# AWSstudy

1장 클라우드의 역할

1.1 시스템 인프라와 기존 시스템 인프라의 문제점

\* IT 기반의 비즈니스를 하려고 한다.

필요한 것?

애플리케이션 + IT 시스템(시스템 인프라)

* SI(System Integration, 시스템 통합

\* 시스템

* 기능적요건: 프로그래밍
* 비 기능적요건: 시스템 성능/신뢰성/확장성/유지보수/보안

애플리케이션 실행 환경

* + 1. 시스템 인프라

: 애플리케이션을 사용자가 365/24 이용할 수 있도록 환경을 뒷받침하는 기술 요소

H/W + OS + MW

그림 1-1(p.18)

* + 1. 시스템 인프라 보유의 문제점
* 고비용: 초기비용 높고, 유지보수 비용도 높다.
* 서버의 자원을 100% 활용하기 어렵다.
  1. 클라우드 시스템
* 시스템 인프라 문제점을 해결하기 위한 방편
* 시스템을 ‘보유’하는 것이 아닌 ‘필요할 때’만 시스템을 이용하고 ‘필요 없을 땐’ 시스템을 ‘반환’
* 인프라를 ‘서비스’로 제공
* ~~~ as a Service

: Iaas, InfraStructure as a Service

: Paas, Platform as a Service

: SaaS, SW as a Service

: AIaas, Artificial Intelligence as a Service

: MLaas, Machine Learning as a Service

\* 기업의 시스템 구성

- 온프레미스(on-premises)

: 회사내 자체 데이터 센터 구축, 시스템 구축/운용/유지보수 직접 수행

- 퍼블릭 클라우드

: 경제적

- 프라이빗 클라우드

: 보안 용어, 독자적 기능/서비스 추가 용이

1.2.2 클라우드 사용과 맞지 않는 경우

\* 매우 높은 가용성이 요구되는 경우

\* 저장 위치를 명확히 하지 않으면 안되는 데이터를 취급하는 경우

\* 특수한 요건이 있는 경우

\* 전체 비용이 높아질 경우

# CloudFirst

: 시스템 설계/구축/이행 때 클라우드 서비스 적용을 제일 먼저 고려/검토하는 방침

2장 AWS 기본과 계정등록

2.1 Amazone Web Service에서 제공하는 서비스

2.2 AWS 클라우드 디자인 패턴

\* 디자인 패턴: ~~할 때에는 ~~하게 하는 것이 좋다~

2.2.1 Multi-Server 패턴

2.2.2 Scale-up 패턴

2.2.3 DB Replication 패턴

2.2.4 Functional FireWall 패턴

2.3 AWS 데이터 센터

2.3.1 리전(Region)

2.3.2 AZ (Availability Zone)

\* 같은 AZ 내 통신

\* 같은 리전 내 통신

\* 다른 리전 간 통신

2.4 AWS 계정 등록 및 사용

\* AWS Free Tier

- 가입 후 12개월

- 무료 이용 한도

2.5 AWS 개발 도구

2.5.1 AWS 관리 콘솔 (AWS Management Console)

2.5.2 AWS 명령 라인 인터페이스 (AWS CLI)

2.5.3 AWS SDK

2.5.4 통합개발환경을 위한 플러그인

- AWS ToolKit for Eclipse

- AWS Toolkit for Visual Studio

3장 웹 서버 구축

3.1 웹의 구조와 HTTP 통신의 기본

3.1.1 웹 애플리케이션이란?

\* 웹 애플리케이션

- 서버에서 실행, 인스톨은 불필요, 어떤 플랫폼에서도 이용 가능하나, 네트워크가 필수이며 성능이 낮다.

\* 네이티브 애플리케이션

- 클라이언트에서 실행, 전용 마켓에서 다운로드 및 인스톨을 한다. 네트워크가 없어도 사용 가능하고, 프로세싱 성능이 높다. 그러나 플랫폼 마다 개발이 필수적이다.

3.1.2 웹 서버에 대한 Request & Response

3.1.3 웹 서버에 접속과 URL 서식

\* URL (Uniform Resource Locater)

Cf.) URI (Uniform Resource Indicator)

\* URL 서식 (p.83 리스트 참조)