış dünya arasındaki ilişkiyi sağlayacak gateway sistemidir.

Proxy server, client'dan aldığı request'leri yürütür ve sonucu yine client'a iletir.

Aynı anda, bu bilgileri cache'ler ve proxy server'da tutulur ve artık bu bilgi proxy servisinden gelir ve iletişim çok daha hızlı olur.



ELB'de proxy protocol enable etmeden önce,

- Load Balancer'ın proxy server'in arkasında olmadığını kontrol etmeliyiz.
 - Proxy protocol hem proxy server'da hem de load balancer'da enable ise, load balancer request'e başka bir header ekler.
- Instance'in, proxy protocol bilgisini işleyebileceğini kontrol etmeliyiz.
- Listener'in, proxy desteği olduğu kontrol edilmelidir.

- Proxy protocol http ve https listener'da desteklenmez.
- **Proxy protocol front ve back end listener SSL ise desteklenmez.**
- Client, request'i proxy server'ın arkasından yaparsa, ELB'de proxy protocol enable edilemez.

SSL front-end:backend listener konfigurasyonu Proxy Protocol header'ı desteklemez.

	OSI Layer	TCP/IP	Datagrams are called
Software	Layer 7 Application	HTTP, SMTP, IMAP, SNMP, POP3, FTP	Upper Layer Data
	Layer 6 Presentation	ASCII Characters, MPEG, SSL, TSL, Compression (Encryption & Decryption)	
	Layer 5 Session	NetBIOS, SAP, Handshaking connection	
	Layer 4 Transport	TCP, UDP	Segment
Hardware	Layer 3 Network	IPv4, IPv6, ICMP, IPSec , MPLS, ARP	Packet
	Layer 2 Data Link	Ethernet, 802.1x, PPP, ATM, Fiber Channel, MPLS, FDDI, MAC Addresses	Frame
	Layer 1 Physical	Cables, Connectors, Hubs (DLS, RS232, 10BaseT, 100BaseTX, ISDN, T1)	Bits

- HTTP/HTTPS request'ler L7 listener üzerinden gelir ve **Proxy Protocol desteklenmez**.
 - **X-Forwarded** HTTP ve HTTPS ile çalışır.
- TCP/SSL request'ler L4 listener üzerinden gelir ve **X-Forwarded desteklenmez**.
 - **Proxy Protocol** TCP/SSL ile çalışır.

Client ile ELB arasında SSL haberleşme gereksinimleri

- SSL Protocol
- SSL Ciphers (Encryption alogritmasıdır. SSL farklı cipher'lar kullanabilir.)
- Server Order Preferences (Enable ise, ELB cipher listesi ile client listesiyle ilk eşleşme kullanılır)
-

AWS ELB SSL security policy için aşağıdakiler desteklenmektedir.

- TLS 1.0
- TLS 1.1
- TLS 1.2
- SSL 3.0
- **TLS 1.3 ve SSL 2.0 (deprecated yani kullanımdan kaldırıldı) desteklenmez.**
- **Farklı region'larda, EC2'ler için, ELB load balance enable edilerek, HA ve fault tolerance olması istenirse**
 - ELB ve EC2 instance'ları aynı VPC'de olmalıdır yani aynı region'da.
 - ELB region spesifiktir.
 - Sadece ELB ile farklı region'lardan, load balance yapılamaz. Bunun için Route53'de gereklidir.

Stateless Application: Session state yani session bilgisini tutmuyor demektir.

Perfect HA Solution için

- Sec Group ve NACL doğru configure edilmiş IGW olmalıdır.
- Aynı region'da olan, en az iki AZ olmalıdır.
- Her AZ için public subnet(s) ve ELB bunlardan birisine tanımlanmış olmalıdır.
- Database seviyesinde private subnet olmalıdır.
- Multi-AZ RDS veya AWS tarafından yönetilen DB engine olmalıdır.
- Her iki AZ için tanımlanmış, ELB ve EC2 ile çalışabilecek auto scaling olmalıdır.

Notlar:

- Single ELB üzerine multiple session sticiness yapılamaz.
- Single ELB, multiple SSL certificate desteklemez.
- ELB SNI certificate desteklemez.
- **Session sticiness sadece HTTP protokolü ile yapılabilir. TCP ile yapılamaz.**
- ELB'nin trafic artışını tespit etmesi ve yeni node eklemesi 1-7 dk arasında olabiliyor.

Sertifikalarla ilgili ek bilgi:

<https://forums.aws.amazon.com/thread.jspa?threadID=145336&start=0&tstart=0>

ELB hakkında:

<https://aws.amazon.com/articles/1636185810492479>

HTTP/HTTP Load Balancer

Use Case	Front-End Protocol	Front-End Options	Back-End Protocol	Back-End Options	Notes
Basic HTTP load balancer	HTTP	NA	HTTP	NA	<ul style="list-style-type: none">• Supports the X-Forwarded headers
Secure website or application using Elastic Load Balancing to offload SSL decryption	HTTPS	SSL negotiation	HTTP	NA	<ul style="list-style-type: none">• Supports the X-Forwarded headers• Requires an SSL certificate deployed on the load balancer
Secure website or application using end-to-end encryption	HTTPS	SSL negotiation	HTTPS	Back-end authentication	<ul style="list-style-type: none">• Supports the X-Forwarded headers• Requires SSL certificates deployed on the load balancer and the registered instances

TCP/SSL Load Balancer

Use Case	Front-End Protocol	Front-End Options	Back-End Protocol	Back-End Options	Notes
Basic TCP load balancer	TCP	NA	TCP	NA	<ul style="list-style-type: none">• Supports the Proxy Protocol header
Secure website or application using Elastic Load Balancing to offload SSL decryption	SSL	SSL negotiation	TCP	NA	<ul style="list-style-type: none">• Requires an SSL certificate deployed on the load balancer• Supports the Proxy Protocol header
Secure website or application using end-to-end encryption with Elastic Load Balancing	SSL	SSL negotiation	SSL	Back-end authentication	<ul style="list-style-type: none">• Requires SSL certificates deployed on the load balancer and the registered instances• Does not insert SNI headers on back-end SSL connections• Does not support the Proxy Protocol header

Önemli Soru:

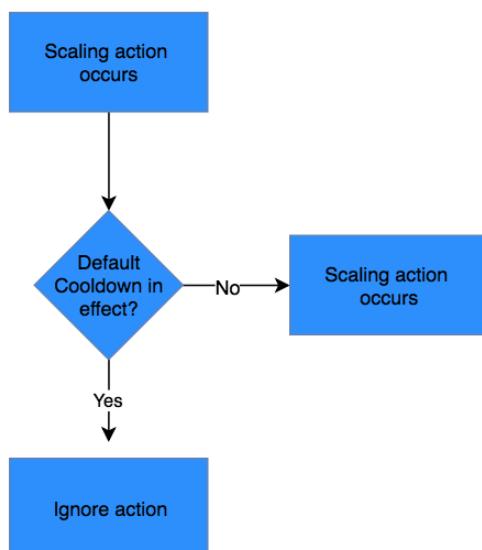
You are assigned to build a solution to host a2-tier application that deals with sensitive data on AWS VPC. As per corporate data security policy, all sensitive data must be protected in-transit and at rest. Authorized employees will be accessing the application, entering data and uploading files which will be stored on the EC2 instances' EBS volumes for processing, then uploaded to S3 buckets. Database tier design and its encryption is out of your scope.

How can you architect this in a multi-AZ, scalable solution while ensuring data protection in-transit and at rest?
(Choose 2)

1. Use Auto Scaling with ELB for Multi-AZ and scalability. Use TCP termination on the ELB and EBS Encryption on EC2 instances, Server Side Encryption on S3 buckets.
2. **Use Auto Scaling with ELB for Multi-AZ and scalability. Use SSL listeners on ELB both for front and backend. EBS encryption, and Server Side Encryption on S3.**
3. **Use Auto scaling with ELB for Multi-AZ and scalability. Use TCP on ELB and SSL termination on EC2 instances. EBS Encryption on EBS volumes, and server side encryption on S3**
4. Use SSL termination on ELB front end. EBS encryption on EC2 instances, and SSE on S3 bucket(s). Use Auto scaling with ELB for multi-AZ
5. Use TCP on the load balancer front end listeners, SSL termination on the Amazon EC2 instances, and Amazon S3 with server-side encryption.

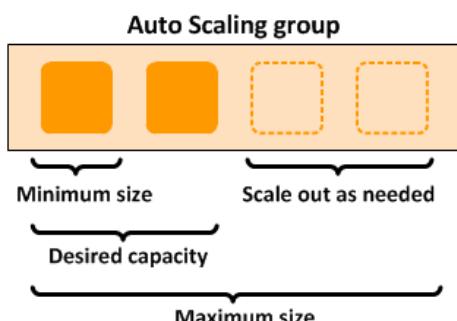
AWS Auto Scaling

- Farklı AZ üzerinde çalışıyor ancak farklı region'lar için işe yaramaz. Region spesifiktir.
- Workload duruma göre EC2 instance sayısı azalır veya artarabilir.
- Artma durumuna Scale Out ve azalma duruma Scale In denir.
 - Scale-In durumu olurken, ilk olarak en eski launch konfigürasyon olan instance terminate edilir.
 - Bütün instance'lar aynı launch konfigürasyonuna sahipse ve bütün AZ'larda aynı sayıda instance var ise, ASG bir sonraki faturalandırma süresine en uzak olan instance'sı seçecektir. Bu seçim, EC2 instance'larının kullanımını max çıkarmayı sağlayacaktır.
- Tanımlanan policy'e göre, her zaman ihtiyaç duyulan EC2 kaynağını sağlar.
- Console, CLI, SDKs veya API üzerinden konfigüre edilebilir
- Auto Scaling, scaling aktivitelerinde cooldown süreci işletir.
 - Default 300 saniyedir.
 - Konfigüre edilebilir.
 - Bir önceki scaling aktivitesi bitmeden, ek bir EC2'nin eklenmesine veya terminate edilmesini engeller.



Komponentleri;

- Launch Konfigürasyonu
 - Instance family, instance type, AMI, Key pair, Block device ve Security group tanımlanır.
 - Launch kongigürasyonu değiştirilecek ise, yeni bir tane oluşturmak gerekiyor. Sadece silinebilir veya kopyanalabilir.
- Auto Scaling Group



- EC2 instance'lar için logical group bilgisidir ve istenildiği zaman edit edilebilir.
- AS group'un yeni instance'ları için hangi subnet'i kullanması gerektiğini belirleyebiliriz.

- Kapasite veya başka herhangi bir sebepten dolayı istenilen AZ'a yeni bir instance launch olamazsa, process başka bir AZ'a instance eklemeye çalışır. O da olmaz ise bir diğerine deneyerek, Scaling Policy'e bağlı kalmaya çalışacaktır.
 - ASG, AZ arasında bir dengesizlik görür ise, Re-Balancing activity devreye girecektir. Mesela EC2 sayıları 3-3-2 olan 3 AZ olsun ve 1 tane daha EC2 eklenmesi gereksin ama 2 EC2 instance'a sahip olan AZ3'de de kapasite olmasın. Bu durumda yeni instance AZ1 veya AZ2 eklenir ve AZ3'de kapasite olduğu andan AZ Rebalance proces'i, bütün AZ'ları 3-3-3 olacak şekilde işletir.
- Scaling Policy
 - ASG hangi durumlarda scale veya shrink olacağı belirlenir. (On-demand/Dynamic/Cyclic/Scheduled)
- Asagidaki kurallar saglanıyor, AWS console veya CLI kullanarak AS Group'a EC2 instance eklenebilir.
 - Instance running state'de olmalıdır. (Stopped veya Terminated olmamalı, Reboot olabilir ama reboot running anlamına gelir.)
 - Kullanılacak AMI (Amazon Machine Images) Amazon'da halen aktif olmalı.
 - Instance başka bir AS Group'da dahil olmamalı.
 - AutoScaling Group ile aynı AZ'da olmalı.
- Bir instance terminate statusune gelirse, bunu tekrar AS Group'a ekleyemeyiz.

İçeriğinde t2.micro EC2 instance'lar olan bir ASG olsun ve ihtiyaç doğrultusunda bunları t2.2xlarge ile değiştirmek isteyelim.

Bu durumda yeni instance type ile, **yeni bir launch konfigürasyonu oluşturup, ASG update edilmelidir.** ASG'a ancak bir tane launch konfigürasyonu tanımlanabilir ve bu launch konfigürasyonu modify edilemez.

Scaling Policy

Üç çeşit scaling policy bulunmaktadır.

Target Tracking Scaling: Çalışması, bir termostatin evin sıcaklığını korumasına benzer. Hedef değer ne ise, ona bağlı olarak kapasite artar veya azalır.

Step Scaling: Scaling ayarına göre davranıştır. Bu scalin işlemi CloudWatch metrikleri ile tetiklenebilir ve eşik değer buna göre belirlenebilir.

Simple Scaling: Tek bir scalin ayarına göre hareket eder.

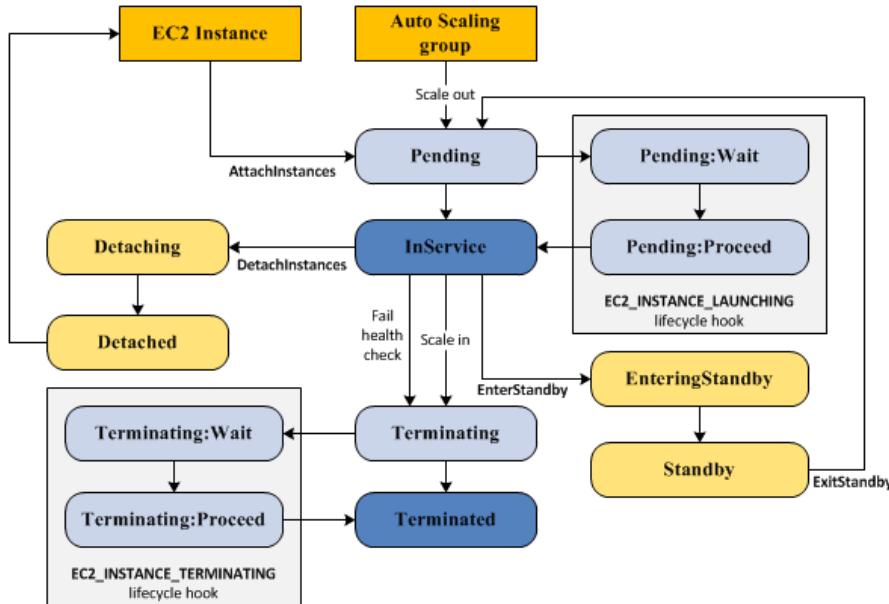
AWS, Simple Scaling'den ziyade Step Scaling kullanılmasını tavsiye eder.

AWS Application Auto Scaling

Bu servis ile, scalable kaynaklar için auto scaling tanımlanabilir.

Örnek olarak, DynamoDB'de yaşanacak yoğunluk ile başa çıkmak için, AWS Application Auto Scaling servisini kullanabilir. Bir kereye mahsus veya yinelenen şekilde action tanımlayabiliriz.

ASG-EC2 Instance States (Cont.)



- Instance standby olursa, hala ASG tarafindan yönetiliyor demektir ve ASG bu node üzerinde Health Check calistirmaz.

ASG ve ELB

- Mevcut AS Group'a bir veya daha fazla ELB eklenebilir.
 - ELB AS Group ile aynı AZ olmalı.
 - ELB eklendikten sonra, mevcut olan veya eklenmis EC2 instance'lar, otomatik olarak ELB register olacaklardır. De-register edilirse de aynı şekilde de-register isi otomatik gerçekleşir.
 - EC2 ve ELB aynı AZ olmalıdır.
 - Connection draining ELB'de enable ise, AutoScaling'de bunu kabul eder.

ASG-Health Checks

- Instance için scheduled termination baslamadan, AWS CLI üzerinden, as-set-instance-health komutu calistirilirsa, Instance tekrar health statusune gelecektir. Bu çok kısa bir zaman aralığıdır ve termination basladıkta sonra çalışmaz.
- Elastic IP ve EBS volume'lari terminate olmuş instance'dan deattach olacaktır. Bunları yeni gelecek instance'a manual olarak attach etmemiz gerekecektir.

ASG ve Spot Instance

- Launch konfigurasyonunda, Spot instance secerék, fiyat belirtebiliriz.
- AS Launch konfigurasyonunda, on-demand ve spot instance beraber olamaz.
- Bid price degisitirilmek istenirse, yeni bir launch konfigurasyon dosyası oluşturulmalıdır.
- Belli AZ'da bulunan AutoScaling için yeni bir spot instance, market fiyatından dolayı eklenemezse, market fiyatının daha düşük olduğu başka bir AZ'e eklemeyi deneyecektir.
 - Eğer ilk eklenmeye çalışılan AZ market fiyatı düşerse, AutoScaling bu iki AZ arasında rebalance yapmaya çalışacaktır.
- Instance terminate, launched, fail to launch, fail to terminate durumlarında email atacak bir yapı kurulabilir.
- İki AS Group'u CLI üzerinden (AWS Console üzerinden olmaz) merge yapılabilir. Merge yapılacak olan AS Group mevcutlardan biri olmalıdır. Yeni bir ASG olmamalı.

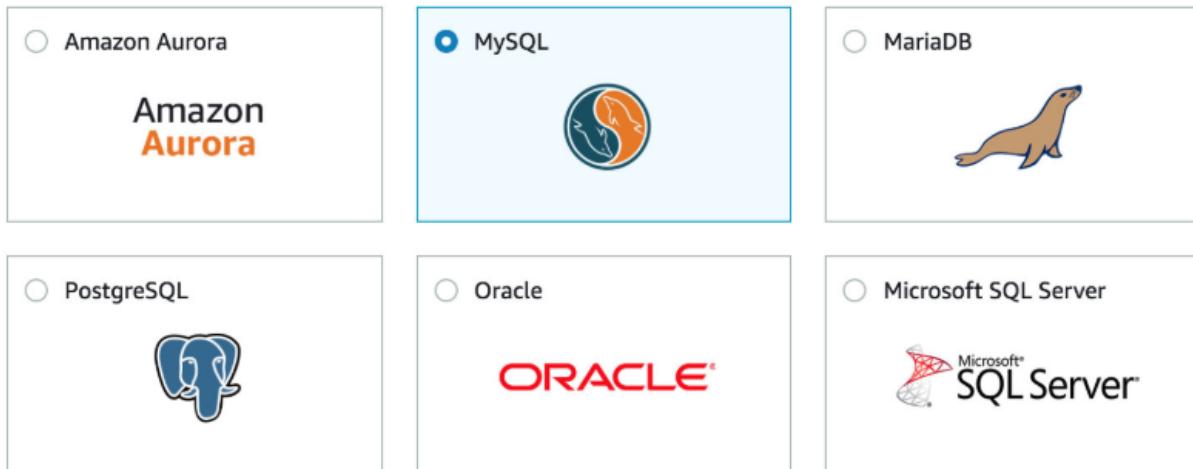
Auto Scaling Policies

- **Manual Scaling:** Min, Desired ve Max instance sayisi manual ayarlanabilir.
- **Schedule Based:** Predictable (Tahmin edilebilir), sistemin davranış bicimi biliniyor ise, ona göre ayarlama yapılabilir.
 - Sürekli çalışmayan ancak düzenli olarak çalışan işler için uygundur. Örneğin, sadece ve gece haftasonları çalışan ETL için kullanılacak ise, Scheduled-Instance doğru seçim olacaktır.
- **Event (Dynamic) Based:** Mesela sistem %40 altına inerse, Scale-In ayarlanabilir ve EC2 instance sayisi azalabilir.
 - Tanımlanacak alarm; CPU, memory, network, ... esas alınarak tanımlanabilir.
 - Alarm monitor etmek ve generate etmek için Cloud Watch kullanılabilir.
 - %70 olacak şekilde bir tanım olsun ve %71 olunca bir instance daha eklenmis olsun. Bu durumdan sonra default 300 saniye olan Cool Down timer baslar ve bu sure sonda %71 olan yoğunluğun durumuna göre yeni alarm oluşturur ve buna göre instance ekler veya eklemez.
- **Dynamic/On-demand - Step Scaling:**
 - Cool Down timer desteklenmez
 - Multiple step vardır. %70, %80, %90 davranışları gibi.
 - Warm-Up timer desteklenir.
- **Target tracking Scaling:**
 - Bir ASG birden fazla policy attach olabilir.
 - Scheduled action için, unique tarih ve saat gereklidir.
 - Aynı tarih ve zaman için birden fazla aktivite ekleyemeyiz.
 - Scaling policy; max kapasiteden daha fazla ve min kapasiteden daha az instance olmasını sağlayamaz. Ancak bu rakamlar arasında işlem yapabilir.

Monitoring Auto Scaling Group

- By default her 5 dakikada bir kontrol edilir. Her 1dk için de yapılabılır (detailed monitoring) ancak bu ücretlendirilir.
- Launch conf AWS CLI ile yapılrsa, detailed monitoring by default enable olur.
- Enable olduğu zaman, AutoScaling servisi Cloud Watch'a metrikleri göndermeye başlar.
- Bu özellikler degistirilmek istenirse, yeni bir launch conf dosyası oluşturulmalıdır.

RDS (Relational Database Service)



- Client information, address, kredi kartı bilgileri, client bilgileri gibi bilgileri barındır.
- DB instance'a yani OS'a erişim yoktur ama engine'e erişim vardır.
 - Operation system yetkileri de isteniyorsa, RDS seçilmemelidir. RDS, AWS tarafından manage edilir ve sadece DB engine'e erişim sağlar. Operation system yetkileri için, EC2 kullanılmalıdır.
- Point-time-in recovery yapılabılır.
- Administration tarafını AWS yönetir, manage'dır. **Fully Managed değildir.**
 - Automated patch, backup gibi admin işleri kullanıcı yapmalıdır.
 - Software updates,
 - RDS kullanıcıları, major versiyon upgrade'leri ve bir çok minor upgrade'leri ne zaman olacağını kontrol edebilirler.
 - Gerektiği taktirde storage ve compute yükseltmesini,
 - Multi AZ seçilmişse; active ve standby arasındaki senkronasyon işlerini AWS yönetir.
- DB instance'lari, weekly maintenance calisir ve eger belirtilmezse, AWS default olarak 30dk olan secilir.
- DB settings, build DB schema, performance tunning işlerini AWS yüklenmez. AWS'in DB engine ulaşım sağlamaşının temel nedeni de budur.
- MS SQL, Oracle, PostgreSQL, MariaDB, AWS Aurora ve MySQL desteklenen relational database'lerdir.
 - Database'ler icin iki tur lisanslama vardır.
 - Bring your own license (BYOL) - Mevcut lisans var ise, aktarılabilir.
 - License included, saatlik olarak ücretlendirilir.
 - **Oracle RMAN ve RAC RDS'de support edilmez.**
- Bir account icin en fazla 40 DB instance olabilir.
 - License included modelinde, 40 tanesinden 10 tanesi Oracle ve MS SQL olabilir.
 - BYOL kullanılırsa, 40 tanesi istenilen DB tipi olabilir.
- Amazon RDS, DB ve Log storage icin EBS volume kullanır. (Instance-Store kullanılmaz)
- Üç tip kullanım vardır;
 - **General Purpose**: Genel orta düzeyli I/O gereksinimi için kullanılır.
 - **Provisioned IOPS RDS Storage**: Yoğun OLTP işleri için kullanılır.
 - **Magnetic RDS Storage**: Düşük DB workload için kullanılır.
- Max storage kapasitesi desteklenen DB'ler için 16TB'dır.

Spesifik ayarları bir grup database'e aktarılmak isteniyorsa, **Parameter Groups** kullanılabilir.
Alter Database komutunu tek tek çalıştırımdan ziyade, bu grup yardımı ile, hepsine uygulanabilir.

Multi-AZ RDS Option

- Aynı region'da farklı AZ'da olmalıdır.,
- Standby'da read ve write yapılamaz.
- Standby'in hangi AZ'da olacağını AWS seçer.
- AWS multi-AZ için provisioned IOPS instance tavsiye ediyor.
- Manual olarak failover ancak reboot sırasında olabilir. "**Reboot with failover**" seçeneği Primary RDS DB tarafında seçilebilir.
- API call ve CLI ile son 14 günlük RDS event'leri gorulebilirken, console üzerinden sadece son gün gorulebilir.
- OS patching, system upgrade ve DB scaling önce standby'da yapılmalıdır.
 - Multi-AZ'da yer alan RDS database'lere storage artırımı gibi bir maintenance yapılacak ise, Standby Maintenance, Failover to Standby, Eski Primary Maintenance yapılır ve artık eski DR sistemini yeni Prod olarak kullanılması devame eder. Tekrar Eski Prod'a failover yapmaya gerek yoktur.
- Manual olarak, support edilen DB instance'larından birine DB console üzerinden upgrade yapılabilir.
 - RDS/DB Instances/Modify DB Instances/ Set DB Engine version
 - By default, degisiklikler bir sonraki maintenance zamanında olur ama istenirse, force an immediate yapılabilir.
- **Multi-AZ'da, synchronous replikasyon yapılmaktadır.**

DB Automated Backups

- AWS Auto backup alınırken, verinin de sağlıklı olup olmamasını kontrol eder.
- Backup'lar S3'de tutulur.
- Multi-AZ için backup'lar standby tarafından alınır. (MariaDB, MySQL, Oracle ve PostgreSQL için gecerlidir.)
- Backup alınabilmesi için DB "Active" state'de olmalıdır.
- Manual backup'lar point-in-time recovery için kullanılamaz.
- Backup'lar daily alınır ve DB transaction log'lari da buna dahildir.
- By default enable'dir.
- RDS silinirse, automated backup'lar da silinir.
- Retention 0 ayarlanırsa, automated backup alınmıyor demektir.
- Automated backup'lar, MySQL için sadece InnoDB storage engine destekler. ISAM'da dahil olmak üzere, diğerleri beklenmeyen sonuçlar doğurabilir.
- Backup'lar direk share edilemez ama kopyası alınarak paylaşılabilir.

Manual Backups

- Point-in-time recovery için kullanılamaz.
- RDS silinirse, manual backup'lar silinmez.
- Backup'lar S3'de tutulur.
- Diger AWS account'lari ile paylaşılabilir.
- DB restore sonrası, sadece default DB parametreleri, security group'lari gelir.
 - Restore sonrası, custom DB parametreleri ve security grupları apply edilmelidir.
- DB snapshot'i, mevcut olana restore edemeyiz, yeni bir tane oluşturulması gerekmektedir.
- Restore sonrası, RDS endpoint'de değişir. Uygulama bunun üzerinden çalışıyorsa, güncellenmelidir.
- Restore sürecinde, storage tipi de degisebilir. (General purpose, Provisioned IOPS, Magnetic)

RDS DB Security and Encryption

- Mevcut un-ecrypted DB instance'i, encrypt edilemez.
 - Yeni bir ecrypted DB oluşturup, un-encrypted data'yı migrate edebiliriz.
 - Yeni bir ecrypted DB oluşturup, backup/restore yapabiliyoruz.
- RDS, DB instance ve App instance arasında SSL haberleşmeyi support eder.
- Encrypted DB'de, snapshot'lar, backup'lar, data ve bu database'den oluşturulan replica da encrypt'dir.
- AWS IAM account, RDS API olan yetkileri kontrol eder.

AWS Storage Gateway ve AWS glacier'da data by default encrypt edilir. Amazon RDS, ECS ve Lamda'da encryption destekler ama enable edilmesi gerekmektedir.

SQL Server DB instance gibi bir database instance oluşturulduğu zaman, Amazon RDS database için SSL sertifikası oluşturacaktır.

SSL sertifikası, sahtecilik saldırısına karşı koruman için, Common Name (CN) olarak DB instance endpoint içerecektir.

SSL kullanarak SQL Server instance'a bağlanmanın iki yolu vardır.

- Bütün bağlantıları SSL'e zorlamak veya
- Belirli bir connection'ı encrypt etmektir.

Force SSL kullanılarak bütün bağlantılar SSL'e zorlanacak ise, statik parametre olan **rds.force_ssl** parametresini true yaparak, parametrenin aktif olması için DB instance'ı reboot etmek gerekmektedir.

Belirli bir connection bağlantısını encrypt etmek için de, Client için RDS Root CA sertifikası alarak, sertifikanın server'a import edilmesi gerekiyor ve uygulamanın da konfigüre edilerek, RDS'e bağlantının SSL olmasını sağlayabiliriz.

RDS Billing

- Saat başına DB instance kullanımı (hem Primary hem de Standby için)
- Storage GB/mo.
- Sadece Magnetic RDS storage için, I/O request/mo. (hem Primary hem de Standby için)
- RDS Provisioned IOPS SSD Instance için, Provisioned IOPS/mo.
- Internet data transfer
- Backup Storage (DB backups, active manual snapshots)
 - 10GB'a kadar olan EBS volume size için automated RDS backup'ları ücretlendirilmez.
- Multi AZ
 - Multi-AZ DB hours
 - Multi AZ Provisioned Storage
 - Double write I/Os (Primary I/O ve Primary -> Standby replikasyonu)
 - Data transferi için ayrıca ödenmez.
- Single AZ (Free tier)
 - Her account için 1 yıllık 750 micro instance hours/months
 - Oracle için BYOL (BringYourOwnLicenses) veya License included gereklidir.

Read Replicas

- **I/O kapasitesin dolması ve daha fazla I/O ihtiyacı gereksinimi olduğundan kullanışlı olabilir.**
- Read replica, Primary RDS instance'ın sadece okuma için kullanılan replicasıdır.
 - Data önce Primary DB instance'a yazılır. Sonra da **asynchronous olarak read replica'ya yazılır.**
 - Birden fazla read replica olabilir.
 - Multi-AZ read replica oluşturulamaz.
 - Console ve API ile oluşturulabilir.
 - Automatic backup'in enable olması gerekmektedir.
 - InnoDB engine support edilir. MyISAM support edilmez. (MyMAPo)
 - MySQL
 - MariaDB
 - PostgreSQL support edilenlerdir.
- Replication Chain en fazla 4 olabilir. (Primary → ReadReplica1 → ReadReplica2 → ReadReplica3)
- Failover veya Switchover durumunda, Read Replica'lar, yeni Primary'den okumaya devam edeceklerdir.
- Primary silinirse, Replica'lar silinmez ve Single AZ stand-alone DB instance olurlar.
- ReadReplica1 promote olarak, read-write instance haline gelirse; ReadReplica2, halen eski ReadReplica1 ve yeni DB instance'dan okumaya devam edecektir
- Primary ve replica arasındaki replication 30 gün boyunca devam etmez ise, replicasyon düzeltilemeyecek şekilde terminate olur.
- MySQL, MariaDB ve PostgreSQL için farklı bir region'da read replica oluşturulabilir.
- Standalone DB instance'a dönüştürülmüş olan promoted replica,
 - Backup retention period
 - Backup window
 - DB parameter group bilgilerini halen tutar ve bu bilgiler bir önceki kaynak Prod sistemi ile aynıdır.

RDS'de, read replica synchronous replikasyon sağlamaz. Asynchronous replication sağlar.

Synchronous replikasyon isteniyorsa, Multi-AZ kullanılmalıdır. Ancak bu AZ'lar kesinlikle aynı region'da olmalıdır.

Multi-AZ Deployments	Read Replicas
Synchronous replication – highly durable	Asynchronous replication – highly scalable
Only database engine on primary instance is active	All read replicas are accessible and can be used for read scaling
Automated backups are taken from standby	No backups configured by default
Always span two Availability Zones within a single Region	Can be within an Availability Zone, Cross-AZ, or Cross-Region
Database engine version upgrades happen on primary	Database engine version upgrade is independent from source instance
Automatic failover to standby when a problem is detected	Can be manually promoted to a standalone database instance

RDS Scaling

- Compute ve Storage size artırlabilir ancak azaltılamaz.
 - Storage scaling online yapılabilir ancak bazı performans sorunlarına neden olabilir.
 - Compute scaling downtime'a neden olabilir.
- Storage type MS SQLServer dışında değiştirilebilir.

Sınav Sorularından Notlar:

- Elasticache sık sık okunan verileri cache'e alarak, okuma yükünün azalmasını sağlayabilir.
- Synchronous replication sadece aynı region'da olabilir. Farklı region'larda support edilmez. Farklı region'lar için, DR çözümleri Asyncronous replica olabilir.
- DB instance'da event takibi isteniyorsa, subscribe yapılması gerekiyor. Bu şekilde, DB instance, DB cluster, DB snapshot, DB cluster, DB parameter group değişim bilgilerini alabiliriz.

- RDS kullanıcıları, major versiyon upgrade'leri ve bir çok minor upgrade'leri ne zaman olacağını kontrol edebilirler.
- Backup window değiştirilirse, etkisini hemen gerçekleştirebilir.
- Backup, Prod RDS'den alınırsa, bir kaç dk için I/O operasyonları suspend olabilir.

Amazon RDS CloudWatch Enhanced Monitoring.

CloudWatch, DB instance için metricleri hypervisor'dan alır. Enhanced monitoring ise instance'daki agent'dan alır. Bu iki sonuç aynı olmayacağından, hypervisor katmanı daha az miktarda veri barındırır, bu nedenle enhanced monitoring daha avantajlıdır.

Spesifik bir process için ne kadar CPU ve memory kullandığını öğrenmek istiyorsak enhanced monitoring kullanılmalıdır.

MariaDB, Microsoft SQL Server, MySQL version 5.5 veya sonrası, Oracle, PostgreSQL database'leri için geçerlidir. RDS child process ve OS process bilgilerini toplayabilir.

IAM Database Authentication

- SSL kullanarak, database'ye gelen ve giden trafiğin encrypt olmasını sağlar.
- **Her bir database'i ayrı ayrı yönetmek yerine, database erişim merkezi olarak kullanılabilir.**
- EC2'de çalışan uygulamalar için, şifre yerine özel profile credential kullanılabilir.
 - Örnek olarak, EC2 ve RDS arasında şifresiz güvenli bağlantı oluşturulabilir.

AWS Simple Storage Service (S3)

Block Storage

- Transactional database, random read/write ve structured database storage için uygundur.
- Hangi tarihte oluşturuldu, ne zaman modifiye olduğu, içerik gibi metadata bilgisi içermez.
- Sadece block adres bilgisinin olduğu index'i tutar, block içerisinde ne olduğunu önemsemeyiz. Sadece gerekligi zaman nasıl getirileceğini önemser.

Object Storage

- Dosyanın tamamını tutar ve bunları blockları halinde bolmez.
 - Büyük objeler için dahi gecerlidir. 5TB'a kadar objeler halinde saklar.
 - Objeler; verinin kendisi veya metadata (data created, modified, security attributes, content type) bilgisi olabilir.
 - Her obje global unique ID sahiptir.
- Objelere örnek olarak; photos, videos, music, static web content, snapshots, archival images olabilir.
- Distributed storage kullanılır.
 - Bu sayede block storage'a kıyasla daha ucuz hardware kullanılabılır.
- Object storage'a örnek olarak; AWS S3, Dropbox, Facebook (Image, Videos), Spotify
- Availability ve durability (dayanıklılık) sağlar.
 - Data kopyaları farklı lokasyonlarda tutulur.
- Object storage bir sürücü gibi direkt EC2'ye mount edilemez.
 - Infrastructure üzerinden erişilemektedir.

Data Consistency Models

İki tur consistence vardır.

- Strong (Immediate) Consistency
 - Farklı client'larin, farklı veri kopyasını okuyarak, aynı bilginin donmesi durumudur.
 - Herhangi bir storage node'da, herhangi bir update olması sonucunda, veri client için available olmadan önce, değişim bütün storage node'larda olması sağlanır.
 - Transactional database ve real time sistemler için uygundur.
 - Scalability and Availability için iyi degildir.
- Eventual Consistency
 - Farklı veri kopyasını aynı zamanda okumak farklı sonuçlara neden olabilir.
 - Blocking mekanizması yoktur, veri obje olarak update edilirse ve o an başka node'dan okuma yapılrsa, aynı data gelmeyecektir.
 - Zamanla diğer node'larda da yapılacaktır ve okumalar eventually consistent olacaktır.
 - Eventual consistency; scalability, availability, data durability sağlar bu nedenle cost storage'i düşürür ve bunlar object storage için require'dır.

