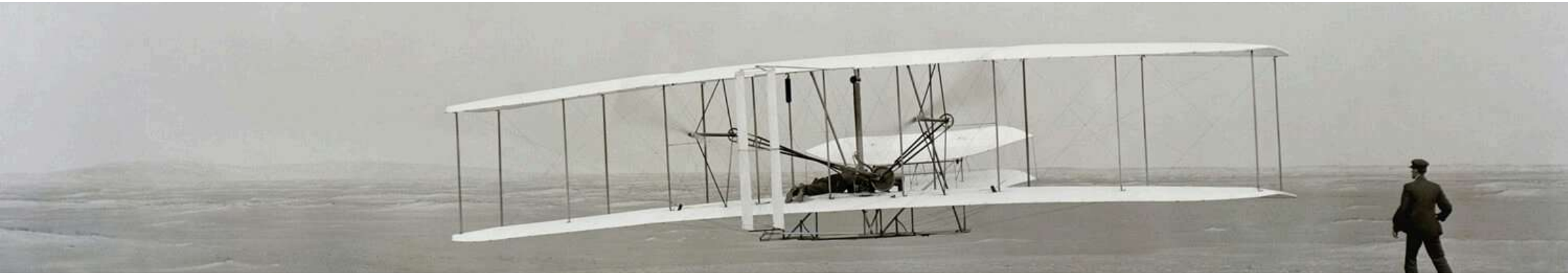


Architecting on AWS

아이콘의 의미 → ☞ : 설명 • : 개념 정의 ✷ : 서비스 ❒ : 서비스 및 개념의 그룹

https://bit.ly/emhong-arch-note

▼ Module 1. Architecture Fundamentals

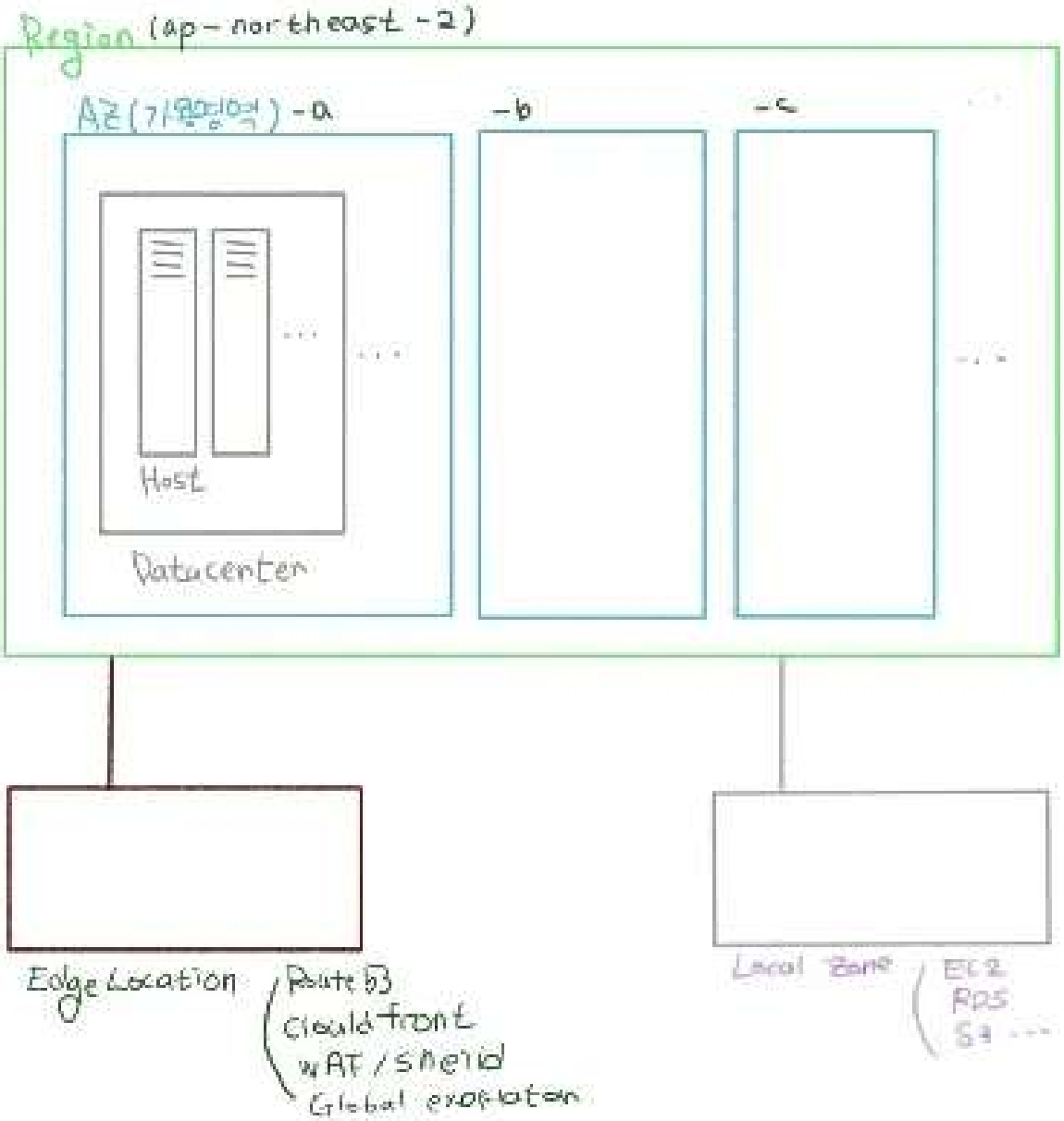
AWS Infra(Global Infra)

