aws에 대한 정보

aws는 서버-클라이언트 방식
aws는 종량과금제를 사용한다. (온디맨드 방식으로 - 필요하면 바로 제공)

클라우드 컴퓨팅 배포 모델 : 클라우드 기반 배포 / 온프레미스 배포 / 하이브리드 배포

1. 클라우드 기반 : 모든 응용프로그램을 클라우드에서 실행. 따라서 기존 애플리케이션을 클라우드로 마이그레이션. 새 애플리케이션에 대한 설계 및 빌드도 클라우드에서.

2. 온프레미스 배포 : 가상화 및 리소스 관리 도구를 통해 리소스 배포. 애플리케이션 관리 및 가상화 기술을 사용하여 리소스 활용도 상승. 프라이빗 클라우드 배포

3. 하이브리드 배포 : 클라우드 기반 리소스를 온프레미스 인프라에 연결. 클라우드 기반 리소스를 레거시 IT 애플리케이션과 통합

또한 AWS의 AZ는 전 세계에 있기 떄문에 전 세계에 배포가 빠르다.

즉 장점을 정리하면
1. 규모의 경제에 있어서 많은 고객이 사용할수록 종량 과금제 요금 감소
2. 선행비용을 가변비용으로 대체
3. 용량 추정 불필요 --〉 인스턴스 수직 확장
4. 몇 분 만에 전 세계에 배포

멀티 테넌시 : 가상 컴퓨터간 기본 하드웨어 공유
같은 호스트에 여러 인스턴스 들이 있지만 분리되어 오직 할당 받은 것만 인식
프로비저닝 : IT 인프라를 생성하는 것

스팟인스턴스 : 90% 할인된 가격으로 온디맨드 형식으로 제공받지만 AWS가 임의로 회수할 수 있음. 즉 잠깐 쓰는 배치 워크로드에 주로 사용

리전수준에서 로드 밸런싱
보통 로드밸런싱은 프론트에서 백엔드로 갈 떄, 백엔드 인스턴스가 추가 되면 그 백엔드 인스턴스는 모든 프론트에게 알리는데, 로드 밸런서(ELB)를 두면 로드 밸런스 하나에만 통보하여 다 높은 가용성을 가져갈 수 있다.

밀결합된 아키텍처 : 시스템의 결합도가 높아 하나가 문제 생기면 다른 것이나 전체가 다 무너지는 것 (모놀리식 애플리케이션)
소결합된 아키텍처 : 하나의 문제가 다른 장애를 유발하지 않음 (마이크로서비스)
--〉 예시가 버퍼임
이 버퍼는 SQS Simple queue service : 다중화되어 버퍼의 값을 잃어버릴리 없음
SNS Simple Notification Service: 알림까지 전달하는데 발행/구독 형태로 전달 (모바일 푸시나 알림)

서버리스 : 기본 인프라를 보거나 액세스 할 수 없음
aws Lambda는 사용자의 코드를 Lambda 함수라는 공간에 넣고 트리거를 통해 관리형 환경에서 자동으로 실행되게 함. 실행시간은 15분이하로 정해짐--〉 빠른처리에 적합
만약 기본 인프라에 접근을 해야하면 좋지 안쓰는 것이 좋음

기본 인프라에나 기본 OS에 접근을 해야하거나 할때는
ECS
EKS 와 같은 컨테이너(Docker) **오케스트레이션 도구**를 사용하는 것이 좋음

컨테이너 : 코드 패키지 (어떤 프로그램을 실행시키기 위한 코드들)
이들도 똑같이 가상화 기술처럼 격리된 공간에서 실행되지만 EC2 호스트에 의해 실행

그게 필요없다면 Fargate를 사용
정리
1. **그냥 EC2 :** 기존 애플리케이션 호스팅, OS에 대한 전체 엑세스
2. **Lambda**단: 기적인 실행 함수, 서비스 중심 앱 혹은 이벤트 기반 앱, 서버를 프로비저닝 또는 관리하고 싶지 않을 때 사용
3. 오케스트레이션 도구인 **ECS, EKS** : Docker 컨테이너 기반 워크로드 실행
3-2 --〉 그 후 내가 이 컨테이너를 EC2로 관리할 것인지 아니면 편하게 Fargate로 알아서 하게 할 것인지 결정하게 되는 것.