S3'de

- Yeni gelen objeler(http put), immediate veya diğer adı ile strong consistency (Read(get)-after-write(put)) ile S3 yazılmaktadır.
 - http put; update veya add olabilir
 - http get; read işlemidir
- Mevcut bir obje üzerinde, update(http put) veya delete işlemi yapılyorsa, bu işlem eventual consistency olacaktır.

Host static websitesi oluşturulacak ve content bucket'a upload edilecek ve region spesifik olacak ise, format aşağıdaki gibi olmalıdır.

`<bucket-name>.s3-website-<AWS-region>.amazonaws.com`

S3 Buckets

S3 internet için bir storage yöntemidir. Internetten herhangi bir yerden, web servis ile datanın saklanması sağlanır.

- S3 bucket region spesifiktir.
- S3'de toplam store edilecek data ve obje sayısı sınırsızdır.
- Object based storage'dır. Block storage degildir.
 - Block storage'a örnek EBS verebiliriz.
- Distributed data-store mimarisine sahiptir.
- Data bucket'larda tutulur.
- Flat container object'dır yani iç içe bucket oluşturulamaz.
- En fazla 5tb olacak şekilde obje store edilebilir.
- **Tek bir PUT ile upload edilebilecek obje max 5 GB'dır.**
- Bucket içerisinde sadece console'dan olmak koşulu ile folder oluşturulabilir.
 - Bu folder'lar gerçek folder degildir. Görünüm o sekildedir.
- Bucket sahibliği değiştirilemez.
- SDK veya API ile erişim tavsiye edilir.
 - Console'da, internally olarak API kullanılır.
- Soft limit en fazla 100 bucket oluşturulabilir ama bu rakam AWS ile iletişim kurularak artırılabilir.
- S3 bucket üzerinde access permission, version status ve storage class gibi özellikleri mevcuttur.
- **S3 bucket name (key) butun AWS region'larında unique'dır.**
 - Bucket oluşturuldan sonra isim değiştirilemez.
 - Bucket silinirse, belki bir süre sonra isim tekrar available olacaktır.
 - Bucket adı 3-63 karakter arasında olabilir.
 - Bucket adları, ona ulaşılma için kullanılan URL içerisinde de geçer.
 - Bucket adlarında upper case olmaz
 - Sayı veya küçük harfle baslayıp bitmelidir.

- "-" (hyphens) kullanılabilir
- Amazon S3, bucket'i configure etmek için çeşitli seçenekler sunar,
 - Bucket konfigurasyonu yapabilmek için, sub-resource desteği sunar.
 - S3 API, console ve SDK ile bunlar oluşturulabilir ve yönetilebilir.
 - By default; bucket, objeler ve ilgili bütün sub-resource private'dir yani oluşturana özeldir.
- Sub-resources
 - Lifecycle; objenin ne kadar süre o bucket'da kalacağı, başka bucket'a taşınacağı veya silineceği bilgisini tutar.
 - Website; Bucket için static websitesi kullanılabilir.
 - Versioning; Bucket bir objenin güncellenmiş yani farklı versiyonlarını tutabilir.
 - Access Control List (ACL); yetkileri tutar.
 - Bucket Policies; JSON formatında tutulur.
- Bucket DNS adı bucket region ve bucket adını içerir,
 - Örnek, <https://s3-eu-west-1.amazonaws.com/cloudbucket1>
 - s3-eu-west-1.amazonaws.com bolumu eu-west-1 için end point'dır.
- Performans için ve maaliyeti düşürmek için S3 bucket musteriye yakın DC oluşturulması önerilir.

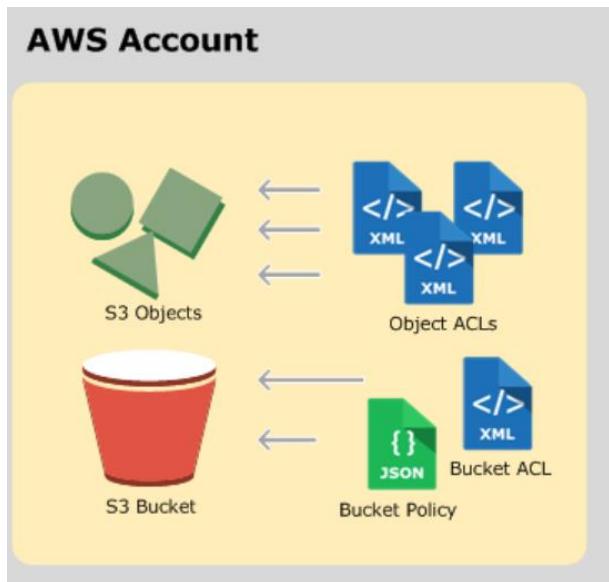
S3 Objects

- Bütün objeler bir unique key ile tutulur.
- Bir objeye erişilmek istediği zaman,
 - Service end point (objenin region bilgisi)
 - Region bilgisi, kasıtlı olarak başka bir region'a taşınmadığı taktirde değişmez.
 - Bucket adı
 - Object adı
 - Opsiyonel olarak, object versiyonu

S3 ACLs and Policies

- By default, bütün S3 resource'ları (bucket ve objects) private'dır.
 - Sadece resource sahibi, resource üzerinde yetki verebilir.
 - A account'u bucket sahibi olsun ve başka bir AWS account'ı (B) o bucket'a obje upload etmiş olsun. Bu durumda obje yetkileri B account'ındadır.
 - AWS account'ın da, AWS Identity ve Access Management (IAM) user oluşturulursa, AWS account parent user olur.
 - Kabaca, account kendi oluşturduğu user'ların parent owner'ıdır.
 - A Account'nın oluşturmuş olduğu bir kullanıcı, B account'ında obje oluşturursa, sahibi A account'ı olur.
 - Bucket owner kimin olduğu önemsemeden, bucket'dan istediği objeyi silebilir.
 - Bucket owner kimin olduğu önemsemeden, objelerein archive'ını alabilir veya restore edebilir.
 - S3 bucket veya objeler için,
 - Herhangi bir account'da yer alan kişisel userlara yetki verebilir.
 - AWS account'lara yetki verebilir.
 - Public yetkisi de verilebilir.
 - Credential'a sahip kullanıcılarla verilebilir. Kullanma hakkı olan veya abone olan kullanıcılarla.

S3 Access Policies



- Bu policy'ler, kimin S3 bucket'ina veya bucket içerisindeki objeye yetkisi olduğunu belirtir.
- Resource-based ve user policy ile mevcut policy'ler kategorize edilebilir.
- Resource-based Policies içerisinde ikiye ayrılır
 - Access Control List (ACLs)
 - Iceriginde bucket veya Object olabilir. Butun bucket ve objelere ait ACL bulunmaktadır.
 - ACL bir nevi basic yetki listesidir.
 - **Basic cunku ACL ile S3 içerisinde full yetkilendirme yapılamaz.** Bunun için resource base policy kullanılmalıdır.
 - ACL içerisinde en fazla 100 yetkilendirme yapılabilir.
 - Bucket Policies
 - Bucket policy ekleyerek, diger AWS account'larina ayrıca bucket icin veya içerisindeki objeler icin IAM kullanicisi yetkisi verilmesini sağlanabilir.
- Account A'nın bucket'ında, account B objesi olsun. Account A bunlar icin read/write/versioning delete gibi yetkiler veremez ama deny access, delete, archive gibi işlemleri yapabilir. Account B yetki vermiş olsa bile, Account A bunu engelleyebilir.
- User Access Policy
 - Amazon S3 resource erişimlerini yönetmek için, AWS Identity and Access Management (IAM) kullanılabilir.
 - IAM ile, IAM user, group ve role oluşturulabilir ve bunlara access policy ekleyerek, S3 dahil AWS resource'ları için yetki vermesi sağlanabilir.
 - Policy, kullanıcı seviyesinde verildiği için, IAM user policy içerisinde public yetki verilemez.
 - User policy'ler bucket ve içerisindeki objeler için yetki verebilir.
 - IAM user, group ve role'e verilebilir.
 - Root account validation yani yetki kontrol kısmını geçer.
 - Diğer kullanıcılar için oncelikle user context ve sonrasında bucket context kontrol edilir.
- Parent AWS account resource(bucket veya object) sahibi ise, user policy veya resource policy ile IAM user'a resource yetkilendirmesi yapabilir.
 - Bucket ve object sahibi aynı ise,
 - Objeye yetki bucket policy verilebilir.
 - Bucket ve obje sahibi farklı ise,
 - Object sahibi, yetki vermek için object ACL kullanmalıdır.
 - AWS account, objenin sahibi ise ve IAM user'in parent account'i ise, user policy ile object yetkilerini değiştirebilir.
- S3 ACLs, bucket ve objeler için erişim yönetimini sağlar.

- Butun bucket ve objeler ACL ile senkronize calisir ve buna sub-resource denir.
 - Bu sub-resource; AWS account'inin ve Pre-defined S3 grubunun sahip oldugu yetkiyi barindirir.
 - Buradan kisisel IAM kullanicilarina yetki verilemez.
- Yeni bir obje veya bucket olusturulursa, S3 default ACL olusturur ve bu ACL resource sahibine full yetki verilmesini saglar.

S3 Pre-defined Groups

Uc tip pre-defined group vardır.

- Authenticated user group
 - Bu gruba verilen yetki, herhangi bir AWS authenticated user'in, resource'a erismesini saglar.
 - Ornek olarak Netflix aboneleri
- All Users group
 - Public gruptur ve dunyadaki herkesin buna erismeni saglar.
 - AWS bu gruba write, write_acp veya full kontrol yetkisinin verilmemesini önerir.
- Log Delivery group
 - Bucket üzerinde write yetkisi verilirse, server access log yazma yetkisi olur.

Permission	When granted on a bucket	When granted on an object
READ	Allows grantee to list the objects in the bucket	Allows grantee to read the object data and its metadata
WRITE	Allows grantee to create, overwrite, and delete any object in the bucket	Not applicable
READ_ACP	Allows grantee to read the bucket ACL	Allows grantee to read the object ACL
WRITE_ACP	Allows grantee to write the ACL for the applicable bucket	Allows grantee to write the ACL for the applicable object
FULL_CONTROL	Allows grantee the READ, WRITE, READ_ACP, and WRITE_ACP permissions on the bucket	Allows grantee the READ, READ_ACP, and WRITE_ACP permissions on the object

- Object seviyesinde tek tek yetki verilmesinin bazi sinirlamaları vardır.
 - Bucket policy en fazla 20kb olabilir ve o yuzden bunun icin kullanisli olmayacaktir.
 - Object ACL en fazla 100 yetki barindirabilir.
- Bucket ACL icin sadece tavsiye edilen kullanim, S3 Log Delivery group'a yetki vererek, bucket içerisinde log objeleri yazilma durumudur.
- ACL Sinirlamaları
 - ACL ile diger AWS account'lara yetki verilebilir ancak kendi accountumiza bulunan kullanicilara yetki veremeyiz.
 - ACL ile AWS account'lara, basic read/write yetkileri verilebilir.

Ne zaman Bucket ve Ne zaman User Policy

- AWS account bucket sahibi ise ve kendi account'unda yer alan kullanicilara yetki verecek ise, her ikisini de kullanabilir.
- AWS account'i bucket ve obje sahibi ise, object yetkileri icin bucket policy yazabilir
- Cross-account seviyesinde butun S3 icin baska bir account'a yetki verilmesi isteniyorsa, Bucket policy tek secenektir.

- User policy, butun S3 operasyonları icin kullanilabilir.
- User policy, o account içerisinde bulunan userların yetkilerini yönetir. Bu yetkilendirme isini her iki policy ile de yapabiliyoruz.
- AWS account kendi account'ı içerisinde bulunan objeleri yönetmek istiyorsa, user policy kullanabilir.

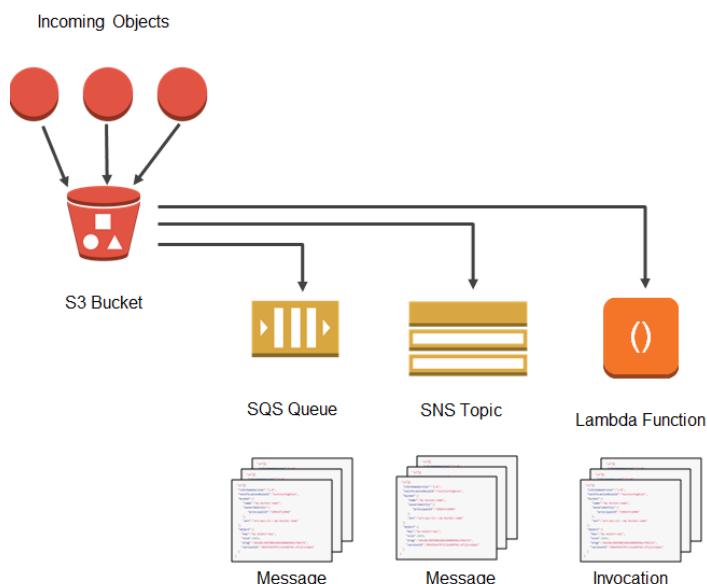
Permission delegation:

- Eger A account'i, B account'ına kendi account' icin yetki verirse, B account'i bu yetkileri dagitabilir.
 - Account B, baska bir account delegasyon yapamaz.

S3 Bucket Versioning

- Ilk amaci, kazara objenin veya datanın silinmesi veya uzerine yazılmasını önlemektir.
- Data retention ve archive icin de kullanılır.
- Bucket versioning enable edildikten sonra, suspend duruma alınabilir ama disable edilemez.
- Feature butun bucket icindir ve obje seviyesinde yapılamaz.
- Bucket versioning enable edildikten sonra, mevcut, yeni gelecek objeler ve update olan objeler korunacaktır.
 - Update anlami; objeler uzerinde uygulanacak http(put, post, copy ve delete) islemleridir.
- By default, http get komutu en son versiyonu getirecektir.
- Versioning enable olduktan sonra,sadece bucket owner kalici olarak objeleri silebilir.
 - Bucket owner disinda, obje silinmeye calisilrsa, gercektan silinmez, "Delete marker" olarak isaretlenir ve obje artik bu sekilde gorunur.
 - "Delete marker" silinebilir ve obje tekrardan available olabilir.
- Versioning cozumunu S3 lifecycle policy ile kullanilabilir.
 - Eski versiyon silinebilir veya daha ucuz bir S3'e tasinabilir.
- Uc tane versioning state'i vardir.
 - Enabled, Suspended ve Un-versioned
- Versioning enabled edilmenden once bucket'a aktarilan objelerin version ID, null olarak gorunur.
- Versioning enable olduktan sonra suspend edilirse, mevcut objeler versiyonu "as is" yani oldugu gibi kalir.
 - Bu objeler bundan sonra update edilemez ve yeni gelenlerin version ID'si de null olacaktir.
 - Yeni gelenlerin ID'si null oldugundan, bunların yeni versiyonları bir oncekini ezecektir ama enable halinde gelen objeler etkilenmeyecektir.

S3 Event Notification Destination: S3 için event notification destination olarak, AWS Lambda fonksiyonu, SQS que veya SNS topic kullanılabilir.



MFA (Multi Factor Authentication) Delete:

- Security katmanında Second layer'dir ve objeler üzerinde max koruma sağlar.
 - Bucket versiyonun degistirilmesi ve permanent olarak silinmesini engeller.
- Kullanılması için,
 - Security credential
 - Physical veya SW-based authentication device gereklidir.
- **S3 objelerinin versioning state değişimi veya objelerin permanently delete işlemleri gibi kazaları önlemek için de kullanılabilir.**

S3 Copying/Uploading S3 Objects

- Multipart Upload
 - Objelerin parçalar halinde ve paralel upload edilesini sağlar.
 - 100mb ve üzeri objeler için tavsiye edilir.
 - Desteklenen boyut, 5mb ve 5TB arasıdır.
 - 5GB üzeri objeler için kullanılmalıdır.
 - S3 multipart upload API'ları ile yapılır.
- 5GB ve altı objeler için single iş çalıştırılabilir.
- Kopyalanan yer başka bir region ise, ayrıca maaliyeti olacaktır.
- Kopyalama işlemi AWS SDK veya Rest API ile yapılabilir.
- Kopyalama işlemini,
 - Yeni bir kopya için,
 - Objeyi rename etmek için,
 - Objenin storage class'ını veya encrypt durumunu değiştirmek için,
 - Objeyi başka bir region'a taşımak için,
 - Objenin metadatasını değiştirmek için,
 - Bu işlemi source objede yapmak ancak UI ile mümkündür.
- Yeni UI ile bu işlemlerin bir çoğu yapılabiliyor.
 - Rename, delete, storage class değişimi, bazı metadata değişimleri, encryption değişimi gibi işlemler yapılabilir.

S3 Tiered Storage Classes

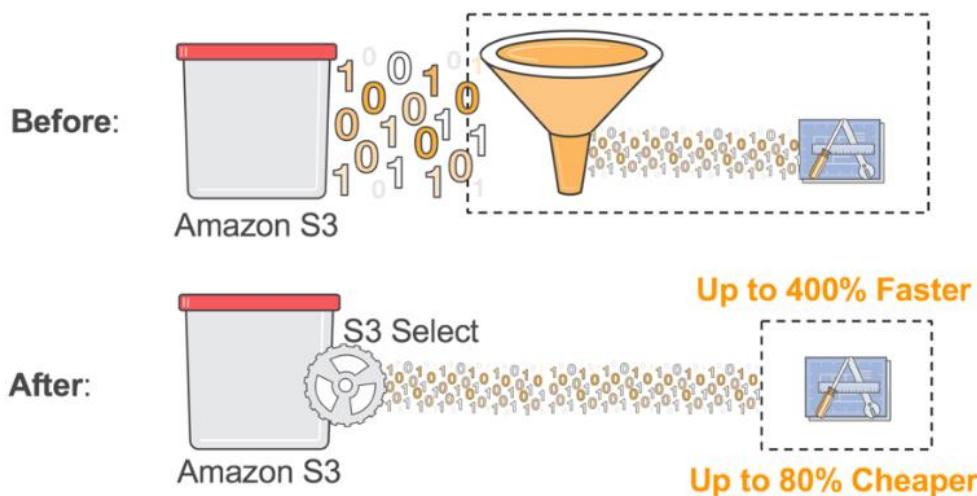
- S3 Standard Storage Classes
 - %99.99 per year availability sunar.
 - 11 9's per year Data durability (obje kaybetme riski) sağlar. (%99.9999999)
 - SSL ile data encrypted in-transit
 - Designed to sustain the concurrent loss of data in two facilities
 - Veri birden fazla bölgeye yazılmadan, available olmaz.
- S3 Infrequent Access Class (S3-IA)
 - Daha az erişilen veriler içindir.
 - Designed to sustain the concurrent loss of data in two facilities
 - Backup ve eski veriler için uygundur.
 - Daha ucuzdur.
 - %99.9 availability per year
 - 11 9s data durability
 - SSL ile data encrypted in-transit
 - Burada duracak veri en az 30 gün tutulmalıdır.
 - 128KB ve daha büyük objeler için uygundur.
 - **İhtiyaç duyulduğu takdirde, hızlı erişim sağlanabilir.**
 - **Standart gibi, low latency ve high throughput performance sağlar.**
- S3 Reduced Redundancy Storage (S3-RRS)
 - Designed to sustain the data loss in one facility
 - %99.99 availability
 - %99.99 data durability
 - Kritik olmayan veriler için uygundur

S3 hali hazırda ek bir konfigürasyon yapmaya gerek duymadan, saniyede 2000 PUT ve 3500 GET request'i karşılayabilmektedir.

S3 Cross-Region Replication Enable olması için;

- Source ve Destination bucket'da versioning enable olmalıdır
- Source ve Destination bucket farklı region'larda olmalıdır.
- Amazon S3 işi yapabilmek için yetkiye sahip olması gerekmektedir.

Amazon S3 Select



Amazon S3 select, S3 objelerinin content'lerinin içeriğini filtrelemek ve sadece ihtiyaç duyulan verinin subset'ini SQL ile almak için tasarlanmıştır.

Bu sayede datafiltrelenerek, veri miktarı azaltılabilir ve böylece maliyet ve latency azalmış olur. CSV, JSON ve Parquet formatında objeler ile çalışabilir. CSV ve JSON, GZIP veya BZIP2 ile sıkıştırılırsa, bunlar için de sonuç alabiliriz.

Python Örnek,

```
r = s3.select_object_content(  
    Bucket='example-us-west-1',  
    Key='sample-data/test.csv',  
    ExpressionType='SQL',  
    Expression="select * from s3object s where s.\"Country (Name)\" like '%Turkey%'",  
    InputSerialization = {'CSV': {"FileHeaderInfo": "Use"}},  
    OutputSerialization = {'CSV': {}})
```

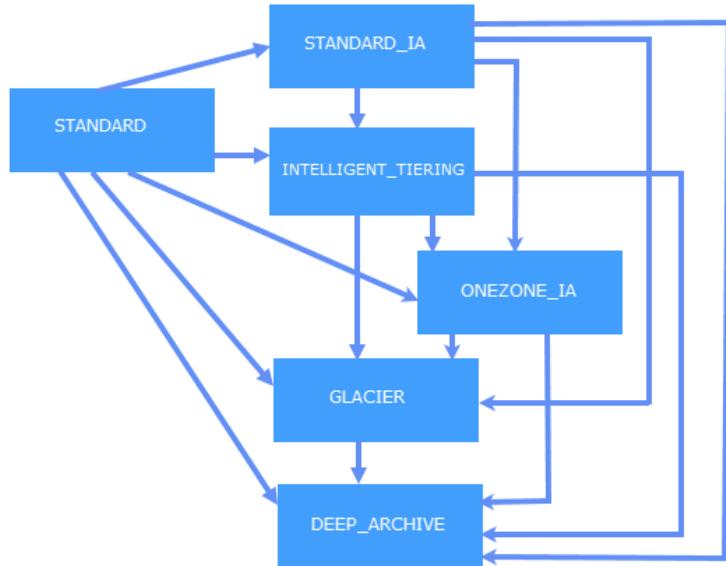
Glacier (Archiving Storage)

- Archiving storage'dır ve az erişilen veriler için uygundur.
- Bu objeler real time access için uygun değildir. Veriyi okuyabilmek için öncelikle geri restore etmek gereklidir.
 - Bu işlem bir kaç saat alabilir.
 - Request edilen data, RRS'e kopyalanır. RRS'de en fazla 24 saat süre ile kalır ve sonra expire olur.
 - Glacier'den alınan data saklanmaya devam eder.
 - Bir obje direkt Glacier'de oluşturulamaz.
 - 11 9's availability
 - No SLA
 - Designed to sustain the concurrent loss of data in two facilities
 - Burada duracak veri en az 90 gün tutulmalıdır.
 - **Buradaki veri AES-256 bit ile automatic encrypt edilir.**
 - Bütün AWS region'larda olmayabilir.
 - Glacier object metadata bilgisini almaz. Client tarafından bulunan database ile bu sağlanmalıdır.
 - Archive alınan her obje için bir index mekanizması vardır ve bu index mekanizması bu bilgiyi barındırır.
 - Hangi archive hangi objeyi tutuyor bilgisi.
 - Media, health care information gibi bilgiler için uygundur.
 - 1 byte ve 40 TB arasında objeler konulabilir.
 - 4GB'a kadar single ve 100MB-40TB atası multipart upload ile yapılabilir.
 - Upload işi synchronous, download işi Asynchronous iterler.
 - Upload olan bir content sonradan update olamaz.
 - Upload işini CLI, SDKs veya API ile yapılabilir.
 - Console üzerinden yapılamaz.
 - Bir obje lifecycle rule ile Glacier'a alınabilir.
 - Glacier yaklaşık 32KB civarında bir dosya oluşturur ve bunda index ve medata bilgisi vardır.
 - Bu işlem her obje için yapılır. Bu nedenle tavsiye edilen,
 - *****Bu şekilde küçük objelerin daha büyük bir objeye grüplendirilmesi ve o şekilde tutulur. Zira bu dosyalar için de ücret ödenecektir ve çok büyük rakamlara ulaşılırsa, size da artar.**
 - Glacier'den archive dosyalarını almanın birden fazla yolu vardır.
 - **Expedited,**
 - Data 1-5 dk arasında alınabilir.
 - Acil request'ler için kullanılır ve en pahalıdır.
 - **Standart,**
 - Daha ucuzdur.
 - Data 3-5 saat arasında alınabilir.
 - Ayda 10GB data request etmek ücretsizdir.
 - **Bulk Retrieval**
 - Data 5-12 saat arasında alınabilir.
 - En ucuzdur.
 - Birden fazla objeyi grüplayarak ve tar veya zip ile sıkıştırıp archive'lamak için, en çok kullanılan yöntemdir.
 - S3 ve Glacier http get request'i support eder.
 - Archive edilmiş dosyasının tamamı yerine, içerisinde ihtiyaç duyulan alınabilir.
 - Archive 1KB ile başlar ama 1MB artarak devam eder.
 - Aynı region'da yer alan EC2 ve Glacier arasındaki data transferi ücretsizdir.
 - Glacier'da veri tutma süresi 90 gündür. Eğer bundan önce silinirse, bu ücretlendirilir.
 - Glacier'den restore yaptıktan sonra,
 - Glacier storage alanı(zaten ödenmiş olmalı), Request ve RRS storage için ücret ödenir.

Verileri arşivlemek için glacier kullandığımızı varsayıyalım ve ihtiyaç halinde 15dk'dan daha kısa sürede ulaşmak isteyelim. Bu durumda **Expedited Retrievals** (hızlandırılmış geri alma) ile verileri hızlıca alabiliz.

Çok büyük dosyalar haricinde (250mb ve üstü gibi), 5 dk içerisinde ulaşabiliriz.

Provisioned Retrieval Capacity ile de ihtiyaç duyulması durumuna karşılık, **expedited retrieval**'ın olmasını sağlayacaktır.



S3 Standard storage class olan storage tipini, STANDARD_IA ve ONEZONE_IA için 30 gün sınırı vardır. Bu sınırlama INTELLIGENT_TIERING, GLACIER ve DEEP_ARCHIVE için geçerli değildir.

S3 Bucket LifeCycle Policies

- Bucket level bir sub-resource'dur yani konfigurasyonudur.
 - Belli bir obje, folder, belli bir tag veya specifik prefix için uygulanabilir.
- İki ayrı tanım yapılabii.
 - **Transition actions:** Belli bir periyoddan sonra başka bir S3 storage'a almak.
 - **Expiration actions:** Belirlenen süre sonunda objenin silinmesi.
- LifeCycle policy'leri archived objeler için kullanamayız.
 - Bunun için workaround, archived objeleri RRS'e alarak, objeyi kopyalamalıyız ve kopyalanana yeni storage tanımlamalıyız.
- Objenin storage class'ını, RRS olarak değiştirememiz.

S3 Encryption

- Encryption için iki yol vardır.
 - **Client side encryption**
 - Data client tarafından encrypt edilir ve S3 bucket'a o şekilde gönderilir.
 - **Server side encryption**
 - Data S3'e yazılmadan önce, S3 servisi tarafından encrypt edilir.
 - Download sırasında data tekrar de-encrypt edilir.
 - Objeye sadece bir çeşit encryption uygulanabilir.
 - Encryption key nasıl manage edildiğine bağlı olarak, 3 çeşit S3 SSE vardır.
 - **SSE-S3**
 - S3 managed encryption key ile server side encryption
 - Her obje unique key ile encrypt edilir.
 - AES-256 bit encryption
 - S3 düzenli olarak master key'i yeniler.
 - **SS-KMS**
 - AWS KMS key kullanan server side encryption
 - Customer master key kullanarak encryption yapar.
 - Default CMK key kullanılabilir veya ayrıca oluşturulan CMK key kullanılabilir.

- Bu servis ücretlidir.
- SSE-C
 - Client tarafından sağlanan key ile server side encryption
 - Diğer ikisinin aksine, client key'i yönetir.
 - Key müşteri tarafından yönetildiğinden, key kaybedilirse, veriye erişim imkansız olur.

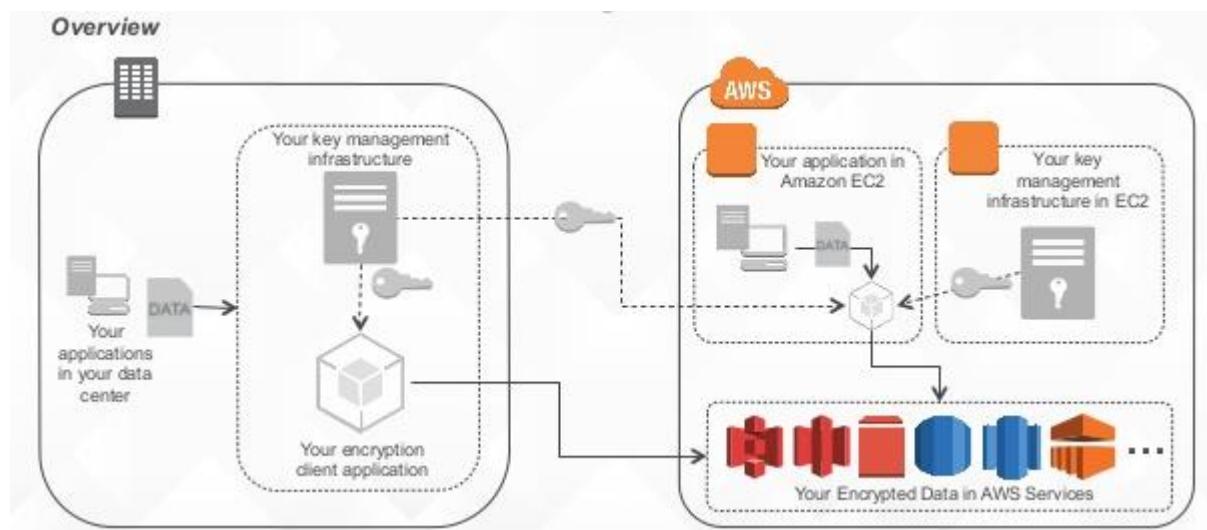
Not: Data eğer EBS'de tutuluyor ise, EBS encryption enable edilebilir. Amazon S3'de tutuluyorsa, client-side veya server-side encryption enable edilmelidir.

EBS'de bulunan belirli bir uygulama için encryption isteniyorsa, EBS encryption'dan ziyade AWS KMS (Key Management Service) API tavsiye edilir.

AWS Key Management Service (KMS): Encryption key'lerinin kullanılmasını ve yönetilmesini sağlayan multi-tenant, managed bir servistir.

EBS encryption, EBS üzerinde ekstra encryption sağlar ve bütün volume için bunu uygularken, tek bir uygulamadan gelen veri encrypt edilemek isteniyorsa, KMS API doğru tercihtir.

Clien-side encryption:



S3 Static Website Hosting&Redirection

- S3 bucket host static içeriğe sahip web siteler için kullanılır.
 - Bu tip içeriklere, PHP, .NET, JSP veya ASP server gibi scriptlerin çalıştırılmasına gereklilik yoktur.
 - Content: HTML pages, images, video, Javascript gibi client-side script'ler olabilir.
- ELB veya Auto Scaling'e gereklilik olmadan, AWS S3 automatic olarak scale eder.
- Müşteri isterse kendi domain'i de kullanabilir.
- HTTPS (SSL) connection desteklenmez
- HTML document döndürür.
- Objeler için sadece Get ve HEAD desteklenir.
- S3 bucket'da website hosting tanımlanmış ise, gelen talepleri aynı bucket içerisinde yer alan başka bir objeye veya external URL'e yönlendirilebilir.
- Bucket level'da gelen bütün talepleri başka bir websitesine yönlendirilebilir.
- Talep içerisindeki, object prefix'ine bakılarak, koşullu yönlendirme de yapılabilir.
- Gelen talepler istenirse, hata olarak da döndürülebilir.

Sharing S3 Objects va Pre-Signed URLs

- Bucket level replikasyondur.
 - Automatic, asynchronous şekilde objelerin başka bir AWS region'a kopyalanmasıdır.
 - AWS Console, CLI, SDKs ve API ile tanımlanabilir.
 - AWS S3 CRR özelliğini, S3 lifecycle management rules ile tanımlayabiliriz.
 - Replica aynı ad ve metadata bilgisine sahip olacaktır.
 - Creation time, version ID, ACL, Storage Class, User-defined metadata
 - İstenirse farklı storage class tanımlanabilir.
 - Data in-transit encrypt olacaktır.
 - Source ve destination ayrı region'larda olmalıdır.
 - Source ve destination'da versioning enable olmalıdır.
 - Sadece bir tane destination olabilir.
 - Yeni bir obje upload, silindiği veya objenin metadaya veya ACL'in de bir değişim olduğu zaman replikasyon devreye girecektir.
 - Belli bir objenin, spesific bir versiyon ID'si silinirse, destination'da bu gerçekleşmez. Bunun nedeni de, kötü niyetli olabilecek bir işlemi engellemektir.
 - Replikasyonun olabilmesi için, bucket owner'ın yetkisinin olması gerekmektedir.
 - Replikasyona dahil olmayacak da durumlar mevcuttur.
 - Replikasyon öncesi oluşan objeler,
 - Replikasyon sonrası versiyonlar, replike olacaktır.
 - İstenirse, Copy API ile öncekiler de kopyalabilir.
 - SSE-C veya SSE-KMS ile oluşan objeler
 - S3 objenin oluşumundan sonra oluşan, müşterinin sağladığı encryption key'leri saklamaz. Bu nedenle bunlar replike olmaz.
 - Object owner'i, source bucket owner'ından farklı ise ve bucket owner bu obje üzerinde yetkiye sahip değilse, bu obje replike edilmez.
 - Bucket level sub-resource update'ler replike olmaz.
 - Örnek, lifecycle
 - Sadece customer hareketleri replike edilir, lifecycle ile alınan aksiyonlar replike edilmez.
 - Manual olarak her iki tarafta aynı lifecycle olması sağlanırsa, aynı olması sağlanabilir.
- Upload request'leri, inter-region data transfer ve destination'da ki storage alanı ücretlendirilir.

S3 Cross Region Resource Sharing (CORS)

Client web application'ın başka bir domain'den resource request etmesidir.

S3 Transfer Acceleration

Merkezi bir Bucket var ise ve dünyanın her yerinden kullanıcılar buraya upload işlemi yapılıyor ise, S3 transfer acceleration kullanılabilir. Dosyalar, Cloudfront Edge lokasyon denilen, kullanıcıların yakınlarında bulunan bir staging alana atılır. Bu staging alanlardan, daha hızlı altyapı ile hedef bucket'a aktarılır.

- Edge location'dan, S3 bucket'a aktarım secure'dur.
- Kullanılabilmesi için, S3 bucket'da Transfer Acceleration enable olmalı.
 - Bir kere enable olursa, sadece suspend yapılabılır. Disable yapılamaz.
 - Yaklaşık 30dk sonra etkisi görülecektir.
- Bucket adı DNS adını barındırmalıdır ve bucket adı ve label arasında, periyot barındırmalıdır.
- PUT/GET request,
 - bucketname.S3-accelerate.amazonaws.com şeklindedir.
- Herhangi bir hız iyileştirme olmamış ise, ücretlendirilme yapılmaz.
- Cloudfront edge alanında data kaydedilmez.
- Multipart upload kullanılabilir.
- Accelerate kullanılma durumunda ve kullanılmama durumunu önceden kıyaslayabilir ve buna göre karar verilebilir.

S3 Performans Considerations – Önemi

Bucket 300'den fazla Put/List/Delete veya 800'den fazla GET request alıyorsa, AWS'e ticket açılmalı ve herhangi bir limite takılmayı engellenmelidir.