- Region (리전) : 개별 지리 영역으로 세 개 이상의 가용영역(AZ) 가 포함
- Availability Zones(가용영역:AZ) : Region(리전) 내의 하나 이상의 데이터 센터의 그룹
- Datacenter(데이터센터) : 수천대의 서버를 호스트, 각 로케이션간 AWS 전용 네트워크를 사용

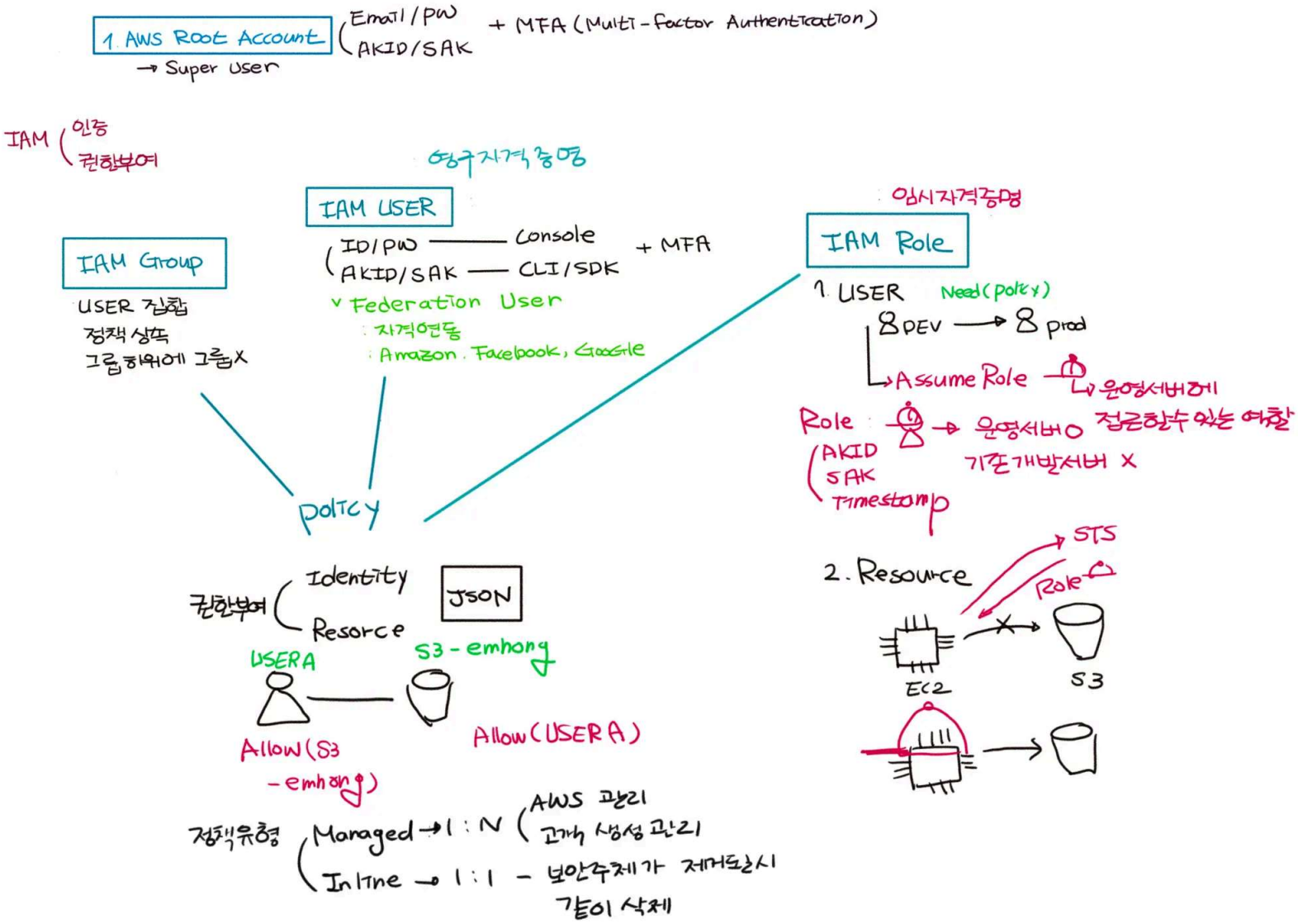
- AWS Local Zones : 사용자와 지리적으로 근접한 AWS Region(리전)의 확장
- Edge Location : 전세계 주요 도시에 위치하는 요청자에게 가장 가까운 지점

Well-Architected Framework

- AWS Well-Architected Tool (Well-Architected Tool)


Region (ap-northeast-2)
AZ(가용영역) -a
-b
-c
Host
Datacenter
Edge Location
Route 53
cloudfront
WAF/shield
Global exploration
Local Zone
EC2
RDS
S3 ...

