Exam 130 minutes
60 questions
Multiple choice
1000 점 중 720 점
시나리오 베이스 질문

* IAM
- AWS 서비스와 리소스에 대한 액세스 관리
- AWS 사용자 및 그룹 생성과 AWS 리소스에 대한 액세스 허용 및 거부 관리
- 멀티팩터 인증
- 일시적인 접근 제공
- 구성
    - User : 사용자
    - Group : 그룹별 권한 허용 가능
    - Policies : User 와 Group, Role 이 사용할 수 있는 권한 설정
- Roles : 정책을 사용자 입맛에 맞게 묶은 것
- Region 에 국한되지 않고 사용 가능
- 루트 계정은 계정을 처음 설정할 때 생성된 계정
- 신규 유저는 생성시 아무런 권한 없음
- 신규 유저는 Access Key Id 와 Secret Access Key 가 할당됨
- Programmatics Access, Console Access 2 가지로 권한 부여 가능
- Access Key Id 와 Secret Access Key 는 콘솔 로그인시 사용 불가능
- 각 키는 최초 생성시만 볼 수 있으며 즉시 보관해야함
- 멀티팩터 인증은 항상 설정해두는 것이 좋음

* S3
- S3 는 스토리지로서 웹 서비스 인터페이스를 통해 사용 가능
- S3 는 Object Base 이며 0 Byte 부터 5TB 까지 업로드 가능
- 모든 Object 은 Bucket 에 저장됨
- S3 의 이름은 반드시 고유해야 함(Universal Name Space)
- S3 에 업로드할 때 업로드에 성공하면 HTTP 200 Code 를 전달 받음
- Object 는 다음을 포함
    - Key : Object 의 이름
    - Value : 데이터이며 연속적인 바이트로 이루어져 있음
    - Version ID : 같은 데이터라도 버전 별로 나눌 수 있기에 중요
    - Metadata : 데이터의 데이터
- Read after Write consistency for PUTS of new Object : 새로운 키로 업로드한 데이터는 분산복제가 끝나기 전까지 읽어올 수 없음(신규 업로드시만 해당)

- Eventual Consistency for overwrite PUTS and DELETES : 빠른 응답성을 위해 분산복제가 완료되기 전에는 과거 데이터를 읽어올 수도 있음(덮어쓰기 PUT, DELETE 해당)
- 99.999999999% 내구성 보장(9 가 11 개임)
- Tiered Storage Available, Lifecycyle Management, Versioning, Encryption, MFA Delete 등의 기능 제공
- ACL, Bucket Policy 로 접근 제어 기능 제공
- CRR(Cross Region Replication) : 한 리전의 S3 를 다른 리전의 S3 에 업로드마다 복제하는 것
- Transfer Acceleration : Cloudfront 의 Edge location(아마존의 백본 네트워크)를 이용하여 빠르게 전송하는 것

* S3 클래스(Object 의 클래스, 오브젝트별로 지정 가능)
- Standard : 99.99% 가용성, 99.999999999% 내구성, 다수의 시설 내 다수의 장비에 중복 저장되도록 설계, 2 개 AZ 가 파괴되어도 유지 가능
- S3-IA(Standard-Infrequent Access) : 자주 접근하지 않지만 필요시 빨리 접근해야하는 데이터를 위한 클래스로 S3 보다 싸나 검색비용이 생김
- S3 One Zone - IA : IA 보다 싸지만 다수의 AZ 에 복사하지 않음
- S3 - Intelligent Tiering : 업로드 주기가 매우 불규칙한 경우의 스토리지
- S3 Glacier : 데이터 저장용 S3 으로 매우 싸나 검색시간이 매우 오래 걸릴 수 있음(3-5 시간까지)
- S3 Glacier Deep Archive : 가장 싼 스토리지 클래스로 12 시간
- 가용성은 Standard 가 99.99%, Intelligent-Tiering, IA 가 99.9%, One Zone-IA 는 99.5%, Glaicer(Deep Dive)는 99.9%
- AZ 의 경우 One Zone IA 를 제외하고 모두 3 개 이상에 저장됨

