* AWS sysops administrator exam
 - 총 7 개의 도메인
    - Monitoring & Reporting
    - High Availability
    - Deployment & Provisioning
    - Storage & Data Management
    - Security & Compliance
    - Networking
    - Automation & Optimization
 - White papers 와 백서가 중요
 - 130 분 진행
 - 65 개의 문제
 - 1000 점 만점 중 720 점 획득
 - 150$
 - 2 년 이상 서비스를 사용해본 사람에게 유리

 1. Monitoring and Reporting
 * CloudWatch
 - Compute, Storage & Content Delivery monitoring
 - SNS, SQS, Opsworks SQS Monitoring
 - CPU, Network, Disk, Stats check
 - 메모리 사용량은 커스텀 메트릭이며 EC2 모니터링의 기본 수집시간은 5 분(1 분까지 단축 가능)
 - 원하는 기간만큼 저장가능하며 기본적으로 반영구적으로 보관
 - 많은 서비스에서 1 분으로 지정되어있음(기본적으로) why??
 - EC2 나 ELB 가 종료된 이후에도 데이터를 검색할 수 있으며 Log 는 영구적으로 보관됨(기본적으로)
 - custom metric Granularity 는 세부적으로 1 분 모니터링이며 표준은 5 분임
 - 온프레미스 서비스에도 사용가능

 * Monitoring EC2 with Custom metric
 - CPU, Network, Disk, Status Check
 - Ram Utilization : Custom metric
 - Custom Metrics : minimum granularity 는 1 분

 * Monitoring & Modifying EBS
 - IOPS 는 초당 입력/출력작업을 나타내는 측정단위로 작업은 KiB 단위로 측정됨
    - 최대 3000IOPS 까지 가능하며 그 이상이 필요하면 PIOPS 를 사용해야 함
 - gp2, io1, st1, sc1
 - gp2 : 짧은 지연시간과 대부분의 워크로드에서 사용, 시스템 볼륨
    - 기본적으로 3IOPS per GiB, 예를 들어 100GiB 볼륨이 있다면 300 IOPS 임
 - io1 : IOPS 퍼포먼스가 중요한 경우 사용(데이터베이스 사용시 유용)

- st1 : 빅데이터 및 로그 처리, IOPS 보단 처리량에 우선된 작업에 유용, 시스템 볼륨 불가
- sc1 : 접근 빈도가 적은 데이터웨어하우스용 EBS, 시스템 볼륨 불가
- Pre-warming EBS Volumes : https://docs.aws.amazon.com/ko_kr/AWSEC2/latest/UserGuide/ebs-initialize.html
- 볼륨 모니터링 지표
    - BurstBalance : 볼륨에 대한 버스트 버킷 잔고를 남은 잔고로 비유하였으며, 버스트 버킷을 모두 사용하면 스로틀링이 시작됨
    - volume read byte : 각 읽기 작업의 평균 크기 보고
    - volume write byte : 각 쓰기 작업의 평균 크기 보고
    - volume read ops : 지정된 기간의 총 읽기 작업 수
    - volume write ops : 지정된 기간의 총 쓰기 작업 수
    - volume total read time : 모든 읽기 작업에서 사용한 총 시간, 5 분동안 700 개의 작업이 완료되고 작업당 1 초가 걸린 경우 값은 700 초
    - volume total write time : 모든 쓰기 작업에서 사용한 총 시간, 5 분동안 700 개의 작업이 완료되고 작업당 1 초가 걸린 경우 값은 700 초
    - volume queue length : 처리되기 위해 기다리고 있는 읽기 쓰기 작업의 수
    - VolumeThroughputPercentage : EBS 에 할당된 총 IOPS 작업에서 전달한 I/O 작업의 비율을 결정하는데 사용하는 메트릭
- 볼륨 상태
    - ok : enabled, normal
    - warning : enabled, degraded or severely degraded
    - impaired(It's not working) : enabled/disabled(volume is offline and pending recovery or waiting for the user to enable I/O), Stalled(impacted) or Not available
    - insfficient-data : Enabled/Insufficient Data, Insufficient Data
- EC2 에 연결된 상태에서 사이즈를 늘리거나 타입을 바꾸고 IOPS 퍼포먼스를 조정할 수 있음(떼지 않아도)

* Monitoring ELB
- CloudWatch metric
- Access log : ELB 로 전달된 요청에 관한 자세하 정보를 캡쳐하여 로그로 남김, IP, 지연시간, 요청 경로, 서버 응답 등을 포함하며 이를 S3 버킷에 저장함
    - 매우 중요함
    - 서버가 삭제되면서 사라지는 로그를 저장할 수 있으므로, Autoscaling 사용시 발생하는 이벤트에 대한 로그 저장 가능
- Request tracing : HTTP 요청을 추적하는 기능
    - ALB 에서만 사용 가능