S3 Billing

- Aynı region'dan olan EC2, S3 transferinde ücretlendirme yoktur.
 - Farklı region'larda ise ücretlendirilir.
- S3 içerisinde transfer ücretsizdir.
- Başka region'lara kopyalama ücretlendirilir.
- S3, Cloudfront transferi ücretsizdir.
- Ücretli olanlar,
 - per GB/month S3 storage
 - Upload request (\$0.05/1000 requests)
 - Put ve Get
 - S3-IA ve Glacier için geri alma request'leri
- Requester ödesin durumu enable ise,
 - Bucket owner sadece S3 storage için ödeme yapar
 - Requester ise aşağıdakileri öder,
 - S3 upload/download
 - Data transfer
 - Torrent gibi public access için geçerli değildir.
 - Bucket leveldir. Object level da yapılamaz.

S3 Monitoring and Event Notification

- Ekstra bir ödemeye gerek duymaksızın, bucket level'da; SNS, SQS veya AWS Lambda function bir veya daha fazla konfigüre edilmelidir. Bu servislerin kullanılması ücretlendirilir.
- Create object, Delete object, Object delete marker oluşturulması gibi bir çok bucket ile ilişkili event ayarlanabilir.
- S3 monitoring işini CloudWatch ile yapabiliriz.
 - Birden fazla metric monitor edilebilir.
 - Sadece bir metric için action alınabilir.
 - S3 requests(Get/Put/..), bucket storage, bucket size, HTTP 4XX, 5xx hataları izlenebilir.
 - Günlük CloudWatch, Bucket-level storage metric'leri ücretsiz enable'dır. Detaylı istenirse, ücretlendirilir.
- CloudTrail S3 API'a yapılan bütün API request'leri yakalar.
 - By default bucket level request'leri yakalar ama istenirse, object seviyesinde de ayarlanabilir.
 - Delete, Get, Put, Post gibi
 - Request kim tarafından yapıldı, ne zaman yapıldı ve ne yapıldı gibi detaylar yakalanır.
 - **CloudTrail event log'ları, Amazon S3 server-side encryption kullanarak, by default encrypt'dir.** AWS KMS ile de encrypt etmek seçilebilir. Bu log dosyalarını bucket'da istenildiği kadar tutulabilir. Bu loglar için notification'da isteniyorsa, Amazon SNS notification servisi kullanılabilir.

Sınav Sorularından Notlar:

- Bir bucket'a sürekli olarak 100 put request'in üzerinde iş geliyorsa, key adlarına **random prefix** eklemek performans için faydalı olabilir.
Bu şekilde S3'de farklı partition'larda tutulabilir. Put/List ve Delete için faydalıdır.
- Aynı bucket sürekli 350 Get/s talep geliyorsa, random prefix yine işe yarayabilir ama Cloudfront kullanarak, static content'in direk S3'den değil de buradan okunmasını sağlamak, sürekli okunan verilerin cache'lenmesini sağlayacaktır ve daha faydalı olacaktır. Yükler occasionally yani bazı zamanlar gelir ise, bir şey yapılmasına gerek yoktur.

Hangi Storage?

Amazon EBS, S3 ve EFS arasından seçim yapılacaksa, kuşkusuz bu ihtiyaca göre belirlenir.

	Performance	Cost	Availability and Accessibility	Access Control	Storage and File Size Limits
Amazon S3	<ul style="list-style-type: none">- Supports 100 PUT/LIST/DELETE requests per second- Scalable to 300 requests per second	<ul style="list-style-type: none">- First 50 TB/month: \$0.0245 per GB- Next 450 TB/month: \$0.0235 per GB- Over 500 TB/month: \$0.0225 per GB	<ul style="list-style-type: none">- 99.99 percent available- Accessible via internet using APIs	<ul style="list-style-type: none">- Access is based on IAM- Uses bucket policies and user policies	<ul style="list-style-type: none">- No limit on quantity of objects- Individual objects up to 5TB
AWS EBS	<ul style="list-style-type: none">- Provisioned IOPS delivers 4000 input/output operations per second	<ul style="list-style-type: none">- Use-based cost structure that varies between regions	<ul style="list-style-type: none">- 99.99 percent available- Accessible via single EC2 instance	<ul style="list-style-type: none">- Security groups- Use-based authentication (IAM)	<ul style="list-style-type: none">- Max storage size of 16 TB- No file size limit on disk
AWS EFS	<ul style="list-style-type: none">- Up to 7000 file system operations per second	<ul style="list-style-type: none">- \$0.30, \$0.33, or \$0.36 per GB-month depending on region	<ul style="list-style-type: none">- No publicly available SLA- Accessible from multiple Availability Zones in the same region	<ul style="list-style-type: none">- IAM user-based authentication- Security groups	<ul style="list-style-type: none">- No limits on size of the system- 52 TB maximum for individual files

Amazon S3,

- Object storage service'dır.
- API ile her yerden erişilebilir.
- EBS kadar iyi latency sağlayamaz.
- Highly available ve highly scalable'dır

Amazon EBS,

- EC2 için kullanılan, block level storage'dır.
- Tek bir EC2 instance'ından, en düşük latency'i sağlar
- 16TB'a kadar arttırılabilir veya yeni bir volume eklenebilir.

EFS,

- File system interface sağlar.
- Strong consistency ve file locking sağlar.
- Binlerce Amazon EC2 için aynı anda erişilebilirlik sağlar.

Sözleşme, izin belgeleri ve finansal belgelerin tutulacağı ve işleneceği güvenilir bir Cloud mimarisine ihtiyaç olsun ve bu gibi bir durumda API'ları host edecek bir server'a ihtiyaç olacak.

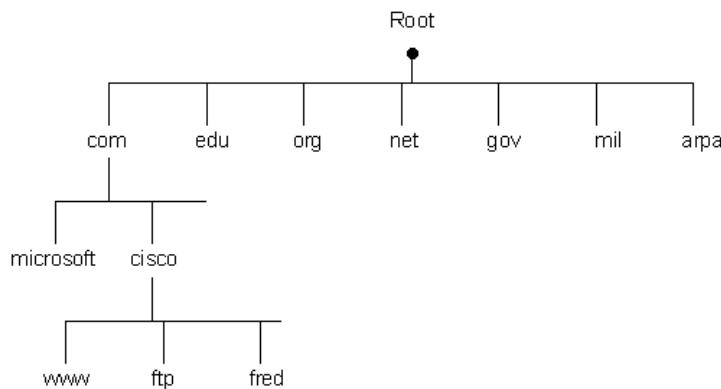
S3 static bir web sitesinin yanı sıra, storage olarak da kullanılabilir ancak gereksinim dinamik olması gereken bir online portal olduğundan, S3 belgeleri alarak çıktı oluşturmak için en iyi seçenek olmayabilir.

S3 ve DynamoDB beraber kullanılabilir ancak bu ikili yanlışca NoSQL veri tabanı sağlamakla sınırlı kalır. Bu ihtiyaç dynamic uygulamayı destekleyeceğinden, **EC2 + ELB** daha doğru bir seçenek olacaktır.

AWS Route 53

DNS server; dns tanimini, IP'ye cevirmek ile sorumludur.

- Internet üzerinde bulunan bir cihaz, web adresine ulaşırken, oncelikle bu dns tanımı dns server'a sorar ve dns server karşılığında bir IP adresine döner.
- Cihaz bu ip adresini kullanarak, erişmek istediği sayfaya ulaşır.



- DNS ağacında her şeyin üzerinde Root "." bulunmaktadır.
 - Bunun altında Generic Top Level Domains(TLDs) adı ile yüzlerce ülke spesifik, geografik spesifik TLD'ler vardır. Bu TLD'ler, root'un branch'leri yani şubeleridir.
 - Örnek; .net, .com, .org, .au, .ca, .nl
 - <https://root-servers.org/> linkinden listeye ulaşılabilir.
 - .com altına inildiği zaman, bunun branch'leri vardır. "amazon" dns tanımı .com'un bir branch'ıdır.
 - amazon altında subdomain olarak "www" vardır
 - Bu üç kısım birleşerek; "www.amazon.com.", yani Full Qualified Domain Name (FQDN) olusur.
- Client www.amazon.com.'a ulaşmak istediği zaman;
 - DNS Resolver'a(ISP) bu adresi sorar ama bu ilk bağlantı ise adres cahce'de değildir ve cevap veremez.
 - İlk olarak Root Name Server'a bu adres sorulur ancak Root Name Server'da bunu bilmez ve kendi altında yer alan .com'dan sorumlu Name Server'a yönlendirir.
 - Parent Name Server, Glue record ile artık IP adresin hangi child Name Server'a ait olduğunu bileyec.
 - Herhangi birisi bu delegation'ı yaptığı zaman, Root Server'da artık .com'u bileyec ve .com'da amazon.com'u bileyec.

Route53 - Name Servers

Route53 kullanıcı request'lerini, EC2 server'ları, ELB veya S3 bucket gibi AWS uygulamalarını bağlar ve kullanıcıları AWS dışında yer alan sunuculara yönlendirir.

- Name Server, DNS querie'lerine karşılık vermek ile sorumludurlar.
- Recursive server veya caching server, DNS Resolver'lardır.
- AWS kaynakları ile Route53'ün aynı region'da olmak gibi bir şartı yoktur.

Route53, 3 ana işlemi sağlar;

- Domain adının register edilmesi
 - Yeni bir domain register edileceği zaman, bunun register edilmesini ve kullanılabilir olmasını sağlar.
- Internet trafiğini, domain'e göndermesini sağlar.
- Kaynaklar üzerinde health check gerçekleştirir.
 - Kaynaklar; Web Server, ELB, CLB olabilir ve istenirse notification alınabilir.

- *** Route53 internet trafiğini bir domain için, register edilen başka bir domain'e gönderilmesini sağlayabilir.

Route53 Konfigürasyonu

- Bir domaine'ye register olunması gerekmektedir. Bu route53'de olabilir başka bir DNS register da olabilir.
- Route53'de Hosted zone oluşturulmalı.
 - Eğer domain route53 ile oluşturulmuşsa, bu kendisinden olacaktır.
- Hosted zone içerisinde record setinin oluşturulması gerekmektedir.
- Route53 delegate işlemi
 - Bu adımda her şeyin bağlanması ve çalışması gereklidir.

Route53 Hosted Zone: Tanımlanmış bir domain için record'ları tutar.

- Domain için ve subdomain için trafiğin nasıl yönlendirmek istediğimizi tutar.
- Public veya private hosted zone oluşturulabilir.
- Route53 otomatik olarak, Hosted Zone ile aynı ada sahip bir Name Server adı oluşturacaktır.

Supported DNS Record Types

- A Record (**Address Record**)
 - www.amazon.com IN A 2.2.2.2
 - IN=Internet A=A record type
 - IPv4 address record
 - Non-Alias type "A" ile sadece IP adresi ayarlanabilir.
- AAAA Record (**IPv6 Address Record**)
 - www.amazon.com IN AAAA 2001:d8b1::1
 - IPv6 address record, domain adını IPv6 ile eşleştirir.
- CNAME (**Canonical Name Record**)
 - web IN CNAME www.amazon.com
 - Mevcut bir kayıda, alias verilmesidir.
 - CNAME'i, DNS namespace'de top node için kullanamayız.
 - amazon.com için CNAME kullanılamaz ama www.amazon.com için veya support.amazon.com için kullanılabilir.
 - Subdomain için CNAME oluşturulursa, bu subdomain için başka bir record oluşturulamaz.
 - Route53 icindir ve dışarıdan görünmez.**
 - LoadBalancer, Cloud Front Distribution, S3 bucket ve Elastic Beanstalk için kullanılabilir ve IP adresi değişse bile hard code içinde alias kullanılabilir.
 - Real time olarak DNS querie'si calistirarak, en güncel IP adresini görür.
 - Aynı hosted zone'da tanım yapılabilir.
 - Trafiği hosted zone'da bulunan bir kayıttan başka bir kayda yönlendirmemizi sağlar.
 - CNAME Route53'deki zone apex'inde kullanılamaz.
 - CNAME, herhangi bir yerdeki DNS kaydını tutabılır.
 - Multi-AZ olan bir RDS veri tabanında, primary fail olursa, CNAME primary'den, standby'a switch olur.
 - Zone içindeki bir CNAME oluşturulamaz.
- NS Record (**Name Server Record**)
 - Name Server için delegation zone kullanılır.
 - amazon.com IN NS ns1.amazon.com
- SOA Record (**Start of Authority Record**)
 - Authority(yetki) record başlangıcı
 - Her Zone'da vardır ve sadece bir tane bulunur.
 - email owner'i kimdir, authoritative server, Zone data degimisi, refreshin time ve TTL bilgilerini barındırır.
- MX Record (**Mail Exchange Record**)
 - Mail exchanger, hangi mail server'in belli bir domain adına gönderimi belirler
 - amazon.com IN MX mail01.amazon.com
IN MX mail02.amazon.com

- CAA (**Certification Authority Authorization**)
- NAPTR (**Name Authority Pointer Record**)
- PTR (**Pointer Record**)
- SPF (**Sender Policy Framework**)
- SRV (**Service Locator**)
- TXT (**Text Record**)
- DNSSEC (**Domain Name System Security Extension**) desteklenmez.

Alias ve Non-Alias

CNAME Records	Alias Records
A CNAME record can redirect DNS queries to any DNS record. For example, you can create a CNAME record that redirects queries from acme.example.com to zenith.example.com or to acme.example.org. You don't need to use Route 53 as the DNS service for the domain that you're redirecting queries to.	An alias record can only redirect queries to selected AWS resources, such as the following: <ul style="list-style-type: none"> • Amazon S3 buckets • CloudFront distributions • Another record in the Route 53 hosted zone that you're creating the alias record in For example, you can create an alias record named acme.example.com that redirects queries to an Amazon S3 bucket that is also named acme.example.com. You can also create an acme.example.com alias record that redirects queries to a record named zenith.example.com in the example.com hosted zone.
You can't create a CNAME record that has the same name as the hosted zone (the zone apex). This is true both for hosted zones for domain names (example.com) and for hosted zones for subdomains (zenith.example.com).	In most configurations, you can create an alias record that has the same name as the hosted zone (the zone apex). The one exception is when you want to redirect queries from the zone apex (such as example.com) to a record in the same hosted zone that has a type of CNAME (such as zenith.example.com). The alias record must have the same type as the record you're routing traffic to, and creating a CNAME record for the zone apex isn't supported even for an alias record.
Route 53 charges for CNAME queries.	Route 53 doesn't charge for alias queries to AWS resources. For more information, see Amazon Route 53 Pricing .
A CNAME record redirects DNS queries for a record name regardless of record type, such as A or AAAA.	Route 53 responds to a DNS query only when the name of the alias record (such as acme.example.com) and the type of the alias record (such as A or AAAA) match the name and type in the DNS query.
A CNAME record appears as a CNAME record in response to dig or nslookup queries.	An alias record appears as the record type that you specified when you created the record, such as A or AAAA. The alias property is visible only in the Route 53 console or in the response to a programmatic request, such as an AWS CLI <code>list-resource-record-sets</code> command.

- Alias ile top node icin tanim yapabiliriz.
- Ornek olarak Load Balancer IP'si degisirse, alias'da ayni anda degisikligi gorur.
- Alias sadece secilmis AWS kaynaklari icin kullanilabilir.
- Sadece ayni Zone icin kullanilabilir.

- Use a **CNAME** record if you want to alias one name to another name, and you don't need other records (such as MX records for emails) for the same name.
- Use an **ALIAS** record if you're trying to alias the root domain (apex zone), or if you need other records for the same name.

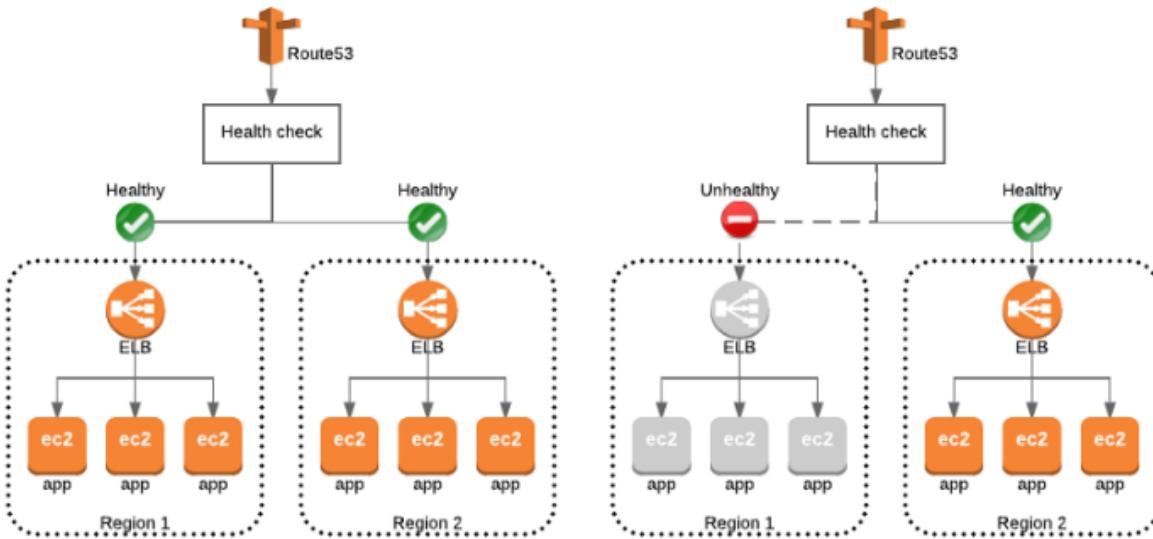
Route53 Routing Policies

- Bir record tanimlandigi zaman, route53 query'e nasil donecegine dair policy tanimlanir.
- Simple routing policy (default)
 - Tek bir resource icin kullanilir.
 - Use case: Bir web server amazon.com icin icerik sunar
- Failover routing policy
 - active-passive failover istendigi zaman kullanilir.
- Geolocation record policy
 - Bir bolgedeki kullaniciların yalnızca o bölgeye erişmesini sağlar.
- Latency routing policy
 - Birden fazla bolgede resource var ise ve talebi, en iyi latency veren resource'a gonderilmek isteniyorsa kullanilir.
- Weighted routing policy

- Trafigi birden fazla resource'a, belirlenen oranlarda gondermek icin kullanilir.
- Asagidaki feature'lar henuz yayinlanmadı.
 - Geoproximity routing policy
 - Bu policy'de vardiya usulu calisir ve trafigi bir bolgeden, bir digerine gonderilmesini saglar.
 - Multivalue answer routing policy

Route-53 DNS failover durumuna karılık da kullanılabilmektedir. Bu şekilde DNS querie'leri olası bir durumda ikinci bir region'da bulunan Route-53'e göndereilebilir.

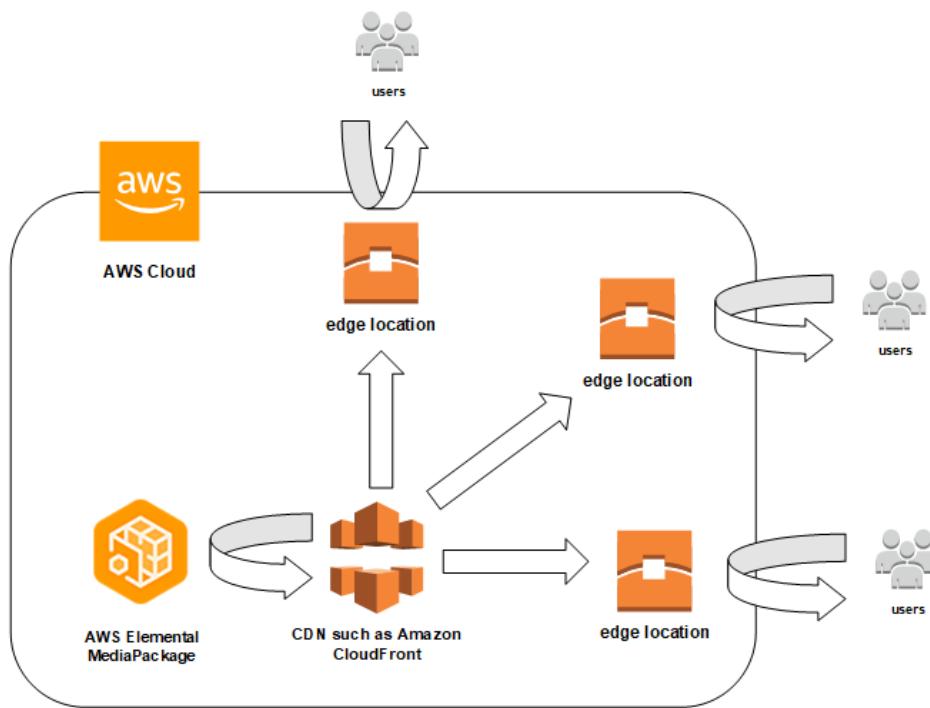
Bu konfigürasyonu yaparken tavsiye edilen, **Active-Active Weighted routing Policy**'dir.



AWS CloudFront (Content Delivery Network)

Content Delivery Network(CDN) olarak gecer ve eger bir content'e sahipseniz ve global location'da distribute etmek istiyorsanız, uygundur.

Amazon CloudFront, client'lara daha düşük delay saglamak için 30 ülkedeki 69 şehirde, oluşan global bir ağ kullanır.

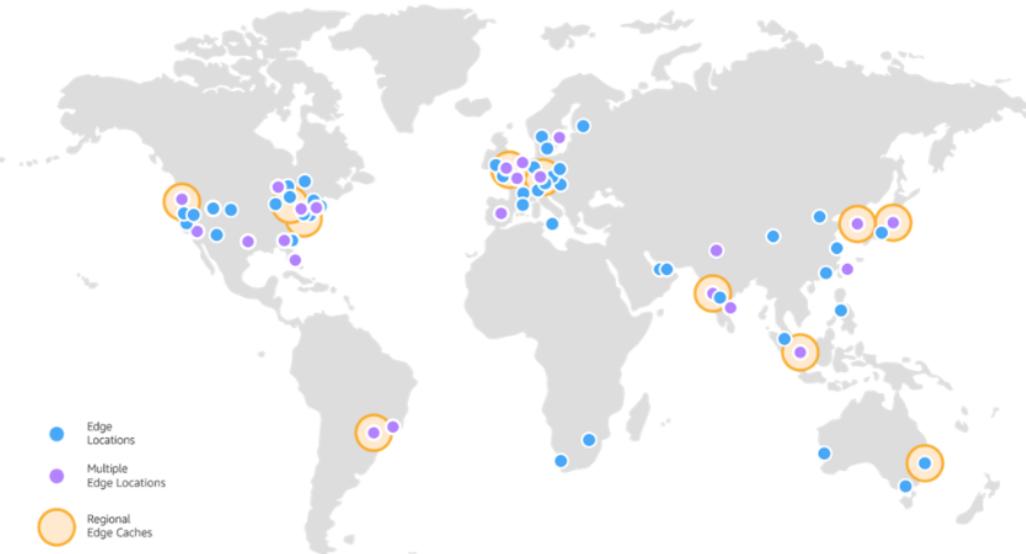


- Cache server'lar, kaynak server ile haberleserek, kullanıcının indirmek veya görmek istediği içeriği getirir.
- CloudFront objeleri upload etmek için kullanılır.
- Html, css, js veya image dosyalarının, dynamic ve static web content'lerinin dağıtımını hızlandıran bir web servisidir.
- Bu request'ler caching server'da kalır ve bir sonraki talepler buradan getirilir.
- DNS, kullanıcı taleplerini onlara en yakın caching server'a iletir.
 - Route53 ile latency based kullanılarak gerçekleşir.
- Use Cases
 - Websitesinin hızlanması
 - Customize user experience
 - Secure content
 - Stream live ve on-demand media
- AWS console, SDKs, API, Windows PowerShell için AWS tool ve command line ile bağlanılabilir.

Static ve Dynamic Content

- Web sayfaları static'de dynamic'de olabilir.
- Static content cache'lenebilir ve HTML sayfasının content'inin tek değişebilme şartı, web developer tarafından yapılacak update'dır.
- Dynamic web sayfalarına örnek; PHP, ASP ve JSP
- Dynamic content on-the-fly generate edilebilir.

AWS CloudFront Edge ve Regional Edge Cache



- Edge location, multiple edge location ve regional edge cache olmak üzere üç çeşit edge location vardır.
- Bir talep geldiği zaman önce edge location'a daha sonra regional edge cache location'da var mı diye kontrol edilir ve eğer orada var ise kullanıcıya oradan gelir ve bu sırada bir kopyası bir sonraki olası talepler için edge location'da bırakılır.
- Objeye gelen talep azalırsa, obje edge location'dan silinebilir ve yerine daha çok talep edilen konur.
 - Regional edge cache daha büyük olduğundan, talep yoğunluğuna göre orada daha fazla kalır.
- Regional edge cache, Amazon S3 için değil, custom origin için kullanılır.
 - Static website kullanılıyorsa ve Amazon S3 bu custom origin olarak kabul edilir.
- CloudFront Cache'de objelerin ne kadar süre ile kalacağını belirleyebiliriz.
 - **Cache-Control max age 0** yapılrsa, bunun anlamı cache süresinin 0 saniye olacağıdır ve data akışı Edge location'dan ziyade origin server'dan gelir.

Signed URL ve Signed Cookie

CloudFront, signed URL ve signed cookie özelliği, content'e kimin erişebileceğini kontrol eder. Eğer belli kullanıcı gurubuna, CloudFront ile URL değişmeden, signed cookie ile private content verilecek ise, kullanıcılarla set-cookie header'ı gönderilerek, content'in sadece onlar için açık olması sağlanabilir.

AWS CloudFront Distributions

- CloudFront kullanılmak isteniyorsa, distribution oluşturulmalıdır.
- En fazla 25 tane S3 veya HTTP server kombinasyonu tanımlanabilir.
- Yetkilendirme sınırlaması yapılabilir.
- **CloudFront için Origin Access Identity (OIA) oluşturarak, S3 içindeki objeler için yetki verebiliriz.**

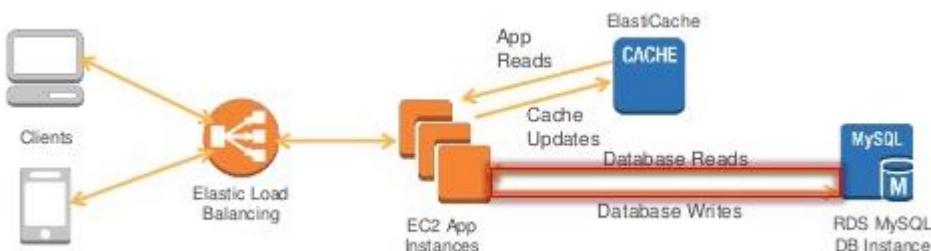
AWS Services

ElastiCache

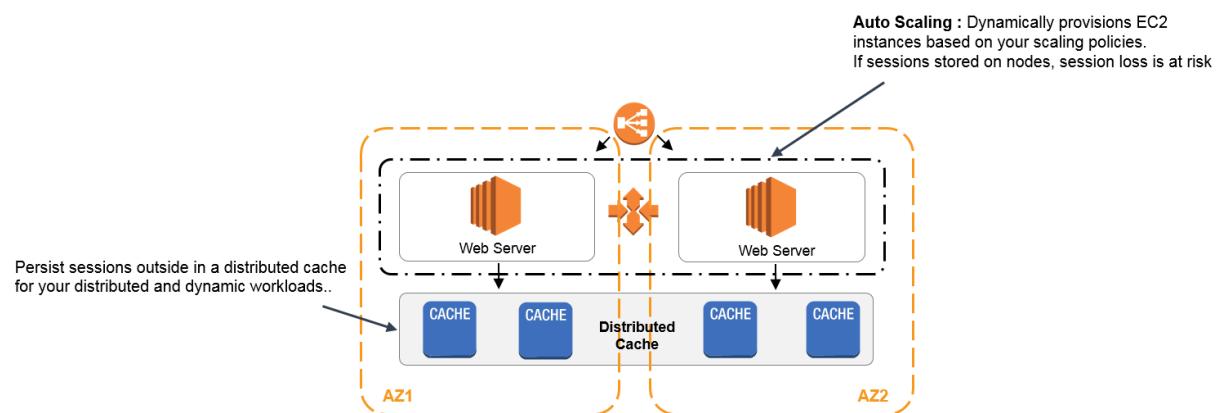
In-memory key-value store: Verinin kopyasına, çok hızlı(milisaniyeden düşük) ve pahali olmayan şekilde ulaşmayı sağlar. Web, App ve DB olmak üzere üç katmanlı bir yapı olsun. Web üzerinden alınacak verinin kaynagi database katmanıdır.

Distributed session management için en iyi tercih ElastiCache'dır.

Database ile App arasında sık kullanılan ve değişim çok seyrek olan veri memory'de tutulabilir ve database tarafından kaybedilen süre engellenmiş ve response süresi milisaniye seviyelerine düşmüs olur.



- ElastiCache EC2 node'ları, internetten veya başka bir VPC'de bulunan EC2'den erişilemez.
- On-demand veya reserved instance olabilir ancak spot instance olamaz.
- ElastiCache node'una erişim, VPC security group ve subnet group tarafından kontrol edilir.
 - ElastiCache cluster'ında subnet group'u değiştirmek şu anlık support edilmiyor.
- ElastiCache node fail olursa, otomatik olarak AWS ElastiCache tarafından replace edilir.
 - AWS tarafından fully managed bir servistir.
- ElastiCache node'larının oluşturduğu cluster; aynı subnet group'da bulunan, birden fazla subnet'e dahil olabilirler.
- İki tane destek verilen caching engine vardır.
 - **Memcached**, sadece cache mekanizmasıdır.
 - **Redis**, en hızlı NoSQL veri tabanıdır, DB olarak kullanılabilir ve open source'dur.
 - İkişi birden aynı cluster içerisinde kullanılamaz.
- Cache populate etmek için birden fazla strateji vardır.
 - Lazy loading, Write through ve TTL ekleme
- **ElasticCache ve DynamoDB session state datasını tutmak için uygundur.**

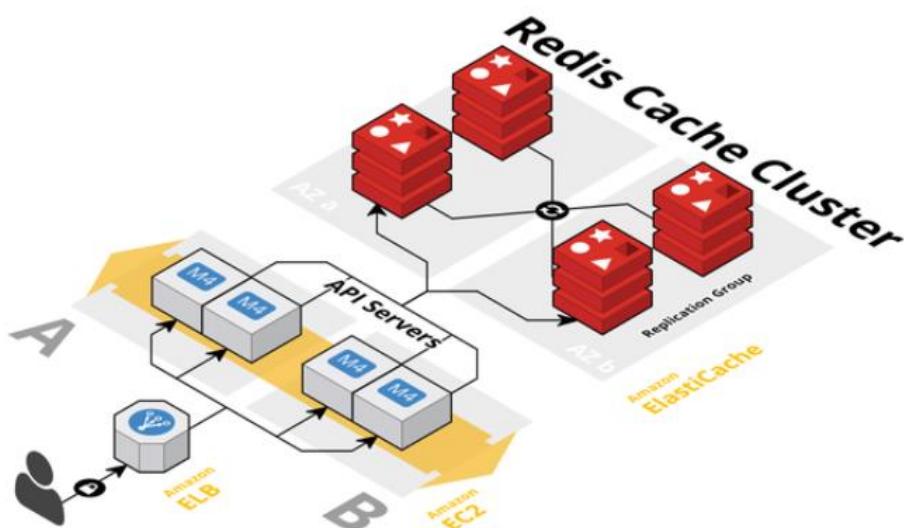


Distributed session data management için, **Amazon ElastiCache Sticky Session**'dan daha uygundur. Sticky session, session datasını yönetmek için kullanılabilir ancak bu data distribute olarak kullanılacak ise, ElastiCache kullanılmalıdır.

ElastiCache for Memcached

- Data store olarak kullanılamaz.
- Node fail olursa, o node üzerinde cache'lenmiş veri de kaybolur.
- RDS, DynamoDB,... gibi veri tabanlarının önünde kullanılır.
- Soft limit cluster basına 1-20 node arasındadır ve region basına en fazla 100 node olabilir.
- Node eklenip, çıkarılabilir.
- Yeni bir family grup cluster istenirse, mevcut cluster ile yapılamaz. Yeni cluster oluşturulmalıdır.
- Multi-AZ, snapshot ve backup/restore desteklenmez.

ElastiCache for Redis



- Data store olarak kullanılabilir.
- Automatic ve manual snapshot teknolojisi kullanılabilir.
 - Yeni bir redis cluster oluşturularak, backup ile data buraya populate edilebilir.
 - Backup sırasında herhangi bir ek API veya CLI calistirilamaz.
 - Snapshot'lar AWS console veya ElastiCache API ile yönetilebilir.
 - Snapshot'lar direkt olarak değil ama alternatif yol ile başka region'a kopyalanabilir.
 - Alınan snapshot'i aynı region'da bulunan S3 bucket'a atılır
 - Kopyalandıran snapshot istenilen region'a atılabilir.
- Master/Slave replication desteği vardır.
- Multi-AZ support edilir.
- Cluster mode disable olursa,
 - 1 primary olur ve 0-5 arası read only replica olabilir.
 - Aynı region'da, farklı AZ'da replikasyon yapılabilir.
 - Replikasyon asynchronous olarak yapılır.
- Cluster mode enable olursa,
 - 15 primary olabilir ve her replika kendine ait 0-5 arası read replikaya sahip olabilir.
 - Data 15 primary arasında dağılıbilir.
 - ElastiCache tarafından primary'nin fail olduğu tespit edilirse,
 - ElastiCache, primary'e en yakın replika'yı promote eder ve yeni primary o olur.
 - DNS kaydı aynı kalır.
 - Bu gibi durumlar için IP yerine endpoint kullanılması yani DNS kaydi tavsiye edilir.
 - Diğer replikalar, yeni primary'den okumaya başlarlar.
 - Cluster mode enable olursa, zorunlu olarak Multi-AZ failover otomatik olarak enable olur.

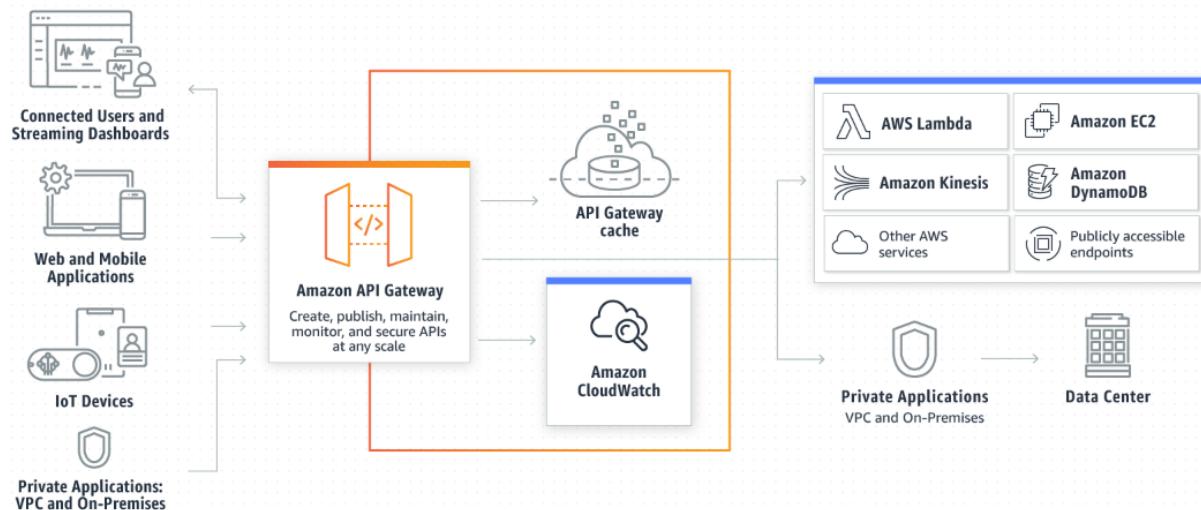
Redis AUTH komutu ilave security sağlamaktadır. Bu komut sonrası kullanıcıların öncelikle bir şifre girmeleri gerekmektedir. Bu şifre ile login olduktan sonra, Redis komutları çalıştırılabilir.