* S3 Security
- 암호화는 Client Side 와 Server Side 로 나뉘며 Client Side 는 Client 에서 S3 로 전송될 때의 암호화(Data at transit)을, Server Side 는 S3 에 저장될 때의 암호화(Data at rest)를 의미함
- Server Side Encryption
    - SSE-S3 : S3 의 고유한 키로 암호화를 실시하며 암호화 주체가 S3 가 되는 방식. 암호 알고리즘은 AES-256.
    - SSE-KMS : Key Management Service 를 이용하여 객체를 암호화하는 방식으로 KMS 고객 마스터 키(CMK)를 활용함. SSE-S3 와 달리 고객에 키를 제어할 수 있음

- SSE-C : 고객(Customer)가 제공하는 키로 암호화를 진행하는 방식으로 제공된 암호화 키를 사용하여 디스크를 쓰거나 해독할 때 객체에 액세스할 때의 모든 암호화를 관리함. 제공된 암호화키는 저장되지 않음

- Client Side Encryption
    - S3 로 데이터를 보내기 전의 암호화
    - KMS 에 저장된 고객 마스터키를 사용하여 암호화
    - 애플리케이션 내 마스터 키를 사용하여 암호화

* S3 Versioning
- 객체의 모든 버전을 저장함
- 한 번 사용하면 해제 불가능, 중지만 가능
- 라이프사이클과 통합하여 사용하며 MFA Delete 제공
- 최신 버전의 오브젝트를 삭제하더라도 versioning 에 남아있음
- 최신 버전이 완전 삭제되면 그 직전 버전이 최신 버전이 됨

* S3 Lifecycyle
- 일정 기간이 지나면 자동적으로 다른 스토리지 티어로 이동하게 하는 서비스
- 현재 버전과 이전 버전(Versioning)에 적용 가능
- 만료 구성(Configure Expiration)을 통해 일정 기간이 지난 현재 버전과 이전 버전의 오브젝트 삭제 가능
- 현재 버전 : 지정된 생성기간 이후 현재 버전의 오브젝트 삭제
- 이전 버전 : 객체의 (이전 버전으로) 지정된 기간 후 이전 버전의 오브젝트 삭제
- 불완전한 멀티파트 업로드를 지원하여 S3 에 제대로 업로드되지 않은 데이터를 삭제할 수 있음

* S3 CRR(Cross Region Replication)
- Source S3 Bucket 와 Destination S3 Bucket 에 Versioning 이 활성화되어야 함
- 복제할 권한을 부여할 IAM Role 필요(GetReplicationConfiguration, ListBucket)
- 복제 구성을 추가하기 전의 파일들은 복사되지 않음
    - 복제 구성을 추가한 후 생성된 객체들을 대상으로 함
- 연속적으로 업데이트되는 파일들은 자동으로 복제됨
- Delete Marker, 암호화된 객체는 복제되지 않음

* S3 Transfer Acceleration
- S3 에 직접 업로드하는 것이 아닌 Cloudfront 의 Edge Location Network 를 이용하여 Edge Location 에 업로드하고 그것을 S3 로 옮기는 서비스

* Cloudfront

- content deivery network 로서 웹페이지나 다른 웹 콘텐츠를 전달
- Edge Location : 콘텐츠가 캐시되기 위한 공간으로 Region 과 AZ 와는 다른 별개의 장소이며 전세계에 퍼져있음
- Origin : 캐시된 데이터들의 원래 서버
- Static, Dynamic, Streaming 서비스 등을 캐시
- Distribution : 캐시하고자 하는 컨텐츠를 배포하고 설정하는 과정
- 모든 컨텐츠들은 TTL 의 주기에 따라 정해진 시간동안 캐시됨
- Cache 를 클리어할 수 있지만 요금 부과됨

* Snowball
- 대용량의 데이터를 안전하게 옮기기 위해 사용되는 솔루션
- S3 에 Import 하거나 Export 가능
- Snowball 이라는 물리적인 실체가 존재하는 기기를 이용하여 Local PC 에서 AWS 로 데이터 전송 가능

