

# Computing Service(1)

---

## 알아둬야 할 서버 기초 지식

### 1. 서버란

- **서버**: 데이터나 서비스를 제공하는 컴퓨터  
**클라이언트**: 그 서비스를 이용하는 프로그램
- 클라이언트: 서버에 데이터 요청  
서버: 클라이언트에 요청에 대한 응답 반환

### 2. 대표적인 서버 유형

- **웹 서버**
  - 웹 사이트를 구성하는 데 필요한 데이터 저장, 시스템 제어 프로그램을 설치한 서버
  - AWS에서 웹 서버 구축 시 EC2와 ECS로 구현하는 경우 많음
- **데이터베이스 서버**
  - 시스템이 취급하는 데이터를 통합 관리하는 데이터베이스 관리 시스템이 설치된 서버
  - 클라이언트에서 특정 데이터 참조/수정/삭제 등 데이터 처리 요구를 요청으로 받아 들이고, 실행 결과를 응답으로 반환
  - **AWS는 RDS나 DynamoDB와 같이 데이터베이스에 특화된 서비스 제공**
- **메일 서버**
  - SMTP 프로토콜, POP3 프로토콜을 이용해 메일 송신/전달/수신하는 서버
  - 일반적으로 역할에 따라 SMTP 서버와 POP3 서버로 나뉨
  - SMTP 서버: 메일을 보내는 역할을 하는 서버

- AWS에서는 EC2에 사용자가 메일 서버 구축 가능

### 3. 웹 서버 작동 방식

- 사용자 측의 웹 브라우저=클라이언트, 웹 서버=서버
- **브라우저-웹 서버**는 **HTTP, HTTPS**라는 프로토콜로 데이터 통신
- HTML을 중심으로, 사이트 자체의 디자인이나 페이지에서 실행되는 프로그램은 서버에 저장
- 웹 서버가 처리하는 주요 데이터

카테고리	데이터	설명
사이트 구성	HTML	- Hyper Text Markup Language - 웹 사이트 자체의 구조를 정의하는 코드, 단락이나 목록 구성, 이미지 파일 사용, 링크 작성 등 사이트의 뼈대가 되는 내용 기입
사이트 레이아웃 및 디자인	CSS	- Cascading Style Sheets - HTML로 정의된 콘텐츠에 스타일 디자인 적용/배치 가능 - 표시되는 문자의 글꼴/색/크기 지정, 애니메이션 효과 가능
스크립트 (서버)	PHP, Ruby 등	- 웹 서버에서 처리하는 프로그램, 요청에 따라 처리 결과를 클라이언트에 반환 - 웹 서버에서 처리되므로 결과는 클라이언트 환경에 의존X
스크립트 (클라이언트)	JavaScript	- 클라이언트(브라우저)에서 실행되는 프로그램 - 클라이언트가 처리하므로 서버 부하X - 클라이언트 환경에 따라 처리 속도와 결과 다를 수 있음
이미지	JPEG, GIF, PNG 등	- 사이트에 표시되는 이미지도 서버에 저장 - 데이터 압축 방식이나 기능에 따라 저장 형식 선택

### 4. 서버 OS란

- 서버에도 운영체제(OS)는 반드시 탑재
  - **OS:** 기기의 관리/제어 수행을 위한 **인터페이스/하드웨어 관리 기능**, 기기에서 동작할 소프트웨어가 공통적으로 이용할 **기본 기능 구현 소프트웨어**
- 리눅스

- 무료, 오픈소스 → 누구나 자유롭게 개발/배포 가능
- 다양한 기업과 단체가 기본이 되는 리눅스 커널을 이용해 추가 개발한 OS를 **리눅스 배포판**으로 제공 중

배포판 이름	설명
Red Hat Enterprise Linux (RHEL)	- 레드햇이 개발한 상용 리눅스 배포판 - 대규모 시스템 등의 서버에서 많이 이용 - 패키지 관리 시스템으로 RPM 이용
CentOS	- RHEL의 복제 OS - RHEL의 상용 부분을 제거한 리눅스 배포판
Debian GNU/Linux	- Debian 프로젝트에서 개발한 리눅스 배포판 - 패키지 관리 시스템으로 deb 이용
Ubuntu Linux	- Debian을 기반으로 만들어진 리눅스 배포판 - 주로 개인 용도로 많이 이용

- 윈도우 서버
  - 마이크로소프트에서 출시한 서버용 OS  
리눅스와 달리 OS 자체의 라이선스와 **CAL(Client Access License)**로 불리는 서버 이용 라이선스를 구매해야 함
  - **GUI(Graphical User Interface)**로 조작하며, 일반적으로 사용되는 데스크톱 윈도우와 사용법이 비슷, **CUI(Character User Interface)**로 조작하는 리눅스 배포판보다 사용 편리
  - 다른 마이크로소프트 제품과의 연계성, AD(Active Directory)와의 연계, 충실한 지원

#### ● OS별 비교

항목	리눅스	윈도우 서버
도입 비용	기본 무료 이용 * 배포판에 따라 라이선스로 있음	라이선스 수수료 외 CAL 비용 필요
필요 사양	상대적으로 낮은 사양에서도 동작	쾌적한 사용을 위해서는 고사양 필요
인터페이스	주로 명령줄에서 명령어 조작	주로 마우스로 조작가능한 인터페이스
전문 지식	명령어에 의한 조작이나 설정 파일 관리 등 전제 지식 필요	데스크톱 윈도우와 비슷하므로 필요 지식 상대적으로 적음