Sınav Sorularından Notlar:

- **DynamoDB**, hızlıdır, data depolayabilir ve session/state data store edebilir.
- Soruda, ElasticCache session data'sını saklar mı diye sorulursa cevap evet çünkü RedisDB'de ElasticCache'in bir çeşididir ve RedisDB data store olarak kullanılabilir.

AWS API Gateway

Endpoint: Communication yani iletişim kanalının bir ucudur.



Backend endpoint access olarak,

- Lambda fonksiyonunu çağırmak,
- Diğer AWS servislerini çağırmak,
- HTTP website veya webpage ulaşmak için kullanılır.

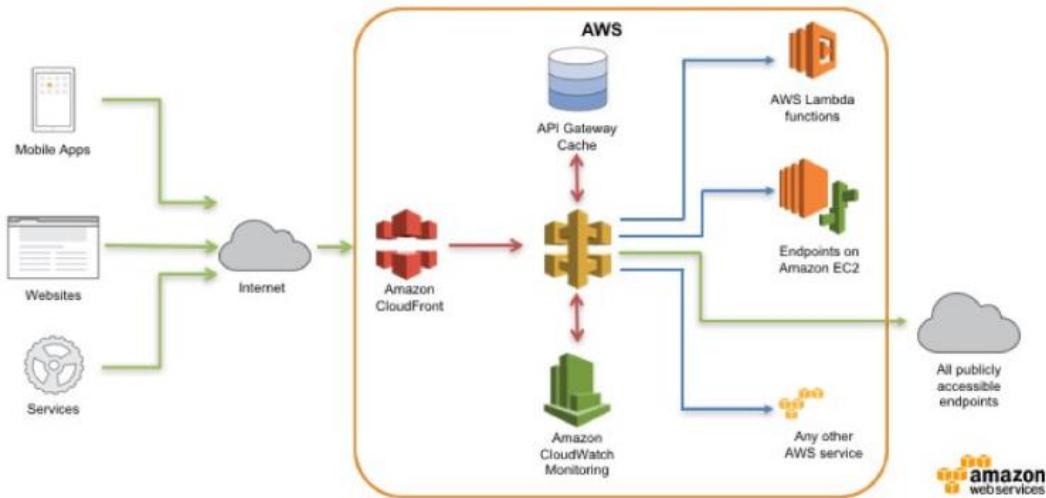
Amazon API Gateway, geliştiriciler tarafından istenen ölçüde API'ler oluşturulup yayılmasını, bunların izlenmesini, bakımın yapılmasını ve güvenliğinin sağlanması mümkün kılar, AWS tarafından fully managed'dır.

API Gateway, yüz binlerce API çağrısının kabul edilip işlenmesi için gerekli olan trafik yönetimini, yetkilendirme ve erişim denetimini, izleme, API sürüm yönetimi dahil olmak üzere tüm görevleri üstlenir.

Bütün AWS mimarisinin arkasında API işlemleri vardır. Console üzerinden yaptığım herhangi bir iş arkada karmaşık API call'ları oluşturmaktadır. Bu işlem kod ile veya HTTPS request ile gerçekleşiyor olabilir.

- API private ve public olabilir.
 - Private olan API'lar, sadece VPC interface endpoint ile olabilir.
 - Public regional veya edge optimized olabilir. (CloudFront ile)
- Mobile cihazların yayılması ve Internet of things (**IoT**) yükselişi, API ile erişilebilen uygulamaları yaygınlaştırdı.
- Amazon API Gateway kullanıldığı kadar ödenen bir servistir ve güvenliğini AWS sağlar.
 - Bu API'ları rahat kullanılması için, HTTPS istenmiyorsa, Amazon API Gateway, JavaScript, iOS ve Android de client SKDs generate edebilir.
- Amazon API Gateway ile yeni servislere hızlıca, düşük maliyet ile bağlanabilir ve kullanıcılar business servislere oluşturmaya odaklanabilir.
- **Amazon API Gateway tarafından oluşturulan bütün API'lar, HTTPS'dir.**

- API Gateway arkasında, AWS Lambda function, EC2, S3, Kinesis, DynamoDB veya başka AWS servileri olabilirken aynı zamanda 3rd party private application'lar da olabilir.
- API Gateway mimarisinin önünde CloudFront bulunur ve global bir hizmet sağlanır. Ama istenirse CloudFront kullanmadan direk client odaklı da olabilir.
- Amazon API Gateway unencrypted HTTP desteklemez.
- Prod, Dev ve Test olmak üzere API'ın birden fazla stage'i olabilir.
- API key'leri developer'lara dağıtılabılır.
- **CloudFront ile DDoS (Denial-of-service attack) feature'u vardır.**
- **Versioning destekler** yani yeni bir sürüm oluşturmak için, API clone'lanabilir.



- Gateway arkası backend side, önü ise frontend side olarak düşünübiliriz.
- Laptop, telefon veya herhangi bir cihazdan süreç başlıyor. Bizim mimarimize göre, call AWS Lambda function veya EC2'ye veya başka bir AWS servisine veya başka bir public endpoint'e gidiyor.
- Default enable değildir ama AWS **Sig-v4** authorize access enable edilerek, API yetkilendirmesi yapılabilir.
- API Gateway'de istenirse API datası için cache mekanizması enable edilebilir ve bu cache mekanizmasının size'ı da belirlenebilir.
 - Response zamanı ve backend tarafına uygulanacak load'ı düşürecektr.
 - İstenirse TTL belirlenebilir.
- Request/Response data transformasyon yapılabılır. JSON gönderip, XML almak gibi...
- API Gateway saniyede olan request'i takip eder.
 - Request sınırı verilebilir. Örnek olarak, API owner saniyede 1000 request olacak şekilde sınırlanır.
 - Bir kaç saniyeliğine 2000 request handle edilsin gibi bir konfigürasyon da yapılabilir.
- API Gateway, backed operasyonlarında bir proxy görevi de görebilir.

Cross-origin resource sharing (CORS): Browser'lardan başlatılan HTTP isteklenirini kısıtlayan bir security özelliğidir. Eğer kullanıcının REST API'ı cross-origin request'ler alıysa, enable edilebilir.

- Farklı domain'lerden kaynak paylaşımını güvenli şekilde sağlar.
- CORS üzerinde HTTP Get, Put, Post gibi methodları enable etmeden önce, API gateway API üzerinde enable edilmelidir.
 - CORS enable edilmez ise, farklı domainden gelen requestler bloklanır.

Sınav Sorularından Notlar:

Soru: API Gateway, server olmadan kodu çalıştırılabilir ve saklayabilen fully managed olan bir servis midir?

Cevap: Hayır

Bu daha çok lambda fonksiyonu tanımıdır ve kod kısmı API gateway için uygun değildir.

Amazon API Gateway Throttling

Throttling limit ayarlanarak, aşırı talep durumlarında kullanılabilir.

Örneğin, REST API'da belirli bir method için saniyede 1.000 request limiti belirleyebilir ve ayrıca Amazon API Gateway'i, birkaç saniye boyunca saniyede 2.000 request patlaması yapacak şekilde yapılandırabilir.

Throttling ile, backend systems ve uygulamalar yoğun trafiğin oluşturabileceği sorunları engellememize yardımcı olur.

AWS Lambda

AWS Lambda, sunucu olmadan kod çalıştırılmasına olanak tanır.

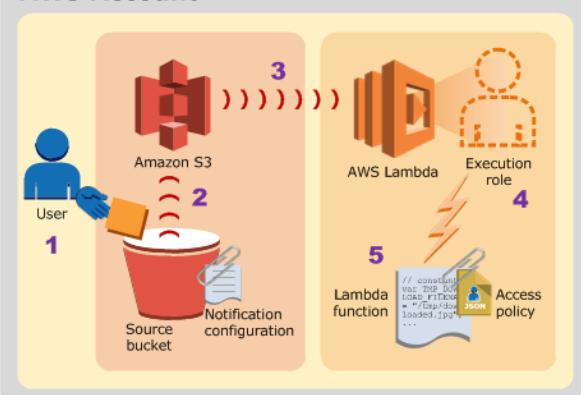
Yalnızca kullanılan işlem süresi için ödeme alınır; kodun çalışmadığı zamanlar için ödeme alınmaz.

Hiçbir management gerekmeden neredeyse her tür uygulama veya backend hizmeti için kod çalıştırılabilir. Kodun yüklenmesi yeterlidir; Lambda kodunu yüksek erişilebilirlikle çalıştmak ve ölçeklemek için gereken her şeyle ilgilenir.

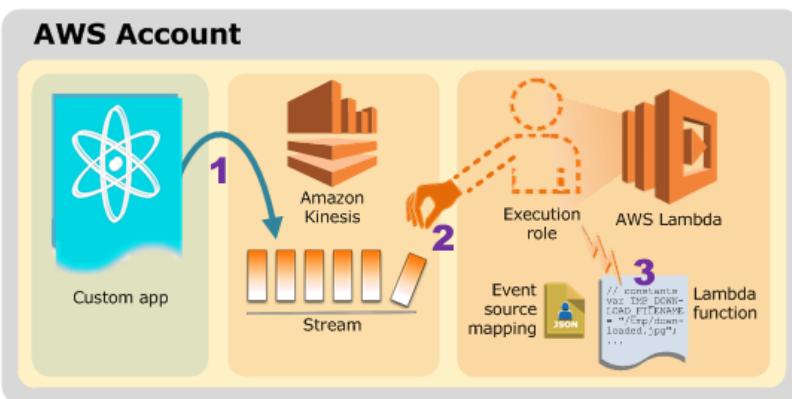
Kodun diğer AWS hizmetlerinden otomatik olarak tetiklenecek şekilde ayarlayabilir veya doğrudan web'den veya mobil uygulamadan çağrırlabilir.

- Hiç bir administration gerekmeden, herhangi bir uygulama veya backend servisi için kod çalıştırılmaya olanak tanır.
 - Provisioning ve capacity compute hesaplamaları (CPU, network, memory)
 - Server ve OS bakımı
 - High availability ve auto scaling
 - Monitoring ve logging
 - Security patch'leri
 - Kodun deploy edilmesi
- AWS Lambda kodun sadece gerekli olduğu zaman çalıştırır ve automatic olarak scale eder.
- Sadece compute sırasında ücret alınır. Kod çalışmıyorken ücret alınmaz.
- Lambda fonksiyonları yalnız da çalışabilir, DynamoDB ve Amazon S3 gibi servislerle de çalışabilir.
- Yapılması gereken sadece kodun lambda formunda supply edilmesidir.
 - Şu anlık; Node.js, Java, C#, Python, Ruby ve Go destekleniyor.
- Compute instance'a veya operation system'a bağlanılamaz.
 - Bu isteniyorsa, EC2 veya Elastic Beanstalk kullanılmalıdır.

AWS Account



- AWS Lambda trigger edilesi aşağıdaki adımlarda olur.
 - Kullanıcı bucket'da objeyi oluşturur.
 - Amazon S3 objenin oluşturulduğuna dair event oluşturur.
 - Amazon S3 execution role ile sağlanan yetkiyi kullanarak, Lambda function çağrıır.
 - AWS Lambda, parametre olarak tanımlanan lambda fonksiyonunu çağrıır



- AWS Lambda, Kinesis Stream ile trigger edileşi aşağıdaki adımlarda olur.
 - Custom app kayıtları Kinesis stream'e yazar.
 - AWS Lambda, stream'i sürekli kontrol ederek, yeni kayıt gelince, Lambda fonksiyonunu çağırır.
 - AWS Lambda, event source mapping ile hangi stream'in, hangi lambda fonksiyonuna uygun olduğunu tespit eder.
- Kodun beklenildiği gibi çalıştığını kontrol etmek için, kodun içerisinde logging statement insert edilebilir.
- Lambda otomatik olarak CloudWatch'e entegre edilir ve koddan, CloudWatch'a bütün logları aktarır.
- Lambda fonksiyonu için sadece memory tanımının yapılması gereklidir.
 - 128MB - 3GB arasındadır.
- 1536MB üstü fonksiyonlar için multiple CPU thread allocate olur.
- Max execution zamanı 900 saniyedir.
 - Ayrıca da tanımlanabilir ve bu süre sonunda AWS Lambda, lambda fonksiyonunu öldürür.
- AWS Lambda'nın, lambda fonksiyonunu execute edebilmesi için, execute yetkisinin olması gereklidir.
- Lambda fonksiyonunun, VPC içerisindeki servislere internet üzerinden erişebilmesi için konfigürasyon yapılması gereklidir. Default olarak bu açık değildir.
 - VPC subnet ID ve Security group ID konfigürasyonu yapılmalıdır.
- Lambda fonksiyonu HTTPS ile de çağrılabılır.

AWS Lambda compute platformu kullanılıyor ve fonksiyon güncellenecek ise, yeni lambda fonksiyonuna trafiğin nasıl kaydırılacağına dair aşağıdaki deployment konfigürasyonlarından birisinin seçilmesi gerekmektedir.

Canary: Yüzde belirterek, tanımlanmış yüzdelik trafiğin yeni versiyon Lambda fonksiyonuna kaydırılmasını sağlanabilir.

Linear: Trafik, her artış arasında eşit sayıda dakika olacak şekilde eşit artışlarla kaydırılır

All-at-once: Bütün trafik orjinal fonksiyondan, yeni versiyona aktarılır.

AWS Lambda Supported Services (Sınav Sorusu)

- AWS Kinesis ve DynamoDB hem stream based hem de Poll(yoklama) based servislerdir.
- Amazon SQS (Simple Queue Service) sadece Poll based servistir.
- Amazon S3
- Amazon SNS (Simple Notificaton Service)
- Amazon Simple Email Service
- Amazon Cognito
- AWS CloudFormation
- Amazon CloudWatch Logs
- Amazon CloudWatch Events
- AWS Config
- Amazon Alexa
- Amazon Lex
- Amazon API Gateway
- AWS IoT Button
- Amazon CloudFront

- *Amazon Kinesis Firehose*
- *Diğer Event kaynakları*

AWS Lambda Limits

Resource	Limit
Function memory allocation	128 MB to 3,008 MB, in 64 MB increments.
Function timeout	900 seconds (15 minutes)
Function environment variables	4 KB
Function resource-based policy	20 KB
Function layers	5 layers
Invocation frequency (requests per second)	10x concurrent executions limit (synchronous – all sources) 10x concurrent executions limit (asynchronous – non-AWS sources) Unlimited (asynchronous – AWS service sources)
Invocation payload (request and response)	6 MB (synchronous) 256 KB (asynchronous)
Deployment package size	50 MB (zipped, for direct upload) 250 MB (unzipped, including layers) 3 MB (console editor)
Test events (console editor)	10
/tmp directory storage	512 MB
File descriptors	1,024
Execution processes/threads	1,024

Lambda fonksiyonları dedicated VPC'ye direk bağlanamaz. Private VPC'ye bağlanması istiyorsak, **VPC subnet ID** ve **Security Group ID** sağlamalıyız.

AWS Lambda, ENIs (**Elastic Network Interface**) için bu bilgilere ihtiyaç duyar ve bu bilgiler ile, VPC içerisindeki diğer servislere güvenli bağlantı sağlamış olur.

Event sayısına göre Lambda fonksiyonu otomatik olarak scale olacaktır. Bu nedenle yeterli ENI kapasitesidir olduğundan emin olunması gerekmektedir.

Her AZ'da da en az bir subnet belirtilmeldir. Bu konfigürasyon doğru yapılmaz ise, [EC2ThrottledException](#) hatası alınabilir.

AWS Lambda Monitoring and Maintenance

Monitoring

- Amazon CloudWatch
 - Execute sayısı, latency per request, hata alan işler gibi monitor edebiliriz.
 - EC2 instance'larda belirli bir metriği CloudWatch yardımı ile monitor edebiliriz. **Network, CPU, Disk okumaları default olarak gelir ancak memory ölçümüleri, manual olarak ayarlanmalıdır.**

Bir dizi EC2 instance olsun ve server'larda yüksek oranda memory kullanımımasına rağmen, scaling group yeni bir instance eklememiş olsun.

Bu gibi bir durumda, instance'lara CloudWatch monitoring script yükleyerek, CloudWatch'a custom metrikler gönderebiliriz. Böylece bu metrikler, Auto Scaling scale-up işini tetikleyebilir.

Bir web uygulaması bazı hatalar alıyor olsun ve bu hatayı instance’ı restart ederek kolayca çözebiliyor olalım. Bu çözümü otomatize etmek için, CloudWatch log’larına bakarak, hatanın keyword’üne uygun bir custom metric tanımlayabiliriz.

Tanımlanan bu custom metric ile bir CloudWatch alarmı oluşturabilir ve EC2 instance’larının restart edilmesini tetikleyebiliriz.

Unified CloudWatch

Amazon EC2 instance’ları ve on-premise server’ların log’larının, CloudWatch log’larına toplanmasını sağlar. Unified CloudWatch agent ile, sadece bir agent ile her iki taraftaki de log’ları ve advanced metriklerin toplanması sağlanabilir. Windows server’lardan log toplanmasını aktif hale getirir.

- AWS X-Ray
 - Splunk gibi çalışır ve performans sorunlarını analiz etmek, saptamak ve optimize etmek için kullanılır.
 - Metadata verilerini toplar ve servislerin upstream, downstream durumlarını bildirir.
 - Request’lerini takip ve analiz edilmesi için kullanılabilir. API Gateway, AWS X-Ray’ın bütün API Gateway endpoint tiplerinin takip edilmesini destekler. AWS X-Ray, Amazon API Gateway ile, X-Ray’ın olduğu bütün region’larda kullanılabilir.
- CloudTrail
 - API call’larını yakalar ve S3’e gönderir.
 - Bu loglarda kim ne yaptı, ne zaman yaptı gibi bilgileri barındırır.
- Server Access Logging
 - Amazon S3 için AWS CloudTrail log ile birlikte Server Access Logging’de kullanılabilir. CloudTrail bucket-level’da ve object-level’da API yakalamayı sağlarken, Server Access Logging, S3’deki object-level işlerde daha fazla görünürlük sağlar.

CloudTrail ve CloudWatch birlikte kullanılarak, bucket-level API aktivitelerini CloudTrail ile yakalayabilir, CloudWatch’da tanımlanacak API spesifik email notification tanımlanabilir.

Ancak S3’e gelen bütün access request’ler (requester, bucket adı, request time, request action, geri dönüş süresi, hata kodları) isteniyorsa, object-level düzeyinde visibility gereklidir ve bunu Server Access Logging sağlayabilir.

Access logging, ELB’de by default disable olan ve isteğe bağlı olarak kullanılabilecek bir servistir. Load Balancer için access logging enable edildikten sonra, ELB bu log’ları yakalar ve bunları compress dosyalar olarak belirttiğiniz Amazon S3 bucket’da saklar.

Access Log, gönderilen request’ler ile ilgili detaylı bilgi sağlar. ALB üzerinden geçen HTTP request’leri detaylı olarak görmek istiyorsak, Access Logs enable edilmesi doğru tercih olacaktır.

AWS Lambda otomatik olarak fonskiyonları izler ve bunu by default CloudWatch'a rapor eder. Bu metrikler, toplam **invocation** (yürütme) request’leri ve hata oranlarıdır.

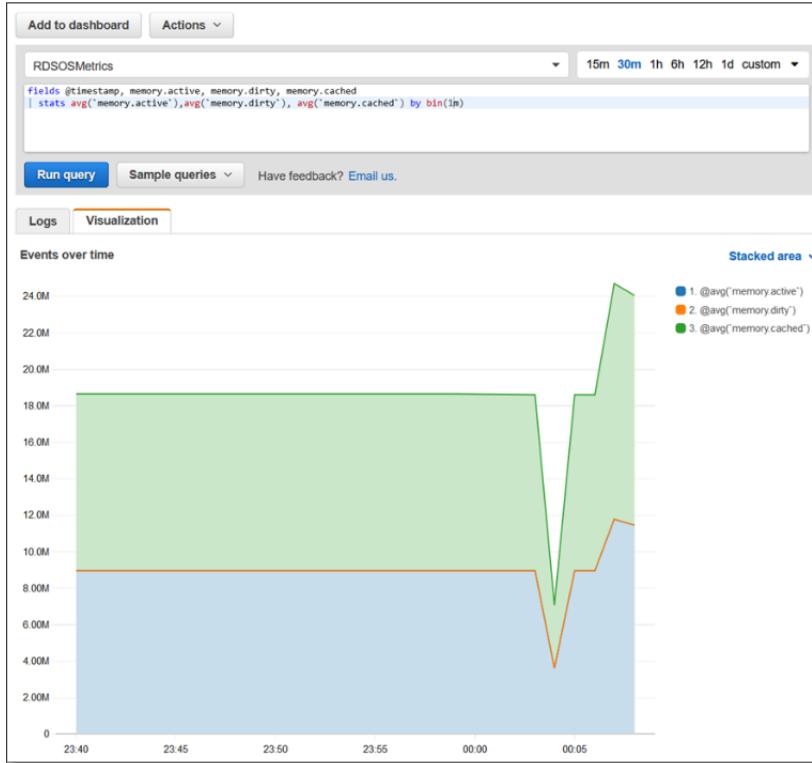
Bunları CloudWatch console haricinde, Lambda console, AWS CLI veya CloudWatch API’ı ile görüntülenebilir.

Cloud Watch Agent Log

Datayı Amazon EC2 instance’lardan, CloudWatch Loglarına göndermek için otomatik bir süreç sağlar. CloudWatch Logs Agent aşağıdaki bileşenlerden oluşur.

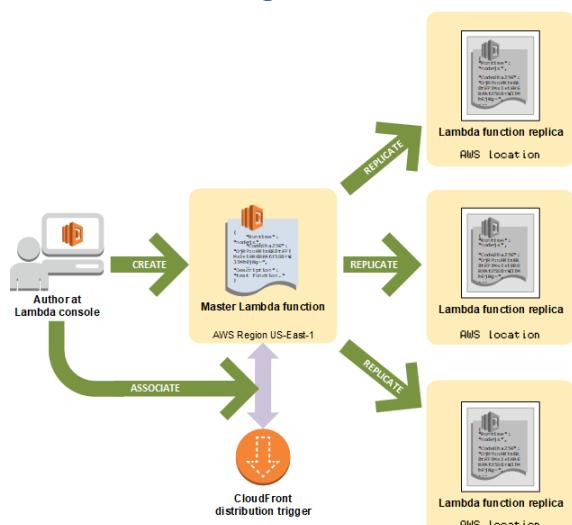
- Logları, CloudWatch loglarına ileten AWS CLI’a bir plug-in
- Logları, CloudWatch loglarına iletme işlemini başlatan bir script (daemon)
- Bu script’in sürekli çalışmasını sağlayan cron job

CloudWatch Logs Insights



Daha etkileşimli bir arama ile birlikte log datanın analiz edilmesine olanak sağlar. Operasyonel sorunları analiz etmemizi sağlayacak sorgular oluşturabiliriz. Bir sorun ile karşı karşıya kalırsak, root cause belirlemek için CloudWatch logları kullanılabilir.

AWS Lambda@Edge



Lambda@Edge, Amazon CloudFront'un kullanıcılarına daha yakın kod çalıştırmanızı sağlayan, performansı geliştiren ve gecikmeyi azaltan bir özelliktir.

- Lambda feature'larının, CloudFront ile birleşmesi ile kullanıcılara global olarak ulaşmasını sağlar.
- Sadece kodu AWS Lambda'ya upload ederek, kodun high available olarak, kullanıcılara en yakın AWS konumunda çalışmasına olacak verir.
- Bir kaç request'den, 1000 request'e kadar otomatik olacak scale olur

- Viewer'a en yakın AWS lokasyonunda request'i process ederek, latency düşürülür ve hizmet kalitesi artar.
- CloudFront event'leri, Lambda@Edge fonksiyonu ile tetiklenir.
 - Lambda fonksiyonu aşağıdaki CloudFront event'leri olduğundan çalışır.
 - Viewer Request, CloudFront viewer'dan request aldığı zaman
 - Origin Request, CloudFront request'i origine'ye forward etmeden önce
 - Origin Response, CloudFront origin'den response aldığı zaman
 - Viewer Response, CloudFront'a viewer'dan response gelmeden önce

Çalışma prensibi:

- Bir lokasyonda lambda function oluşturulur
- CloudFront distribution, cache yapısı seçilir, bir veya daha fazla CloudFront event'i tanımlanır.
- Trigger yani event tanımlandıktan sonra, Lambda fonksiyonu dünya üzerinde replike etmeye başlar.
- Yeni bir versiyon oluşturulduğu zaman, diğer lokasyonlara da replike olur.

Sınav Sorularından Notlar:

- AWS Lambda tarafından sağlanan event source mapping, aşağıdaki AWS servisleri tarafından Lambda function trigger olarak kullanılabilir.
 - DynamoDB ve Kinesis
- Non-stream event source için her event için aynı anda sadece 1 tane Lambda function event gerçekleşebilir.
 - Bir lambda function event başına bir tane yürütme gerkekleştirebilir.
 - Her event paralel olarak en fazla 1000 thread is yapabilir.

AWS Redshift

- AWS'in fully managed olan ve Cloud ortamında olan, petabyte mertebesinde veri ile çalışabilen bir DWH servisidir.
- OBBC ve JDBC kullanan, hızlı sorgulama imkanı sunan, structure veri ile çalışan bir database'dır.
- Bir çok fiziksel resource ile paralel iş yapabilir.
- AWS management console veya API ile çok hızlı bir şekilde buyutulebilir ve kucultulebilir.
- Replication block ve sürekli çalışabilen backup ile node fail durumunda veya component failover durumunda recover edebilir.
 - Backup'lar için 35 güne kadar retention tanımlanabilir. Default 1 gündür.
- Redshift'e aynı anda bir çok kaynaktan gelen veriyi store edebilir ama aynı anda bir çok kaynaktan, çok büyük veriler gönderilemez.
 - Bu işi Kinesis yapabilir.
- Klasik RDBMS'lerden 10 kat daha hızlıdır.
- AES-256 bit encryption destekler.
 - Encryption key management'i kendi yönetir ama istenirse HSM veya AWS KMS ile kullanıcı kendi manage edebilir.
 - Client uygulaması ve Redshift arası SSL encryption yani HTTPS desteklenir.
- AWS Redshift node'lara direk erişim yoktur.
- Columnar data storage'dır.
 - Data, satır sıralamasından ziyade, kolon sıralaması ile tutulur.
- Daha az I/O gereksinimi vardır.
- Advanced compression desteği vardır.
 - Otomatik şekilde gerçekleştirir.
- Massive Parallel Processing (MPP), data ve query butun node'lara dağılır ve parallel olarak çalışabilir.

AWS Redshift Backup/Restore ve Monitoring

- Single 160GB Redshift DWH node'u en küçük sistemidir.
- Cluster için, leader ve compute node ihtiyacı vardır.
 - Leader, client bağlantılarını ve query'lerin iletilmesini sağlar.
 - Compute node, query'lerin compute edilmesini sağlar.
- En fazla 128 compute node olabilir.
- Kullanıcının belirlediği periyota göre veya default 1 gün olacak şekilde backup veya snapshot retention ayarları.
 - Retention period 0 yapılrsa, snapshot disable olur.
- Eğer cluster silinirse,
 - Manual backup'lar silinmez
 - Cluster silinme sırasında, final snapshot alınabilir
- Redshift'in sadece 1 AZ desteği vardır.
- Snapshot'lar aynı AZ veya farklı AZ'da restore edilebilir.
- Compute utilization, Storage utilization ve read/write trafiği AWS Cloudwatch API veya Amazon console ile ücretsizdir.
 - AWS CloudWatch ile User-defined metrikler de eklenebilir.

AWS Redshift HA, Data Durability, Scaling ve Billing

- Verinin en az üç kopyası tutulur.
 - Orijinal data
 - Compute node'un replikası
 - S3 üzerindeki backup
- Asynchronously olarak snapshot'lar başka bir region'da bulunan S3'e replike edilebilir.
- Fail node olursa, otomatik replace edilir.
 - Yeni node DB eklenene kadar, cluster querie'ler için erişilemez olacaktır.
- Redshift en kısa sürede replacement node'u ayaga kaldırıracak ve ilk olarak en çok erişilen verileri S3'den geri getirecek ve en kısa sürede available olacaktır.
- Single node cluster'lar, data replikasyonu desteklemez.
 - Failover durumunda, cluster snapshot'dan restore edilmelidir.
- Scale sırasında yeni cluster kurulup, data tasınana kadar, cluster'a kısa süreli erişim olmayacağından emin olmalıdır.
- Leader node'dan ücretlendirme olmaz, compute node'lar per node/per hour şeklinde kullanıldığı kadar odenir.
- Backup'lar için S3 kullanım ücreti odenir.
- Aynı AWS region'da, Redshift'den, S3'e data transferi ücretsizdir. Diğer veri transferleri ücretlidir.

Amazon Redshift için disaster recovery planı yapılacak ise, **Cross-Region snapshot** enable edilerek, snapshot'ların tanımlanan region'lara kopyalanması sağlanır.

Enable edildiği zaman, bütün yeni manuel ve automatic snapshot'lar belirlenen region'a gönderilir.

Redshift WLM (Workload Management)

Bir parametre group oluşturulduğunda, by default aynı anda 5 query çalışabilen bir que yapısı oluşur. WLM özellikleri değiştirilebilir ve daha fazla que eklenebilir.

Eklenen her que, değiştirilmediği taktirde default WLM konfigürasyonu kullanır. WLM konfigürasyonu değiştirilmek istenirse, bir parametre grubu oluşturulmalı ve bu grubu custom WLM konfigürasyonuna ilişkilendirilmelidir.

Bu sayede, que'da bulunacak querie sayısı ve yönlendirmeleri belirlenebilir

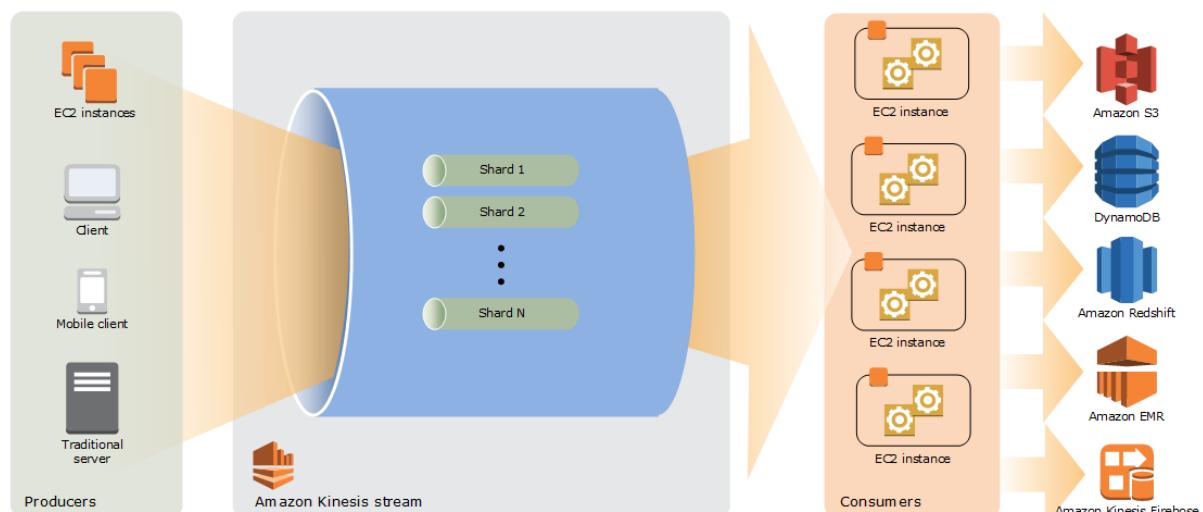
Amazon Redshift'de, Enhanced VPC routing kullanıldığı zaman, VPC ve data repository arasında bütün copy ve unload trafiği yakalanabilir.

AWS Services Kinesis

Streaming of Data (veri akışı);

- Data generate edilir ve bir çok veri kaynağından(1000 veya 100 binlerce), sürekli olarak küçük veriler halinde(Kb veya MB) gönderilir.
- Kinesis, IoT ve Bigdata Analytics için datayı stream eden yani akışını sağlayan platformdur.
- AWS managed bir servistir.
- Saatte 100 binlerce veri kaynağından terabyte mertebesinde veri yakalar ve depolar.
- Birden çok veri kaynağından, aynı anda data getirmekten söz ediliyorsa, Kinesis en uygun yoldur.
- Üç tane managed servis barındırır;
 - Kinesis Streams
 - Kinesis Firehose
 - Kinesis Analytics
- Kaynak verisine örnekler;
 - IoT Sensor verileri
 - Müşterinin mobil ve web uygulamalarından gelen log dosyaları
 - e-ticaret satın almaları
 - Oyun içi aktiviteleri
 - Social media
 - Finans, hisse senedi verileri
 - GSM GPRS verileri

Kinesis Streams



- Producers, veri kaynakları
- Consumers, Amazon Kinesis Stream'den verileri alır
- Shard, veri gruplarını belirler
 - Her bir shard 1MB/s input ve 2MB/s output'a kadar iş yapabilir.
 - Saniyede 1000 Put records destekler
- Real time olarak büyük çaplı verilerin alınması ve process edilmesini sağlar.
- Process edien veriler;
 - Dashboard'lara gönderilebilir.
 - Alarm generate edilmesi sağlanabilir.
 - Dinamik olarak değişen fiyatları analiz edebilir.
- Kinesis Stream kendisi gidip veriyi S3, DynamoDB veya başka storage alanlarına yazmaz. Bunlar için Kinesis Firehose kullanılmalıdır.
- Retention period default 1 gündür ve ilave ödeme ile 7 güne kadar uzatılabilir.

- Amazon Kinesis Stream, data producer'dan gelirken, otomatik olarak veriyi encrypt eder.
- Encrypted stream'i yazarken, producer ve consumer uygulamaları master key'e ulaşabilmelidir.
- Amazon Kinesis Streams three facilities, synchronously olarak replikasyon yapabilir.

Kinesis Data Stream, bir veri kayıt sırasına sahiptir. Datalar, Kinesis Data Stream tarafından atanın bir sequence numarasına sahiptir. Bu sayede, gönderilen yüksek hacimli mesajları teslim alabilirler.

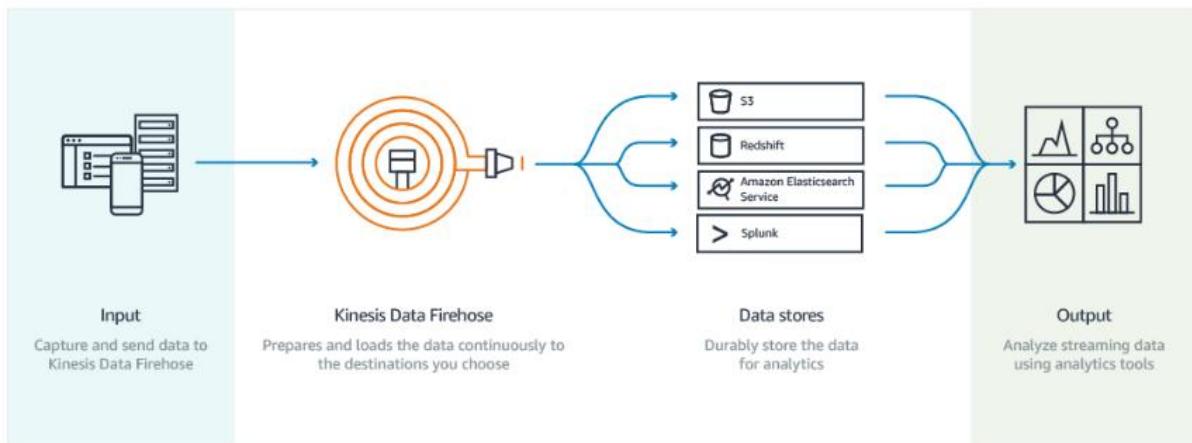
Böyle bir yapıya örnek verecek olursak, günde binlerce mesaj alınması gereken bir yapı olsun ve bu veriler process edilmek üzere Amazon EMR'a gönderilmesi istensin.