* Storage Gateway
- On-premise Software 를 Cloud Base Storage 에 연결하여 사용하는 것
- File Gateway(NFS), Volume Gateway(iSCSI, Stored, Cached), Tape Gateway(VTL) 등이 있음
- File Gateway : NFS mount point 를 통해 S3 bucket 에 저장됨, 한 번 옮겨지면 Object 처럼 관리됨, S3 에 다이렉트로 저장됨(for flat file)
- Volume Gateway : 자주 사용하는 데이터는 Cache Storage 에 저장하고 나머지 데이터를 S3 에 백업(Cached Volume), 기본 데이터를 로컬에 저장하는 한편 데이터를 비동기식으로 S3 에 백업(Stored Volume)
- Tape Gateway : VTL(가상 테이프 라이브러리)를 지원하는 Storage Gateway 로 가상 테이프 데이터는 S3 나 S3 Glacier 에 저장될 수 있음

* EC2(Elastic Compute Cloud)
- 가상 서버를 제공하는 서비스
- 요금체계 : On demand, Reserved, Spot(시작과 종료시간이 자유로울 때, 매우 저렴한 서버 요금이 요구될 때)
    - 온디맨드(On Demand) : 필요할 때 바로 생성하여 사용하는 방식으로 1 시간 단위로 과금이 이루어짐, 1 분을 사용하더라도 1 시간 과금을 물리는 방식
    - 스팟(Spot) : 경매 방식의 인스턴스, 최초 생성시 기준가격이 화면에 나타나며 화면의 가격보다 높은 가격을 제시하면 계속 사용이 가능함. 그러나 다른 사람이 더 높은 가격을 입찰했다면 인스턴스가 종료됨. 불시에 중단되어도 상관없거나 각종 테스트에 적합

- 예약(Reserved) : 12 개월 또는 36 개월 단위로 예약하여 사용하는 인스턴스로 온디맨드에 비해 가격이 대폭 할인됨. 장기적으로 사용할 경우 추천, 예약 인스턴스이기 때문에 사용하지 않더라도 요금이 부과됨

- Spot 의 경우, EC2 에 의해 종료되면 부분 사용 시간의 요금은 부과되지 않지만 스스로 끌 경우 부과됨

- 우발적 종료 보호는 반드시 사용해야 함

- EBS(Elastic Block Storage) 기반의 인스턴스에서는 인스턴스가 종료될 때 Root Device Volume 도 함께 종료됨

- Root Device Volume 은 기본적으로 암호화될 수 없으므로 서드 파티 툴을 이용하여 루트 볼륨을 암호화할 수 있음

- 추가적인 볼륨은 암호화가 가능

* Security Group

- EC2 인스턴스의 인바운드/아웃바운드 트래픽을 제어하는 가상 방화벽

- 설정을 변경하면 즉시 적용됨

- 체적인 포트와 IP 허용은 가능하나 차단은 불가능(Network ACL)

- 한 개 이상의 Security Group 부착 가능

- Security Group 을 새로 생성시 모든 인바운드 트래픽은 기본적으로 Block 됨

- 기본적으로 모든 아웃바운드 트래픽은 허용됨

- 다수의 EC2 가 하나의 Security Group 공유 가능

- 다수의 Security Group 이 하나의 EC2 공유 가능

- Stateful 특징을 가지기 때문에 Deny Rule 은 없음

* EBS(Elastic Block Storage)

- 영구적인 블록 스토리지 볼륨으로 EC2 인스턴스의 디스크로 사용됨

- 가용성을 위해서 AZ 내에서 자동으로 복제됨

- 볼륨 유형으로는 General Purpose(SSD), Provisioned IOPS(SSD), Optimised Hard Disk Drive, Cold hard Disk, Magnetic 이 존재함

- 범용 SSD(gp2) : 시스템 부트 사용 가능, 대부분의 워크로드에서 사용

- 프로비져닝된 IOPS SSD(io1) : 지속적인 IOPS 성능이나 16,000 IOPS 이상의 볼륨당 처리량을 필요로 하는 경우 적합(DB 워크로드)