리전 선택의 주요 4가지 : 규정 준수, 근접성, 기능 가용성, 가격

한 리전에는 여러개의 가용영역이 존재해서 재난이나 피해에 따른 장애를 대비할 수 있음.
중요한 것은 절 때 한 리전에 정해진 데이터는 외부에 나가지 않음

아마존의 CDN : CLoudFront
--〉 콘텐츠를 더 빠르게 전송하기 위해 복사본을 캐싱하는 사이트 : 엣지 로케이션
-〉 js나 웹 이미지 파일 속도도 증가시켜줌

아마존의 DNS : route53

AWS outposts라고 매우 소규모적인 리전을 사용자의 작업실이나 회사에 두게하도록 하고 AWS 인프라 및 서비스를 온프레미스 데이터 센터로 확장

아마존의 수많은 서비스들은 API 호출을 통해 통신이 이루어짐
이것들은 AWS 관리 콘솔 , AWS CLI, SDK, 기타 다양한 도구로 사용 가능
1. 관리 콘솔 : 비용 청구, 쉬운 인터페이스 , 하지만 수동적인 도구
2. CLI를 사용하면 거의 같은 환경을 생성시 오류가 나타나지 않음 (터미널로 작성)
3. SDK는 다양한 프로그래밍 언어로 Aws에 접근 가능 (API를 사용하지 않아도 됨)
4. 관리도구 : 밑의 2가지
aws elastic beanstalk : EC2 기반 환경을 프로비저닝하게 만들어줌
aws CloudFormation : 다양한 Aws 리소스를 정의,관리하는 자동화 도구

※CloudFront와 cloudFormation의 차이를 알기

VPC : 싸용자가 정의한 가상 네트워크에서 AWS 리소스를 논리적으로 격리된 AWS 콘텐츠를 프로비저닝 할 수 있음. 오직 정의된 사용자만 접근 가능
즉 통신을 담당하는 것은 퍼블릭 서브넷으로, 데이터나 그런 것들은 프라이빗 서브넷을 할당

VPC는 인터넷 게이트웨이(공개된 출입구와 같음)으로 외부와 통신
그런데 퍼블릭 게이트웨이를 두면 어느든 접근하려고 하므로 가상 프라이빗 게이트웨이를 활용하게 됨.
이 가상 프라이빗 게이트웨이는 정해진 네트워크나 회사 네트워크에서 오는 트래픽만 접근할 수 있는 게이트웨이다. --〉 VPN으로 연결되어진다.
근데 문제는 아무리 이것이 비공개고 암호화되어있어도 통신하기 위해서는 공용 인터넷(큰 도로)를 지나게 된다. 즉 속도가 저하될 수 있고 완전한 보안을 만족하지는 않는다.
---〉 Aws direct connect를 사용가능
aws direct connect는 데이터센터에서 aws로 이루어지는 전용 광섬유 회선을 이용가능
이건 진짜 사람 불러서 물리적인 회선을 제공하는 것이라 완벽한 보안 및 속도 저하 예방도 가능

아무리 서브넷에 ACL을 통해 패킷을 걸러내더라도 서브넷 내에서 어떤 EC2 인스턴스에 갈 것인지는 보안이 없어서 aws는 보안 그룹이라는 틀을 만들었다.
보안 그룹은 오직 정해진 종류의 패킷만 통과할 수 있다.

내부 인스턴스끼리의 통신을 생각하자 그렇다 하더라도 두 인스턴스가 서로 다른 서브넷에 존재할 수 있다. 한 서브넷의 인스턴스는 보안 그룹과 ACL을 잘 통과하더라도 그 두 인스턴스의 주소가 승인목록에 있지 않으면 통신 할 수 없다.