Kinesis Data Stream bu işi yaparken, mesajların hiç biri kaybolmaz, hiç bir mesajda duplication oluşmaz ve EMR'da bu mesajları sequence numarası sayesinde aynı sırada işleyebilir.

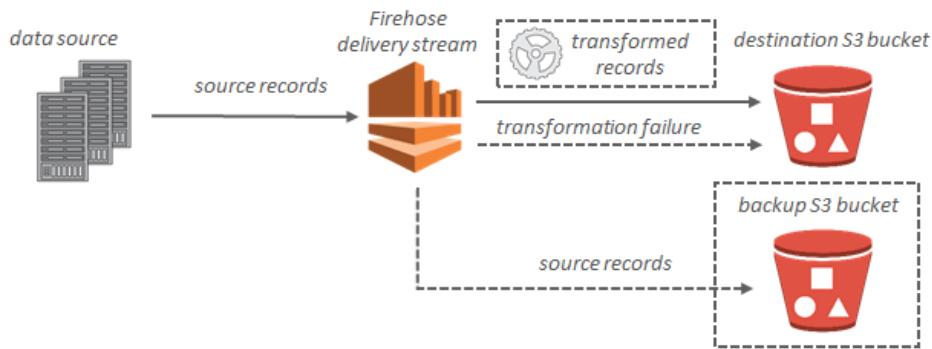
Kinesis Data Firehose

Amazon Firehose ile kolayca veri yakalanabilir, transform edilebilir, streaming veriyi S3'e atabilir.

Delivery service gibi çalışmaktadır ve adı yangın hortumundan esinlenmiştir. İstenilen bir yere data püskürtüyormuş gibi düşünebiliriz.



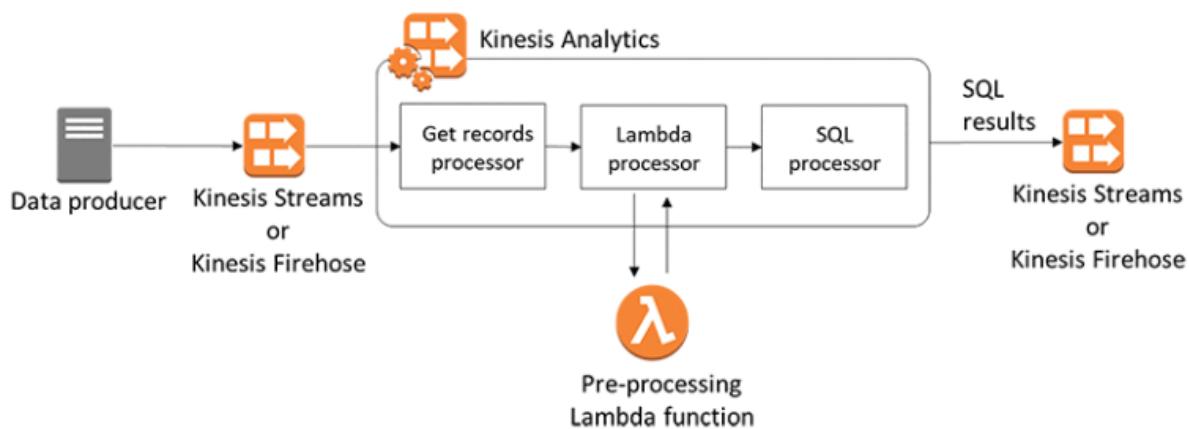
- AWS tarafından Fully managed'dır.
 - Real-time olarak veriri otomatik yakalayabilir.
- Real-time streaming verisini Amazon S3, Amazon Redshift, Elasticsearch ve Splunk gibi hedef noktalara taşımmasını sağlar.
- **Kinesis firehose veriyi taşımadan önce transformasyon işini de yapabilir.**
- Uygulama yazılımasına veya resource'un manage edilmesine gerek yoktur.
- Veriyi sıkıştırılabilir, encrypt edebilir ve böylece veri destination'a yazılmadan önce güvenliğini sağlamış ve veriyi küçültmüştür.
- Elastic olarak scale olabilmektedir.
- AWS Kinesis Firehose veriyi three facilities, synchronously olarak replikasyon yapabilir.
- Destination erişilemez olursa, 24 saatte kadar veriyi bünyesinde tutabilir.
- Örnek olarak aşağıdaki akışı düşünebiliriz.



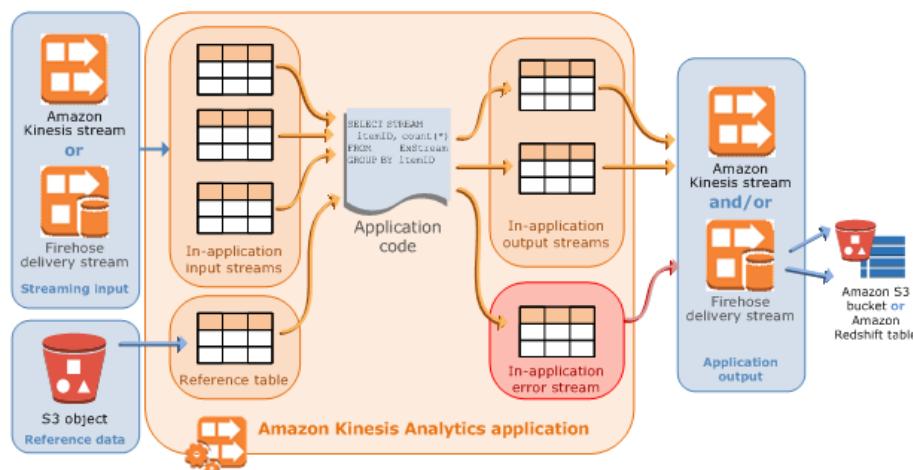
- Data transformasyonu enable ise, source data başka bir S3 için backup alınabilir.
- Server side encryption enable edilebilir.
 - Bu ancak Kinesis stream data source ise olabilir.

Kinesis Analytics

- AWS tarafından Fully managed'dır.
- Amazon Kinesis Analytics, standart SQL ile veriyi process ve analiz eder.
- Source Kinesis Stream ve Kinesis Firehose olabilir.



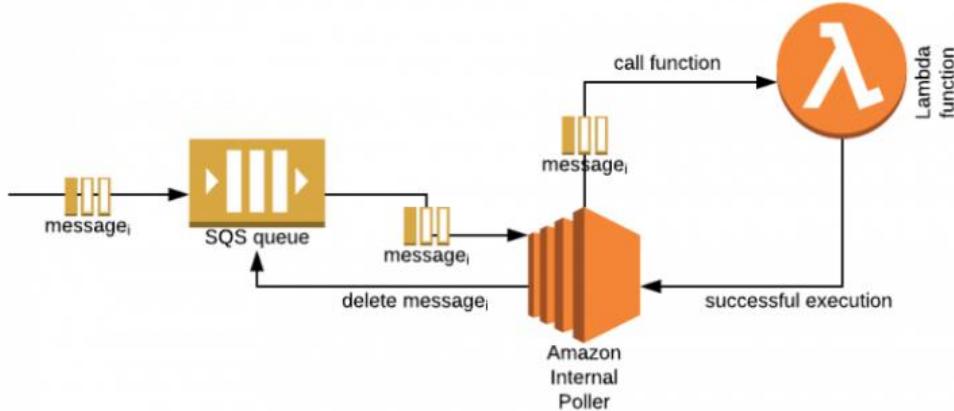
- Destination olarak da; Amazon S3, Redshift, Elasticsearch ve Kinesis Stream olabilir.



- Application code, SQL kodudur.
- Kinesis Analytics, source'dan okumak için ve destination'dan yazmak için yetkiye ihtiyacı vardır.
 - Bunun için IAM role kullanılabilir.

- Near real time olarak sürekli çalışabilir.
- Use cases;
 - Zaman bazlı analytics
 - Real-time dashboard'lar
 - Real-time metrikler oluşturulabilir.

AWS Services Simple Queue Service (SQS)

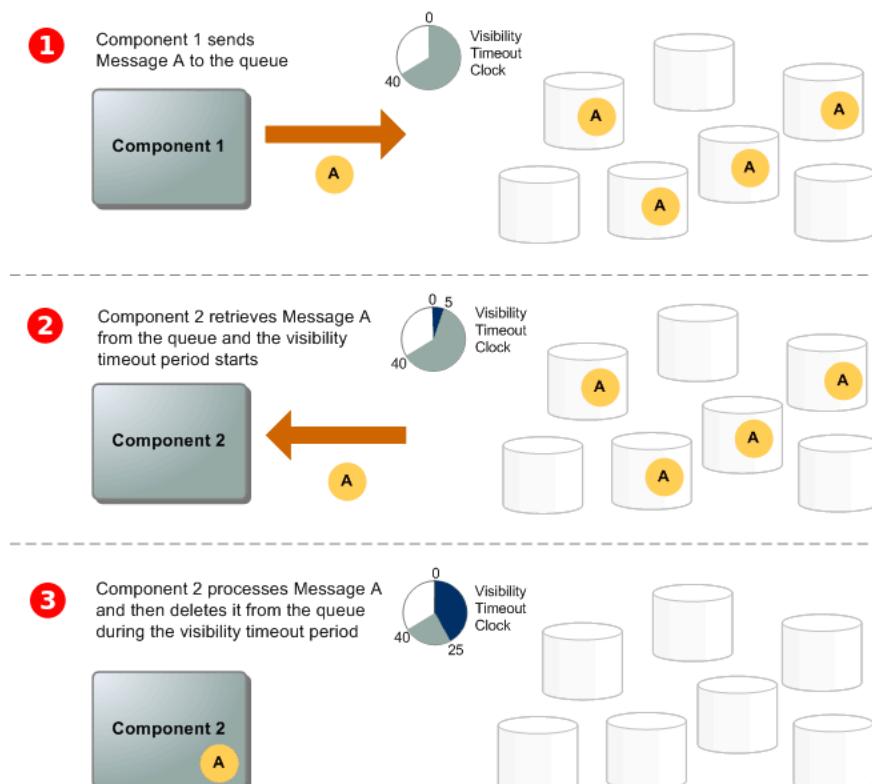


- SQS uygulamalar arası gerekli ayrimı sağlayarak, uygulama component'larini yatay ölçeklenmesini kolaylaştırır.
- Hızlı, güvenilir, fully managed message queue servisidir.
- İşlenmeyi bekleyen mesajları saklayan ve mesaj que'larına erişmeyi sağlayan bir web servisidir.
- Application ayrışmasına izin verir bu şekilde bir component'de bir hata meydana gelirse, çok büyük sorunlara neden olmayacağından emin olabiliriz.
- SQS queue mesajlarını okumak ve process etmek için, EC2 instance'i üzerindeki uygulamaları kullanılabılır.
- SQS mesajlarını işleyen, EC2 processing için Auto Scaling kullanılabilir.
- EC2'de bulunan bu uygulamalar, SQS message/job process edebilir ve bu sonuçları başka bir SQS que'ya veya başka bir AWS servisine gönderebilir.
- Birden fazla farklı SQS tipi vardır;
 - Standart Queue
 - Çok yüksek (unlimited) throughput sağlanır.
 - En az bir kez ulaşılması sağlanır.
 - Duplicate olabilir.
 - Best effort ordering
 - FIFO (FirstInFirstOut) Queue (Bütün region'larda yoktur.)
 - Limited throughput - saniyede 300 transaction
 - **Sadece 1 kere processing**
 - Duplicate olabilir.
 - Sıkı sıralama (First-in-First-out)
- 1 milyon request başına ücretlendirme yapılır.
- Request herhangi bir SQS action'ı olabilir.
 - Maximum 256kb olacak şekilde, 1-10 arası mesaj olabilir
 - SQS mesajları; max 256kb veya 10 mesaj olacak şekilde gönderilebilir, alınabilir ve silinebilir
 - Her 64kb bir chunk'dır ve bir chunk bir request'dır.
 - SQS mesaj boyutu, 1kb-256kb arasında olabilir.
- Store etmek için S3 kullanılıyorsa, S3 için de ücretlendirme yapılır.
- Aynı region'da bulunan SQS ve EC2 arası data transferi ücretsizdir.
 - Aynı region'larda ise, her iki tarafında data transfer rate'leri ile ücretlendirilir.

SQS Polling types and SQS Timers

- SQS polling base bir servistir.
- Short(default) ve long olmak üzere, iki tip polling vardır.
 - Short que'da, que boş olsa dahi, request hılcıca dönmelidir.
 - Sadece available olan ve server'larda, bir subset sorgulanır.
 - **ReceiveMessageWaitTime** 0'dır.
 - Long que daha az maaliyetli ve daha az request gönderir.
 - Boş response'ların hepsini eler.
 - ReceiveMessageWaitTime en fazla 20 saniye olabilir ve arada istenen değer atanabilir.
- Que'da olan mesajlar için retention period default 4 gündür ve 1 dk ile 14 gün arasında ayarlanabilir.
 - **Retention zamanından sonra otomatik silinir.**
- Mesajların que'ya gönderilmesi ve okunması aynı anda yapılabilir.
- SQS; Redshift, DynamoDB, EC2, ECS, RDS, S3 ve Lambda ile kullanılabilir.
- Birden fazla SQS farklı priority alacak şekilde tanımlanabilir.
 - Bu durumda, app component'i önce yüksek priority olanı alır ve hiç high priority job kalmaz ise normal priority'e geçer.
- SQS visibility timeout;
 - Birden fazla que process'in aynı mesajı okumaması için, ilk que process'i (consumer) mesajı okuduğu zaman, visibility timeout parametresi kadar read lock koymasıdır.
 - Bu süre zarfında, mesaj yerine ilettilmiş olabilir ve bu durumda mesaj artık silinebilir.
 - Process visibility timeout'dan daha fazla sürebilir ve bu durumda süreç tamamlanmadan read lock kalkabilir.
 - Bu süre zarfında, mesajın yazılması beklenen EC2 process sırasında fail olabilir ve bu durumda mesaj bir süre sonra unlock olur ve başka bir process mesajı iletmek için süreci tekrar başlatır.
 - Max 12 saatdir
 - Consumer'ın mesajı process etmek için daha fazla süreye ihtiyacı var ise, visibility timeout'u değiştirebilir.
 - 15 dakikaya kadar mesajlara delay koyulabilir.

SQS Message Lifecycle



- Bütün mesaj que'ları tek bir region'da, HA olarak ve birden fazla AZ'da tutar.
- Yetkilendirme için IAM policy kullanılabilir.
 - Kim que'ya mesaj gönderebilir, kim mesajları alabilir gibi...
- HTTPS destekler
- Server side encryption yapılabılır.

SQS Limits, Queue Names ve Logging

- In-flight messages
 - Mesaj consumer application'a iletilmiş ama halen silinmemiş olabilir.
 - Standart que için bu rakam 120.000 ve FIFO için 20.000'dir
 - Bu rakamlara ulaşılırsa, OverLimit hatası alınır ve process edildikten sonra mesajların silinmesi gereklidir.
 - FIFO için hata dönmez.
- SQS que adları region'da AWS accountunda unique olmalıdır ve 80 karakter olabilir.
- SQS mesajları region dışında paylaşılamaz.
- AWS SQS, CloudWatch üzerinden izlenebilir.
 - Her 5 dakikada bir ücretsiz olarak metrikleri gönderir ve detailed monitoring şu anlık available değildir.
- AWS SQS, Cloud Trail'in yakadığı API call'ları ile loglanabilir.
 - Request'i kim yaptı, hangi IP'den yapıldı, ne zaman yapıldı gibi...

AWS Services DynamoDB

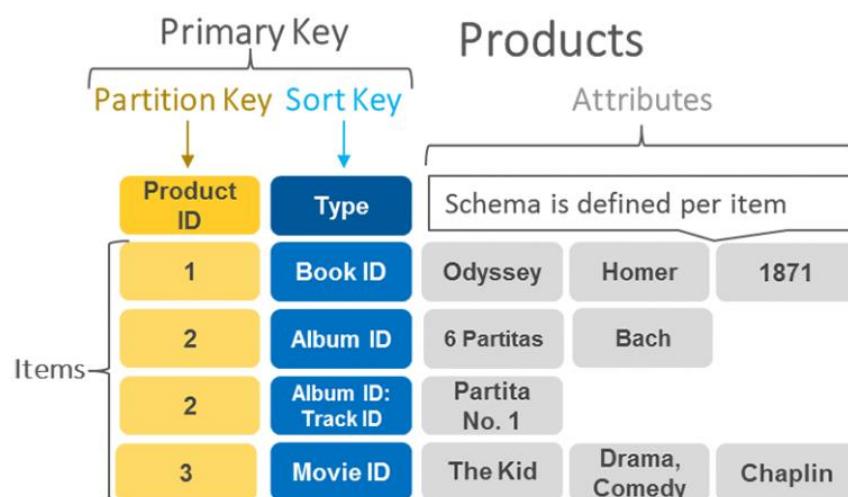
Unstructured Data: Tipik olarak text içerir ama tarih, rakam, email mesajları, word dokumanları, video, photo, audio, sunum, web sayfaları da içerebilir.

Semi Structured Data: Relational database gibi degildir ancak daha kolay analiz için, organizasyon özellikleri barındırır. Örnek olarak XML veya JSON dokumanını verebiliriz.

DynamoDB bir NoSQL veri tabanıdır yani unstructured veri barındırır. NoSQL veri tabanları schema değişimine daha elverişlidir.

NoSQL veri tabanlarında, relational veri tabanları gibi katı schema mantığı yoktur. Sıklıkla değişen schema yapısı ile baş edebilir. Tablo ve collection'a rahatlıkla satır ve element eklenebilir.

Business; high-traffic querie'lere, low latency gereksinimi var ise, NoSQL daha uygun bir seçim olacaktır.



DynamoDB schema flexibility sunar. Daha fazla önemli queri'lere göre tasarlanabilir. Complex ve hiyerarşik yani aşamalı veriyi tek bir item'da saklayabilir.

- Fully managed'dir ve hem document hem de Key-Value store support eder.
- Cok hızlı ve beklenen performansi sunar.
- Use cases:
 - Mobil, Web, Oyun, Ad-tec uygulamaları
 - Internet of things (IoT)
- Schema mantığı flexible'dır. Her item kendi attribute sahiptir.
- Complex query ve join destegi yoktur.
- Three facilities (veri 3 kopya halinde tutulur) olarak otomatik data replikasyonunu yapar.
- Herhangi bir failover durumunda, otomatik failover gerçekleşir.
- SSD volume ile calisir.
 - Low latency
 - High I/O
- Read/Write kapasitesine göre, number of server belirlenebilir.
- Read consistency icin, eventualy (default) veya strong consistency destegi vardir.
- ElasticCache, DynamoDB session state datasını tutmak için uygundur.

Tables:

{ "PersonID": 101, "LastName": "Smith", "FirstName": "Fred", "Phone": "555-4321" }
{ "PersonID": 102, "LastName": "Jones", "FirstName": "Mary", "Address": { "Street": "123 Main", "City": "Anytown", "State": "OH", "ZIPCode": 12345 } }
{ "PersonID": 103, "LastName": "Stephens", "FirstName": "Howard", "Address": { "Street": "123 Main", "City": "London", "PostalCode": "ER3 5KB" }, "FavoriteColor": "Blue" }

- Butun tabloların kendi item'i vardır.
- Item bir nevi row'dur.
- Item en fazla 400kb olabilir.
- 1 tabloda bulunacak item sayısı için limit yoktur.
 - Tabloda bulunan bazı itemların altında da başka bilgiler olabilir yani nested attribute olabilir.
 - Address altında, Street bolumu olabilir ve en fazla 32 level olabilir.
- **S3'de bulunan objeler için store pointer olarak kullanılabilir.**
- Tabloda bulunan veriyi PK index yapısı tutulmalıdır.
- PK kullanılarak, GET/PUT işlemleri yapılabilir.
 - PK alanında birden fazla kolon olabilir ve PK kolonları tabloda bulunan her item'da bulunmalıdır.
- **DynamoDB'den veri okurken, kullanıcılar eventually consistent veya strongly consistent belirtebilirler.**
- Read Capacity
 - Eventually consistence istenirse, 1 Read Capacity Unit 8 kb/sec performas gösterir.
 - Strong consistence istenirse, 1 Read Capacity Unit 4 kb/sec performas gösterir.
 - 1000 kb/sec Read kapasitesi istenirse ve strong consistence istenirse; $1000/4=250$ RCU (ReadCapacityUnits) gereklidir.

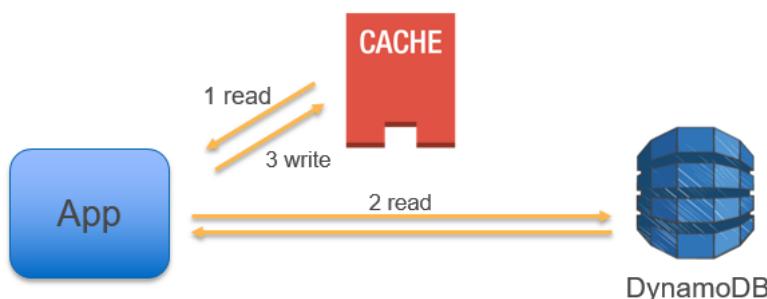
- Bir item 4kb'dan fazla ise, size'ina gore RCU olacaktir. 17kb sahip item icin Strong consistency durumunda 5 RCU gerekecektir.
- Write Capacity
 - Bir write capacity, 1 write/sec karsilik gelir.
 - 4kb yazim islemi icin 4 WriteCapacityUnit gerekecektir.
- **Her tablo read/write kapasitesine sahiptir. Bu kapasiteye ulasilrsa, HTTP 400 hatasi olusacaktir.**
- Bu rakamlardan dusuk yazma ve cok okuma DynamoDB doğru bir secim olacaktir.
- Yuksek yazma ve dusuk okuma DynamoDB icin maaliyetli olacaktir.
- Throughput, S3 kullanimi, Region disi data transferi icin ucretlendirilir.
- Ayni region'da I/O kullanimi ucretlendirilmez.
 - Her ay ilk 25 Read Capacity ucretsizdir.
- Scalability
 - Bir tus ile read/write kapasitesi arttirilabilir.
 - Bir gunde en fazla 4 kez scale down yapilabilir.
 - **Tablo basina 10.000 units/sec read ve write yapabilir. Data fazlasina ihtiyac var ise, AWS ile iletisime gecilmelidir**
 - API ve AWS console ile DynamoDB I/O throughput arttirilabilir.
 - Tablo icin tanimlanan read/write kapasitesi asilirsa, DynamoDB bu request'i sinirli tutacaktir.
- Limits
 - Bir account, bir region'da 256 tablo olusturabilir.
 - Tablo size icin limit yoktur.
 - Tablo ve account basina throughput default limit region'a gore degisebilir
 - US East;
 - 40.000 read/write capacity per table
 - 80.000 read/write capacity per account
 - Diger regionlar
 - 10.000 read/write capacity per table
 - 20.000 read/write capacity per account
 - Degisitirilmek istenirse, AWS ile iletisime gecilmelidir.
- DynamoDB'den verileri RedShift'e aktarimi yapilabilir. Bu aktarim sirasinda DynamoDB'den yapılan okumalar ucretlendirilir.
- Hive ile DynamoDB de beraber kullanilanlar ve SQL tipinde sorgular calistirilabilir, join islemleri yapilabilir.

DynamoDB Time-to-Live (TTL), uygulamamizin web session'larini kolayca yönetmemizi sağlar.

Expired item'ları tablolardan silmek için belirli bir zaman (TTL) ayarlamamıza olanak sağlar.

Küçük data elementlerini veya Amazon S3 objeleri gibi file pointer'ları kaydetmek için en uygun secedmdir.

Amazon DynamoDB Accelerator (DAX)



DAX, saniyede milyonlarca istek islense bile DynamoDB'nin performansini 10 kat daha geliştirek, milisaniye düzeyindeki yanıt süresini mikrosaniye düzeyine indiriren, fully managed ve yüksek oranda erişilebilir, in-memory cache yapısıdır.

DynamoDB Best Practices:

- Item size'in küçük tutulması (400kb max)
- Serial data tutuluyorsa; günler, haftalar ve aylar için ayrı tablo kullanılabilir.
- Çok sık erişilendataları ayrı bir tablo olarak tutulabilir.
- 400kb üstü tabloları S3'de tutulabilir.

DynamoDB Streams

DynamoDB stream ile tabloda oluşan aktiviteler yakalanabilir.

DynamoDB ve ASW Lambda entegre çalışabilirler. Bu sayede Lambda ile trigger oluşturup, otomatik olarak DynamoDB stream'de yer alan event'lere karşılık verilebilir.

Örnek verecek olursak, DynamoDB tablosuna yeni bir giriş yapıldığında, Lambda fonksiyonu tetiklenebilir ve process olmuş veriyi doğrulamak için bazı testler yapılabilir.

AWS EMR (Elastic MapReduce):

Amazon S3 DataLake storage ile, Apache Spark ve Apache Hadoop feature'larının kullanılmasını olanak sağlar.

Amazon EC2 instance'larda bulunan büyük miktarda verinin işlenmesini kolay, hızlı ve hesaplı hale getiren bir Hadoop altyapısı sağlar. Bu EC2'lerin işletim sistemlerine erişilebilir.

EMR'de Apache Spark, HBase, Presto ve Flink gibi open source tool'ları kullanarak, Amazon S3 ve Amazon DynamoDB gibi diğer AWS veri kaynaklarındaki verilerle beraber kullanılabilir.

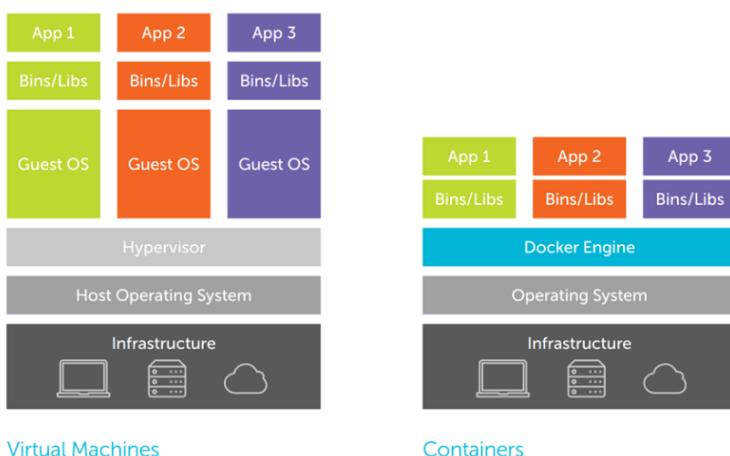
AWS Services EC2 (Elastic Container Service) (ECS)

Docker(Container):

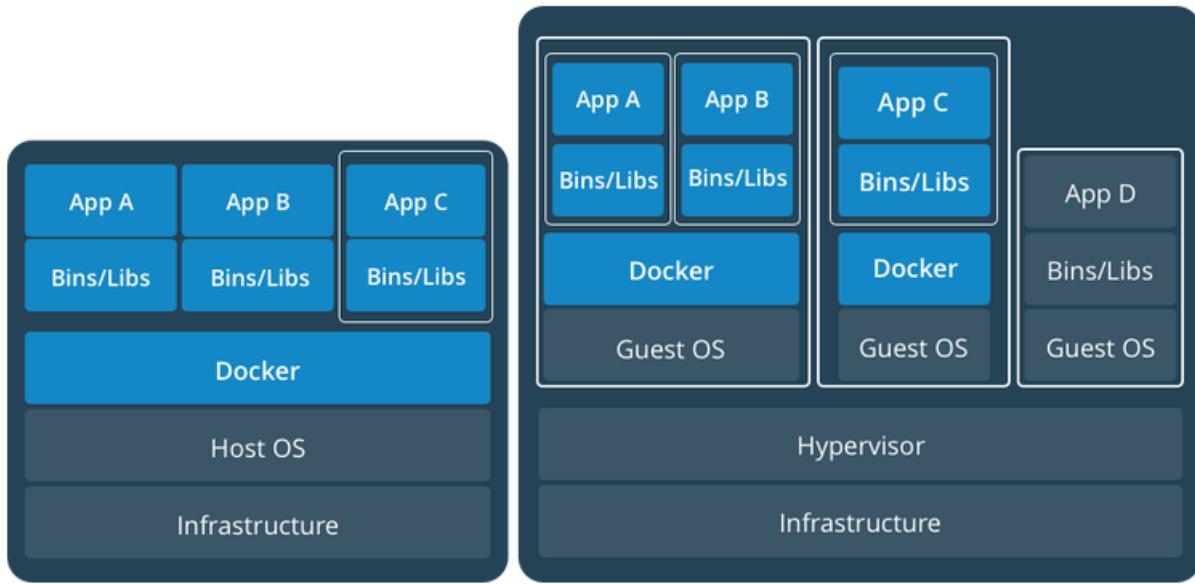
İşletim sistemi seviyesinde sanallaştırma sağlayan bir programdır. İlk sürümü 2013'te yayınlanmıştır. Konteynerler birbirinden izole edilmiş ve bağımsız halde çalışabilmektedir.

Tüm konteynerler tek bir işletim sistemi çekirdeği üzerinde çalışır ve bu sayede sanal makinelerden daha az yüksek sahip olur.

Virtual Machine ve Container arasındaki fark;



- Her ikisi de kaynak izolasyonu sağlar ancak container'lar hardware'den ziyade işletim sistemini sanallaştırır. Bu nedenle container'lar daha taşıınabilir ve verimlidir ancak container'lar ile de farklı işletim sistemleri kullanılamaz. VM'de ise farklı işletim sistemleri çalıştırılabilir.
- Container size MB mertebesinde ve GB mertebesinde olduğundan, containers çok hızlı başlar.
- AWS'de docker çalıştırıldığı zaman, developer admin sağlanmaktadır. Herhangi bir scale'de, distributed application rahatça çalıştırılabilir.
 - Amazon ECS, EC2 instance'larda bulunan container'lara bağlanmak için task tanımı olan docker imaj kullanır.
- AWS hem open source olan Docker Community Edition (CE) hem de Docker Enterprise Edition(EE) desteği sunar.
- Docker ve VM aynı yapıda da kullanılabilir ve AWS, Azure, IBM şu anda bunu sağlıyor.



- Docker uygulamalar birbirinden izole ederek, security katmanını da ayırmış olur.
 - Uygulama katmanında bir bug olursa, sadece o container etkilenir.

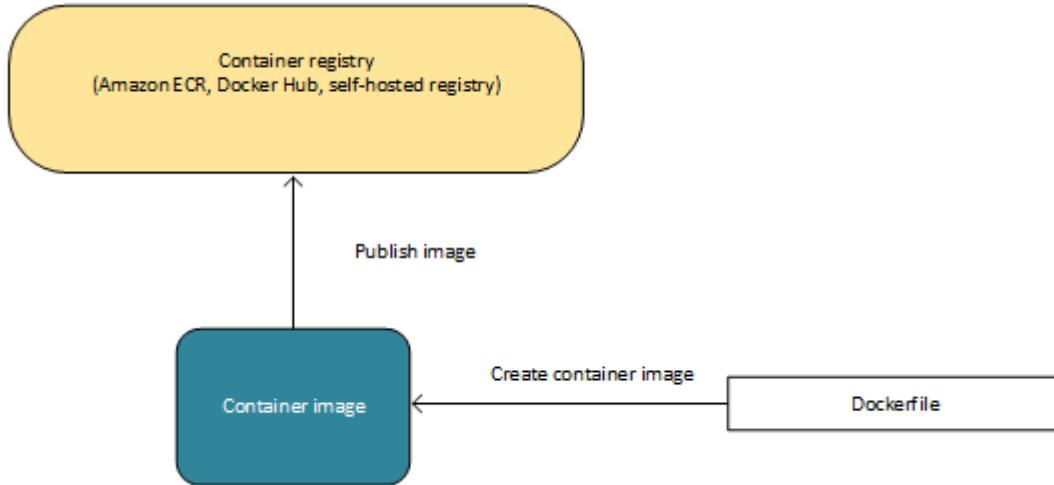
Kubernetes: Google tarafından GO dilinde geliştirilmiş Cloud Native Computing Foundation tarafından desteklenen mevcut container haline getirilmiş uygulamalarınızı otomatik deploy etmek, sayılarını artırmak azaltmak gibi işlemler ile birlikte yönetmenizi sağlayan bir container cluster aracıdır.

Docker Enterprise Edition: Muhtemelen ticari amaçla kullanılan container management çözümüdür. Uygulamaların Enterprise Linux, Windows ve AWS gibi Cloud ortamları üzerinde çalışması için, test ve entegre edilmiştir.

Amazon Elastic Container Service (ECS);

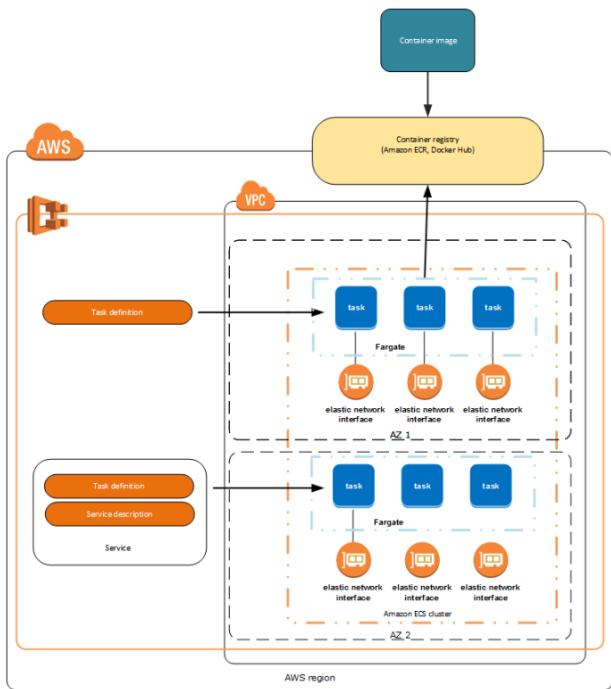
- Scalable, hızlı bir container management servistir.
- Rahatlıklar çalıştırılabılır, durdurulabilir ve Docker cluster'daki container'lar yönetilebilir.
- Cluster, server olmadan kullanılabilir ve bu durumda Amazon ECS tarafından yönetilir. Buna Fargate launch tipi denir.
- Daha fazla kontrol isteniyorsa, Amazon EC2 kullanılabilir. Bu durumda da EC2 launch tipi kullanılmış olur.
- Amazon ECS;
 - Container-base uygulamaları basit API call ile başlatılabilir ve durdurulabilir.
 - Merkezi bir servis ile, cluster durumu alınabilir.
 - Bir çok EC2 feature'ı erişim sağlar.

- Cluser'da kaynak ihtiyacına, izolasyon policy'e, availability talebine göre, placement group schedule edilebilir.
- Infrasacture yönetimindeki scaling yani ölçeklendirmeyi ECS yapmaktadır.
- Batch ve ETL workload'lar ölçeklendirilebilir, yönetilebilir ve microservice modeli ile karmaşık mimariler oluşturmaya olanak sağlar.
- ECS regional bir servistir ve aynı region'da bulunan birden fazla AZ'da çalışabilir.
- Mevcut VPC'de veya yeni bir VPC'de oluşturulabilir.
- Cluster up and running olduktan sonra, task definition ve servis tanımı yapılabılır ve bununla hangi Docker container imajının cluster'da çalışacağı belirlenebilir.
- Container imajları, AWS infrastructure içinde veya dışında bulunabilecek container registried'de tutulur.