- CloudTrail logs : ELB API 에 호출된 요청에 대한 자세한 정보를 제공하고 로그 파일을 S3 에 저장

    - source IP 가 무엇인지 어디서 요청되었는지 누가 요청을 만들었는지 언제 요청이 만들어졌는지 확인 가능

- CloudWatch vs CloudTrail

    - Cloudwatch 는 퍼포먼스를 모니터링

    - CloudTrail 은 API Call 을 모니터링

* Monitoring ElastiCache

- https://docs.aws.amazon.com/ko_kr/AmazonElastiCache/latest/red-ug/CacheMetrics.html

- CPU 사용률

    - Memcached : 멀티 쓰레드, 90%까지 요청을 핸들하며 90%가 넘으면 노드를 추가함

    - Redis : 멀티 쓰레드가 아니며, 사용률을 노드 수에 나누어 측정함. 예를 들어 코어가 4 개인 경우 90%이면 각 22.5%가 됨

- Swap Usage : 램 사용률이 높을 때 디스크에 공간을 할당하여 사용하는 비율, 보통 램의 사이즈와 동일함

    - Memacached : 50Mb, 50Mb 이상이 될 경우 Memcached connection overhead 파라미터를 증가시켜야함

    - Redis : SwapUsage 메트릭이 없음, 예약 메모리를 대신 사용

- Evicition : 시스템 내 빈 공간에서 제거해야 할 아이템을 제거하고 새로운 아이템을 넣을 때 발생

    - Memcached : 추천되는 세팅은 없으며 스케일 업(메모리 증가) 혹은 스케일 아웃(노드 증가) 가능

    - Redis : 추천되는 세팅은 없으며, 스케일 아웃(노드 증가) 가능

- Concurrent Connection

    - Memcached & Redis : 추천되는 세팅이 없으며 커넥션 수에 스파이크가 발생한다는 것은 스파이크가 정말로 발생한 것이거나 app 이 커넥션을 열지 못하는 것

* CloudWatch Custom Dashboards

- 각 리전의 모니터링 지표를 대시보드에 표현 가능 위젯을 추가하기 위해서는 리전을 바꾸어서 위젯을 추가해야함

- 저장버튼을 눌러야 함

* AWS Organizations

- 다수의 계정을 한 번에 관리하게 해주는 서비스로 계정의 그룹을 만들고 정책 부여 가능

- AWS 서비스에 대한 접근 제어, 계정 생성 자동화 및 관리

- 접근 제어 : SCP 를 통해 가능, IAM 가 있더라고 SCP 로 덮어쓰기 가능
- 자동화 : Organization API 를 통해 계정 생성을 자동화 가능
- 결제 통합 : 각 계정간의 요금을 하나로 통합하여 관리 가능

* Tagging & Resource Groups
- Key / Value 값으로 리소스에 부여됨
- 메타데이터
- 태그 번호는 Autoscaling, Cloudformation, Elastic Beanstalk 에 상속가능
- Resource Group
    - 태그를 사용하여 태그가 연결되어있는 리소스를 그룹화하고 하나 또는 그이상의 태그를 공유하는 리소스를 그룹화할 수 있음
    - 지역, 이름, 헬스 체크 등을 포함할 수 있음

* EC2 Pricing - Refresher
- On demand
    - 방해받지 않으면서도 짧은 기간이나 예측할 수 없는 워크로드 적합
    - 테스트 및 개발용도로 첫 시도에 유용
    - 1 시간 혹은 1 분 단위의 요금 가능
- Reseved
    - 지속적이고 예측 가능한 사용률
    - 예측 가능한 사용량에 유용
    - Standard RI 는 온디맨드에 비해 75% 저렴
    - Convertible RI 은 온디맨트에 비해 54% 저렴
- Spot
    - 시작과 종료시간이 유동적인 앱에 유용
    - 매우 저렴함
    - 추가적인 대규모 용량을 긴급하게 사용할 시 유용
- Dedicated host
    - 고객 전용 EC2 인스턴스 용량을 갖춘 물리서버
    - 소프트웨어 라이센스 사용 가능
    - 온디맨드 사용가능

* AWS Config
- 리소스 인벤토리, 설정 내역, 설정 변경 알림 등을 제공
- 보안 감사, 분석, 추적 가능
- 스냅샷 생성 및 컨피그 변경을 로그로 생성
- 준수사항 체크 자동화 가능
- 서비스 config 변경 - > Event(Config) - > S3, Lambda function
- 리소스, 스냅샷, 스트림 생성 가능

- 각 계정의 로그 설정 후 S3 에 저장 가능, SNS 로 알림 가능
- Resource Type, IP, Compliance, Timeline 보기 가능
- 리전 기반으로 각 리전마다 설정해야 함
- 타임라인을 통해 언제 생성되어 언제 변경되었는지 확인 가능
- 트리거
    - 주기적 트리거
    - 설정 변경시 트리거
- Config 는 IAM Role 을 필요로 함
    - 기록된 리소스 읽기 권한
    - S3 쓰기 권한
    - SNS 접근 권한
- 소수의 관리자만이 Config Full Acces 권한을 갖는 것이 좋음
- FAQ 참조!!