▼ Module 2. Account Security



AWS 계정 루트 사용자

☞ AWS 계정을 생성할 때 만들어지는 최고 권한의 계정으로 AWS 로 로그인하는 데 사용하는 AWS 계정 루트 사용자 자격 증명을 생성

☞ 계정을 생성할 때 입력한 이메일 주소와 암호를 사용

☞ 모든 리소스에 완전히 무제한으로 액세스 가능

☞ 일상적인 액세스에는 루트 사용자 자격 증명을 **사용하지 않는 것을 권고**

단일 계정 관리

✷ AWS Identity and Access Management(IAM)

☞ 인증 / 권한 부여를 관리

- **IAM User (사용자)**

  ☞ ID/PWD(Management Console Access) 혹은 AKID/SAK(Programing 방식) 를 이용하여 영구자격증명 획득

- **IAM Group (그룹)**

  ☞ IAM User 의 집합으로 Group 단위의 Policy(정책) 부여 가능 및 Policy(정책) 상속

다중 계정 관리

✷ AWS Organizations

☞ 여러 AWS 계정을 조직에 통합하고 중앙에서 관리할 수 있는 계정 관리 서비스

☞ 계정을 조직 단위(OU)로 정책을 적용

☞ 서비스 제어 정책(SCP)

IAM 권한 경계

SCP(서비스 제어 정책)

IAM Policy (정책)

✷ AWS Identity and Access Management (IAM)

☞ IAM 엔티티(IAM User, Group, Role) 에게 허용된 권한을 정의

☞ 여러 AWS 계정을 조직에 통합하고 중앙에서 관리할 수 있는 계정 관리 서비스

☞ 계정을 조직 단위(OU)로 정책을 적용

☞ 서비스 제어 정책(SCP)

IAM 권한 경계 : IAM 엔티티(IAM User, Group, Role) 가 수행 할수 있는 최대 권한을 설정, 반드시 IAM Policy (정책) 와 함께 사용해야 함

SCP(서비스 제어 정책) : 조직의 권한을 관리하는 데 사용할 수 있는 조직 정책 유형

✷ 정책 분류 #1

1. 관리형 정책 : IAM 권한을 조직 전체에서 사용할 수 있으며 IAM 엔티티(사용자, 그룹 또는 역할)에 연결할 수 있는 자격 증명 기반 정책
   a. AWS 관리형 정책 : AWS 에서 생성 및 관리하는 관리형 정책
   b. 고객 관리형 정책 : 사용자가 자신의 AWS 계정에서 생성 및 관리하는 관리형 정책
2. 인라인 정책 : 직접 생성 및 관리하며, 단일 사용자, 그룹 또는 역할에 직접 포함되는 1:1 관계를 가지는 정책으로 재사용이 불가능

☞ **IAM 및 AWS STS 할당량**

https://docs.aws.amazon.com/ko_kr/IAM/latest/UserGuide/reference_iam-quotas.html

◦ 정책분류 간의 관계

정책 분류 #1	장책분류 #2	예제
자격 증명 기반 정책	관리형 정책 인라인정책	User A 에게 S3 bucket 에 접근할 권한을 부여
리소스 기반 정책	인라인정책	S3 bucket 에 User A 라는 사용자가 접근할 수 있게 허가

▼ Module 3. Networking 1

명시적 거부 >> 명시적 허용 >> 암묵적 거부

- **Role (역할)**
  - ☞ 임시자격증명을 부여하는 것으로 AWS 리소스에 대한 액세스 권한을 위임함.
    - 역할 수임하는 방법

> *IAM 역할을 사용해 신뢰하는 계정과 다른 AWS 신뢰받는 계정 간에 신뢰 관계를 설정*

> *신뢰 관계를 생성한 후 신뢰받는 계정의 IAM 사용자 또는 애플리케이션은 AWS Security Token Service(AWS STS)에게 AWS 리소스에 액세스할 수 있는 임시 보안 자격 증명을 요청*

> *AWS STS로 부터 임시자격 증명을 반환 받아 사용*

VPC (Virtual Private Cloud)

✷ Amazon VPC

☞ 사용자의 AWS 계정 전용 가상 네트워크로써 AWS 클라우드에서 다른 가상 네트워크와 논리적으로 격리됨.

☞ Region 레벨 서비스로 하나의 리전을 선택해서 생성하고 여러 가용영역을 포함.

- CIDR(Classless Inter-Domain Routing)

  ☞ VPC와 서브넷에 IPv4 주소와 IPv6 주소를 CIDR(Classless Inter-Domain Routing) 블록 형태로 지정 할당

  ☞ CIDR 블록에는 마침표로 구분된 0~255의 십진수가 최대 3개인 4개 그룹이 있으며, 그 뒤에 슬래시와 0~32의 숫자가 표시됩니다. 예: 10.0.0.0/16

  ☞ AWS에서 지정가능한 CIDR 범위는 /16 ~ /28 까지임

✷ Subnet

☞ 서브넷은 VPC의 IP 주소 범위로써 네트워크 영역을 분할함.