항목	리눅스	윈도우 서버
지원	기본 지원 없음 * 유료 배포판은 지원 있음	마이크로소프트 지원 시스템
기타	배포판/패키지 자유롭게 선택 가능 해 원하는 기능/특징 구현 가능	Active Directory/SQL 서버 같은 마이크로소프트 소프트웨어와 연계성 우수

## 5. 서버 가상화

- 서버의 가상화: 하나의 하드웨어에서 여러 OS 동작 가능  
→ **서버 가상화**: 하드웨어를 나눠 독립된 가상의 서버를 만드는 것
- OS 종류와 실행할 소프트웨어에 맞는 자원 할당을 고려해야 함

## 6. AWS에서의 가상화

- **Amazon EC2**(Amazon Elastic Compute Cloud): AWS의 대규모 서버에서 가상화 수행  
사용자는 용도에 따라 OS 종류, CPU/메모리 크기 선택해 인스턴스(가상 서버) 생성 가능

# EC2로 손쉬운 가상 서버 생성

## 1. Amazon EC2: 가상 서버 서비스

- Amazon Elastic Compute Cloud(이후 EC2)는 몇 분 만에 가상 서버 생성 가능한 서비스
- 온프레미스: 하드웨어 준비부터 OS 설치까지 전부 사용자 몫  
**EC2: 가상 머신 생성 시 OS 함께 설치 → 가상 머신 사용 준비 완료 시 바로 사용 가능**
- 서버 사양 자유롭게 선택 가능, 머신 생성 후에도 변경 가능  
→ 자원의 변경/삭제가 쉬워 부담없이 서버를 생성해 테스트 가능, 가용성 확보 설정 가능
- EC2: 가상 서버를 인스턴스 단위로 관리

→ 사용자가 인스턴스 유형(=가상 서버 사양)을 결정해 생성 시 인스턴스 유형/기간에 따라 요금 발생, 정지해두면 요금 발생X

## 2. 가상 서버 생성

- EC2 인스턴스 생성 시 설정해야하는 내용
  - **Amazon 머신 이미지(AMI)**
  - **인스턴스 사양(인스턴스 유형)**
  - **배포할 네트워크**
  - **데이터를 저장할 스토리지 용량**
  - **사용 권한 설정(보안 그룹)**
- Amazon 머신 이미지: OS와 소프트웨어가 설정된 템플릿, AWS에서 미리 준비한 AMI 제공
- 인스턴스 유형: 선택 시 가상 서버의 성능 결정, 생성 후 변경 가능
- VPC 선택 → 스토리지 용량(EBS) 설정 → 보안 그룹 선택
- 인스턴스 생성 이후 관리 시스템 기능을 이용해 연결 가능 (AWS Systems manager)

## 3. 인스턴스 유형으로 서버 성능 결정

- 인스턴스의 성능 → 인스턴스 유형 이름으로 알 수 있음
- 인스턴스 유형 이름 규칙 (ex. m6g.medium)
  - 메모리와 CPU 중 어느 것을 우선시 하느냐와 같이 인스턴스의 특징에 따른 문자열 존재
  - 세대 숫자가 클수록 성능이 좋음
  - Graviton2(CPU 이름)은 g, 네트워크 강화는 n 추가 기능 문자가 없는 경우도 있음
  - 크기가 클수록 고성능, 가격도 크기에 따라 높아짐
- EC2 인스턴스 패밀리

- T 유형 인스턴스는 베이스라인이라는 정해진 CPU 사용률을 초과해(버스트) 이용 가능

패밀리	특징
T	버스트 가능한 범용 유형
M	균형 잡힌 범용 유형
C	컴퓨팅 최적화 (vCPU 수가 많음)
R	메모리 최적화 (메모리 탑재량이 많음)
P	고속 컴퓨팅
I	스토리지 최적화

#### 4. 다양한 EC2 요금제

- 온디맨드 인스턴스: 일반적으로 사용하는 EC2 인스턴스 (이용 시간과 유형에 따라 요금 발생)
- 예약 인스턴스: 요금을 선불로 지불하고 이용
- 절감형 플랜(Savings Plans): 기간 약정 조건으로 할인받는 요금 모델
  - 예약 인스턴스에 비해 유연하게 요금 설계 가능
  - 예약 인스턴스에서는 인스턴스 패밀리와 크기 지정해야 함 but 절감형 플랜은 지정하지 않고 여러 패밀리나 크기를 할당해 구매할 수 있음
- 스팟 인스턴스: AWS 이용하지 않는 자원을 활용해 인스턴스를 생성/사용할 수 있는 모델
  - 용량 반환, 지정 비용 초과 시 AWS 측에서 중지 가능

구매 유형	특징	이용 용도
온디맨드	일반 구매 방법, 이용 상황에 따라 요금 변동	이용 상황에 변화가 있고 사용 예정일을 모르는 경우
예약, 절감형 플랜	1년/3년의 이용 요금 미리 지불, 온디맨드에 비해 최대 72% 할인	프로덕션 가동 등 1년 이상 사용 예정인 경우
스팟	이용하지 않는 부분을 이용 예기치 않게 인스턴스 정지 가능 온디맨드에 비해 최대 90% 할인	개발 환경과 같이 인스턴스를 중단해도 영향이 작은 경우

## 5. 그 밖의 EC2 관련 요금

- 네트워크 외부로의 데이터 통신, 데이터 저장 스토리지 등에서 요금 발생