* CloudWatch vs CloudTrail vs Config
- CloudWatch 는 퍼포먼스를 모니터링(CPU, Network, Disk 등)
- CoudTrail 은 API Call 을 모니터링
- Config 는 AWS 환경의 상태를 기록하고 변화를 통지함

2. Deployment & Provisioing
* EC2 Launch Issues
- InstanceLimitExceeded error : 리전 내에서 사용가능한 숫자를 넘어섬, 리전당 20 개 가능
- InsufficientInstanceCapacity error : 인스턴스를 가동할만큼 하드웨어가 충분하지 않을 경우 발생하는 에러
    - 몇 분 후에 다시 시도
    - 좀 더 적은 수의 인스턴스를 요청
    - 다른 타입의 인스턴스를 요청
    - 예약 인스턴스 구매 시도
    - 구체적인 AZ 를 정하지 않고 구매 요청서를 제출

* EBS Volumes And IOPS
- EBS : 파일 시스템, 데이터베이스, OS 를 운영하는데 사용
- IOPS 는 전적으로 볼륨의 사이즈에 의존함(IOPS hitting 에 한계가 발생하면 볼륨사이즈를 늘려야 함)
- SSD
    - gp2 : 시스템 볼륨에 사용되는 일반적인 버전
    - io1 : IOPS 에 프로비져닝된 버전, NoSQ / RDS / 낮은 지연속도용

- IOPS 는 SSD 볼륨의 성능을 측정하는데 사용
    - gp2 볼륨은 3 IOPS/GB 이며 100 ~ 최대 16000 IOPS 까지 가능
    - io1 볼륨은 50 IOPS/GB 이며 최대 64000 IOPS 까지 가능
- gp2 의 IOPS 한계를 뛰어넘는 워크로드가 발생한다면 어떻게 되는가
    - 리퀘스트 큐를 사용하고, 앱에 따라서 느려지는 현상을 볼 수 있음
    - 볼륨 사이즈를 증가시킬 수 있음, 그러나 5.2TB 에 이미 다다랐다면 이미 최대치에 도달한 것
    - 16,000 IOPS 이상 필요하다면 스토리지 클래스를 io1(Provisioned IOPS)로 바꾸어야 함

* Elastic Load Balancer
- NLB 은 다른 로드밸런서에 비해 낮은 지연시간을 보여주어 성능에 극대화되어있음
- CLB 는 HTTP/HTTPS LB 및 X-Forwared-for 와 sticky session 을 위해 사용됨
    - 제한된 L4 기능을 사용할 수 있음(TCP)
- Pre-warming 기능을 사용해 미리 확장 가능
    - 시작과 종료 날짜, 초당 요청 예측, 전형적인 요청의 사이즈 필요
- ALB 는 워크로드에 대응하여 자동으로 규모를 확장하지만 source-IP ALB IP 로 변경됨
- NLB 는 서브넷별로 static IP 를 부여받으며 IP 를 고정시킬 수 있음

* ELB Error Messages
- ALB, CLB 는 기본적으로 성공 응답이 200 OK 임
- 실패한 request 는 4xx, 5xx error message 가 돌아옴
- 4xx message 는 클라이언트 사이드의 문제를 암시하며 5xx message 는 서버 사이드의 에러를 의미
- 400 error : 헤더가 malformed 인 경우(사이즈가 비정상적으로 큰 경우 등)
- 401 error : User access denied
- 403 Forbidden : 요청이 ACL 에 의해 차단됨
- 460 error : ELB 가 응답을 돌려주기도 전에 클라이언트가 커넥션을 닫음
- 463 error : ELB 가 x-forwarded-for 에 30 개 이상의 IP address 를 받음(비정상적 요청)
- 500 error : 서버 error(ELB 내 error)
- 502 Bad gateway : 서버가 비정상적 응답을 보내거나 커넥션을 닫음
- 503 Service Unavailable : 대상그룹에 등록된 대상이 없음(대상그룹 자체가 없음)
- 504 Gateway timeout : 서버 혹은 DB 문제로 응답하지 않음
- 561 Unauthorized : 사용자 인증시 ID Provider 로부터 에러 코드를 받음