- 처리량 최적화된 HDD(st1) : 시스템 부트 사용 불가능, IOPS 가 아닌 처리량을 기준으로 하며 자주 액세스하는 워크로드에 적합한 저비용 HDD 볼륨, 빅데이터나 데이터 웨어하우스에 사용

- Cold HDD(sc1) : 시스템 부트 사용 불가능, 자주 액세스하지 않는 대용량 데이터 처리에 적합, 스토리지 비용이 최대한 낮아야 할 경우 사용

* Volume & Snapshots
- EC2 Instance 와 EBS 는 같은 AZ 에 속함
- 사용중에도 볼륨 타입과 사이즈를 변경할 수 있음
- 변경중에도 인스턴스를 중지하거나 내릴 필요 없음
- 볼륨을 다른 AZ 로 옮기는 방법은 스냅샷을 생성하여 옮기는 방법이 있음
- 다른 Region 으로 옮기기 위해서는 AMI(Amazon Machine Image)를 복사하여 옮기는 방법이 있음
- Root Device Volume 은 인스턴스 삭제시 같이 사라지지만, 추가적인 볼륨들은 인스턴스가 삭제되어도 남아있음(옵션을 통해 삭제 가능)
- 볼륨은 EBS 내에 존재하며, 스냅샷은 S3 에 저장됨
- 스냅샷은 복사된 볼륨의 복사된 시점임
- 스냅샷은 끊임없이 증가함 : 마지막 스냅샷 이후로 변경된 부분만 S3 로 이동함
- Root Device 로 사용되는 EBS 볼륨의 스냅샷을 생성하기 위해서는 인스턴스를 잠시 멈춰야 함
- 볼륨과 스냅샷 둘 다 AMI 를 생성할 수 있음

* EBS vs Instance Stored
- AMI 에서 생성된 인스턴스의 root device 는 EBS Snapshot 에서 생성됨(EBS volume), AMI 에서 생성된 인스턴스의 root device 는 S3 에 저장된 템플릿에서 생성됨(Instance store)
- EBS 기반 인스턴스는 Stop 시 모든 데이터가 EBS 에 저장됨
- Instance store 기반의 인스턴스는 Stop 이 불가능함
- Instance store volume 은 Ephemeral Storage 로도 불림
- 기본적으로 Root volume 은 인스턴스 삭제시 사라지지만 EBS 의 경우, 제거되지 않게 할 수 있음

* Root device volume 의 암호화
- 최초 인스턴스 생성시 Root volume 은 암호화 불가능
- Volume 의 스냅샷을 생성하여 copy(암호화 옵션클릭)하여 암호화된 스냅샷 생성, 암호화된 스냅샷을 이용하여 AMI 생성, 이를 이용하여 인스턴스 생성
- 암호화된 볼륨의 스냅샷은 자동적으로 암호화됨
- 암호화된 스냅샷에서 생성된 볼륨은 자동적으로 암호화됨
- 스냅샷은 공유 가능하지만 오직 비암호화되었을 때만 가능

* CloudWatch
- AWS 클라우드 리소스와 AWS 에서 실행되는 애플리케이션을 위한 모니터링 서비스
- Cloudwatch 는 성능을 모니터링

- EC2,Autoscaling, ELB, Route 53 Health Check, EBS Volume, Storage Gateway, Cloudfront 등의 성능 관제를 지원
- CPU, Network, Disk, Status Check 등의 항목 존재
- CloudTrail 은 console action 과 API Call 을 기록하여 어느 유저와 계정이 얼마나 요청을 하는지 확인 가능케 함
- Cloudwatch 는 성능을 감시하지만, Trail 은 요청을 감시함
- CloudWatch 는 기본적으로 5 분 간격으로 모니터링(EC2, Standard Monitoring)하지만 세부 모니터링 사용시 1 분으로 변경 가능
- 세부 모니터링은 EC2 생성시 별도로 활성화해주어야 하며, 하지않으면 Standard Monitoring 적용

* Elastic File System
- 네트워크 파일 시스템(NFSv4)를 사용하여 파일 스토리지 서비스
- 사용중인 스토리지에 대해서만 과금
- 페타바이트까지 확대 가능
- 수천 개의 커넥션 설정 가능
- 데이터는 다수 AZ 에 걸쳐 저장됨
- Autoscaling 이 자동으로 적용되기 때문에 미리 크기를 프로비져닝할 필요 없음