보안 그룹은 ACK 패킷이 나가는 것을 모두 허용함.
하지만 ACL은 그렇지않음.(승인목록을 확인함)
상태 비저장 목록을 통해 한번 잘 들어갔던 패킷은 보안그룹이 그냥 통과시킴

기본 네트워크 액세스 제어 목록은 상태 비저장이며 모든 인바운드 및 아웃바운드 트래픽을 허용한다 즉, 아무것도 기억하지 않고 각 방향(인바운드 및 아웃바운드)으로 서브넷 경계를 통과하는 패킷만 확인합니다.

route 53 : DNS로서 가용성이 뛰어남

Amazon Elastic Block Store 줄여서 EBS : 블록 스토리지, 16tib까지 지원
- 백업에 우수, 초기는 전부 백업, 이후 변경사항만 백업--〉 스냅샷으로 복원
- 단일 가용영역 내에서 이루어지는 스토리지
S3 : 모든 데이터 비정형 까지 되는 스토리지, 무한대 스토리지, 서버리스
- 데이터를 3개 이상에 시설에 같이 저장, 또한 한 시설에 두 개의 별도 물리 장치에 복제
- 99.999999999퍼센트의 내구성 제공
- 버킷의 URL로 웹페이지를 호스팅할 수 있음
S3-IA: 액세스 빈도가 낮은데 필요할 때 빠르게 액세스하는데 사용. 재해 데이터 센터 등
Amazon S3 Glacier : 거의 데이터를 저장만하고 건드리지 않을 때

객체 스토리지는 데이터, 메타데이터, 키로 이루어져 있음

아마존 EFS (파일 시스템)
- 여러 인스턴스가 동시에 접근 가능

EBS는 가용영역 수준 리소스라서 같은 가용영역에 있어야 EC2에 연결할 수 있음. 또한 볼륨이 자동으로 확장되거나 축소되지도 않음
그 반대로 EFS는 읽기 쓰기가 동시에 가능하며, LInux 파일 시스템으로 이루어져 있음. 또한 리전 수준의 리소스여서 같은 리전에 있으면 EC2에 연결 가능

amazone RDS : 키- 값으로 저장하지 않음
-Amazon Aurora 데이터베이스에 데이터 저장

dynamoDB : 서버리스 데이터 베이스
- 데이터베이스 관리 알아서 해줌
- 데이터를 중복 저장 그래서 가용성이 높음
- 성능도 존나 빠름
- SQL문 사용 안함. --〉 성능 향상에 큰 도움
- 즉 NOsql 데이터 베이스임 (확장성 좋음)
- 복잡한 SQL 쿼리 사용 불가. 파티션 이용
- 키- 값으로 저장

과거에 대한 데이터 즉 **기록 데이터**는 매우 양이 커질 수밖에 없어 기존 데이터베이스로는 감당하지 못하게 된다. 또한 기록 데이터는 비정형도 존나게 많아서 힘들다. 그래서 데이터 웨어하우스가 등장
Redshift : 데이터 웨어하우스

DMS : 마이그레이션 서비스
- 마이그레이션이 지속되도 데이터베이스 기능은 정상 작동
- 앱 중지 시간 최소화

이종 마이그레이션 서비스는 SCT를 통해 스키마와 데이터베이스 구조를 같게 만들고 그 다음 DMS를 통해 마이그레이션 하는 것이다.

Aws는 클라우드 자체의 보안을 책임지고 고객은 클라우드 내부의 보안을 책임진다.