☞ 서브넷 생성시 5개의 주소는 이미 예약됨.

☞ Subnet의 종류

1. Private Subnet
   a. 인터넷과 연결되지 않은 Subnet 으로 기본으로 생성 하면 Private Subnet 으로 생성됨
2. Public Subnet
   a. 인터넷과 통신 가능
   b. 통신을 위해서는 3가지 요건이 필요
      i. Internet Gateway
      ii. Public IP : 리소스에 부여
      iii. Routing table

✷ Internet Gateway

☞ 수평 확장되고 가용성이 높은 중복 VPC 구성 요소로, VPC와 인터넷 간에 통신할 수 있게 해주는 일종의 관문

✷ Routing Table

☞ 패킷의 경로를 지정해 주는 것으로 네트워크 트래픽이 전달되는 위치를 제어

☞ VPC의 각 서브넷을 라우팅 테이블에 연결, 테이블의 각 라우팅은 목적지 및 대상을 지정

✷ Elastic IP

☞ 인터넷에서 연결 가능한 퍼블릭 IPv4 주소

✷ Elastic Network Interface

☞ VPC에서 가상 네트워크 카드를 나타내는 논리적 네트워킹 구성 요소

✷ NAT(Network Address Translation)

☞ 프라이빗 서브넷의 리소스가 인터넷, 다른 VPC 또는 온프레미스 네트워크에 연결되도록 허용, 외부로 부터 들어오는 트래픽은 받을수 없음.