* Relation Database System
- 관계형 데이터베이스를 지원하는 서비스
- 종류로는 SQL Server, Oracle, MySQL Server, PostgreSQL, Aurora, MariaDB 등이 존재함
- Read Replicas, Multi-AZ(AZ1 - Primary Database, AZ2 - Secondary) 지원
- OS 에 접근하여 제어할 수 없음
- OS 와 DB 패치는 전적으로 아마존이 책임짐
- RDS 는 Serverless 서비스가 아님(Aurora 는 Serverless 임)

* RDS Backup, Multi-AZ, Read Replicas
- 백업은 1 일에서 35 일까지 가능(기본값은 7 일)
- 기본적으로 활성화되어 있으며 백업 데이터는 S3 에 저장됨(10Gb 의 RDS 인스턴스가 있다면, S3 에 10Gb 의 스토리지가 있는 것과 같음)
- 암호화는 MySQL, Oracle, SQL Server, PostgreSQL, MariaDB & Aurora 에 KMS 를 사용하여 제공됨
- Read Replica, Backup, Snapshot 모두 암호화됨
- 'A' AZ 의 DB 를 통해 서비스하다 'A'가 죽으면 'B' AZ 의 'B' DB 로 서비스(즉 리부팅을 통해 failover)
- 자동으로 failover 됨

- Multi AZ 는 MySQL, Oracle, SQL Server, PostgreSQL, MariaDB 의 서비스만 가능
- Read Replica(RR)는 로드 밸런싱을 위한 읽기 전용 DB
- Read Replica 는 MySQL, PostgreSQL, MariaDB, Aurora 서비스만 가능(MPMA)
- RR 는 부하 분산용이지 DR 이 아님, 자동백업이 반드시 활성화되어있어야 함
- 어느 DB 든지 최대 5 개까지 보유 가능하며, Read Replica 의 Read Replica 을 가질 수 있음(지연시간 존재)
- 각 Read Replica 은 자신만의 DNS 엔드 포인트를 가지며 Multi-AZ 지원 가능
- Read Replica 은 DB 로 승격 가능
- Read Replica
    - 반드시 백업이 활성화되어있어야 하며, Multi-AZ 를 지원함
    - 다른 리전에도 둘 수 있음
    - Aurora 혹은 MySQL 가능(Read Replica 의 Read Replica)
    - 마스터로 승격 가능하지만 RR 이 사라짐

* Dynamo DB
- 빠르고 유연한 NoSQL DB 서비스
- SSD 스토리지에 저장되며, 지리적으로 나누어진 3 곳의 데이터 센터(AZ)에 분산저장됨
- 테이블, 항목, 속성 등의 구성요소로 나뉘며 테이블의 각 항목을 나타내는 고유 식별자인 '기본 키'가 있음
- Dynamo DB 의 일관성 모델 : Eventual Consistent Reads, Strongly Consistent Reads
- Eventual Consistent Reads
    - 모든 데이터의 복사본은 수 초 내에 도달 가능하나 업데이트되지 않은 데이터가 전달될 수 있음, 읽기를 반복하면 최신 데이터를 반환함
- Strongly Constitent Reads
    - 결과값을 리턴할 때 읽기 전 모든 변경점을 반영한 상태로 리턴함
- Autoscaling 지원

* Redshift
- Leader Node : 커넥션 관리와 쿼리 유입 담당
- Compute Node : 데이터 저장 및 쿼리 수행, 계산 담당(128 개까지 구성가능)
- Cluster : Leader Node 와 Compute Node 의 집합
- Single Node : 한 개의 Node 가 'Leader Node'와 'Compute Node' 역할을 모두 맡는 Node
- Multi Node : 'Leader Node'와 'Compute Node'가 분리되어 있는 Node
- Massively Parallel Processing(MPP)
    - 쿼리 성능을 높이기 위한 기능