루트사용자가 Multi-factor authentication (MFA)를 활성화해서 토큰 제공
혹은 IAM을 통해 권한을 루트가 부여하여 정해진 권한만 부여 가능
IAM은 JSON을 통해 부여하는 것인데, 일시적으로도 부여 가능
IAM으로 권한을 Aws 서비스를 부여할 수도 있음

aws organization
- 여러 aws 계정을 관리하는 중앙 위치
- 0통합 결제
- 계정의 계층적 그룹화
- 루트 계정도 서비스 제어 정책에 영향을 받음

SCP는 조직 단위(OU)와 개별 멤버 계정을 적용

aws artifatct를 통해 특정 규정을 준수하도록 할 수 있음

ELB는 Slowroris 공격에 효과적인 대응
aws waf와 aws shield 는 충분히 보안 공격에 대응함

Aws key management는 dynamoDB에 접근하기 위한 키를 관리할 수 있다
SSL 암화는 거의 모든 aws 서비스에서 제공

amazon inspector : aws 서비스나 서비스에서 개발한 응용프로그램의 보안 규정 준수 여부를 판단함.

Amazon CloudWatch : aws 서비스를 모니터링
- 서비스의 상태에 따라 경고를 알림형태로 알려줄수 있음
- 게다가 SNS를 통해서도 받고, SNS를 통해 경고를 줄 수도 있음
- 또한 대시보드 기능도 제공
- 즉 평균 업무 비용 (MTTR) 감소 -〉 비즈니스 가치 증대
- 애플리케이션 및 운영 리소스 최적화를 위한 인사이트 확보

aws CloudTrail은 상태 변화 변경 등 감사 관점에서 대부분의 변화들을 기록하는 서비스
즉 거의 모든 것을 로깅하고, 로그를 저장하는 서비스이다
또한 대시보드에서 원하는 지표(서비스)에 액세스할 수 있다.

AWS trusted Advisor : 보통 Aws 사용한다 해도 모든 서비스를 완벽히 사용할 수 없다. 하지만 이는 비용 최적화, 성능, 보안, 내결함성, 서비스 한도에 대해 조언해 주는 서비스 이건 무료다. 일부 서프트 수준이 높은 건 결제해야하는 듯

통합 결제로 대량 구매 할인을 적용받기 위해 계정 간 사용량을 결합할 수 있음

Aws support는 AWS 비용 측면에서 효율적으로 사용하기 위한 플랜들을 제공한다
- Basic , Developer, Business, Enterprise 로 4가지 플랜이 있다.
- basic은 무료 이나 그 이상은 다 유료
-기술 지원 관리자(TAM)에게 도움받고 싶으면 Enterprise
AWS trusted Adviser는 Business 플랜 이상부터 제공한다.

AWS Marketplace는 타사 소프트웨어를 검색,배포, 관리를 통합시켜놓은 옵션이다.
- 즉 원클릭으로 편하게 함.
- 물론 라이선스 구입에 돈을 쓰긴함.
- 하지만 온디맨드 및 종량제 옵션도 제공하는 것이 대부분이라 개꿀임

온프레미스 환경이나 애플리케이션을 aws에 마이그레이션하려면 aws Cloud Adoption Framework(AWS CAF)를 사용한다.
즉 마이그레이션 하는데 도움을 준다
총 6가지 관점에서 도움을 준다
- 비즈니스 , 인력, 거버넌스 , 플랫폼, 보안, 운영

: 비즈니스 관점은 비즈니스 전략과 IT 전략을 분리하는 모델에서 IT 전략을 통합하는 비즈니스 모델로 전환하는 데 도움이 됩니다.
: 운영 관점은 비즈니스 이해당사자의 요구 사항을 충족하도록 IT 워크로드를 운영 및 복구하는 데 중점을 둡니다.
: 인력 관점은 인사 관리(HR) 직원이 클라우드 기반 역량을 포함하도록 조직 프로세스와 직원 기술을 업데이트하여 팀이 클라우드 채택을 준비하는 데 도움이 됩니다.
: 플랫폼 관점은 비즈니스 목표 및 관점에 따라 AWS 인프라를 설계, 구현 및 최적화하는 데 도움이 되는 관점

