



10.23 메모

🕒 Created	@2024년 10월 23일 오전 9:19
📁 Class	10.21 ~ 10.24 AWS 기초

AWS 기초 (S3, Lambda, RDS)

- **S3(Simple Storage Service):**
 - AWS의 객체 스토리지 서비스로, 대규모 데이터를 안전하게 저장하고 관리.
 - 버킷을 통해 데이터를 관리하며, **버킷 정책**이나 **ACL**로 접근 제어 가능.
- **Lambda:**
 - **서버리스 컴퓨팅 서비스**로, 다른 AWS 서비스와 연동되어 트리거를 통해 자동으로 코드 실행.
 - 다양한 언어 지원 (Python, Node.js, Go 등) 및 최대 15분 동안 실행 가능. 최대 10GB 메모리 제공.
- **RDS(Relational Database Service):**
 - AWS에서 제공하는 **관계형 데이터베이스 서비스**.
 - **MySQL, PostgreSQL, MariaDB** 등 여러 DB 엔진 지원.
 - **자동 백업과 스냅샷**, 자동 스케일링 기능 제공.

AWS 기초 (VPC)

- **VPC(Virtual Private Cloud):**
 - AWS 계정 전용 **가상 네트워크**로, 사용자가 정의한 프라이빗 네트워크 환경.
 - **EC2, RDS, Lambda** 등의 AWS 리소스를 VPC 안에 구축 가능.
- **Subnet:**
 - 네트워크를 분할하는 기술로, **퍼블릭 서브넷**과 **프라이빗 서브넷**으로 나누어 관리.
 - **퍼블릭 서브넷:** 외부 인터넷과 연결되는 네트워크.

- **프라이빗 서브넷**: 내부 통신을 위한 네트워크.
- **Gateway**:
 - **Internet Gateway**: VPC와 인터넷 간의 통신을 가능하게 하는 요소.
 - **NAT Gateway**: 프라이빗 서브넷의 인스턴스가 인터넷과 통신할 수 있도록 사설 IP를 공용 IP로 변환.
- **보안 그룹(Security Group)**:
 - 인스턴스 단위의 방화벽으로, **인바운드** 및 **아웃바운드** 트래픽을 제어하며 **Allow**만 설정 가능.
- **NACL(Network Access Control List)**:
 - 서브넷 단위의 방화벽으로, **Allow**와 **Deny** 규칙을 설정 가능.

boto3

- **boto3**: AWS에 접근하기 위한 **Python SDK**로, S3, EC2, DynamoDB 등 다양한 AWS 서비스와 상호작용 가능.
 - *AWS 리소스 호출(API)**을 간단하게 수행할 수 있으며, **자동화** 및 **스크립팅**에 활용 가능.
 - 정기적 백업이나 EC2 인스턴스 시작/중지 등을 코드 내에서 구현하여 **AWS 서비스 통합**과 **확장성**을 제공.
 - **IAM**과 연동되어 권한 인증 후 안전하게 접근.

Port:

- 네트워크에서 데이터가 컴퓨터에 들어오거나 나가는 **출입구**.
- 특정 서비스나 프로토콜을 **식별**할 수 있으며, 보안 목적으로 특정 포트를 열거나 닫아 **트래픽 제어** 가능.

네트워크 트래픽 흐름

- **라우터**: 우체국 역할.
- **포트**: 우편번호나 식별 번호와 비슷한 역할.
- **게이트웨이**: 우체국의 출입구처럼, 네트워크 데이터의 출입구 역할.
- **IP 주소**: 집 주소처럼 네트워크에서 데이터를 주고받는 위치.

AWS 아키텍처

- **AWS 서비스와 인프라** 구성에 대한 **청사진(설계도)**.
- **PPT**나 외부 도구를 사용하여 설계하며, **DevOps**와 **개발자**가 함께 작성하는 것이 이상적.
- **고가용성 디자인**: AWS의 VPC와 관련된 인프라 설계를 통해 고가용성을 달성.