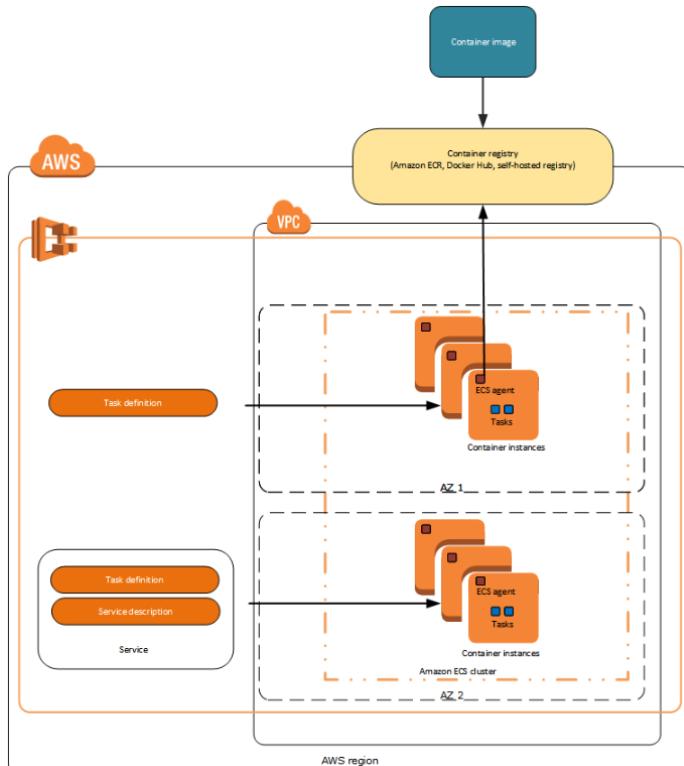


- İmajlar, Dockerfile'lar ile oluşur ve bu container içinde bulunan component'leri tanımlar. Bir çeşit script'tir.
- Dockerfile bir text dökümanıdır ve kullanıcının command line'dan çalıştırabileceği bütün komutları barındırır.
- Docker, docker file'i okur ve imajı oluşturur.
- Docker Image:
 - AWS AMI'a benzer ve container'a ait snapshot'dır.
 - Docker build komutu ile oluşturulur.

ECS Launch Tipleri:



- **Fargate Launch:** Arka planda çalışan bir altyapıya tanımlamaya gerek olmadan, container uygulamasının çalışmasını sağlar. Sadece task tanımı yapılması yeterlidir.



- **EC2 Launch:** Amazon EC2 cluster'ında, container uygulamalarının çalıştırılmasını sağlar.
 - Daha fazla kontrol sağlar ancak EC2 cluster'ı kullanıcı tarafından yönetilmelidir.
 - Her ikisi de public ve private repository destekler.

ECS Task Tanımı

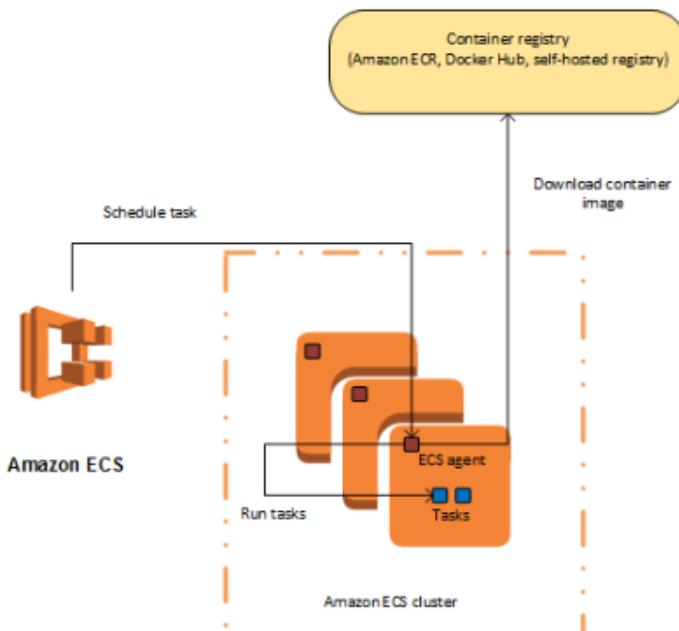
- Blueprint tanımı gibi veya template gibi container'ların nasıl olacağına dair tanımlardır.
- Task tanımı JSON formatından text dosyasıdır. En fazla 10 olacak şekilde container tanımı vardır.
- Task tanımı, aynı container'in 10 defa değil, farklı containerların tanımına sahip olabilir.
 - Bir tanesi Web/Apache ile ilgili olabilirken, diğer uygulama olabilir.

Task tanımı;

- Hangi container kullanılıyor ve repository nerede bulunuyor.
 - Container instance için hangi portlar açık olmalıdır.
 - Kullanılacak data volume.
 - Hangi launch tipini kullanılacağına dair, spesifik parametreler.
-
- Amazon ECR(Elastic Container Registry), AWS Docker registry servisi yönetir. Müşteri Docker CLI kullanarak imajları yönetebilir.
 - ECR imajları S3'de tutulur ve encryption için S3 SSE kullanılır.
 - ECR container imajlarının transferi için HTTPS kullanır.
 - ECS'de bulunan uygulama için task definition oluşturulduktan sonra, cluster için task sayısı belirtilebilir.



- Container agent'lar bütün altyapı kaynaklarında çalışarak, haberleşmeyi ve yapının ilerleyişini sağlarlar.



ECS ve IAM role;

- Default olarak IAM role ECS resource oluşturma/modifiye etme veya ECS API için task oluşturmak için yetkili değildir.
- IAM role ile, container instance seviyesindeki yetkilendirme için kullanılabilir.
- Container instance başlatmadan ve cluster'a register etmeden önce, o instance'ların launch olunca kullanılabilmeleri için IAM role oluşturulmalıdır.
- Bu role sadece EC2 launch tipi kullanılıyorsa apply edilebilir.
- Container instance'ında çalışan container'ların, container instance profilinde yer alan credential'a erişmesi engellenmelidir.
 - AWS container instance role'de yetkilerin sınırlanırmasını önerir.
 - Task içinde bulunan container, listelenenden daha fazla yetkiye ihtiyaç duyarsa, bu taskların onların IAM role'unda tanımlanması tavsiye edilir.
 - AWS ECS'de, IAM task role kullanılarak, task level'da erişim kontrolü için kullanılabilir.
 - IAM console'da, Amazon EC2 Container Service Task Role kullanması için role oluşturulabilir.
- Task için IAM role oluşturulduktan sonra, bunu task definition'da tanımlayabiliriz. Task'da bulunan container, bu yetkileri kullanabilir.

AWS Active Directory Services

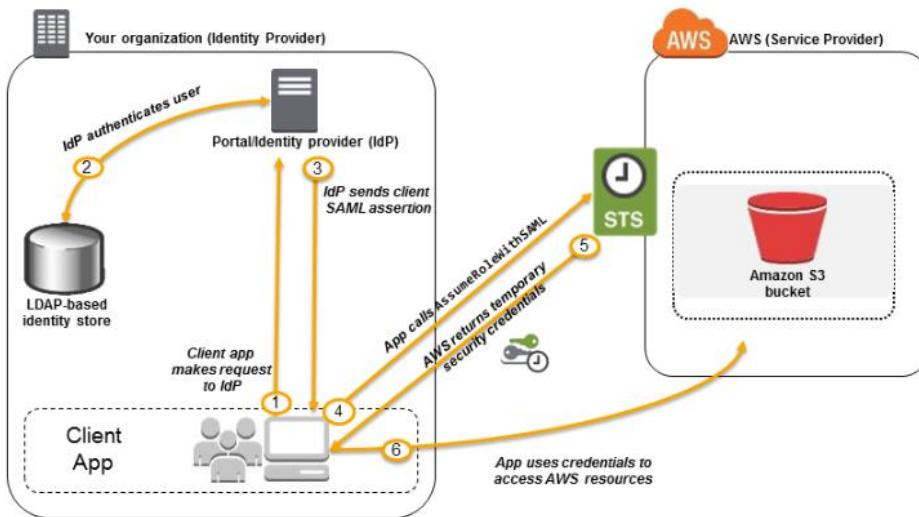
- Kullanıcılar, gruplar, cihazların bilgilerinin bulunduğu directory'dir.
- AWS Directory Service birden fazla directory kullanımını destekler.
 - Mevcut Microsoft AD
 - Lightweight Directory Access Protocol (LDAP)
- Directory Servislerini bütçeye göre feauture belirlenerek seçim yapılabılır.
- AWS Directory Service aşağıdaki servisleri barındırır.
 - Microsoft Active Directory Service
 - Simple AD
 - AD Connectory
 - Amazon Cloud Directory
 - Cognito
- 5000'den fazla fazla kullanıcı var ise ve/veya AWS hosted directory ile On-Premise directory ile trust bir ilişki içinde olsun isteniyorsa en doğru seçimidir.
- AD Connectory, mevcut on-premises AD ile kolayca bağlanır. Eğer mevcut on-premises directory ile AWS servisleri birlikte kullanılacak ise, en iyi seçenek.
 - Ucuzdur ve diğer directory genel feature'ları ile uyumludur.
- Simple AD ve AWS Directory Service for Microsoft Active Directory için snapshot teknolojisi de vardır.
 - AD connector için yoktur çünkü AD directory proxy'den daha fazlası değildir. Sadece on-premise için authentication sağlar. Authentication için Gateway gibidir.

AWS Microsoft Active Directory

- AWS tarafından fully manage'dır.
- Microsoft Active Directory'nin cloud'da çalışan halidir.
- Microsoft SharePoint, SQL Server ve bir çok .NET uygulaması ile birlikte çalışabilir.
- AWS Microsoft AD ile, Cloud'da bulunan mevcut AD ile on-premise arasında trust ilişki yapılabilir.
- AWS Microsoft AD standalone olarak da kullanılabilir.
- SAML authentication olmadan, AWS Management Console'a erişmek için AD credential'ları kullanılabilir.
- AWS Microsoft AD, Amazon WorkSpaces, Amazon WorkDocs, Amazon QuickSight, Amazon Chime, Amazon Connect ve RDS for SQL Server desteği vardır.
- Security feature'larını da barındırır.
 - Fine-grained password policy management,
 - LDAP encryption,
 - Mevcut Radius-based MFA infrastructure için Multi-factor authentication

- Loglama için CloudTrail, SNS notification, günlük otomatik snapshot ve recovery kısmını da destekler.
- Domain control ekleyerek, performans ve redundancy artırılabilir.
- Aynı region'da iki AZ deploy edilerek, HA olması sağlanır.
- Otomatik ve manual snapshot desteği vardır.
- İki farklı edition olarak gelir,
 - Standart Edition, 30.000 directory objesine kadar kullanılabilir ve en fazla 5000 çalışan için uygundur.
 - Enterprise Edition, 5000.000 directory objesine kadar kullanılabilir.
- İstenirse kullanıcı EC2 instance kullanarak AWS Cloud'da, kendi Microsoft Service AD yapabilir.
 - Bu AD, on premise AD eklenebilir ve authentication replikasyonu yapılabilir.
 - Primary domain controller olarak promote edilebilir.
 - Bu replikasyon modeli aynı zamanda AWS environment ile on-premise arasında VPN bağlantısına ihtiyaç duyar.
 - Daha az güvenlidir.
 - AWS Microsoft AD, on-premise AD, sadece trust relationship mode varken desteklenir.

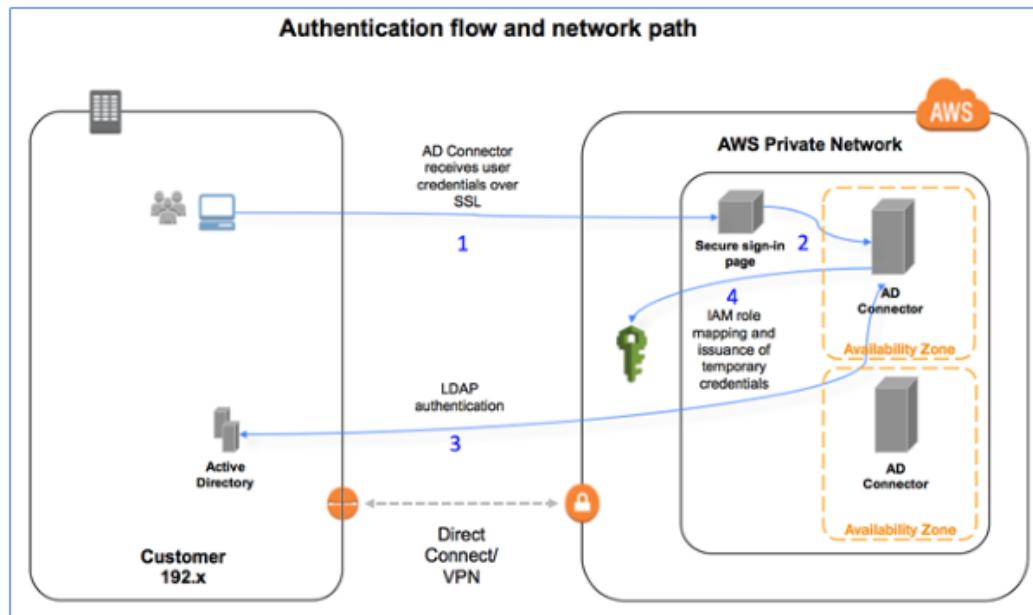
On-premise'de, Microsoft AD kullanılıyorsa, Security Assertion Markup Language (SAML) desteklendiği anlamına gelir. Bu durumda API'lar için SAML-Based Federation ayarlanabilir ve bu şekilde kolayca on-premise'de bulunan credential'lar ile AWS kaynaklarına bağlanılabilir.



AWS Simple AD;

- Standalone, full managed directory'dir.
- AWS uygulamaları için, kullanıcı oluşturma ve erişim kontrolü sağlar.
- Group policy'ler oluşturulabilir ve uygulanabilir ve EC2 instance'a güvenli erişim sağlanabilir ve Kerberos-based single sign-on (SSO) desteği vardır.
- Linux domain veya Windows based EC2 instance'a katılabilir.
- Monitoring, otomatik ve manual snapshot desteği vardır.
- AWS Simple AD, Amazon WorkSpaces, Amazon WorkDocs, Amazon WorkMail ve Amazon QuickSight desteği vardır.
- AWS Console'a ve resource'lara bağlanmasına olanak sağlar.
- İki farklı size sunar;
 - Small, 500 user ve 2000 objeye kadar.
 - Large, 5000 user ve 20000 objeye kadar.
- Düşük ölçek, düşük maaliyet, Samba-4 uyumlu uygulamalar, LDAP uyumluluğu olan uygulamar ile kullanılabilir.
- Sınırlı özellik vardır.
 - Replication, Multi-factor authentication, DNS dynamic update, schema extension, DNS dynamic update desteklenmez.
- RDS SQL server ile uyumlu değildir.
- Diğer domain'ler ile trust relationship desteklemez.

AD Connector



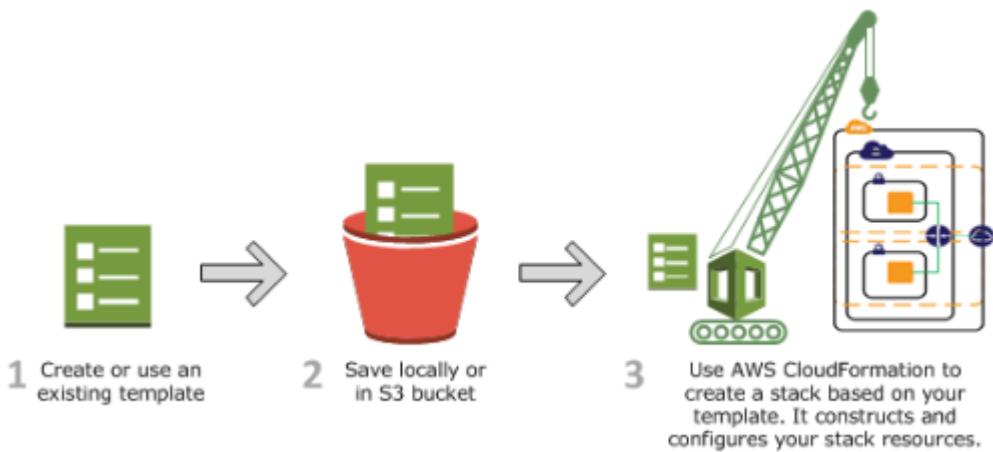
- Mevcut on-premise Microsoft AD'den, Amazon WorkSpaces, Amazon WorkDocs, Amazon Chime, Amazon Connect Amazon QuickSight, EC2 gibi AWS uygulamalarına kolayca bağlanmayı sağlar.
- İki farklı size sunar;
 - Small, 500 user'a kadar.
 - Large, 5000 user'a kadar.
- VPC'nin, VPN ile veya AWS direct connect ile on-premise network'e bağlanması gerekmektedir.
- Kullanıcı AWS uygulamalarına login olduğu zaman, AD Connector sign-in request'i on-premise Active Directory domain'e iletir.
- AWS application'a kullanıcı oluşturulursa, AD Connector mevcut AD liste ve grup oluşturmak için okur.
- RDS SQL server ile uyumlu değildir.
- AWS uygulamaları için multi-factor authentication enable edilebilir.
- Veriler on-premise'den replike edilmez, Connector sadece kontrol amaçlı ihtiyaç halinde select işlemi yapar.
- Snapshot desteği yoktur.
- Mevcut on-premise AD ile AWS servisleri kullanılmak isteniyorsa, en iyi çözümüdür.
- AWS AD Connector, On-Premise MS Active Directory kullanıcılarına STS credentials sağlamak için IAM Rollerini kullanır.
 - Konfigürasyon IAM role ile yapılır.

AD connector aracılığı ile on-premise ve VPC entegre olduktan sonra, Active Directory'de yer alan kullanıcıılara veya gruplara IAM rolü atanabilir.

Sınav Sorularından Notlar:

- LDAP replica authentication talepleri için kullanılabilir. Yani bir nevi read_only olarak çalışabilir.

AWS CloudFormation



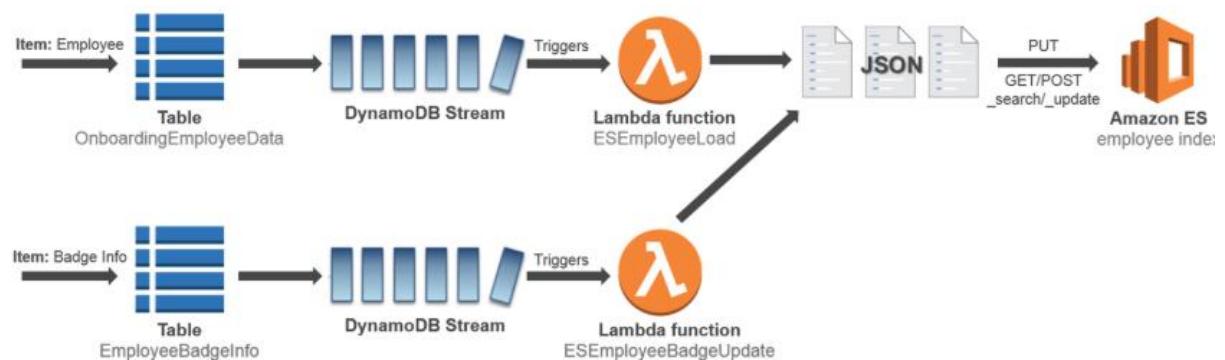
JSON formatında oluşturulan blueprint dosyası ile oluşturulam template veya sık kullanılan template'lerden kullanarak, yeni bir mimari tasarılanabilir. Mimari değişim söz konusu olursa, version kontrol ile eski versiyonlara da ulaşılabilir. AWS CloudFormation için ek bir ödeme yoktur. Sadece oluşturulan AWS resource'ları için ücret ödenir.

CloudFormation template'de birden fazla section bulunmaktadır.

```
{  
  "AWSTemplateFormatVersion" : "version date",  
  "Description" : "JSON string",  
  "Metadata" : {  
    template metadata  
  },  
  "Parameters" : {  
    set of parameters  
  },  
  "Mappings" : {  
    set of mappings  
  },  
  "Conditions" : {  
    set of conditions  
  },  
  "Transform" : {  
    set of transforms  
  },  
  "Resources" : {  
    set of resources  
  },  
  "Outputs" : {  
    set of outputs  
  }  
}
```

Outputs bölümü, kümenin özelliklerini her görüntülediğinizde döndürülen değerleri tanımlar. Örnek olarak, S3 bucket adı için output declare edilebilir ve "**aws cloudformation describe-stacks**" çağrılırsa, AWS CLI komutu adı gösterecektir.

Amazon Elasticsearch



Elasticsearch'u uygun ölçekte ve sıfır kesinti ile dağıtılmasını sağlar. Fully managed'dır, petabyte mertebesinde veri scale edebilen, HA ve scalable bir servistir.

Snapshot, data encryption desteği vardır ve kullanıldığı kadar ödenecek bir servistir. Reserved instance seçeneği de sunar.

Log analytics, Clickstream analytics, App monitoring, Full-text Search, Root-Cause analizi, security information event management gibi Real-Time analytic işlemleri yapılabilir.

Application Load Balancer (ALB)

AWS ELB servisi altında üç tip load balancer barındırır.

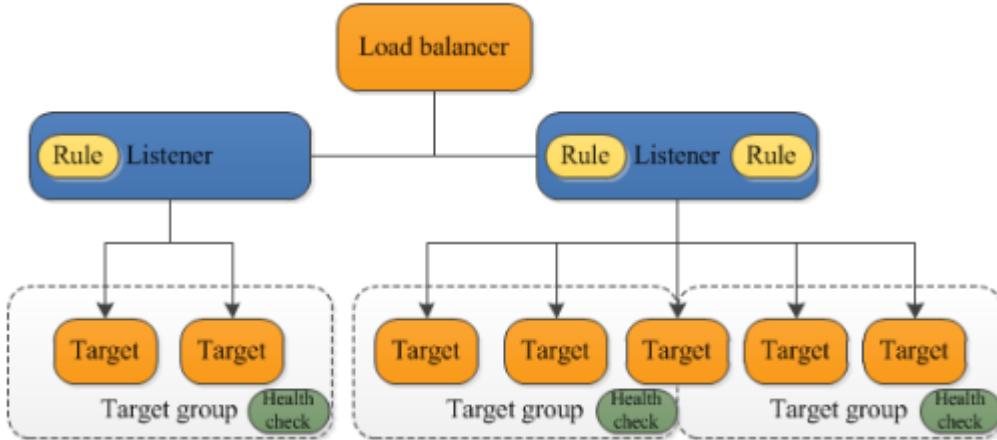
- Classic Load Balancer (CLB),
- ELB olarak bildigimiz Load Balancer'dır. TCP(Layer 4) ve HTTP/HTTPS(Layer 7) desteği vardır. Artık gelistirme yapılmıyor.
- Application Load Balancer (ALB), Q3 2016. Sadece HTTP/HTTPS(Layer 7) destekler
- Network Load Balancer (NLB), Q3 2017. Sadece TCP(Layer 4) desteği vardır.

AWS CLB Listeners

- TCP/SSL (Layer 4) ve HTTP/HTTPS (Layer 7) desteği vardır.
- Bir CLB için en fazla 100 tane listener tanımlanabilir.
- Front-End ve Back-End ile 1:1 static mapping yapılabilir.
- CLB'de min 2 ayrı LB olmalıdır. Bu demek oluyor ki, efor ve maaliyet iki katı olacaktır.

ECS Service

- ECS, task tanımı ile, ECS cluster'in da belirlenen sayıda instance çalışmasına olanak sağlar.
- Herhangi bir task fail olur ve durursa, ECS service scheduler başka bir instance'a ayağa kaldıracaktır ve sayıya bağlı kalacaktır.
- Opsiyonel olarak load balancer arkasında da çalışabilir.
 - Load balancer yükü task'lar arasında dağıtır.
- Single task definition'ı ile başlatılan container'lar, aynı container instance'a yani EC2'ye yerleşir.
- CLB arkasına birden fazla container istenirse,
 - Service definition'da, birden fazla host port tanımı yapılabilir.
 - Aynı database'e birden fazla listener oluşturmak ile aynı şeydir.
- ECS şu anlık bir tane LB desteği sunuyor.



- ALB, CLB aksine, fiziksel EC2 instance'ından ziyade, arkasında hostname ve port ikilisi ile register olabilir.
- hostname/port ikilisi ile logical group oluşturulabilir.
- ALB'de 100'e kadar rule oluşturulabilir.
- CLB aksine, front end listener'lar ile target group'lar ile one to one eşleşme yoktur.
- ALB component'leri;
 - ALB
 - Listeners
 - Gelen talepleri dinler ve rule'lara göre, target group'lara yönlendirir.
 - 50 tane listener tanımlanabilir.
 - HTTP/HTTPS desteklenir.
 - Target Groups
 - Her target group sadece 1 tane LB ile eşleşebilir.
 - Regional'dır.
 - Auto Scaling group bütün target group'lar için ayrı ayrı scale olabilir.
 - Bir target group'da 1000 tane target olabilir.
 - Bir target group'da 1 tane Lambda function olabilir.
 - Target
 - Target; EC2 instance, Microservice, ECS container'da olan App, IP adresi veya Lambda function olabilir.
 - IP adresi public olamaz.
 - IP adresi aynı VPC'de başka bir ALB olamaz.
 - Bir target birden fazla target group'a dahil olabilir.
 - Rules (Condition, Action ve Priority)
 - Rule'lar, Listener ve target group arasında gerçekleşir.
 - 1 ALB'de, 100 rule tanımlanabilir.
 - Client request'i rule ile eşleştiği zaman, action kısmı yapılır.
 - Listener'larda tanımlanır.
 - Priority ve action her rule'da vardır ancak host ve path condition opsyoneldir.
 - Her listener'da, default rule olmalıdır.
 - Condition hiç bir rule ile eşleşmezse, default rule uygulanır.
 - Her rule'in priority'si vardır ve bu sıraya göre uygulanır. İlk önce en düşük öncelikli rule uygulanır.
 - Default rule en yüksek önceliğe sahip olan rule'dur.
 - Default rule condition'a sahip olamaz.
 - Listener silinirse, bütün rule'lar da silinir.

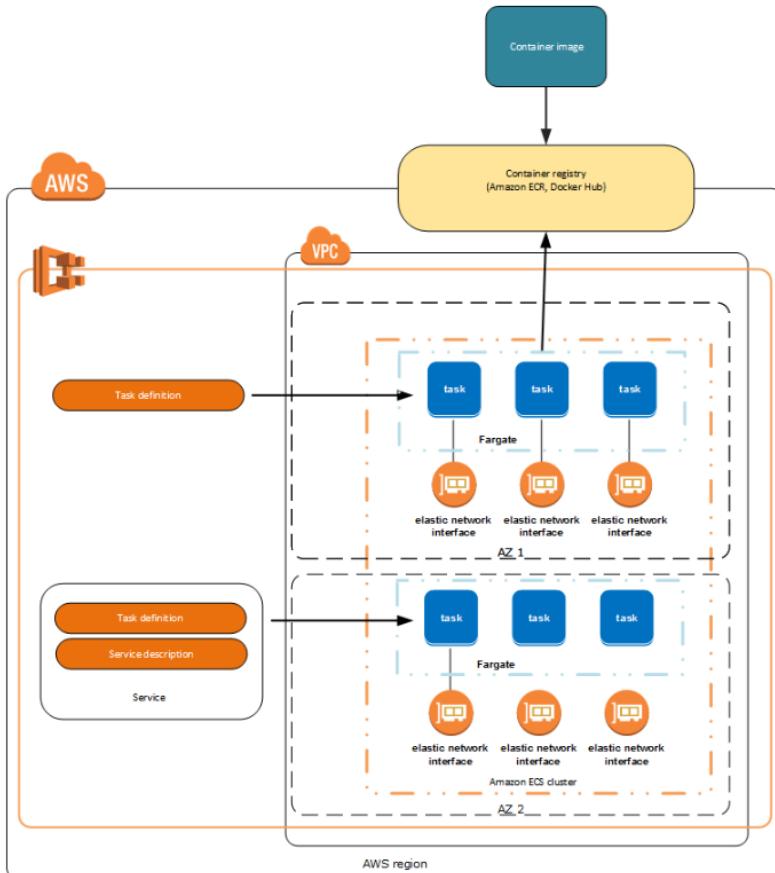
Content-Based Routing;***

- Uygulama birden fazla özel servise sahipse yani Microservice ise, ALB request'i servis özelinde yönlendirir.
- İki tip content routing vardır.
 - Host-Based
 - Domain-Name based de denedilir. www.amazon.com dediğimiz domain-based_dir.

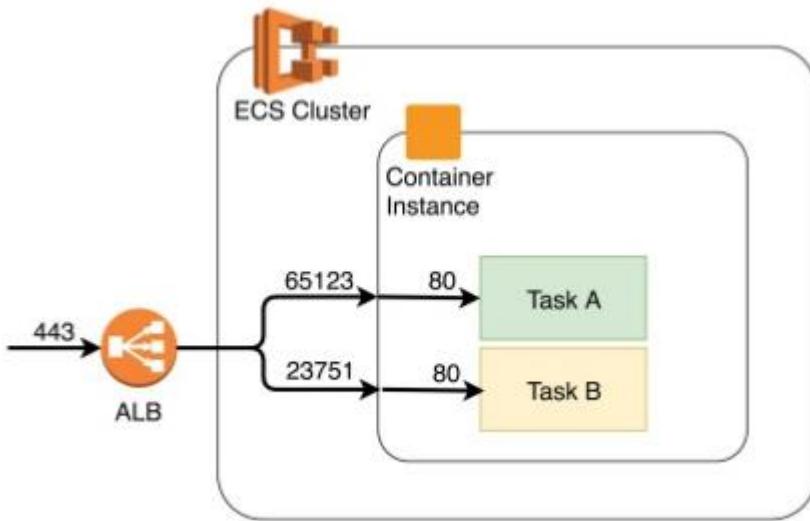
- Path-Based
 - www.amazon.com/videos gibi bir lin olursa, bu path-based olarak kabul edilir.
 - Client talebini, gelen URL path'indeki HTTP header ile yönlendirilebilir.
- Her rule'da en fazla 1 tane Host-Based ve 1 tane Path-Based routing olabilir.
- 1 tane Host-Based ve 1 tane Path-Based, combine edilebilir.

ALB - Containers and Microservices Support

- Her EC2 instance'ina bir veya daha fazla micro-service kurulabilir. Kurulan her servis farklı farklı port'lardan çalışacaktır.
- Butun servisler için bir tane ALB kullanılabilir.



- Amazon ECS, ECS cluster'ında, bir task tanımının belirlenen sayıda instance'i aynı anda çalıştırmanıza olanak sağlar.
 - ECS servisi opsiyonel olarak, ELB olarak da konfigüre edilebilir ve belirlenen instance'lar arasında trafiği dağıtır.
- ALB daha fazla özellik sundugundan, ECS'den daha yeteneklidir.
 - ALB, container'ların dynamic host port kullanmasına izin verir.
 - ALB, path-based routing ve priority rule destekler.
 - Bunun sayesinde, birden fazla servis aynı listener port'u, tek bir ALB'de kullanabilir.
 - ALB, container'lar için dynamic host port eslemesine izin verir.
 - Dynamic port mapping'i, aynı instance'da yer alan, tek bir servis için birden fazla task için kullanılabilir.
- ECS sadece bir tane LB veya target group belirtebilir.



- **Dynamic host port mapping ile farklı servisler aynı portu kullanır. Bunları ayıran özellik onlerinde dynamic olarak farklı portların bulunmasıdır.**
 - Bu sayede iki serviste 80 portunu kullanabilir.
- ALB dynamic host mapping ile, gelen talepleri, aynı container'da bulunan bir veya daha fazla port'a yönlendirebilir.

ALB vs CLB

- ALB WebSocket protocol'u destekler. CLB desteklemez.
 - WebSocket, tek bir TCP bağlantısı üzerinden tam çift yönlü iletişim kanalı sağlayan iletişim protokolüdür.

HTTP stateless request/response protokolüdür.

HTTP protokülünün bu yapısı çok fazla etkileşim içerisinde olan web uygulamaları için uygun değildir. HTTP 1.1'den önce her request server'a yeni bir connection yaratılırdı.

HTTP 1.1 ile birlikte birden çok request/response çifti için tek bir tcp connection'ı kullanılabilir hale geldi. Bu yeni yapı http-keep alive ya da http persistent connection olarak adlandırılır.

Websocket tek bir tcp connection'ı üzerinden çift yönlü ve full duplex mesajlaşmayı sağlar. Websocket ile birlikte clienttan request gelmesine gerek olmadan server'daki değişiklikler client'a iletiler hale geldi. Full duplex iki yönlü simultane iletişime izin verir.

- ALB, HTTP/2 destekler. CLB desteklemez.
 - HTTP/2 aynı anda birden fazla request yapabilir.
 - HTTP/2 ile 128 request paralel olarak gönderilebilir.
- Cross zone load balancing ALB'de by default enable'dır. CLB'de enable edilmesi gerekmektedir.
- ALB gelişmiş health check ve CloudWatch metric destekler.
 - ALB health check improvement sağlar. Bu sayede 200-399 arası hata kodları konfigür edilebilir.
- ALB access log'larında daha fazla bilgi sunar.
- **ALB WAF(Web Application Firewall) desteği sunar.**
- Internet facing ALB IPv4 ve Dualstack destekler.
 - Buna rağmen ALB, hedef ile IPv4 üzerinden haberleşir.
- Internal facing ALB sadece IPv4 destekler.
- CLB backend server authentication destekler ama ALB desteklemez.
- CLB EC2 Classic desteği sunar ama ALB sadece VPC EC2 destekler.
- ALB'de default açık değildir ama Deletion Protection açılabilir ve ALB'nin silinmesi onlenebilir.
- ALB, SNI (Server Name Indication) destekler, CLB desteklemez.
 - SNI ile her birisi kendi sertifikasına sahip TLS secured application'lar kullanılabilir.
 - Aynı secure listener'a birden fazla sertifika bağlanabilir.
 - AWS ACM ile integre olabilir.

- Connection Idle Timeout'da bazi farklılıklar vardır.
 - Load Balancer'in iki farklı connection'i vardır.
 - Client ve LB arasındaki front-end connection
 - ALB ve CLB her ikisinde de, idle timeout süresi dikkate alınır ve bu sure zarfında paket alışıverisi olmaz ise, connection timeout hatası alır.
 - Target ve LB arasındaki back-end connection
 - EC2 instance'ları için, Web Server settings'lerinde HTTP keep-alive enable edilebilir.
 - Enable edilirse, keep-alive süresi expire olana kadar ALB back-end connection'i canlı tutacaktır.
- Sticky session her ikisinde de desteklenir.
- ALB sadece load balancer-generated cookie destekler.
- Health check sonucu Availability Zone'da sağlıklı target yok ise, LB request'i yine de bütün target'lara gönderir.

ALB Monitoring

- CloudWatch,
 - ELB, request olduğu zaman, 60 saniye aralıklarla metrikleri CloudWatch'a gönderir.
 - ALB'de olup da CLB'de olmayan, HTTP request'leri takip edilebilir.
 - ALB request aldığı zaman, göndermeden önce, X-Amzn-Trace-Id header'i ekler.
 - LB ve target arasında yer alan herhangi bir servis veya uygulama da bunu yapabilir.
- AWS CloudTrail
- Access Logs

ALB Limits

- 1 Load balancer için en fazla 50 listener olabilir.
- 1 Load balancer için en fazla 1000 Target olabilir.
- 1 Load balancer için default rule haric en fazla 100 rule olabilir.
- 1 Load balancer için default certificate haric en fazla 25 certificate olabilir.
- 1 Load balancer'a en fazla 100 target register olabilir.
- 1 Load balancer için 1 tane target group olabilir.
- 1 Target Group için 1000 target olabilir.
- Bir rule için 2 condition olabilir. (1 host condition, 1 path condition)
- 1 rule için 1 tane action olabilir.
- 1 action için 1 tane target group olabilir.