aws로 마이그레이션하는 전략은 6가지가 있다. (6R 전략)
-시간 비용 등을 우선 순위를 기준으로 제공한다
1. 리호스팅(리프트 & 쉬프트) : 그냥 모든 애플리케이션을 그대로 AWS에 이전
 - 최적화는 없지만 전환으로 비용 감소 할 수도 있음
2. Replatforming (Rift& shift 및 수정) : 몇까지 코드를 최적화하는 것이지만, 핵심코드는 변경하지 않음. 즉, DB 같은 것은 플랫폼만 옮겨도 성능이 향상됨.
3. 폐기(Retire) : 보통 사용되지 않거나 대체되는 애플리케이션을 사용안함.
4. 유지(Retain) : 그래도 일부 앱은 계속 유지될 수 있으므로 냅두지만 일정한 기간을 둠.
- 오직 비즈니스에 적합한 애플리케이션만 마이그레이션하는 것이 좋기때문이다.
- 궁긍적으로 마지막에는 폐기됨
4. 재구매(Repurchasing) : 이전 쓰던 공급업체의 라이선스 사용을 중지하고 새로운 기업의 제품을 구매하는 것.
 - 이는 일부 선행비용이 발생할수도 있음. 기다려야한다거나
6. Refactoring : 온프레미스에는 가능하지 않은 기능 추가나 변경이 필요한 경우에 사용
- 계획 및 인적 노력 비용이 가장 큼

일부 고객이 데이터를 AWS로 가져와야하는 것은 네트워크로 할 수 있지만 이는 대역폭에 따라 매우 오래걸리거나 비용도 매우 클 수 있다.

1. AWS Snowcone : 8TB
2. AWS Snowball edge
  - compute option : 42TB
  - storage optimized option : 최대 80TB
3. AWS Snow mobile : 13.7 미터의 컨테이너에 담겨져 트럭으로 견인
 약 100PB에 해당 하고 보통 데이터 센터를 마이그레이션할 때 이동
 - 전담 보안 및 실시간 감시, 경호원도 동원됨.
 - 보안 및 변조 방지로 하게 되어 있음. 256비트 암호화 키

Aws sageMaker : 기계학습
Amazon Augmented AI
Amazon A2I
Amazon Lex : 채팅 봇
Amazon Textract
AWS DeepRacer

AWS Well-Architected 프레임워크
- 고객이 구성한 애플리케이션이
- 운영 우수성, 보안, 안정성, 성능 효율성, 비용 최적화
- 에서 잘 짜여졌는 지 , 결함은 없는지 알려줌

aws 클라우드 사용 이점
1. 고정비용을 가변 비용으로 대체
2. 규모의 경제로 얻게 되는 이점
3. 용량 추정 불필요
4. 속도 및 민첩성 향상
5. 데이터 센터 윤영 및 유지 관리에 비용 투자 불필요
6. 몇 분만에 전 세계에 배포

가용 영역 : 글로벌 인프라의 완전히 격리된 파티션
AWS Elastic Beanstalk :애플리케이션을 신속하게 배포하고 확장하는 데 사용되는 서비스
AWS CloudTrail : AWS 환경에서 발생한 사용자 활동 및 API 호출에 대한 세부 정보를 검토할 수 있는 서비스
Amazon S3를 통해 객체의 액세스 패턴을 모니터링하고자 합니다. 어떤 스토리지 클래스를 사용해야 합니까? - S3 Intelligent- Tiering

AWS 서비스 및 애플리케이션에 대한 작업을 자동화하는 데 사용되는 도구는 무엇입니까?
- AWS 명령줄 인터페이스

Amazon EC2 인스턴스에 대한 인바운드 및 아웃바운드 트래픽을 제어하는 Virtual Private Cloud(VPC) 구성 요소
- 보안 그룹

athena, **glue**, s3 - data 엔지니어링

s3에서 파일 올려서 Athena로 가져와서 glue로 ETL 해보는 것

aws sloution architect

s3 권한 정책은 거의 대부분을 허용하는 정책이다.
하지만 IAM을 이용하면 필요한 권한만 부여할 수 있다 --〉 Least privilege