* ELB CloudWatch Metrics

- 메트릭 항목은 두 종류로 나뉨
    - metric for general health : HealthyHostcount, HTTPCode_Backedn_2xx
    - metric for performance : Latency, ReuqestCount, SurgeQueueLength, SpilloverCount
- ELB 자체와 백엔드의 서버를 모니터링하며 30 초의 수집간격을 가짐(HealthCheckIntervalSeconds)
- BackendConnectionError : 백엔드 인스턴스로의 실패한 커넥션 갯수
- HealthyHostCount : 정상으로 등록된 인스턴스 갯수
- 2xx,3xx,4xx,5xx Code : 해당 에러 갯수
- Latency : 지연시간 표시
- RequestCount : 1 분 혹은 5 분간 성공/완료된 커넥션 갯수
- SurgeQueueLength : ELB 에서 인스턴스로 전달되지 못하고 쌓여있는 Request 의 수(Classic only)
- SpilloverCount : Surgequeue 가 쌓여서 거부된 요청의 갯수(Classic only)

* Systems manager(SSM)
- AWS 인프라를 제어하고 작업을 자동화
- Cloudwatch 와 통합하여 문제를 감지하거나 운영 데이터를 볼 수 있게 해줌
- Run command
    - 하나 이상의 EC2 에 사용자정의 커맨드를 실행할 수 있게 해줌
    - 인스턴스에 관한 조작이나, 볼륨 부착/제거, 패치 실행 등
- 인벤토리와 리소스 그룹을 이용하여 실행 가능
- Run Command 를 통해 관리중인 리소스에 명령을 실행할 수 있음

3. Elastic and Scalability
- Elasticiy : 탄력성, 수요에 따라 필요할 땐 늘어나고 필요없을 땐 줄어드는 특성
- Scalability vs Elastcity
    - EC2
        - Scalability : 예약 인스턴스를 이용하여 인스턴스 사이즈를 증가시키는 것
        - Elasticity : Autoscaling 에 기반하여 EC2 의 숫자를 늘리는 것
    - DynamoDB
        - Scalability : 무제한의 스토리지 양
        - Elasticity : 트래픽 폭증시 IOPS 를 추가 증설하는 것이고, 감소 후에 그 수를 줄이는 것
    - RDS
        - Scalability : 인스턴스 사이즈를 증가시키는 것, small - > medium
        - Elasticity : 불가능, 수요에 따라 스케일 in/out 불가
    - Aurora

        - Sc : 인스턴스 타입 변경
        - El : Aurora Serverless
    - Elasticity : 단기적 관점에서 수요에 따라 확장/감소(Scale)
    - Scalability : 장기적인 관점에서 인프라 증설(Scale out)

* RDS Multi-AZ Failover
- Multi-AZ 는 장애시 대응을 위한 것이기 때문에 성능과는 무관함
- RDS 의 경우, MySQL, Oracle, PostgreSQL 엔진이 동기화된 복제본을 갖고 있음
- MS SQL Server 의 경우 미러링을 위해 동기화된 logical replication 을 사용함
- 백업과 복구 모두 Standby 에서 진행하여 성능 저하를 피함
- RebootDBInstance API Call 을 통해 강제로 Failover 할 수 있음

* RDS & Using Read Replica
- 읽기 성능을 비약적으로 향상시키기 위한 기능(Scale out)
- 엔진의 자체적이고 비동기적인 복제본을 통해 생성되고 업데이트됨
- 소스 DB 가 패치/백업 등의 작업으로 인해 서비스 제공이 불가능할 때 R/R 가 서비스 제공
- MySQL, PostgreSQL, MariaDB, Aurora(모든 엔진에 대하여 다른 리전에 R/R 생성 가능)
- Read replica 생성시 스냅샷이 생성되는데 Multi-AZ enable/disable 에 따라 Secondary/Primary DB 가 스냅샷 생성을 수행
    - disable 일 경우, 약 1 분간 I/O 지연현상 발생
- R/R 는 StandAlone 으로 승격 가능
- MySQL, PostgresSQl, MariaDB 는 5 개까지 보유 가능
- 다른 Region 에도 두는 것이 가능
- Snapshot 과 Backup 은 R/R 에서 가져올 수 없음
- automated backups 이 활성화되어있지 않으면 R/R 생성 불가

* ElastiCache
- DB 의 느린 디스크 쿼리를 대신하여 보다 빠른 검색을 위한 웹 서비스
- Social networking, gaming, media sharing, QA portal 등에 낮은 지연시간과 무거운 워크로드에 사용
- Memcached / Redis 는 key-value 기반 캐쉬라 list 나 정렬 같은 데이터구조 제공에 효율적
- Redis 는 Multi-AZ 와 Read Replica 를 지원
- CPU Utilization, Swap Usage, Evictions, Concurrent Connections
- 자주 변경되지 않으면서 무거운 작업에 적합
- 지속적인 OLAP 트랜젝션으로 인한 지연일 경우 Redhsift 가 최적