CLB'den ALB'ye Migration Avantajları

- 1 task için birden fazla port gerekiyor ise,
- Path-based ve host-based desteği,
- 1 EC2'de, ayrı port'ları kullanan, birden fazla uygulama desteği,
- IP adresi ile register olabilme,
- Genişletilmiş CloudWatch seçenekleri,
- EC2 instance'ları için target group atayabilir ve Auto Scaling group oluşturabilir.
- Containerized application,
 - ECS task schedule ederken, kullanılmayan port secebilecek ve bu port ile task'a target group ile register olabilir
- Access log'lar compress'lidir ve daha fazla bilgi barındırır.
- Daha da artırmış LB performansı

Network Load Balancer (NLB)

- Client'tan gelen farklı port ve sequence numarasına sahip TCP connection'lar, farklı target'lara gönderebilir.
- NLB Layer 4 kullanır.
- TCP ve TLS listener destekler.
- Saniyede 1 milyon request ile diğer ELB'lere kıyasla daha fazla connection kaldırabilir.
- NLB, EC2 Instance ID veya IP adresi ile çalışabilir.
 - IP adresi ile tanımlanan target VPC dışarıdan olabilir.
 - Bütün target'lari IP adresi ile veya EC2 ID ile tanımlanabilir ama karışık olamaz.
- Diğer ELB'lerin aksine, Load Balancer'da static IP adresini destekler.
 - Müşteri firewall kural tanımı yaparken kolaylık sağlar.
- İki AZ olan bir LB'de, AZ'dan bir tanesinde sağlıklı target olmaz ise, o AZ ait subnet DNS kaydından silinir ve bütün yük sağlıklı çalışan AZ iletilir.
- Diğer ELB'ler gibi UDP desteklenmez.
- Birden fazla port kullanarak, belli bir instance veya IP'yi aynı target group'a tanımlanabilir.
- Deletion protection, Cross-Zone Load Balancing ve Access logs default olarak disable'dır.
- TLS listener desteklenir.
- Listener'lar da WebSocket kullanılabilir.
 - Listener oluşturulduktan sonra rule tanımlanabilir.

Supported Target Types;

- ELB'nin aksine, Lambda fonksiyonu desteklenmez.
- EC2 instance ID ve IP desteklenir.
- Target public ip'ye sahip olamaz.
- NLB source IP preservation destekler. Bu özellik diğer ELB'lerde bulunmamaktadır.
 - Target type olarak Instance ID kullanılırsa, NLB client'in source IP'sini muhafaza eder ve bunu target'a sunar.
- NLB Proxy Protocol (Gateway sistemi) V2 destekler.
 - Target group level seviyesinden konfigüre edilir ve default disable'dır.
 - Target type olarak IP adresi seçiminde kullanılmıştır ve client source IP adresi gereklidir.
 - Application package'lardan header'i parse edebilmelidir.
- NLB Active ve Passive (Network) health check'e sahiptir.
 - Active health check, LB periyodik olarak register olmuş bütün target'lara request göndererek status'lerini öğrenir.
 - Passive (Network) health check disable edilemez. Target'lardan gelen response'ların hızını gözlemler.
- CloudWatch desteği vardır.
 - CW metrikleri alarm oluşturmak için kullanılabilir.
- VPC flow log kullanılabilir.
- TLS request'lerde detaylı bilgi almak içim, Access Log kullanılabilir.
- ELB API'ya yapılan istekleri Cloud Trail ile alınabilir.

AWS IAM



- IAM role AWS kaynaklarına güvenli bir biçimde bağlanılmasını sağlar. IAM role'ü, kimlerin login olabileceğini ve kimlerin kaynaklar üzerinde hak sahibi olduğunu kontrol etmek için kullanabiliriz.
- AWS'e mail adres ve şifre ile ilk bağlanan account kullanıcısı root kullanıcısı olarak geçer.
 - Günlük işler için kullanılmaması tavsiye edilir.
- Full admin hakkı ile başka kullanıcılar da oluşturulabilir.
- Multi-factor authentication kullanılabilir.
- Identity federation destekler.
- Identity information assurance
 - AWS CloudTrail kullanılıyorsa, resource'lar için kim request gönderdiğini görebilirsiniz. Bu bilgi IAM identities merkezlidir.
- PCI (Payment Card Industry) DSS Compliance destekler.
 - Güvenli bir şekilde kredi kartı kullanımını ve bilgilerin tutulmasını sağlar. Netflix buna bir örnektir.
- Bir çok AWS servisi ile entegredir.
- Eventually consistent, yani tutarlıdır.
 - Bir çok AWS servis ile tutarlıdır.
 - IAM data'yı dünya çapında high available olarak replike edebilir.
 - Değişiklikler user, group, role veya policy oluşturma ve update etme olabilir.
 - Replikasyon biraz zaman alabilir.
- Uygulamalar için kritik olan code path'lerinin dahil edilmemesi önerilir.
 - Bunlar için ayrıca bir initialization veya rutin ve daha az freksanslarda yapılması tavsiye edilir.
- IAM ile yapılan aksiyonlar ücretsizdir.

Belli bir sayıda IAM user'ın S3'e ulaşmak istediğini düşünelim. Bu durumda tek tek yetki vermekten ziyade, IAM group oluşturup, kullanıcıları bu gruba eklemek gerekmektedir.

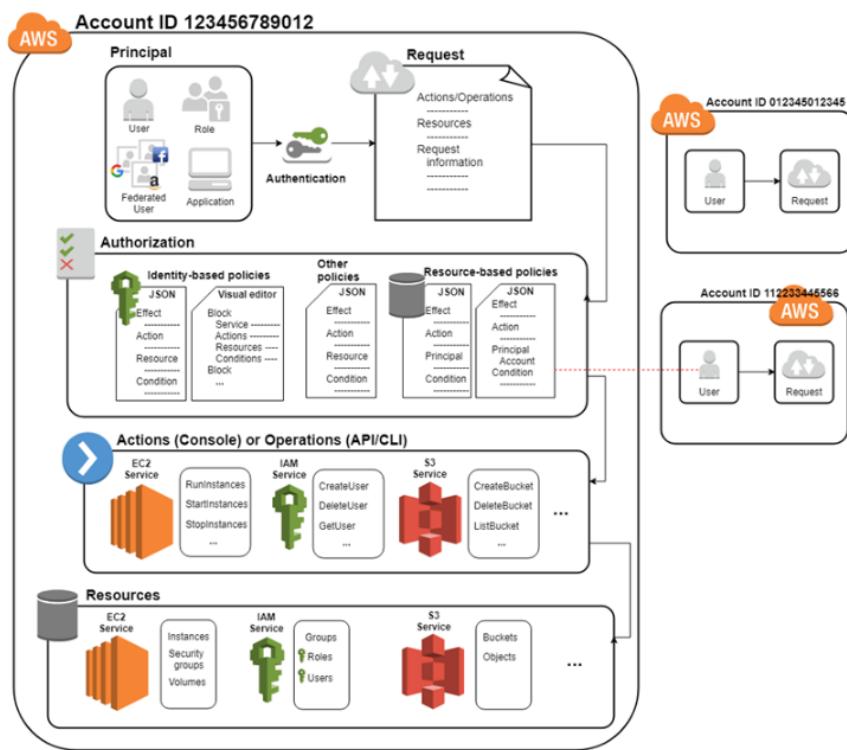
IAM user console'dan oluşturulursa, console password veya access key seçilmek zorundadır.

User AWS CLI veya AWS API ile oluşturulursa, hiç bir yetkisi olmayacağındır. IAM user oluşturulduktan sonra S3, Lambda, DynamoDB veya başka bir AWS resource'u kullanacak ise, user için **Access Key** oluşturulmalı ve gerekli yetkiler user'a verilmelidir.

IAM Policy ve Tag

UAT ve Prod ortamları arasında yetki ayrimı yapılım isteniyorsa, Tag ve IAM Policy kombinasyonu doğru seçim olacaktır. Ortama göre Tag belirleyebilir ve IAM Policy buna bağlı olarak condition eklenebilir.

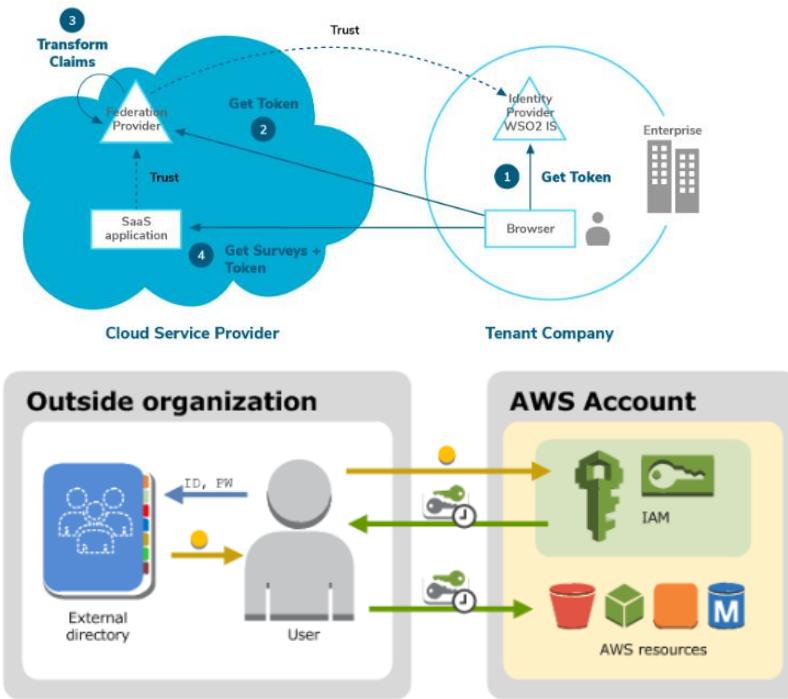
IAM Elements



IAM infrastructure aşağıdaki elementleri barındırır.

- Principal
 - Tanımlanan resource'a kim erişmeye çalışıyor. User, role, federated user, application olabilir.
 - Admin IAM user, ilk oluşan principal'dir. Root user principal değildir.
- Request
 - Request oluşturan principal ne talep etti.
 - Principal bilgilerini de barındırır. Hangi environment'dan talep geldi, vs
- Authentication
 - Kimlik doğrulama kısmıdır ve kullanıcının login yetkisi var mı diye kontrol edilir.
- Authorization
 - Login olan kullanıcın yapmaya çalıştığı şey için yetkisi var mı diye kontrol edilir.
 - IAM policy kontrol edilir ve bu policy'ler JSON formatında tutulur.
 - İki tip policy vardır.
 - User (Identity) - based policy, principal'lar için permission tanımıdır.
 - Resource-based policy, resource'lar için permission tanımıdır.
- Actions
 - Yapılmaya çalışılan eylem nedir.
- Resources
 - Hangi resource üzerinde işlem yapılmıyor.

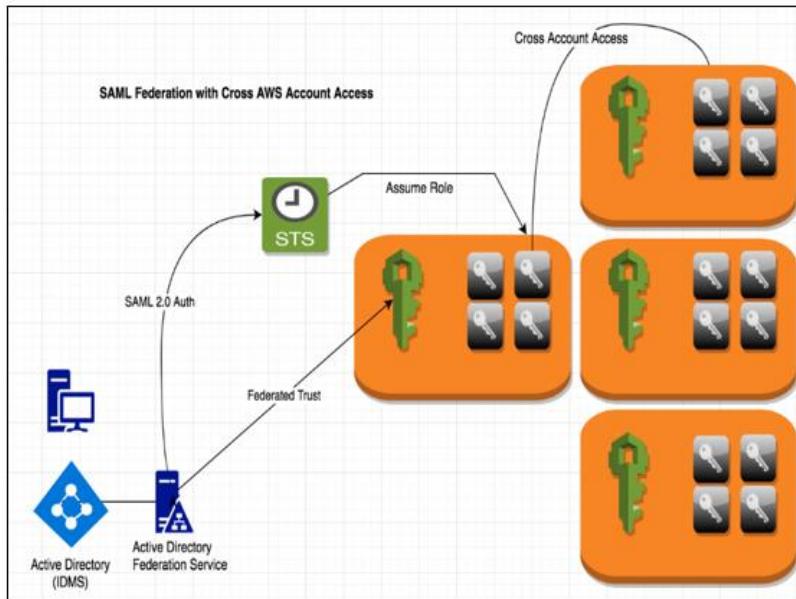
Identity Federation



Bir account başına 5000 IAM user sınırı vardır. Yani Netflix gibi AWS ortamında çalışan şirketler bütün müşterilerinin bağlanması onlara için IAM user vermezler.

Müşteriler ile Netflix arasında trust ilişki olması gerekmektedir. Bu var ise, Bu kullanıcıları AWS içerisinde federate edebiliriz.

Use cases:



- Corpotare directory'de olan kullanıcılar için kullanılabilir.
 - Corpotare directory SSO (single-sign on) ile AWS console'a erişim sağlanabilir.
 - **AD'de SAML 2.0 uyumlu olmalıdır. Değil ise, Identity broker application kurulması gerekmektedir.**
- Kullanıcılarda internet identitity var ise,
 - Eğer mobile veya web tabanlı bir uygulama var ise, kullanıcılar; Amazon, Facebook, Google gibi internet identity provider aracılığı ile erişim sağlanabilir.

- AWS identity federation için AWS Cognito kullanılmasını tavsiye eder.

Amazon Cognito

Web ve mobil uygulamalarına hızlı ve kolayca kullanıcı kaydı, oturum açma ve erişim denetimi eklemenize olanak sağlar.

Amazon Cognito, milyonlarca kullanıcıya ölçeklenir ve Facebook, Google, Amazon gibi sosyal kimlik sağlayıcıları ve SAML 2.0 aracılığıyla kurumsal kimlik sağlayıcıları ile oturum açmayı destekler.

- IAM role'leri ve policy'leri, IAM varlıklar AWS'de globaldir.
- MFA

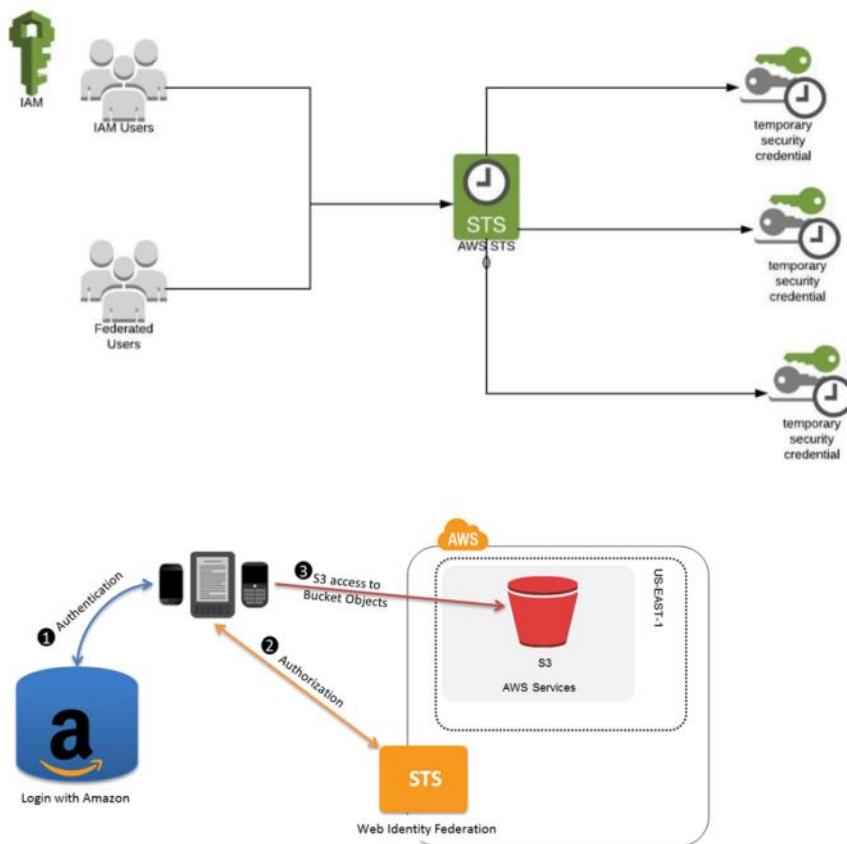


- One time Access veya kısa süreli giriş talepleri ve ihtiyaçları için MFA en iyi çözüm olacaktır.
- CLI ve API call yapabilmesi için, kullanıcıların Access Key'e sahip olması gerekmektedir.
- **Amazon Cognito ID, kullanıcıya geçici ve sınırlı yetki vermek için kullanılabilir.**

AWS IAM best practices:

- IAM kullanıcı ve gruplarına sadece ihtiyaç duydukları yetkiler verilmelidir.
- Root kullanıcısına ve yetkili IAM kullanıcılarına MFA konfigür edilmelidir.
- Şifreler belli periyotlarda değiştirilmelidir.

AWS Security Token Service (STS)



DynamoDB'ye bağlanacak bir uygulama olsun. Best practice olarak bu uygulamanın bağlantısı temporary olmalıdır.

Bir kullanıcı uygulamayı başlattığı zaman, temporary credentials kullanıcı için yetkilendirilir ve kullanmadığı zaman veya expire time'a ulaştığı zamana kadar devam eder.

Böyle bir yapı ancak **Security Token Service** ile olabilmektedir.

- End user uygulamaya bağlanır.
- Amazon tarafından ID token alır ve bunun sayesinde authenticate olur. Amazon burada IdP(Identity Provider) olarak görev yapar.
- Alınan ID token verilir ve karşılığında Cognito token alınır.
- Alınan Cognito token verilir ve karşılığında temporary AWS credential'i alınır.
- Alınan temporary AWS credential ile DynamoDB'ye bağlanılabilir.

Web Identity Federation

Custom sign-in veya kendi kullanıcı kimliğinin yönetilmesi gerekmekz.

Bunun yerine uygulamanın kullanıcıları; Amazon, Facebook, Google veya herhangi bir başka OpenID connect (OIDC) ile uyumlu IdP (Identity Provider) kullanarak oturum açabilir, authentication token alabilir ve bu token'i, istenilen AWS kaynaklarına erişim izni verecek temporary security credential ile değiştirebilir.

IdP kullanarak, uzun süreli kimlik bilgilerinin dağıtımasına gerek kalmaz.

STS: AWS identitiiy ve IAM user için temporary ve limitli yetkilere sahip credential alınmasına olanak sağlayan bir web servisidir.

- Cross region iş yapmak isteyen IAM user yetkilendirmesi bu şekilde gerçekleşir.

- Identity federation, delegation, cross-account access ve IAM role'leri için kullanılması uygundur.
- AWS STS global bir servistir.
 - Opsiyonel olarak AWS STS request'i region spesifik de gönderilebilir.
 - By default bütün region'lar enable'dir ancak istenilen region disable edilebilir.
- STS'e, STS query API ile erişilebilir.
- AWS CLI ve Microsoft PowerShell for AWS command line toolları da kullanılabilir.
- AWS Console ile generate edilemez.

AWS STS API Actions and Permissions

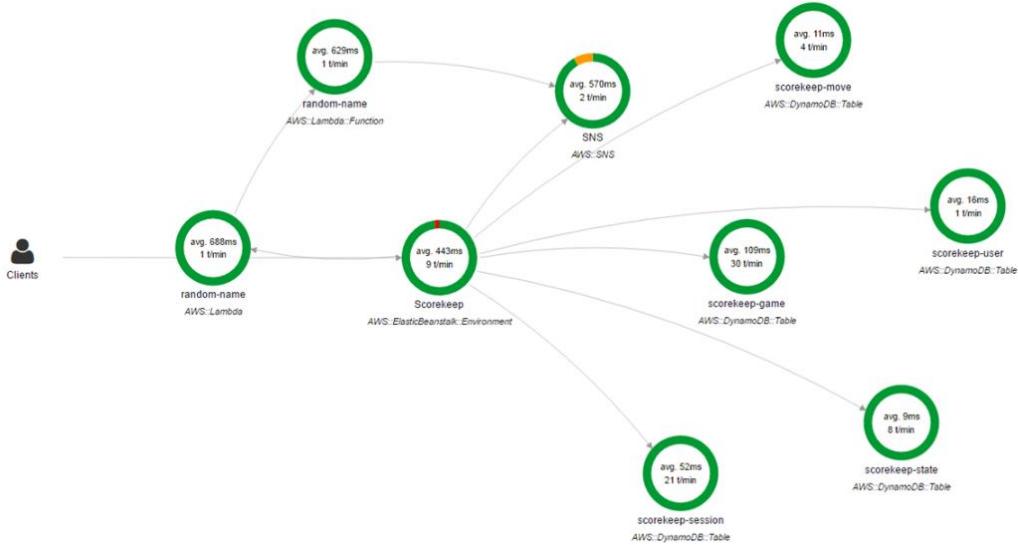
- STS session token alabilmek için birden fazla API sahiptir.
 - **AssumeRole**
 - IAM user veya mevcut temporary security credential çağrırlabilir.
 - Cross Account erişimleri için kullanılabilir.
 - Cross Account non-authenticated kullanıcılar için kullanılamaz.
 - AD connector olarak kullanılabilir.
 - **AssumeRoleWithSAML**
 - Corporate authenticated userlar içindir. SAML authentication'ı geçebilen herhangi bir user çağrırlabilir.
 - **AssumeRoleWithWebIdentity**
 - Web identity token'ı geçebilen herhangi bir user çağrırlabilir.
 - Uygulama, non-authenticated user ile S3 gibi herhangi bir AWS resource'na erişmek isteyen kullanıcılar kullanabilir.
 - **GetSessionToken**
 - IAM user veya AWS account root user çağrırlabilir.
 - **GetFederationToken**
 - IAM user veya AWS account root user çağrırlabilir.
- AssumeRole, AssumeRoleWithSAML ve AssumeRoleWithWebIdentity default 1 saatdir. GetSessionToken ve GetFederationToken default 12 saatir ve max 36 saat ve min 15 dakika olarak ayarlanabilir.
- Verilen yetkiler geri alınamaz. Expiration süresine kadar aktif olarak kalırlar.

AIM Use cases:

- EC2 SSH erişimi
- AssumeRoleWithWebIdentity ile Web ID authentication
- GetSessionToken ile IAM userlarının MFA kullanarak AWS servislerine erişmesi
- AssumeRoleWithSAML kullanarak SAML authentication
- AssumeRole ile Cross Account erişimi
- AssumeRole ile AD connector olarak kullanılabilir.
- GetFederationToken ile IAM user veya AWS root user için ID authentication.
- RDP ve SSH için kullanılmaz.

İpucu: Soruda AWS Microsoft AD veya LDAP database'den bahsetmiyorsa, varmış gibi öngörmemeliyiz.

Amazon X-Ray



Splunk gibi çalışır ve performans sorunlarını analiz etmek, saptamak ve optimize etmek için kullanılır.

AWS X-Ray, birden çok uygulamayı analiz edip bunların hatalarını ayıklamasına yardımcı olur. X-Ray ile, performans sorunları, hataların kök nedeni ve onları gidermek için kullanılabilir.

Uygulamanızdan geçen request'leri uçtan uca sunar ve uygulamaların bileşenlerinin bir haritasını gösterir. X-Ray'i kullanarak üç katmanlı basit uygulamalardan binlerce hizmet içeren karmaşık mikro hizmet uygulamalarına kadar birçok uygulamayı hem geliştirme hem de üretim aşamasında analiz edilebilir.

Kabaca bağlı çalışan bir çok uygulamayı ağaç mimarisinde göstererek, hata sinyali gönderen uygulama ve dallarının belirlenmesini sağlar. Bu sayede uygulamanın yarattığı etki kolayca analiz edilerek, olası probleme çözüm sağlanması da kolaylaşır.

AWS Config

AWS Config, AWS kaynaklarınızın yapılandırmalarını incelemenizi, denetlemenizi ve değerlendirmenizi sağlayan bir hizmettir.

Config, devamlı olarak AWS kaynak yapılandırmalarınızı izler ve kaydeder; kayıtlı yapılandırmaları istenen yapılandırmalara göre değerlendirmenizi otomatikleştirmenizi sağlar.

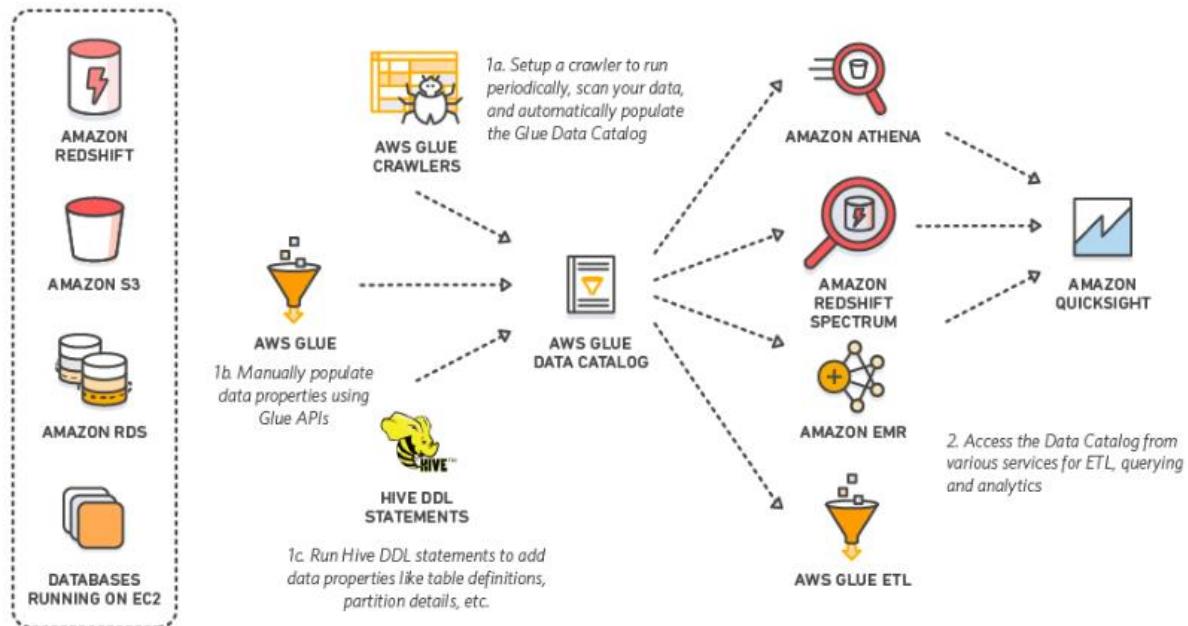
Config ile AWS kaynakları arasındaki ilişki ve yapılandırmalarındaki değişiklikleri inceleyebilir, ayrıntılı kaynak yapılandırması geçmişlerine bakabilir ve dahili yönerilerinizde belirtilen yapılandırmalara göre genel uyumluluğunu belirleyebilirsiniz.

Bu sayede mevzuat uyumluluğu denetimi, güvenlik analizi, değişiklik yönetimi ve operasyonel sorun gidermeyi daha basit hale getirebilirsiniz.

Benefits:

- İstenilen konfigurasyonun oluşması için AWS resource'larını değerlendirir.
- Desteklenen resource'lar için anlık konfigurasyon snapshot'ı alabilir.
- Account içerisinde yer alan resource'ların konfigurasyonu geri alabilir
- Bir veya daha fazla resource'un historical konfigurasyonunu geri alır.
- Resource oluşturduğumda, modify edildiği veya silindiği zaman notification oluşturur.
- Resource'lar arası ilişki görülebilir.
- User ve group'lar için IAM permission geçmişi takip edilebilir.
- Security Group konfigurasyonu incelerek, connectivity ve package drop sorunları incelenebilir.
- AWS config konfigurasyon geçişini retention period'a göre tutar ve bu süre min 30 gün max 7 yıldır.

AWS Glue ve AWS Athena



AWS Glue ETL işleri yapan, AWS tarafından yönetilen bir hizmettir. Python ve Scala desteği sunan Glue, AWS'de store edilen verileri kullanarak, analiz ve raporlama için kullanılmaktadır.

Amazon Athena, Amazon S3'te standart SQL kullanarak veri analizi yapmanızı kolaylaştıran etkileşimli bir sorgu sistemidir. Athena sunucusuz olduğundan yönetilmesi gereken bir altyapı yoktur ve yalnızca çalıştırıldığınız sorgular için ödeme yaparsınız.

Amazon S3'de bulunan verileri kullanarak, standart SQL ile sorgulama yapılmasına olanak sağlar. Karmaşık ETL mimarilerinden ziyade, daha basit SQL sorguları ile verilerden sonuç kümesi oluşturulmasını sağlar.

Athena; AWS Glue ve Amazon Quicksight ile entegre çalışabilir. AWS Glue ile farklı kaynaklardan, analiz yapılmasına da olanak sağlar ve Amazon Quicksight ile visualization katmanın oluşturulmasını sağlar.

AWS Resource Access Manager (RAM)

AWS Resource Access Manager ile güvenli ve kolayca AWS resource'larını başka AWS account'ları ile paylaşılmasına olanak sağlar.

AWS Transit Gateways, Subnets, AWS License Manager konfigurasyonu ve Amazon Route 53 Resolver rules resource'larını AWS RAM ile paylaşılabilir.

Bir çok şirket faturalama ve yönetimleri ayırmak için birden fazla account kullanmaktadır. RAM sayesinde aynı resource'u birden fazla oluşturmadan, operasyonel maaliyetin kısılmamasını sağlar.

- AWS RAM, RAM console, API query, AWS CLI ve Windows PowerShell için olan AWS tool'ları ile erişilebilir.
- AWS CloudWatch Event ile entegre çalışabilmektedir.
- AWS CloudTrail tarafından loglanabilir.

AWS Budget: Bütün AWS kaynaklarına ait budget'ların yönetilmesini sağlar.

AWS Cost Explorer: AWS maaliyetlerini ve kullanım sürelerinin görülebileceği bir interface'dir.

AWS Cost Allocation Tags: Kullanıcı veya AWS'in, bir AWS kaynağına atadığı bir etikettir. Her etiket bir anahtar ve bir değerden oluşur. Bir anahtarın birden fazla değeri olabilir. Kaynaklarınızı düzenlemek için etiketleri ve AWS maliyetlerinizi ayrıntılı bir düzeyde izlemek için kullanılabilir. DEV/PROD gibi Tag'ler eklenebilir.

Bring Your Own IP Addresses (BYOIP)

Public Ipv4 IP adreslerinin bir kısmını veya tamamını kurum içi ağdan AWS hesabına getirilmesini sağlar.

A Route Origin Authorization (ROA)

RIR(Regional Internet Registry) ile oluşturulabilecek bir belgedir. IP adres range'i barındırır.

AWS hesabına IP adres aralığı getirme yetkisi vermek için, RDAP açıklamalarında kendinden imzalı bir X509 sertifikası yayınlanması gereklidir. Sertifika, AWS'nin sağladığınız yetkilendirme içeriği imzasını doğrulamak için kullandığı ortak bir anahtar içerir.

On-Premise'de, whitelisted IP adresi var ise ve AWS'e taşınmak isteniyorsa, öncelikle ROA oluşturulmalı ve ardından taşınabilir.

VPC Flow Logs

Network'de gelen ve giden IP trafigini yakayabilir.

Belli bir IP instance'a erişemiyor ise, bunun analiz edilmesine yardımcı olur.

Amazon Aurora

- MySQL ve PostgreSQL ile uyumlu bir relationale database'dir.
- Standart MySQL veri tabanından 5 kat ve standart PostgreSQL veri tabanından 3 kat daha hızlıdır.
- Security, availability ve reliability için 10 kat daha az maaliyetlidir ve Amazon RDS tarafından full managed'dır.
- Database instance başına 64TB kadar kullanılabilir ve 15 low-latency read replika desteği sunar.
- Point-time recovery, S3'e alınabilecek continuous backup ve 3 AZ replikasyon'a kadar destek sunar.

Amazon Aurora tek bir instance'dan ziyade, cluster mantığında çalışmaktadır. Bir Aurora cluster'ına bağlandığınızda, belirtilen hostname:port ikilisi, endpoint olarak adlandırılan bir handler'a işaret eder. Aurora, bağlantıları soyutlamak için bu endpoint mekanizmasını kullanır. Bu nedenle bazı instance'ları available olmadığından, hardcoded olarak bütün hostname'leri yazmamız veya kendi logic'imizi yazmamız gerekmektedir.

Bazı Aurora instance'ı veya instance'ları farklı roller için kullanılıyor olabilir.

Örneğin, primary instance bütün DDL ve DML işlerini kabul ederken, 15'e kadar olabilecek replika'lar read-only taleplerini karşılayabilir.

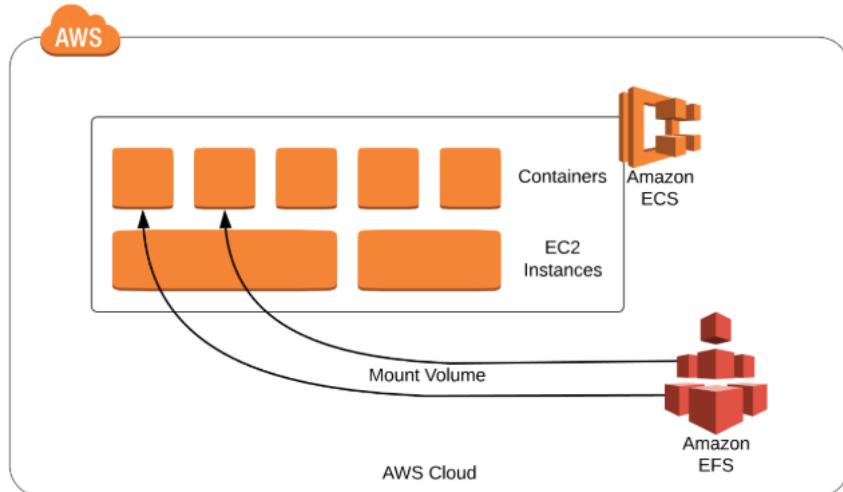
Custom endpoint, DB instance'larını read-only veya read-write dışındaki kriterleri temel alan load-balanced database taleplerini sağlar. Belirli bir AWS instance'ını veya belirli bir instance grubunu kullanan custom endpoint tanımlanabilir. Belirli bir grup kullanıcıya da bu group verilebilir.

Örneğin, internal kullanıcılarla rapor oluşturma veya bir defalik sorgulama için düşük kapasiteli instance'a yönlendirirken, production trafigini yüksek kapasiteli instance'lara gönderilebilir.

Failover: Failover durumunda, Amazon Aurora Replica var ise CNAME replikaya assign olur ve 30 saniye içerisinde yeni primary olarak çalışmaya devam eder.

Eğer replika yok ise ve tek bir Aurora ile çalışiliyorsa, Aurora ilk olarak aynı AZ'da yeni bir DB instance oluşturmaya çalışacaktır. Eğer bunu yapamaz ise, başka bir AZ'da deneyecektir ve süreç 15 dakikanın altında bir sürede tamamlanacaktır.

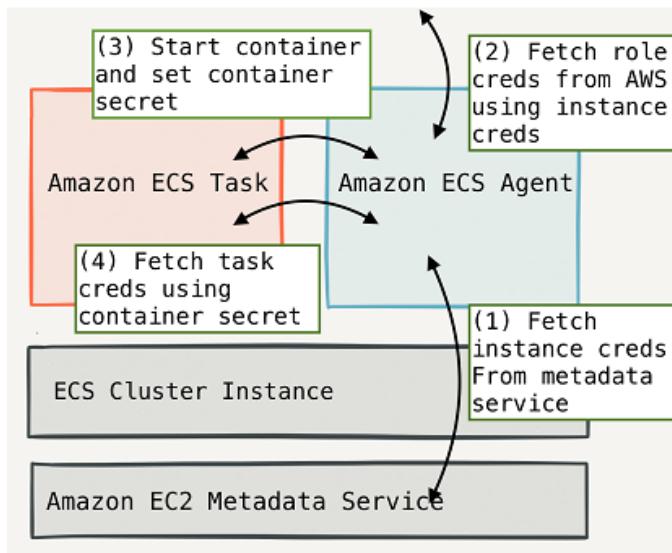
Amazon EFS



- Fully managed'dır. Simple, scalable linux shared file system'dır.
- Otomatik olarak büyüyebilir ve otomatik olarak shrink yapar.
- General purpose ve Max I/O olmak üzere iki çeşidi vardır.
- EFS file share'e aynı anda birden çok EC2 instance erişebilir. Bu durum EBS için geçerli değildir.
 - Bu nedenle aynı anda birden çok okumalar için EFS performansı daha iyi sonuç verir.
- Kullanıldığı kadar ödenen bir servistir.