- Backup : 기본적으로 하루동안 보존하나 35 일까지 설정 가능
- Compute Node 내 원본과 복사본은 S3 에 저장함
- 스냅샷을 S3 에 복사하여 저장함
- 아직까지는 Multi AZ 는 지원하지 않음(1 AZ 가능)

* Aurora
- 오픈 소스 데이터베이스를 기반으로 하는 DB 엔진
- 사이즈는 10GB 부터 64TB 까지 있음
- 32vCPU 와 244GB 메모리까지 확장 가능
- 각 AZ 당 2 개의 데이터 복사본을 가지고 있음
- 에러를 스스로 찾아내 복구함
- Replica 의 경우 최대 15 개까지 가능함
- 데이터 손상 없이 Failover 가능함
- 백업과 스냅샷이 퍼포먼스에 영향을 주지 않음
- 각 AZ 마다 2 개의 데이터 복사본을 포함하고 있으며 최소 6 개를 가지고 있음
- 다른 계정과 Aurora 스냅샷을 공유할 수 있음
- Aurora Replica 와 MySQL Replica 2 가지 종류의 레플리카가 있음(자동 Failover 는 Aurora Replica 만 가능)

* ElastiCache
- 읽기 중심의 애플리케이션 워크로드 혹은 컴퓨팅 중심의 워크로드의 지연시간과 처리량을 비약적으로 향상시키는 인 메모리 캐싱 서비스(In Memory Caching Service)
- Amazon ElastiCache 는 Memcached 혹은 Redis Protocol 과 호환되는 서버 노드를 쉽게 배포할 수 있도록 지원
- DB 와 웹 애플리케이션의 성능 향상을 위해 사용
- 서버 리소스를 프로비저닝하는 것부터 소프트웨어를 설치하는 것까지 인 메모리 환경 설정과 관련된 작업 지원
- EC2 인스턴스처럼 노드별 유형이 나뉨(ex. m4.large, m5.large 등)

* Route 53
- Domain Name System(DNS)를 제공하는 서비스
- UDP 53 포트를 사용하기에 'Route 53'으로 명명됨
- 도메인 네임을 발급하고 레코드를 생성하는 등의 작업을 수행 가능
- AWS 이외의 기관에서 이미 도메인 네임을 보유한 경우에도 AWS 로 이전가능
- Alias Record 와 CNAME 의 차이
    - CNAME 과 달리 Alias 는 DNS 네임스페이스의 최상위 노드에 별칭 레코드 생성 가능

- CNAME 은 쿼리 이름이 일치하지 않아도 쿼리를 리디렉션
- Alias 는 쿼리 이름과 유형이 일치하는 경우와 별칭 레코드를 만든 호스팅 영역의 다른 레코드만 가리킬 수 있음
- Domain 등록시 최대 3 일까지 걸림

* Routing of Route 53
- Simple : 동일 레코드 내에 다수의 IP 를 지정하여 라우팅 가능, 값을 다수 지정한 경우 무작위로 반환함
- Weighted : Region 별 부하 분산 가능, 각 가중치를 가진 동일한 이름의 A 레코드를 만들어 IP 를 다르게 줌
- Latency-based : 지연시간이 가장 적은, 즉 응답시간이 가장 빠른 리전으로 쿼리를 요청함
- Failover : Active/Standby 설정에서 사용됨, Main 과 DR(Disaster Recovery)로 나누어 Main 장애시 DR 로 쿼리
- Geolocation : 각 지역을 기반으로 가장 가까운 리전으로 쿼리 수행, 레코드 생성시 지역을 지정할 수 있음
- Geoproximity : Traffic flow 를 이용한 사용자 정의 DNS 쿼리 생성 가능
- Multivalue Answer : 다수의 IP 를 지정한다는 것은 Simple 과 비슷하지만, Health check 가 가능함(실패시 자동 Failover)
- Health check : 각 A 레코드별로 Health check 를 두고 Health check 에 실패하면 Route 53 에서 제거됨, SNS 통지 가능