* Aurora
- 각 AZ 마다 2 개의 데이터 복사본을 두어 6 개를 유지(각 AZ 마다 최대 3 개까지 증가시킬 수 있음)
- 10GB 에서 시작해 최대 64TB 까지 증가(Storage Autoscaling)
- 64vCPUs, 488GiB of memory
- Data block 과 디스크를 끊임없이 점검해 손상될 경우 자가 치유 실시
- Master 가 Cluster volume 에 있는 각 복사본에 쓰기 작업을 실시하여 데이터를 공유
- R/R 는 최대 15 개까지 보유 가능
- Aurora Replica 는 Aroroa Replica 와 MySQL Read Replica 두 가지 종류가 있음
- 쓰기 작업이 이슈라면 Scale up, 읽기 작업이 이슈라면 Sclae out 을 해야함
- Aurora Serverless : Aurora Autoscaling 구성
- Cross region Read Replica 생성 가능(Multi-AZ 구성 추천)
- 암호화는 default 옵션이며 Failover 는 티어가 낮을수록 우선순위가 됨
- 한 번 암호화가 발동되면 모든 R/R 도 암호화됨

* Troubleshooting Autoscaling
- 키 페어가 없을 때
- 보안 그룹이 없을 때
- 오토스케일링 컨피그가 올바르게 작통하지 않을 때
- 오토스케일링 그룹이 더이상 없을 때
- 인스턴스 타입이 AZ 에서 지원하지 않을 때
- AZ 가 더이상 없을 때
- 유효하지 않은 EBS 가 맵핑될 때
- 계정에서 오토스케일링 서비스가 허용되지 않을 때
- 인스턴스 스토어 AMI 에 EBS 를 붙이려고 할 때

4. Storage & Data Management
* S3
- 오브젝트 기반의 스토리지
- 0 Bytes ~ 5TB 저장 가능(오브젝트 당)
- 무제한의 스토리지
- 업로드 성공시 200 OK 코드를 돌려받음
- Read after write consistency for PUTS of new Objects(first time)
- Eventual Consistency for overwrite PUTS and DELETES (전파하는데 시간이 소요됨)
- Key, Vaoue, Version ID, Metadata, Bucket Policy, ACL, CORS 등의 구성요소가 있음

- RRS 는 One-Zone IA 와는 다르게 내구성이 99.99%이며 쉽게 복제될 수 있는 데이터를 저장함(가용영역이 3 개 이상임)
- 요금은 GB 당 저장요금, REQUEST(Get, Put, Copy), Inventory, 분석, Object 태그, S3 에서 반출시 발생(Cloudfront)
- FAQ 를 꼼꼼히 확인해야 함

* S3 Life Cycle
- S3 Standard 에서 IA 혹은 One-Zone IA 로 옮길 경우 최소 30 일이 경과해야함
- 예시
    - 90 일 이후에 IA 로 이전
    - 1 년 후에 Glacier 로 이전
    - 1 년 후에 만료시킴, S3 는 자동적으로 만료된 오브젝트를 삭제
- Access log 가 활성화되어있으면 로그파일을 축적할 수 있음
- 수명주기는 오브젝트가 생성한 날짜를 기준으로 함

* MFA Delete
- Versioning 은 오래된 오브젝트의 관리를 가능하게 함
- 즉 우발적인 삭제로부터 보호함
- 즉 버전관리가 활성화 된 상태에서는 '삭제' 액션은 삭제를 의미하는 것이 아닌 Delete marker 를 적용하는 것을 의미
- 영구삭제를 위해서는 Version ID 를 삭제 요청에 포함시켜야 함
- MFA delete 사용을 위해서는
    - MFA device 를 통해 받은 유효한 코드를 사용해야함
    - Versioning 이 활성화되어있어야 함

* S3 encryption
- 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. 데이터 암호화 알고리즘은 AES-256 을 사용함
    - 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 암호화를 의미함
- KMS 에 저장된 고객 마스터키를 사용하여 암호화 혹은 애플리케이션 내 마스터 키를 사용하여 암호화
- 암호화시에 HTTP Header 에 x-ams-server-side-encryption: AES256 적용
- 암호화를 강제하고 싶으면 헤더에 위의 파라미터를 포함하지 않는 요청은 거부하는 버킷 정책을 만들면 됨(Client-side)

* EC2 Type - EBS vs Instance Store
- EC2 가 최초 런칭되었을 때는 Instance Store 를 디스크로 사용하였으나 차후 EBS 가 나옴
- Root Device Volume 은 EBS, Instance Store 모두 사용 가능
- Instance Store Root device 의 최대 사이즈는 10GB, EBS Root device 는 최대 1~2TB 까지 가능(OS 에 따라)
- Root Device Volume 은 인스턴스 삭제시 같이 삭제되며 다른 EBS 볼륨은 그대로 남음
- Instance Store 의 Root Device Volume 은 인스턴스 종료시 자동으로 종료되며 막을 수 없고 다른 Instance Store 볼륨도 같이 종료됨
- Root Device Volume 에서 스냅샷을 생성하기 위해 반드시 인스턴스를 멈추게 실행해야 함
- Image 와 Snapshot 모두에서 AMI 생성 가능
- 볼륨은 항상 EC2 Instance 와 같은 AZ 에 있음
- 스냅샷 공유는 암호화되어있지 않을 때만 가능