☞ NAT Gateway : 관리형, 가용성과 대역폭을 제공

☞ NAT Instance : 비관리형으로 EC2 인스턴스에서 NAT 디바이스를 생성

- NAT Gateway vs NAT Instance

VPC Traffic Security

✷ NACL

☞ 서브넷 수준에서 특정 인바운드 또는 아웃바운드 트래픽을 허용/거부 정책 지원

☞ 상태비저장 - 인바운드/아웃바운드시 모두 정책 확인

☞ 기본 : 모두 허용

☞ 규칙 번호가 가장 낮은 규칙부터 평가됨 즉, 규칙에 일치하는 트래픽이 있으면 이와 모순되는 상위 규칙이 있더라도 적용

✷ Security Group

☞ 리소스에 대한 인바운드 및 아웃바운드 트래픽을 제어, 허용 규칙만 지원

☞ 상태 저장 - 인바운드 된 트래픽의 경우 아웃바운드 시 정책 미확인

☞ 기본 : 인바운드 → 모두 거부, 아웃바운드 → 모두 허용

☞ 트래픽 허용 여부를 결정하기 전에 모든 규칙을 평가함

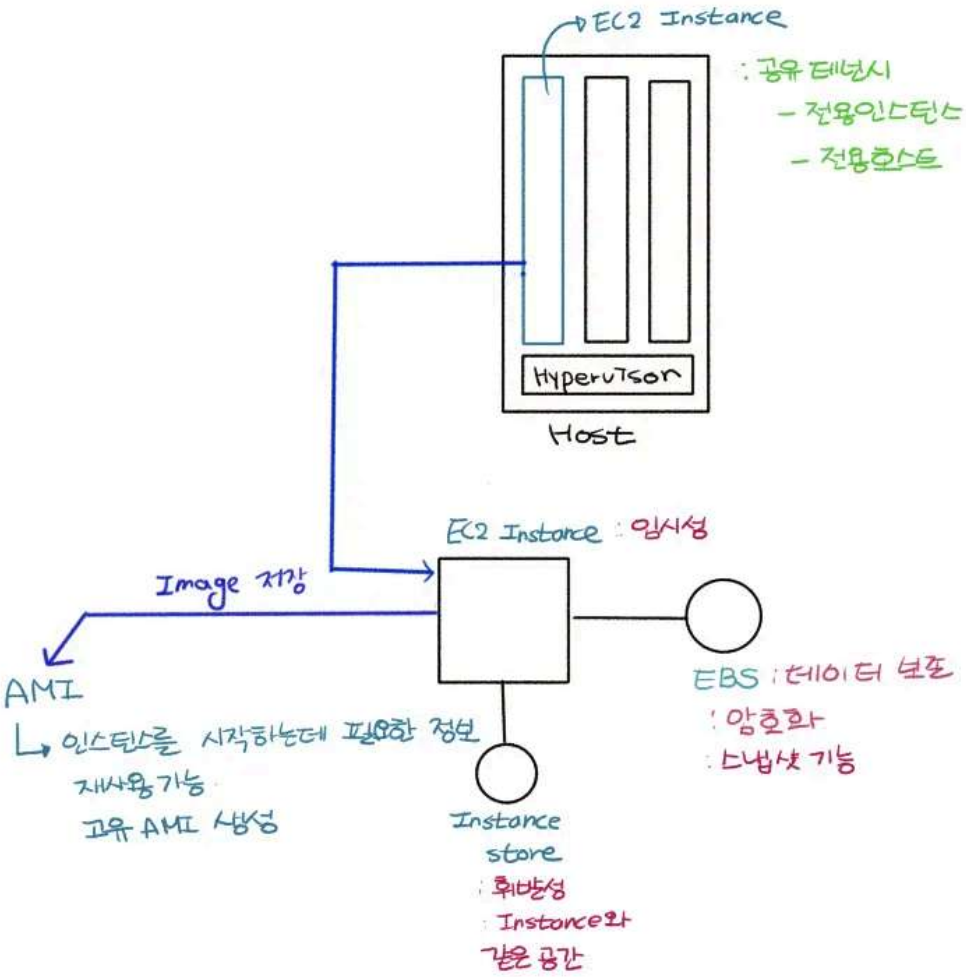
- NACL vs Security Group

Security Group	NACL
인스턴스 레벨	서브넷 레벨
허용 규칙	허용 및 거부 규칙
상태 저장	상태 비저장
트래픽 허용 여부를 결정하기 전에 모든 규칙을 평가함	가장 낮은 번호의 규칙부터 순서대로 규칙을 평가함

▼ Module 4. Computing

Amazon EC2(Elastic Computing Cloud)

☞ EC2 Instance : 클라우드에서 실행되는 가상 서버, AMI에서 인스턴스를 바로 시작



☞ 시작시 확인 사항

✷ AMI (Amazon Machine Image)

EC2 Instance 를 시작하는 데 필요한 정보를 제공하는 AWS에서 지원되고 유지 관리되는 이미지

• UserData : EC2 Instance 생성시에 단한번만 수행되는 스크립트

• Metadata : 실행 중인 EC2 Instance를 구성 또는 관리 하는데 사용할 수 있는 해당 EC2 Instance 관련 데이터, 인스턴스 내부에서만 접근 가능

• Tag : 키와 값으로 이루어 진 단순한 레이블. 인스턴스 관리 등에 사용

• 테넌시

1. 공유 테넌시 : 여러 AWS 계정이 동일한 물리적 하드웨어를 공유
2. 전용 인스턴스 : 인스턴스가 단일 테넌트 하드웨어에서 실행으로 하드웨어 수준에서 물리적으로 격리, 단, 동일한 AWS 계정의 다른 인스턴스와 하드웨어를 공유할 수 있음
3. 전용 호스트 : 고객 전용의 EC2 인스턴스 용량을 갖춘 물리적 서버로써 인스턴스가 완전히 전용으로 사용, 소켓 및 물리 코어 수 표시 여부 제공, 시간에 따라 지속적으로 동일한 물리 서버에 인스턴스 배포 허용

• 배치그룹

클러스터 배치그룹	분산 배치 그룹	파티션 배치그룹
단일 가용 영역 내에 있는 인스턴스의 논리적 그룹	각각 고유한 하드웨어에 배치된 인스턴스 그룹	EC2는 각 그룹을 파티션이라고 하는 논리 세그먼트로 나누어 배치
짧은 네트워크 지연 시간, 높은 네트워크 처리량	서로 떨어져 있어야 하는 중요 인스턴스의 수가 적은 애플리케이션	HDFS, HBase, Cassandra 같은 대규모 분산 및 복제 워크로드

☞ 인스턴스 유형 및 크기 : 패밀리, 세대, 추가 기능 및 크기를 기준으로 이

✷ AWS Compute Optimiser

AWS 리소스의 구성 및 사용률 지표를 제공 , 리소스가 최적 상태인지 여부를 보고하고, 최적화 권장 사항을 생성하여 비용을 절감하고 워크로드의 성능을 개선하는데 이용

☞ EC2 요금옵션

1. 온디맨드 인스턴스 : 장기 약정 없이 초 단위로 컴퓨팅 용량에 대해 비용을 지불
2. Savings Plans
   a. Compute Savings Plans: 인스턴스 패밀리, 크기, AZ, 리전, OS 또는 테넌시와 관계없이 EC2 인스턴스 사용량에 적용되며, Fargate 또는 Lambda 사용량에도 적용
   b. EC2 Instance Savings Plans : AZ, 규모, OS 또는 테넌시에 관계없이 해당 리전의 패밀리 내에서 인스턴스 간에 사용량을 변경할 수 있는 유연성을 제공
3. 스팟 인스턴스 : 애플리케이션이 실행되는 시간을 유연하게 조정할 수 있고 애플리케이션을 중단할 수 있는 경우에 선택

Amazon EBS(Elastic Block Storage)