AWS Secret Manager ve AWS Systems Manager Parameter

Parameter Store Kullanarak Secret Saklamak;

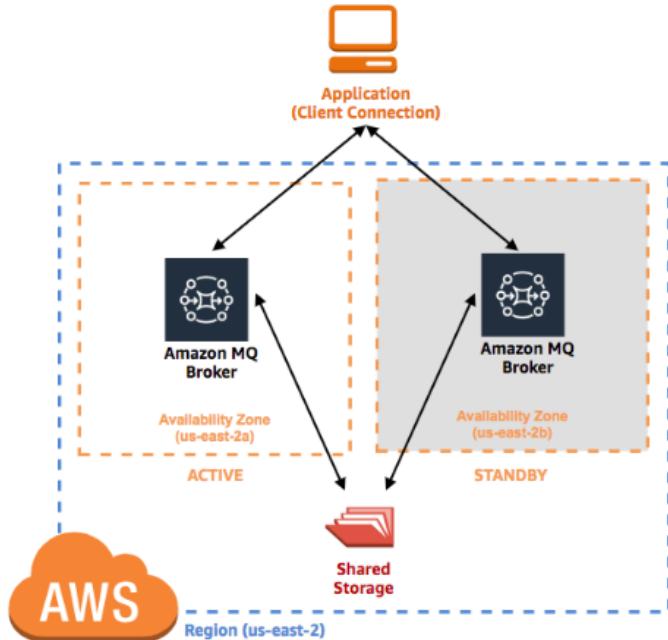


Amazon ECS ve Fargate launch tipleri için sensitive verilerini AWS Secret Manager'a veya AWS System Manager Parameter Store'a kaydederek, bunları container tanımlarına yönlendirerek, hassas verilerin container definition'a eklenmesi sağlanabilir.

- Hassas verileri container'lara entegre etmek için secrets container parametresi kullanılır.
- Bir container'ın log konfigürasyonundaki sensitive bilgilere refer etmek için, secretOptions container parametresi kullanılır.

Bu şekilde environment variable'lar secret olarak tutulabilir. Bunu resource-based policy yerine IAM role ile kullanmak daha uygun bir yoldur.

Amazon MQ (Managed message broker service for Apache ActiveMQ)



On-premise çalışan bir message broker servisi var ise ve AWS'e geçirilmek isteniyorsa, MQ seçilmelidir. Çünkü MQ industry standart API'ları ve protokollerini destekler ve bu sayede uygulamadaki mesajlaşma kodu yeniden yazılmasına gerek olmadan, Amazon MQ'ya geçirilebilir.

Yeni bir uygulama baştan tasarlanacak ise, Amazon SQS veya Amazon SNS önerilir. Mikroservice, distributed sistemleri ve sunucusuz uygulamaları ayırmak, ölçeklendirmek ve güvenilirliği artırmak için Amazon SQS ve SNS'yi kullanılabilir.

Amazon SQS industry standart API'ları ve protokollerini desteklemez. SNS bir pub/sub mesajlaşma servisi olarak daha uygundur.

AWS Shield

EC2, ELB, CloudFront ve Amazon Route-53 üzerinde yapılabilecek olası saldırılarından üst seviyede korunmak için kullanılmaktadır.

Standart ve Advanced olmak üzere iki çeşittir.

Standart ücretsizdir. Web sitelerini ve uygulamaları en yaygın, en sık karşılaşılan web ve aktarım katmanı **DDoS** (Distributed Denial-of-Service) saldırılarına karşı savunma sağlar.

DoS (Denial of Service) atakları websitesini veya uygulamaları kullanılamaz hale getirebilirler. Bunu yaparken de web ve resource'ları tüketen teknikler kullanabilirler. Bu gibi bir durumu önlemek için aşağıdaki yönetmeler uygulanabilir.

- Hem statik hem de dinamik content'i dağıtmak için Amazon CloudFront kullanmak.
- EC2 instance'lar için, ASG ve ALB ile private subnet kullanarak RDS trafiğini de sınırlamak
- CloudWatch ile Network In ve CPU kullanımını takip etmek

AWS Shield Standard hizmetini Amazon CloudFront ve Amazon Route 53 ile birlikte kullandığınızda bilinen tüm altyapı (3. ve 4. Katman) saldırılarına karşı geniş kapsamlı erişilebilirlik korumasına sahip olunur.

EC2, ELB, Amazon CloudFront, AWS Global Accelerator ve Route 53 kaynakları üzerinde çalışan uygulamaları

hedef alan saldırılara karşı daha üst düzey koruma sağlamak için **AWS Shield Advanced** çözümü kullanılmalıdır. AWS Shield Advanced ile birlikte AWS WAF'da gelmektedir.

AWS Web Application Firewall (WAF)

AWS WAF, web uygulamalarınızı uygulama erişilebilirliğini etkileyebilecek, güvenliği tehlikeye atabilecek veya aşırı kaynak kullanabilecek yaygın web açıklarına karşı korumanıza yardımcı olan bir web uygulaması güvenlik duvarıdır.

Perfect Forward Secrecy

CloudFront ve **Elastic Load Balancer** için şifreli dinlenmesine karşılık güvenlik katmanıdır. Bu veriler yakalansa dahi kodun çözülmesini önlüyor ve ek SSL/TLS şifreleme sunar.

AWS Snowball



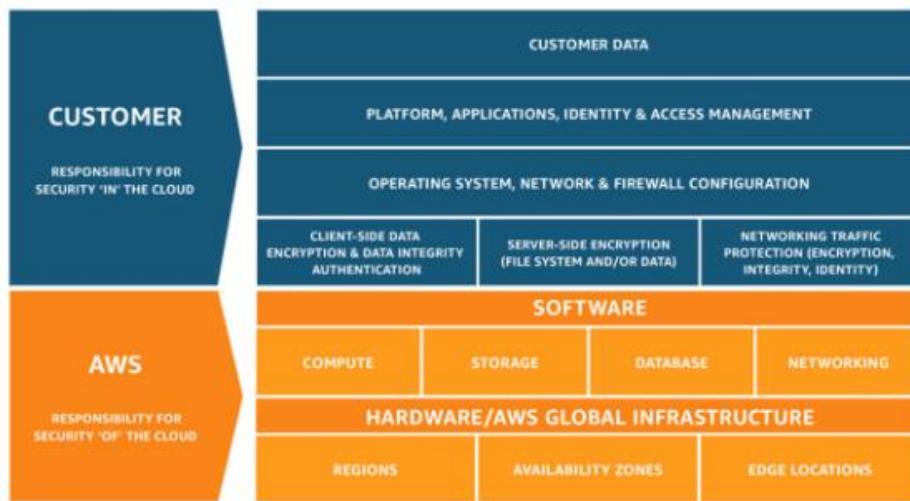
Petabyte mertebesinde veriyi AWS Cloud'a ve dışına taşımak için güvenli cihazlar kullanan, petabayt veri taşıma çözümüdür.

Large-scale veri transferinde, network maaliyeti, transfer süresi ve security endişelerini ortadan kaldırır. Snowball ile veri aktarımı basit, hızlı, güvenli ve daha az maaliyeti olabilir.

AWS Snowball Edge

AWS Snowball 80TB veriyi (72TB usable) tek bir cihazda taşıyamaz. Bu nedenle 72TB ve üstü veriler için AWS snowball'dan ziyade, snowball edge tavsiye edilir.

AWS Shared Responsibility Model



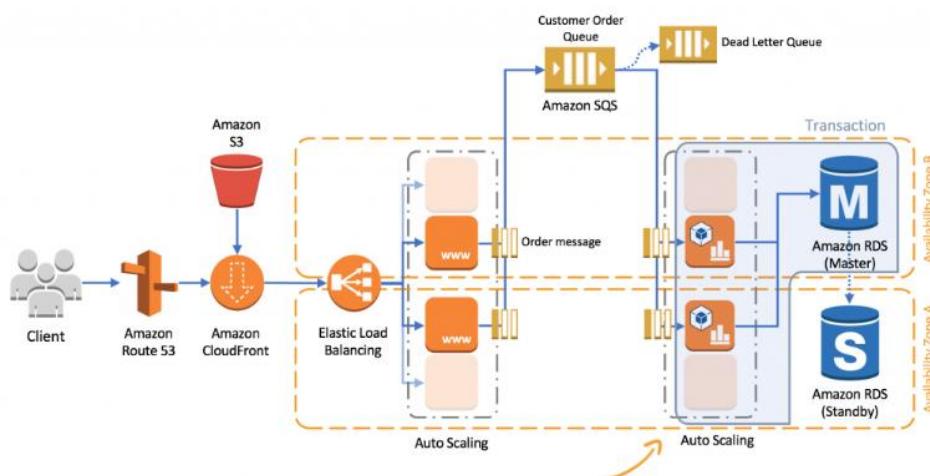
Güvenlik gereksinimlerinin müşteri ve AWS arasında paylaşılmış durumudur.
Aşağıdaki güvenlik noktalarından AWS sorumludur.

- Facilities
- Physical security of hardware
- Network infrastructure
- Virtualization infrastructure

Müşteri ise aşağıdakilerden sorumludur,

- Amazon Machine Images (AMIs)
- Operating systems
- Applications
- Data in transit
- Data at rest
- Data stores
- Credentials
- Policies and configuration

Decoupled Architecture

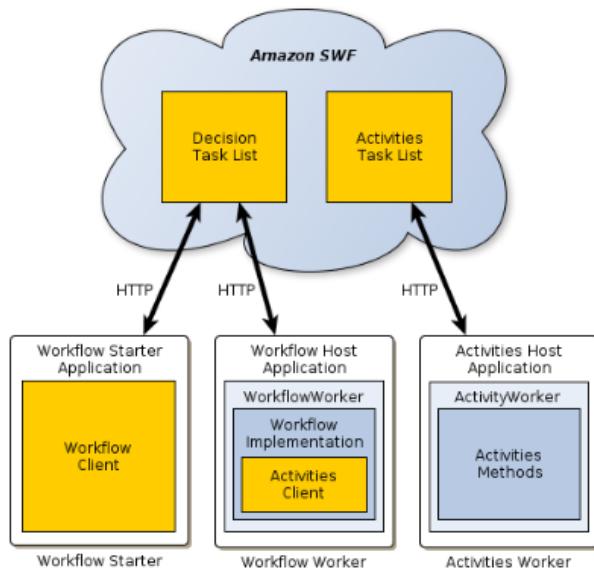


Amazon SQS ve Amazon Simple Workflow Service (SWF) decoupled architecture olarak kullanılabilecek servislerdir.

Bu şekilde her iki servisi aynı mimaride kullanarak, farklı katmanlarda ayrı akışlar sağlanabilir.

Amazon Simple Workflow Service (SWF): Developer'ların paralel ve sequential olarak işlerinin çalışmasına olanak sağlar. Cloud'da yer alan job coordinator olarak düşünülebilir.

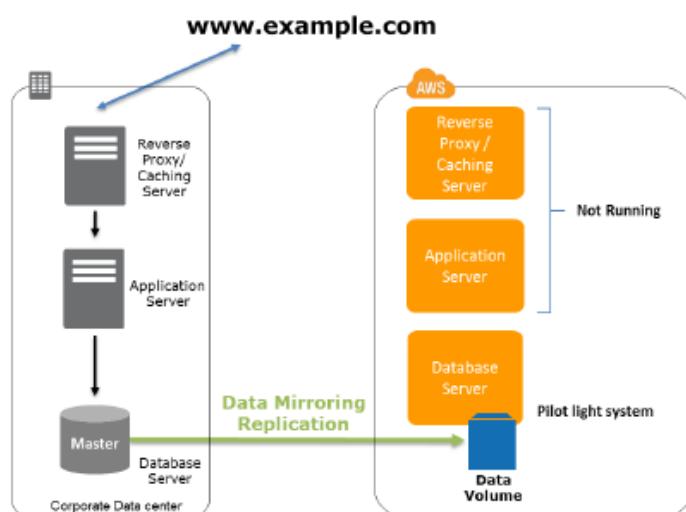
- **Decision Task:** Decider'a, workflow execution durumunu bildirmekle sorumludur.



Örneğin, uygulama adımları 500 milisaniyeden uzun sürerse, processing durumunu takip etmemiz, bir görev başarısız olursa kurtarmamız ve yeniden denememiz gerekebilir. Bu durumda bu gibi işi yapmak için SWF kullanabiliriz.

SWF'de, by default her workflow execution'ı max 1 yıl çalışır. Bu nedenle bazı workflow'lar manual müdahale gerektirebilir. Mesela bir workflow'a max duration time 1 gün verilirse, 1 günden sonra idle execution'a düşecektir.

AWS Pilot Light

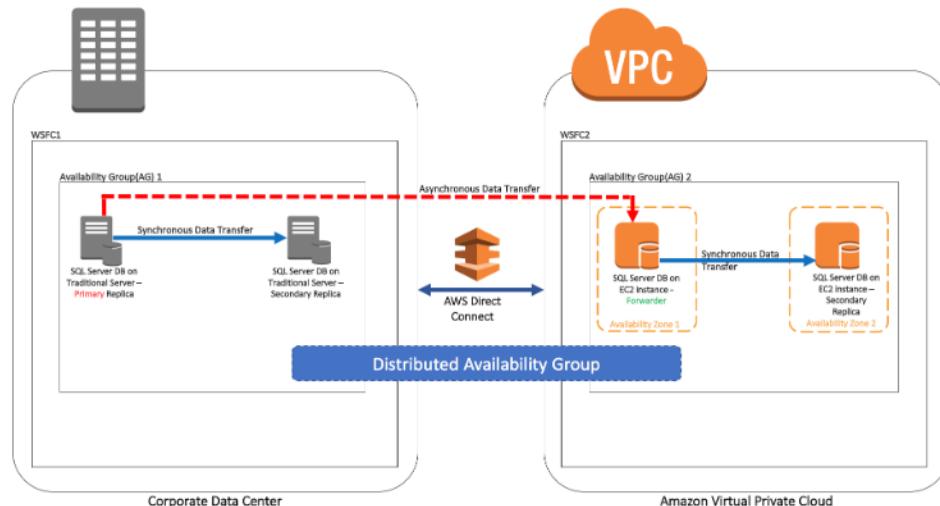


Minimum DR çözümüdür. Gaz ısıtıcısından esinlenerek bu şekilde bir isim verilmiştir. Altında yatan felsefe ise, **bir gazlı ısıtıcıdan gelecek küçük bir alev, bir evi ısıtabilecek bir fırını ateşleyebilir.**

Bu senaryo, bir yedekleme ve geri yükleme senaryosuna benzer. AWS'de çalışan en kritik elementleri sürdürmek için kullanılabilir.

Recovery ihtiyaç olduğu zaman, kritik element için hızla tam ölçekli bir prod ortamı sağlanabilir.

Hybrid Cloud Architectures with AWS



Hybrid Cloud Architecture ile on-premise kaynakları ile public cloud kaynaklarının beraber kullanılmasıdır. Bu mimari ile uygulamalar ve data cloud'a migrate edilebilir ve datacenter kapasitesi arttırılabilir.

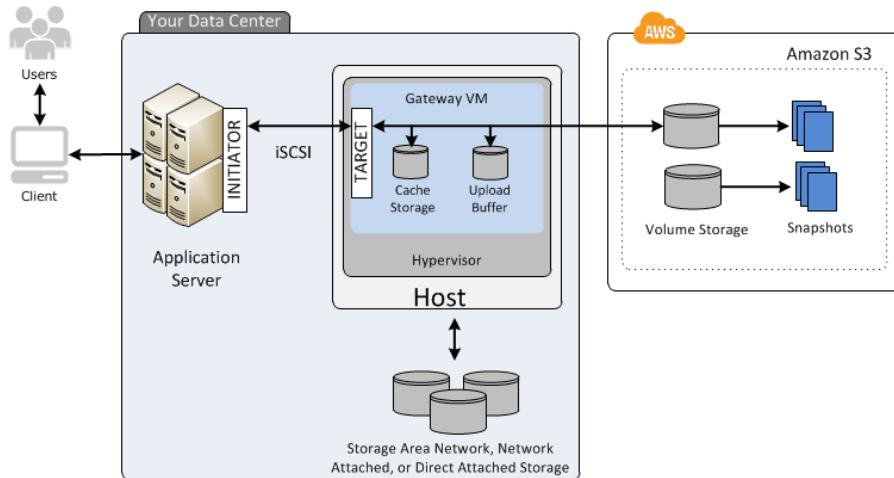
AWS Elastic Beanstalk



Apache, Nginx, Passenger ve IIS gibi sunucular üzerinde Java, .NET, PHP, Node.js, Python, Ruby, Go ve Docker ile geliştirilmiş web uygulamalarını dağıtıp ölçeklendirmek için kullanılır.

Beanstalk ile, uygulamalar altyapıya gerek kalmadan hızlıca AWS'e deploy edilebilir. Application deploy edildikten sonra Elastic Beanstalk otomatik olarak capacity provisioning, load balancing, scaling ve uygulama health monitoring işlerini yapacaktır.

AWS Storage Gateway

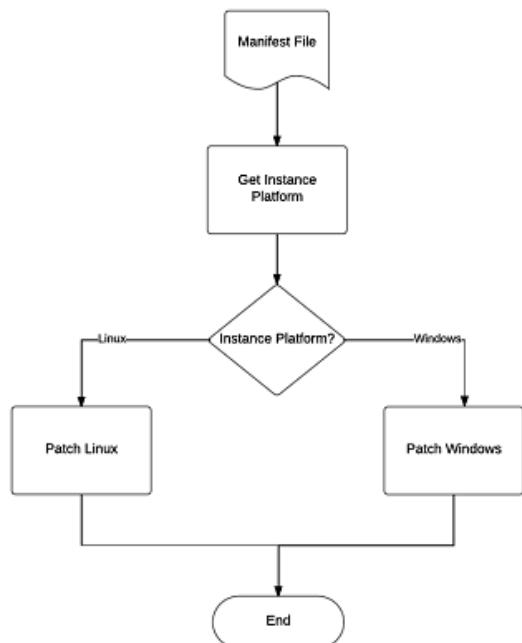


On-premise software appliance ile Cloud-Based storage'ın birlikte kullanılması olanağ sağlar. Bu sayede on-premise'de bulunan bir yazılım, storage olarak AWS ortamını kullanmış olur.

AWS Storage Gateway Cached Volumes

Data S3'de muhafaza edilir ve çok sık erişilen data subset'i on-premise network'ünde saklanır. Cached volume, on-premise storage'da ki maaliyetin düşürülməsinə olanağ sağlar ve on-premise storage ihtiyacını en aza indirir. Ayrıca sık kullanılan verilerde low-latency sağlanır.

AWS Step Functions



Modern uygulamalar için serverless orchestration sağlar. Orchestration; iş akışını birden fazla aşamaya bölgerek, flow logic ekleyerek ve adımlar arasındaki input output'a bakarak, akışı yönetir.

Arada yer alan adımlarda sorun yaşanırsa, uygulama kaldığı yerden devam edebilir. Böyle bir akış ile patch management, infrastructure seçimi ve data synchronization ile process iyileştirilebilir.

Amazon Resource Names (ARNs) ve AWS Service Namespaces

Amazon Resource Names, unique olarak AWS servislerini belirler. IAM policy, Amazon RDS tags ve API call gibi AWS'de açıkça bir kaynak belirtmek için kullanabiliriz.

Örnek:

```
<!-- Amazon RDS instance used for tagging -->  
arn:aws:rds:eu-west-1:123456789012:db:mysql-db
```

IAM policy veya ARN oluşturulduğu zaman, bir namespace kullanarak, bir AWS servisi tanımlarız. Örneğin, Amazon S3 için namespace s3 ve Amazon EC2 için ec2'dir. Action ve resource tanımlarken namespace kullanılır.

Örnek:

```
"Effect": "Allow",  
"Action": "s3.*",  
"Resource": "arn:aws:s3:::example_bucket/marketing/*"
```

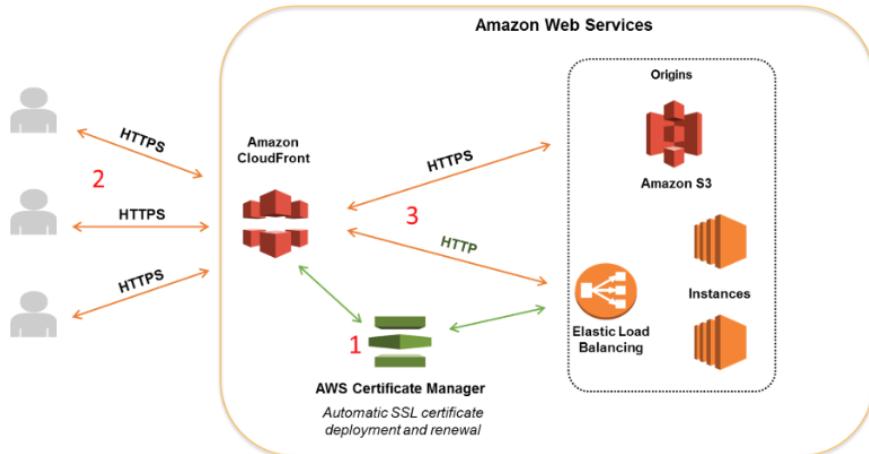
AWS Resource ID: Resource'ları yalnızca Amazon EC2 konsolunda bulmak için kullanılır.

AWS Data Pipeline



Farklı AWS compute ve storage servisleri ile on-premise data source arasında belirli aralıklarla verileri process etmeye ve taşımamıza yardımcı olan bir web servistir.

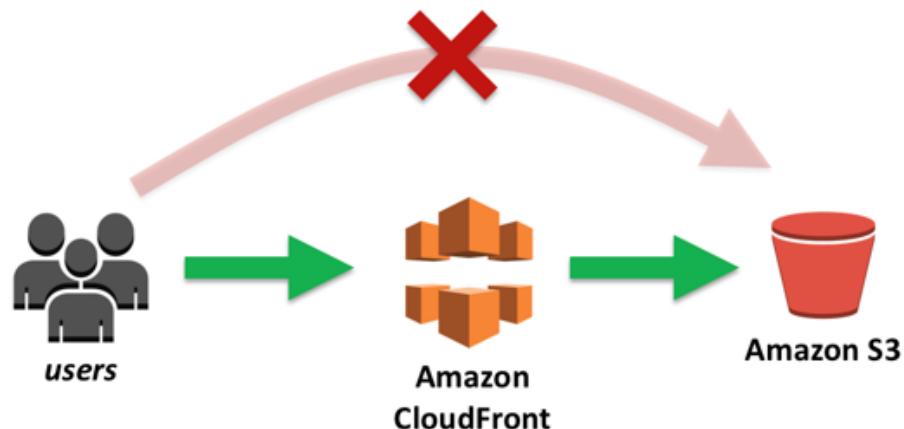
AWS Certificate Manager



AWS Certificate Manager, AWS kaynakları ve internal kaynakları public ve private kullanmak için SSL/TLS sertifikalarını kolayca tedarik edilmesine, yönetilmesine ve deploy edilmesine olanak sağlayan bir hizmettir.

Bir sistem için SSL çözümü uygulanmak istensin ve bu ihtiyaç için 3rd party bir CA'dan sertifika alınmış olsun. Bu durumda bu sertifikaların ACM veya IAM certificate store'a atılması gerekmektedir.

Origin Access Identity (OIA)

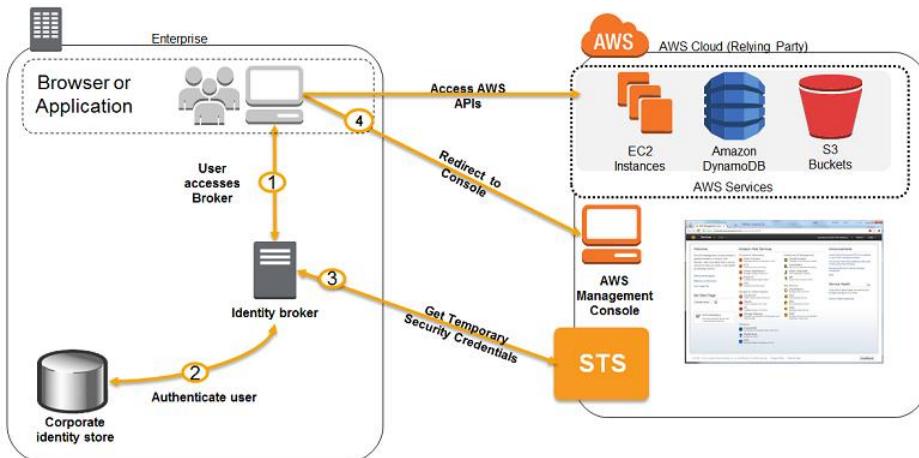


S3 içerisinde erişimlerin kısıtlanması sağlanır.

Amazon S3 bucket’da yer alan dosyalara erişimi sınırlamak için CloudFront signed URL’ler veya signed cookie oluşturulur ve ardından origin access identity (OIA) adı verilen özel bir CloudFront kullanıcıı oluşturup, distribution ile ilişkilendirilebiliriz.

Ardından permission’ları, CloudFront’u kullanıcıların dosyalara ulaşması için OIA’yi kullanabileceği şekilde yapılandırabiliriz. Böylece bir yapıda kullanıcılar doğrudan bir URL kullanmazlar.

Identity Broker Application

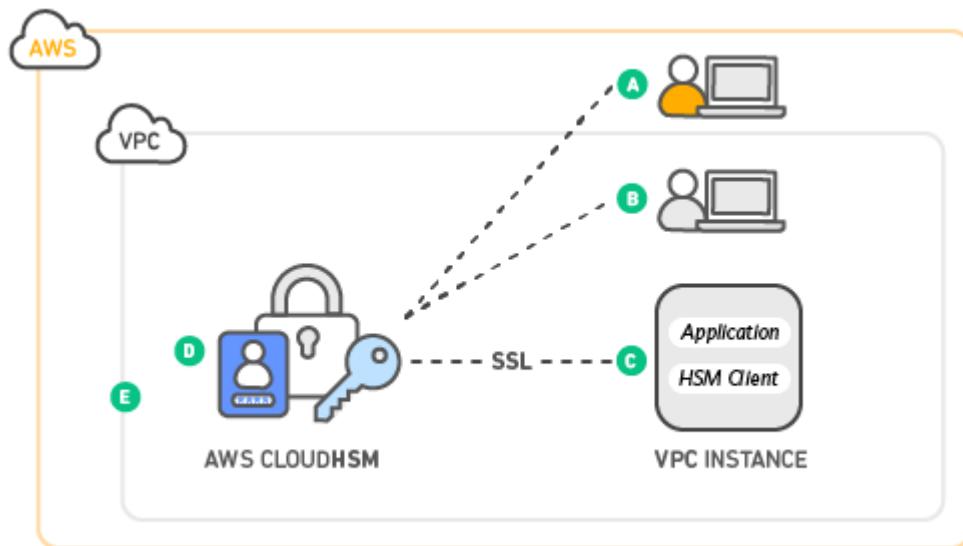


Identity store SAML 2.0 ile uyumlu değilse, custom identity broker uygulaması oluşturulmalıdır. Broker uygulaması kullanıcıların kimlik bilgilerini doğrular, AWS'deki kullanıcılar için temporary credential ister ve ardından AWS kaynaklarına erişime izin verir.

AWS OpsWorks

Aws OpsWorks, Chef ve Puppet'in managed instance'larını sağlayan bir configuration management servisidir. Chef ve Puppet, sunucuların yapılandırmasını sağlayan otomasyon platformlarıdır. OpsWorks, Chef ve Puppet'i kullanarak, sunucuların Amazon EC2 instance'ları veya On-Premise compute ortamlarında yapılandırma, dağıtım ve yönetim biçimini otomatize edilmesini sağlar.

AWS CloudHSM (Hardware Security Module)



Cloud based hardware security module'dur ve kendi encryption key'lerimizi, kolayca generate edip, kullanmamıza olanak sağlar.

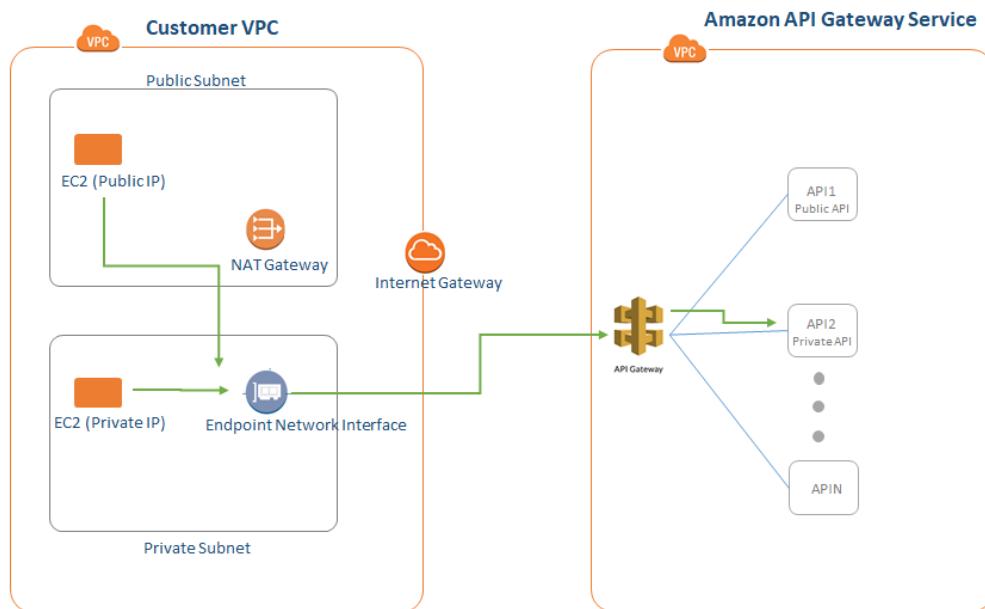
Olası bir duruma karşı bi AZ'da değil, birden fazla AZ'da key storage alanı sağlar. Kullanıcı kontrol edebilir ve yönetebilir. Kullanıcının kendi VPC içerisinde yer alır ve diğer AWS network'ünden izole durumdadır.

Amazon, kullanıcıların HSM credential'larına veya kullanıcı key'lerine erişimi yoktur. Bu nedenle eğer credential kaybedilirse, recover edilemez. Amazon ayrı AZ'da olmak üzere, iki veya HSM kullanılmasını tavsiye ediyor.

AWS CloudHSM, özel bir FIPS 140-2 seviye 3 HSM sağlamaktadır.

Amazon Workspace: Güvenli bir cloud desktop hizmetidir. Bir kaç dakika içerisinde Windows veya Linux desktop tedarik edebilir ve dünyanın dört bir yanındaki çalışanlara binlerce masaüstü sağlanabilir.

VPC Endpoints



Support edilen AWS servislerine, Internet Gateway, NAT device, VPN bağlantısı veya AWS Direct Connection bağlantısı gerektirmeden VPC'yi private olarak bağlanılmasını sağlar.

VPC içerisindeki instance'lar, Public IP'ye ihtiyaç duymaz ve VPC ve diğer servisler arasındaki trafik Amazon network'ünden çıkmaz.

Interface endpoint ve gateway endpoint olmak üzere iki çeşit endpoint vardır.

Interface endpoint, elastic network interface'dir. Private ip adresi, trafik hedefli bir entry point'e hizmet eder.

Gateway endpoint, route tablosunda yer alan spesifik bir route'un, support edilen AWS servisine yönlendirilmiş bir gateway'dir.

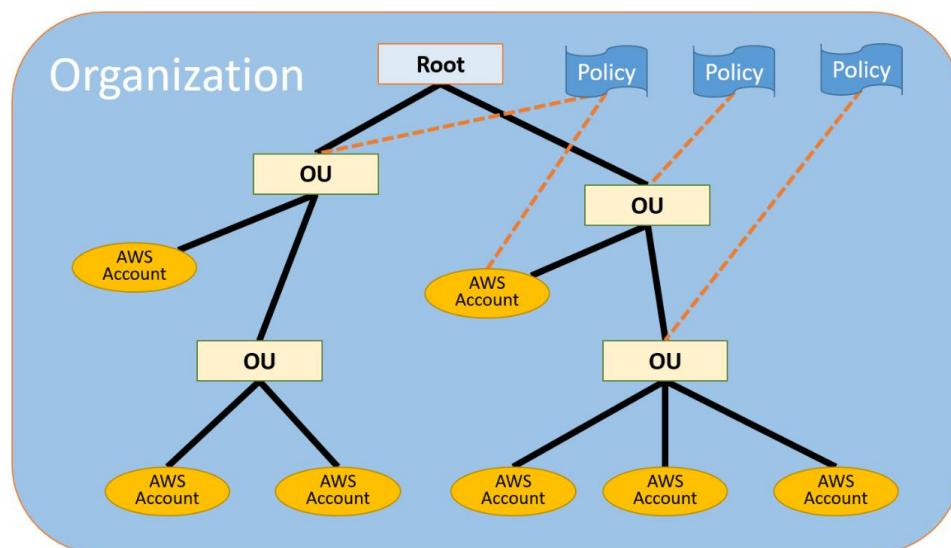
Multicast Network Capability

Birden çok data distribution'a izin veren bir network özelliğidir.

Multicasting ile bir veya daha fazla source, network paketlerini multicast grup içinde bulunan üyelere gönderir. Amazon VPC multicast veya broadcast networking'i desteklememektedir.

Bu şekilde bir ihtiyaç var ise ve mevcut ortam AWS'e taşınacak ise, **Overlay Multicast** kullanılabilir. Overlay multicast, VPC gibi tek bir noktaya IP yönlendirmesini destekleyen bir network yapısı boyunca IP seviyesinde çok noktaya yayın oluşturma yöntemidir.

AWS Organizations



Birden fazla AWS account'ı için policy based management sağlar.

Grup oluşturabilir, account oluşturmayı otomatize edebilir ve bu gruplar için policy tanımlanabilir.

Organizations custom script ve manual bir şey gerektirmeden, **Service Control Policies (SCPs)** ile birden fazla AWS account'unu merkezi bir elden yönetilmesini sağlar.

AWS CodeDeploy

On-Premise instance'lara, EC2 instance'lara uygulama deployment'ları otomatize eden bir deployment servisidir.

Hızlı bir şekilde yeni feature'ların release edilmesine, Lambda fonksiyonlarının güncellenmesine, uygulama deployment sırasında downtime oluşma ve manual deployment sırasında oluşabilecek karmaşasının engellenmesine olanak tanır.

Egress-Only Internet Gateway

VPC'deki instance'lardan IPv6 üzerinden, internet'e outbound connection'ı sağlar. Sadece IPv6 trafiği için kullanılır. IPv4 ile ilgili benzer bir işlem yapacak ise, NAT Gateway kullanılmalıdır.

Blue-Green Deployment

AWS CodePipeline kullanarak, otomatik olarak **Blue-Green** deploy sürecini gerçekleştirir. Yaklaşık 15 dakika içerisinde kesintisiz CI/CD pipeline oluşturur.

Bir uygulama AWS Elastic Beanstalk ortamında geliştirildiğinde, aynı ortam içerisinde paralel iki ayrı ama aynı işleri işleyen Blue-Green alanı oluşturur ve riskleri azaltır.

Bu mimaride Blue alan Production environment'dır ve mevcut trafiği üstlenir. CI/CD pipeline mimarisi, live Elastic Beanstalk alanı olan Blue'nun bir clone'u oluşturur ve buna da Green environment denir.

Ardından pipeline URL'leri bu iki ortam arasında değiştirir.

CodePipeline uygulama kodunu orjinal environment'a deploy ederken, test ve maintenance süreci başlar ve temporary clone environment trafiği üstlenir.

Blue environment'a başarılı tamamlandıktan ve gözden geçirme ve code testing tamamlandıktan sonra, pipeline Blue ve Green environment'ları tekrar değiştirir ve Blue environment trafiği tekrar üstlenir ve pipeline green environment'ı terminate eder.