* Encryption and Downtime
- AWS Resource 는 생성시에만 암호화 설정이 가능
- EFS 의 경우, 암호화를 하고 싶다면 새로운 EFS 를 생성한 후 데이터를 마이그레이션해야 함
- RDS 의 경우, 암호화를 하고 싶다면 새로운 RDS 를 생성한 후 데이터를 마이그레이션해야 함
- S3 의 경우, 이미 생성된 버킷이라도 아무때나 암호화 설정이 가능하나 이미 저장된 객체에는 해당되지 않음

* CloudHSM vs KMS
- 두 기능 모두 데이터를 암호화하는 키를 관리하고 저장하는 솔루션
- KMS
  - 멀티 테넌시가 이슈가 되지 않는 환경에 적합, 하드웨어를 공유하는데 적합한 키관리 시스템
  - Free tier 사용 가능

    - symmetric 환경만을 지원
- CloudHSM
    - 전용 하드웨어 보안 모듈이며, 다른 테넌트들과 하드웨어를 공유하지 않음
    - VPC 내에서 독점적인 제어를 받음
- asymmetric 의 뜻은 암호화/복호화에 쓰인 키가 다르다는 의미

* AMI
- 루트 볼륨 템플릿 (OS, App)
- 인스턴스를 시작할 수 있는 계정 정의
- 시작시 EBS 볼륨로 연결할 디바이스 매빙
- 다른 리전에서 AMI 를 가지고 시작하려면 AMI 을 복사하여 해당 리전에 옮겨두어야 함

* Shareing AMI
- 타 계정을 구체적으로 지정하여 제한된 계정에 공유 가능
- 공유된 계정은 소유권과 제어권을 가짐
- AMI 소유자는 반드시 볼륨 생성 권한을 추가해주어야 함
- 암호화된 AMI 를 다른 계정과 바로 공유할 수는 없음
    - 스냅샷을 복사하여 키를 이용해 재암호화한 후 새로운 AMI 를 만들고 AMI 과 암호화 키를 함께 공유해야 함

* Snowball & Edge
- 대용량의 데이터를 쉽고 안전하게 옮길 수 있도록 해주는 물리적 장비
- 256-bit 암호화 지원 by default
- 로컬 네트워크에 연결하여 스노우볼이 자동적으로 데이터를 암호화하고 옮김
- 테라/페타 바이트 이상의 데이터를 옮기되, 인터넷 네트워크를 쓰고 싶지 않은 경우
- 물리적으로 격리된 환경이나 고대역폭의 인터넷 사용이 불가능 한 경우
- Edge 는 100TB 장비로 추가적인 용량과 엣지 컴퓨팅 제공
    - AWS 로 데이터를 옮기기 전에 로컬 프로세싱이 필요한 경우 사용

* Storage Gateway
- AWS 스토리지와 그곳으로 연결되는 온프레미스 소프트웨어로 이루어짐
- File gateway
    - S3 에 파일을 저장하고 NFS, SMB 제공
    - S3 에 마운트하는 파일시스템과 비슷
    - S3 의 모든 기능 사용 가능하며 온프레미스 스토리지에 비해 가격이 저렴
- Volume Gateway
    - iSCSI 프로토콜을 사용하는 스토리지
    - Stored Volume 과 Cached Volume 으로 나뉨

- 온프레미스에 저장 후 EBS 스냅샷의 형태로 AWS 에 백업하는 것, 그리하여 지연시간이 낮음

- 온프레미스에 두고 S3 에 데이터를 저장하는 것으로 나뉨

- Tape Gateway
  - Virtual Tape Library
  - S3 에 저장되며 Glacier 에 백업가능하고, VTL 을 이용하여 액세스됨

* Athena
- 표준 SQL 을 사용하여 S3 에 저장된 데이터를 쿼리하고 분석가능케 하는 서비스
- 쿼리된 TB 데이터당 요금 부과
- ETL 프로세스가 필요 없음
- 비지니스 리포트를 생성하며 클릭 스트림 데이터에 대해 쿼리 실행 가능
- Serverless