☞ EC2 인스턴스에 사용할 수 있는 블록 수준 스토리지 볼륨

☞ 범용 SSD, 프로비저닝된 IOPS SSD / 처리량 최적화 HDD 및 콜드 HDD

☞ 증분 스냅샷, 암호화

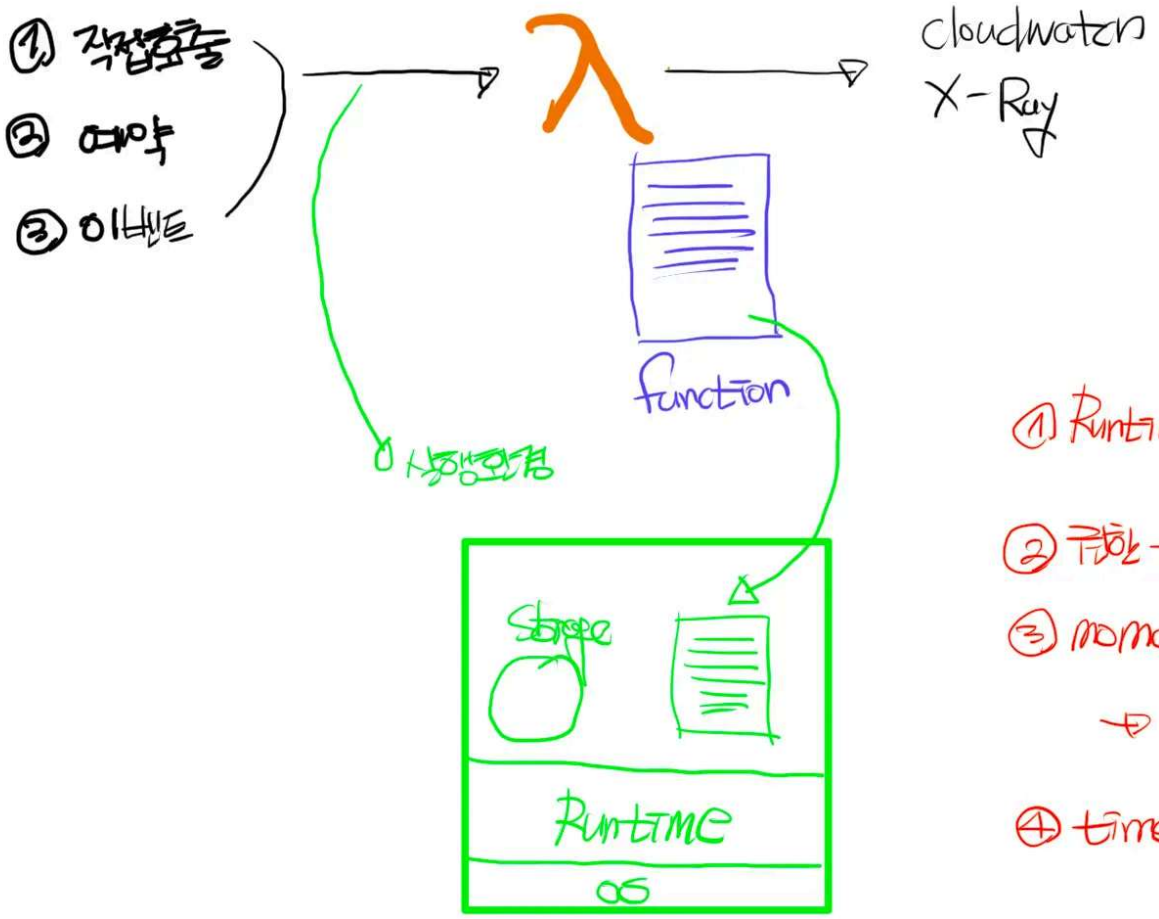
AWS Lambda

☞ 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있게 해주는 서버리스 컴퓨팅 서비스

☞ 15분동안만 실행 / 10GB의 메모리 제약이 있음

☞ CPU는 비례해서 증가함

*cold start / warm start



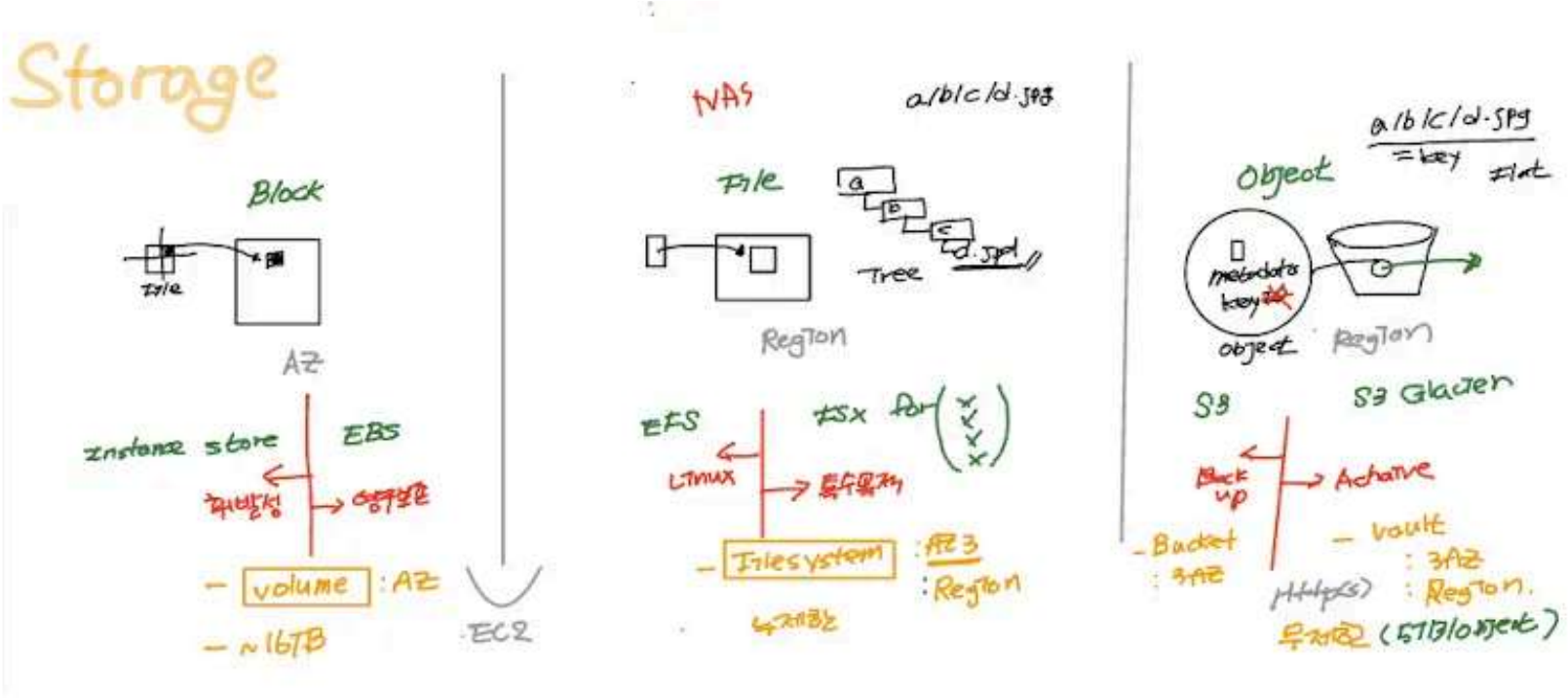
① Runtime (기본 구성 / 사용자지정 런타임)

② 권한 - Role

③ memory : 128MB ~ 10GB
→ cpu / Network 자동

④ timeout : ~ 15분

▼ Module 5. Storage



📕 객체 스토리지 Amazon S3(Simple Storage Service)

☞ 데이터 형태에 구애 받지 않고 Object(객체)라는 단위로 저장

☞ 3 AZ 저장, 고가용성

☞ 구성요소

1. Bucket(버킷) : 저장된 Object(객체)에 대한 컨테이너로써 생성시 리전을 선택하게 되며 이름은 세계적으로 유일하게 지정하여야 함