* VPC
- 가상 사설 네트워크 기능을 제공하는 서비스
- 서브넷, 라우팅 테이블, ACL 등의 기능을 제공
- 1 Subtnet 은 1 AZ 에 상응함
- 인스턴스의 가상 방화벽인 Security group 은 Stateful, Network ACL 은 Stateless 성격을 지님
  - Stateful : 인바운드 정책과 아웃바운드 정책 중 어느 한쪽만 허용되어도 통신 가능
  - Stateless : 인바운드 정책과 아웃바운드 정책 모두 허용되어야 통신 가능
- Region 당 5 개의 Elastic IP(공인 IP)를 사용할 수 있음
- VPC 하나당 하나의 IGW 만 가질 수 있음
- VPC Peering
  - VPC 와 VPC 를 연결하는 것
  - Region 내 VPC 를 연결하거나 Region 간 VPC 를 연결할 수 있음
  - 다른 계정간 VPC 연결 또한 가능

- Star 형 구조로 설정가능, 즉 하나의 중앙 VPC 에 4 개 VPC 를 연결하는 것
- VPC 를 생성할 때, 라우팅 테이블, ACL, Security group 은 자동으로 생성됨
- 아무리 AZ 가 같다하더라도 계정이 다르면 엄연히 다른 네트워크에 해당함

* NAT Instance
- Private Subnet 내의 리소스가 외부 인터넷 통신이 가능하도록 Source IP NAT 를 지원하는 인스턴스
- NAT 인스턴스를 생성할 때는 Source/Destination Check 설정을 Disable 해야 함
- NAT 인스턴스는 반드시 Public Subnet 에 존재해야 함
- Priavet Subnet 에서는 반드시 NAT 인스턴스로 라우팅을 잡아주어야 함
- 인스턴스 사이즈에 따라 성능이 달라지므로 병목현상이 발생하면 인스턴스사이즈를 변경해주어야 함
- 인스턴스가 다운될 경우 혹은 부하가 많을 때를 대비하여 Autoscaling 과 같은 기능을 같이 해주면 좋음

* NAT Gateway
- NAT 인스턴스와 동일한 역할을 수행하는 서비스
- EC2 인스턴스를 이용한 NAT 인스턴스와 달리 하나의 서비스로 존재함
- AZ 내에 시스템 상의 이중화가 되어있음(고객 입장에서는 보이지 않음)
- 5Gbps 에서 45Gbps 까지 확장 가능함
- NAT 인스턴스와 달리패치가 필요 없음
- Security group 의 영향을 받지 않음
- NAT 인스턴스와 마찬가지로 라우팅 테이블을 반드시 업데이트 해야함

* Network ACL
- 서브넷의 트래픽 정책을 제어하는 설정
- 우선순위를 적용할 수 있으며 값 100 단위로 정책을 넣는 것을 권고
- Stateless 이므로 Inbound 정책과 Outbound 정책이 모두 열려야 트래픽이 허용됨
- Custom ACL 은 기본 Deny 되어있음(Default ACL 은 기본 Allow)
- Ephemral Port 를 위해 별도로 열어주는 것이 좋음(NAT Gateway 의 경우, 1024 ~ 65535 포트를 사용함)
- 같은 포트에 대하여 다른 값의 규칙을 걸면 우선순위가 높은 규칙이 먼저 적용
- 하나의 ACL 정책을 여러 서브넷에 적용 가능하지만, 서브넷은 하나의 ACL 만 적용 가능

* Direct Connect
- 온프레미스와 AWS 를 연결해주는 네트워크 전용선 서비스
- Direct Connect Location(DX)을 통해 연결함
- AWS Region --- Direct Connect Location --- Customer

* VPC Endpoint
- NAT Gateway, Internet Gateway, VPN Connection 등의 서비스 없이 다른 AWS 서비스와의 연결을 가능케하는 서비스
- Interface Endpoints : 가상의 Network Interface(Private IP)를 생성하여 트래픽이 지나갈 End Point 를 제공
- Gateway Endpoints : VPC Gateway 를 이용하여 다른 AWS 서비스로 연결(오직 S3 와 Dynamo DB 만이 사용 가능)