5. Compliance
* DDOS
- 분산 서비스 거부 공격, DoS 공격과는 달리 다수의 메커니즘에 의해 실행됨
- Amplification/Reflection 공격 : 공격자가 다수의 NTP, SSDP, DNS, SNMP 서버에 요청을 보내 트래픽을 증폭시킨 후 공격대상에 이를 보내는 공격법(Source IP Spoofing)
- Application 공격 : GET Flooding 공격, 다수의 커넥션을 생성한 뒤 최대한 오래 유지하여 지연시킴
- AWS Shield : ELB, Cloudfront, Route 53, WAF, Autoscaling, Cloudwatch 등을 DDoS 공격으로부터 방어하는 서비스
  - Standard 와 Advanced 로 나뉨
  - Standard 는 추가 비용없이 사용가능하며 굳이 활성화하지 않아도 이미 활성화되어 있음
  - SYN/UDP Floods, Reflection attacks, 각종 Layer 3/4 공격을 막아냄
  - AWS Shield Advanced : Shield 의 강화 버전으로 EC2, ELB, Cloudfront, Route 53 등에 추가적인 보호를 제공하지만 비용을 추가로 내야 함
- 보안 백서를 읽어보는 것이 중요

* STS
- AWS Resource 에 접근할 임시 권한을 유저에게 부여하는 서비스
- IAM 사용자가 사용할 수 있는 장기 액세스 키 자격 증명과 거의 동일한 효력을 지님
- 임시 자격 증명이라는 이름답게 단기적으로 자격을 증명하는 서비스
- 장점
  - 사용자에 대한 AWS 자격 증명을 정의하지 않아도, 특정 리소스에 대한 액세스 권한을 사용자에게 제공할 수 있음

- 더 이상 필요하지 않을 때 교체하거나 직접 취소시킬 필요 없음

- STS Scenario 예시

1. username 과 password 를 입력

2. 앱이 Identity Broker 를 호출하면 Broker 가 username 과 password 를 캡쳐

3. Broker 가 LDAP 디렉토리를 이용하여 사용자의 자격을 유효화함

4. Broker 가 다시 IAM Credentail 을 이용하여 새로운 GetFederationToken 기능을 호출

(이 호출에는 반드시 IAM Policy 와 유효 시간, 임시 자격을 부여할 권한이 담긴 정책을 포함해야함)

5. STS 가 앱에게 4 가지 값(Access Key, Secret access key, token, duration)을 반환하고 새 토큰을 발행하기 위해 IAM 정책을 확인함

6. Broker 가 임시 보안 증명을 앱에 반환

7. 앱이 S3 에 요청을 보내기 위해 임시 보안 증명 사용

8. S3 는 IAM 을 이용하여 증명을 확인하고 S3 버킷에 요청을 보내는 것을 허용

* Logging in AWS

- AWS 의 로깅 서비스
- CloudTrail, Config, CloudWatch Logs, VPC Flow Logs 등이 존재함
- CloudTrail 와 Config 의 차이점에 대해 알아보는 것이 좋음
- CloudTrail 은 '누가' 해당 서비스를 접근하였고, 사용하였고, 변경했는가를 중점적으로 봄
- Config 는 '어떤' 서비스가 접근되었고, 사용되었고, 변경되었는가를 중점적으로 봄

* Hypervisors

- VM 을 실행하는 컴퓨터 소프트웨어
- 하나 이상의 VM 을 실행하는 컴퓨터를 Host machine 라 하며 실행되는 VM 을 Guest machine 이라 함
- EC2 는 Xen Hypervisor 상에서 실행되며 PV,HVM 으로 나뉨
- 하이퍼바이저는 관리자만 접근 가능, AWS 직원은 EC2 에 접근할 권한이 없음

* 전용 인스턴스 vs 전용 호스트

- 전용 인스턴스 : 한 고객에게 종속되는 하드웨어 위에서 실행되는 VPC 내 인스턴스를 말함
  - 다른 계정들과 다른 하드웨어를 사용함
- 공통점 : 전용 하드웨어를 사용하여 제공
- 가장 큰 차이점

- 전용호스트는 추가적인 가시성과 인스턴스들이 물리 서버에 어떻게 배치되는지에 대한 권한을 줌(소켓, 코어, 호스트 ID)
- 전용 호스트는 고객이 보유한 소프트웨어 라이센스를 사용할 수 있게 해줌
- 전용 인스턴스는 다른 AWS 인스턴스(서비스)들과 같은 하드웨어를 사용함

* System Manager EC2 Run Command
- 태그를 이용하여 한 그룹의 서비스에 명령을 내리는 서비스
- 온프레미스의 서비스에도 가능함
- 시스템 매니저 문서에 명령어와 파라미터가 정의되어있음
- 에이전트가 설치되어 있어야 함

* AWS Config with S3
- s3 bucket public write prohibited : 전역 쓰기 접근을 허용하는 버킷을 자동으로 식별
- s3 bucket public read prohibited : 전역 읽기 접근을 허용하는 버킷을 자동으로 식별

* Inspector vs Trusted Advisor
- Inspector : 앱의 보안 정도를 자동으로 평가해주는 서비스
  - 타겟 생성 - > 에이전트 설치 - > 템플릿 생성 - > 실행
- Trusted Advisor
  - 비용, 보안, 가용성, 성능을 확인