2. Object(객체) : Amazon S3에 저장되는 기본 개체, 객체는 객체 데이터와 메타데이터로 구성, 각각의 객체는 5TB 로 제한
3. Key(키) : Bucket(버킷) 내 Object(객체)에 대한 고유한 식별자
    a. `https://DOC-EXAMPLE-BUCKET.s3.us-west-2.amazonaws.com/photos/puppy.jpg :` URL에서 `DOC-EXAMPLE-BUCKET`은 버킷의 이름이고 `photos/puppy.jpg` 은 키

☞ Versioning

동일 버킷 내에 여러 개의 객체 변형을 보유하는 수단

객체를 삭제할 경우 Amazon S3는 객체를 영구적으로 제거하는 대신 삭제 마커를 삽입, 따라서 언제든지 이전 버전을 복원할 수 있음

☞ S3 엑세스 관리

Amazon S3는 Bucket(버킷) 및 객체에 대한 액세스 감사 및 관리 기능을 제공, 기본적으로 S3 버킷 및 객체는 프라이빗

1. 버킷 정책 : Bucket(버킷)과 해당 Bucket(버킷)의 객체에 액세스 권한을 부여할 수 있는 리소스 기반 정책(JSON)
2. 엑세스 포인트 : 데이터 액세스를 대규모로 관리하기 위해 전용 액세스 정책이 포함된 명명된 네트워크 엔드포인트를 구성
3. 데이터 암호화

파일 스토리지 EFS (Elastic File Store)