* VPC Flow Logs
- VPC 내 Network Interface 에 지나다니는 IP Traffic 에 대한 정보를 캡처하는 서비스(Wireshark 같은)
- 다른 계정의 VPC 에 Peering 된 VPC 는 Flow Logs 활성화 불가능함
- Flow Logs 에 태그 불가능함
- 다음 트래픽은 기록되지 않음
    - 인스턴스가 생성될 때 DNS 와 연결되면서 생성되는 트래픽
    - Window License 인증을 위해 생성된 트래픽
    - 인스턴스 메타데이터를 위한 169.254.169.254 로의 트래픽
    - DHCP 트래픽

* Bastion Host
- 내부 인스턴스에 접속하기 위한 외부 인터넷이 연결된 서브넷의 인스턴스

* Elastic Load Balancer(ELB)
- 부하 분산 서비스, 로드밸런서, L4 스위치에 해당하는 서비스
- Application Load Balancer(ALB) : HTTP, HTTPS 에 특화된 Load Balancer
- Network Load balacner(NLB) : TCP 에 특화된 Load Balancer
- Classic Load balancer(CLB) : ALB, NLB 가 나오기 전의 Load Balancer
- HTTP, HTTPS 와 Sticky, X-Forwared-For 같은 특정 기능 사용 가능
- DNS A 레코드가 자동으로 할당되기 때문에 IP 를 할당할 필요 없음
- Sticky Session : 특정 EC2 에 세션이 고정되도록 해주는 기능
- Cross Zone Load Balancing : 가용영역(AZ) Load Balancing 이 아닌 인스턴스 개체 수를 기준으로 Load Balancing 함
- 가용영역별(AZ)로 한 이후 ELB 가 자신의 가용영역이 아닌 다른 가용영역의 Pool member 에 Load Balancing 함

* SQS(Simple Queue Service)
- 메시지 큐 서비스(Push 가 아닌 Pull 기반 서비스)
- 서비스 요청을 저장하고 대기열을 만들어 처리할 수 있도록 하는 서비스

- 각 컴포넌트들을 분할하여 독립적으로 운영하게 함으로써 한 서비스가 실패할 경우에도 서비스 요청을 보존할 수 있게 함
- 열람중에 서비스가 처리되면 사라지지만 처리되지 않으면 다른 열람자가 읽을 수 있도록 보존됨
- 각 메시지는 최대 256KB 의 텍스트로 구성될 수 있음
- 1 분 ~ 14 일간 저장가능하지만 Default 값은 4 일임
- Standard Queues : 표준 서비스로 초당 무제한에 가까운 요청을 처리할 수 있으며 최소 한 번 처리를 보장하나 순서는 보장하지 않음. 그러나 중복으로 처리될 수 있음
- FIFO Queues : 선입선출, 순서를 정확히 지켜 처리함, 초당 300 개로 제한됨

* SNS(Simple Notification Service)
- Notification 를 통보하는 서비스로 애플, 구글, 파이어폭스, 윈도우 디바이스 등에 알림을 보낼 수 있음
- SQS 와는 다르게 Push Base 서비스

* Elastic Transcorder
- 미디어 변환 서비스
- S3 Bucket - > Lambda Function - > Elastic Transcoder - > S3 Bucket

* API Gateway
- 개발자로 하여금 API 를 배포, 유지, 관리하는 것을 도와주는 서비스
- API Gateway 와 Lambda 를 연동하여 사용하기 용이함
- EC2, ECS, Elastic Beanstalk 등의 서비스에 액세스할 수 있도록 하는 "현관문" 역할을 함
- RESTful API, HTTP API, REST API, WebSocket API 등의 생성 옵션이 있음

* Cognito
- 계정 동기화 및 인증 서비스, 모바일을 위한 인증 서비스 제공
- 앱 로그인, 유저 권한 제공, 모바일 디바이스의 데이터 동기화

* Lambda
- Serverless 서비스로 서버없이 코드를 대신 실행하는 서비스
- Scaling 이 자동적으로 이루어짐
- Lambda Function 은 독립적이므로, 1 Event 는 곧 1 Function 에 해당함
- Lambda Function는 다른 Lambda Function을 트리거할 수 있음