* Shared responsibility model
- AWS 는 클라우드의 보안을 책임지지만, 고객은 클라우드 내 보안을 책임짐
- 고객은 콘텐츠와 플랫폼, 앱, 시스템, 네트워크를 보호할 수단을 고를 권한을 가지고 있음
- AWS 의 책임 범위
  - 인프라스트럭쳐
  - 하드웨어, 소프트웨어, 네트워킹, 시설
  - RDS 와 같은 관리되는 서비스(RDS OS 업그레이드, Database 업데이트, PHP 업데이트)
- 고객 책임 범위
  - 업데이트나 보안 패치 같은 것(EC2 OS 패치, 안티바이러스, 보안 그룹 설정)
  - 방화벽 설정

* Security Summary
- DDoS 가능 서비스 : Cloudfront, Route 53, ELB, WAF, Autoscaling, Cloudwatch

- Shield : 기본활성화, 추가옵션은 3 천달러
- IAM : Json 을 통해 커스터마이징 가능하며 연결시 즉시 활성화
- MFA : 루트 계정과 사용자 계정 모두 가능 적용, STS 를 통해 활성화

7. Networking & Route 53

* DNS(Domain Name System)

- SOA(Start Of Authority) : Zone Data 를 제공하는 네임 서버로 독자적인 관리 정책으로 Zone 을 관리하는 관리자
    - 예를 들어 인터넷 전체 Domain 중 aws.com 이름 아래의 Domain Zone 은 .com Domain 영역에 속하긴 하나 독자적인 관리 정책으로 aws.com 이 관리되고 있음을 의미
- NS Record : 탑 레벨 도메인 서버로서 사용되는 네임서버 레코드(DNS 서버가 참조하는 다른 DNS 서버)
    - 자신의 DNS 서버에서 쿼리를 요청받은 domain 에 대한 정보를 알아내지 못할 때, 이 NS Record 에 정의된 서버로 가서 주소를 얻어옴
- A Record : IP 로 번역되는 DNS 레코드로 IP 로 일대일 대응되는 Domain Name 을 뜻함
- TTL : Resolving server 에 캐시되는 시간을 뜻하며 짧을 수록 DNS 레코드가 업데이트되는 속도가 빠름
- CName : 하나의 도메인에 다른 이름을 부여하는 방식, 예를 들어 example.com 은 www.example.com 과 cdn.example.com 으로도 연결가능
    - 사람으로 따지면 이름에 여러 개의 별명이 붙는 것과 같은 이치
- Alias Record : CName 과 비슷하지만 Alias 는 정해진 AWS Resource 로만 리디렉션 가능
    - 호스팅 영역과 이름이 동일한 별칭 레코드 생성가능(CNAME 은 불가능)

* DNS Routing

- Simple Routing Policy : 다수의 IP 에 대응하여 하나의 레코드만 연결가능, Route 53 은 랜덤 순서로 모든 값을 전달
- Weighted Routing Policy : Weight 는 합친 값을 어떻게 잡느냐에 따라 다름, 100 을 전체로 한 후 나누면 됨, 해당 Policy 를 활성화 하고 Set ID 를 다르게 주면 가능
- Latency Routing Policy : 네트워크 지연시간이 가장 짧은 곳으로 라우팅, 이 옵션을 사용하려면 Latency resource record set 를 생성해야 함
    - 각 Region 별로 Set ID 를 생성하고 동일한 Policy 를 지정해준 됨
- Failover Routing Policy : Route 53 이 헬스 체크를 실시함, Primary 가 다운되면 Secondary 로 Failover 를 실시

- Geolocation Routing Policy : 지역 기반 라우팅, Europe 의 사용자가 요청하면 European 고객이 있는 EC2 로 라우팅을 실시함. 즉 요청자의 지역에 기반한 라우팅

* VPC Flow Logs
- VPC 내 네트워크 인터페이스를 통하는 모든 IP 트래픽 정보를 캡쳐하는 서비스
- VPC, Subnet, Network 3 단계에서 생성됨
- 플로우 로그에 태그를 생성할 수 없음
- 다른 계정의 VPC 는 플로우 로그를 생성할 수 없음(피어링된 VPC)
- 플로우 로그가 생성된 이후 설정을 변경할 수 없음
- 수집 제한 대상
    - 아마존 DNS 서버로 전달되는 트래픽
    - 라이센스 활성화를 위한 트래픽
    - DHCP 트래픽
    - 169.254.169.254 로 향하는 트래픽
    - VPC Router 로 향하는 트래릭

8. Automation
* CloudFormation
- Parameters : 사용자정의값 기입
- Condition : 조건부 리소스 할당
- Resource : 의무조건, AWS Resource
- Mapping : 리전 같은 사용자정의 맵핑(Region : AMI)