a. 서버측 암호화

☞ 3 AZ 저장, 고가용성

i. AWS Key Management Service(SSE-KMS)와 함께 서버 측 암호화 사용

☞ EFS : Linux 용 파일스토리지로써 탄력적 파일 시스템

ii. Amazon S3 관리형 암호화 키(SSE-S3)로 서버 측 암호화 사용

☞ FSx for Windows File Server/ Lustre / Netapp ONTAP / OpenZFS

iii. 고객 제공 키(SSE-C)로 서버 측 암호화 사용

데이터 마이그레이션

b. 클라이언트 측 암호화

✷ Storage gateway : 클라우드 스토리지 액세스를 온프레미스에 제공하는 하이브리드 클라우드 스토리지 서비스

i. 클라이언트 측 암호화를 사용하여 데이터 보호

1. Volume

☞ 수명주기 정책 : 객체가 비용 효율적으로 저장되도록 관리

2. File

☞ 객체 복제 : 동일 리전 복제 / 교차 리전 복제

3. Tape

☞ 스토리지 클래스 : S3 standard - S3 Standard IA - S3 One Zone -IA - S3 Glacier

✷ DataSync : 온프레미스와 AWS 스토리지 서비스 사이에서 데이터 이동을 자동화 및 가속화

✷ S3 Glacier : 비용 효율적 스토리지

✷ Snow 패밀리 : 오프라인 방식의 마이그래이션 서비스

▼ Module 6. Database

✷ **Amazon RDS**

☞ EC2 유형과 스토리지 용량을 선택

• Multi-AZ

1. Instance : primary - standby, 동기식 복제, 자동 승격
2. Cluster : primary - read replica - read replica , 동기식 복제, 자동 승격

- Read Replica(max 5개/region)

  AZ, Region 모두 가능, 수동 승격

✷ **Amazon Aurora**

☞ EC2 유형 선택

☞ Instance / Storage 영역이 분리

• Read Replica(max 15개/region) : AZ, Region 모두 가능, 자동승격

✷ Aurora Serverless : Pay as you go, Instance 부분이 요청시 생성

Database

✷ Amazon RDS

1. 특징

- PostgreSQL / MySQL / MariaDB

  Oracle / SQL Server / Db2

- Amazon Aurora

- ~64TB

- 수직확장가능 (인스턴스 : Rebooting

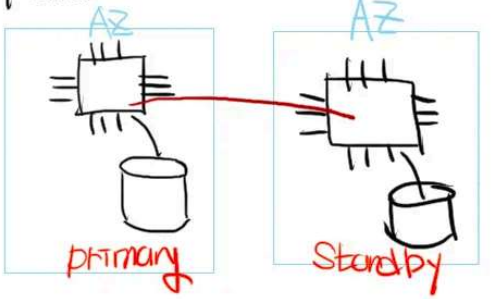
  EBS : 데이터 정보보관가능

5. 자동백업

- Enabled Automated backup 활성화

- 최대 35일 저장

2. Multi - AZ



① 동일한 유형선택

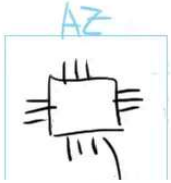
② primary Endpoint로 접근

③ 동시에

④ 자동장애

⑤ 장애복구 다음

3. Cluster



① 3개의 AZ

② Endpoint가 2개

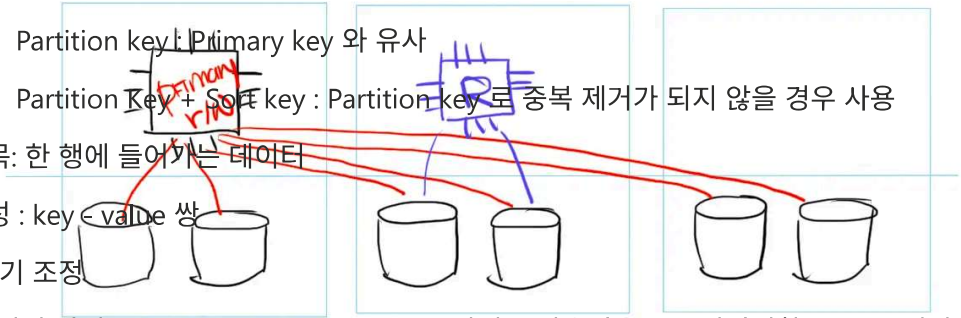
Database

✷ Amazon DynamoDB

☞ AWS의 대표적인 비관계형(NoSQL), 완전 관리형 데이터베이스, 유연한 스키마

☞ 구성 요소

- 테이블 이름
- 기본키
  1. Partition key : Primary key 와 유사
  2. Partition key + Sort key : Partition key 로 중복 제거가 되지 않을 경우 사용
- 항목: 한 행에 들어가는 데이터
- 속성 : key - value 쌍



• 용량 및 크기 조정

1. 프로비저닝 방식 : RCU(4KB/s) / WCU(1KB/s) 단위로 사용량을 프로비저닝함, 오토 스케일링을 이용하여 자동으로 용량 확장
2. 온드맨드 방식 : 사용하는 만큼 비용 지불

• 일관성 법칙

1. 최종 일관성 : 쓰기 작업이 완료되지 않아도 읽기가 가능하므로 1초 미만의 일관성이 맞지 않는 경우가 발생
2. 강력한 일관성 : 쓰기 작업이 완료 되지 않은 경우 읽기 불가능

• 글로벌 테이블

☞ 테이블을 사용할 수 있는 AWS 리전을 지정하고 지속적인 데이터 변경 사항을 모든 해당 리전으로 전파,

☞ 비동기식으로 데이터 연동, 글로벌 테이블을 만들기 위해서는 스트림 생성이 사전 조건임

▼ Module 7. Monitoring and Scaling

✷ CloudWatch

- Metric(지표): 시스템 성능에 대한 데이터, 리소스에 대한 무료 지표를 제공
- Alarm(경보): CloudWatch 지표를 감시하거나 CloudWatch 지표를 기반으로 작업을 호출