Computing Service(2)

가상 서버를 안전하게 외부에 공개

1. 서버 외부 공개
    - 공개된 응용 프로그램은 인터넷을 통해 접속해서 사용할 수 있어야 하므로 EC2와 인터넷 간 통로를 설정해야 함
    - EC2를 인터넷에 공개하려면 3가지 조건을 충족해야 함
        - **EC2를 퍼블릭 서브넷에 배치**

          → 퍼블릭 서브넷: 인터넷과 통신할 수 있는 서브넷

          프라이빗 서브넷: 인터넷과 직접 통신할 수 없는 서브넷, AWS 내에서 통신

          서브넷: EC2를 배치하는 네트워크
        - **퍼블릭 IP 주소를 EC2에 부여**

          → 퍼블릭 IP: 인터넷과 통신할 수 있는 IP 주소

          IP 주소: EC2의 위치를 나타내는 주소
        - **보안 그룹에서 외부로부터의 접근을 허가**

          → 웹 사이트라면 80번, 443번 포트 허가
    - 보안 그룹
        - 특정 인스턴스가 특정 주소/특정 대역/전체 인터넷과 통신할 수 있도록 규칙을 만들어 이 규칙에 따라 통신 제어
        - EC2 인스턴스를 전 세계에 공개하고자 한다면 **80/443** 포트를 이용해 통신을 허가해야 함

2. 서버 접근 제어
    - 보안 그룹
        - 접속을 허가할 지에 대한 **접근 제어에 사용**

    - 온프레미스의 방화벽 기능 수행
- **인바운드 규칙: 외부 → EC2 통신**

  아웃바운드 규칙: EC2 → 외부 통신

  둘 다 통신을 허용할 네트워크와 포트 번호 지정 → **포트 번호: 사용하는 통신 종류**로 결정
- 보안 그룹에 지정하는 규칙: 허용만 할 수 있음, **차단은 설정 불가**
- 보안 그룹은 용도마다 따로 생성하고 키-값에 별칭 입력

부하에 따라 서버 자동 추가/삭제

1. 서버 자동 추가/제거
    - **Amazon EC2 Auto Scaling(이후 Auto Scaling)** 기능 사용 시 서버 추가/제거 자동 수행 가능
    - **스케일 아웃(Scale-out): 서버를 추가하는 것**

      스케일 인(Scale-in): 서버를 제거하는 것
        - **대상 추적 조정 정책:** 목표 사용률을 지정해 지정 값을 유지하게끔 인스턴스 수 자동 조절

          → 무료, 사용자는 늘어난 인스턴스의 비용만 지불
        - **스케줄 스케일:** 사용자 접속 수와 같은 기준을 이용한 자동 추가 및 삭제 설정 가능, 일정을 세워 인스턴스 수 조절
        - **Auto Scaling:** 장애 대비용으로 사용

          → 지정된 인스턴스 수 유지, 예기치 못한 장애로 인스턴스 중지 시 자동으로 인스턴스 생성
        - **예측 스케일링 기능:** 서버 수의 수요 예측
        - **버전 관리:** 인스턴스 시작 시 템플릿이 될 부팅 설정

서버 없이 프로그램 실행

1. **서버리스 컴퓨팅 서비스 Lambda**

- AWS Lambda(이후 Lambda): 서버리스 컴퓨팅 서비스

  → 인프라 관리할 필요X, 프로그램 코드 준비와 업로드만 하면 됨

- 서버리스: AWS에서 서비스가 실행될 인프라 관리, **사용자가 관리할 서버X**
- Lambda의 장점
  - **보안:** AWS에서 OS와 미들웨어 등의 기반 시스템 총 관리
  - **비용:** EC2는 사용하지 않아도 기동하고 있는 시간만큼 요금 발생, Lambda는 코드 실행
    시에만 요금 부과
  - **가용성:** 복수의 가용 영역에서 실행, 설정하지 않아도 고가용성/장애 대응성 유지
  - **확장성:** 동시에 다수 처리할 경우 자동으로 AWS가 관리하는 처리용 인스턴스가 시작되
    며 확장

- EC2의 장점 (Lambda와 비교하여)
  - 온프레미스의 응용 프로그램을 AWS로 이전하는 경우 OS 설정 등 그대로 사용 가능
  - 인스턴스 유형/OS/네트워크 등을 자유롭게 설정할 수 있는 **유연성**
  - 대량의 트래픽, 접속 상시 처리시 EC2 쪽이 저렴할 수 있음
  - 서버에 프로그램을 배포한다는 기존 방식으로 개발 진행 가능

2. 함수 생성 및 실행

- Lambda에서는 **함수라고 하는 단위**로 프로그램 코드 관리/처리
- 함수는 AWS 관리 콘솔 및 여러 도구로 생성 가능
- 별도의 환경을 준비하지 않고 쉽게 실행할 수 있다는 점이 Lambda의 장점
  단, 화면 상에서 편집 가능한 코드는 컴파일이 불필요한 일부 언어만 해당

JSON이란

1. **JSON(JavaScript Object Notation)**은 데이터를 표현하기 위한 **데이터 교환 형식**
2. 자바스크립트의 객체(데이터) 표기 방법에서 유래
3. { }안에 여러 개의 키-값 쌍을 정의
4. { } 안에 다시 { }를 계층적으로 포함하거나 [ ]를 이용해 배열 값을 넣을 수도 있음