

제01장

# On-Premise와 Cloud

DevOps

# 학습목표

1. 웹 개발 진행 과정을 이해할 수 있다.
2. OnPremise와 Cloud의 차이점을 이해할 수 있다.

# 목차

1. 웹 개발 진행 과정
2. OnPremise와 Cloud

```

each: function(e, t, n) {
    var r, i = 0,
        o = e.length,
        a = M(e);
    if (n) {
        if (a) {
            for (; o > i; i++)
                if (r = t.apply(e[i], n), r === !1) break;
        } else
            for (i in e)
                if (r = t.apply(e[i], n), r === !1) break;
    } else if (a) {
        for (; o > i; i++)
            if (r = t.call(e[i], e[i]), r === !1) break;
    } else
        for (i in e)
            if (r = t.call(e[i], e[i]), r === !1) break;
    return e;
},
trim: b && !b.call("\uffff\u00a0") ? function(e) {
    return null == e ? "" : b.call(e);
} : function(e) {
    return null == e ? "" : (e + "").replace(C, "");
},
makeArray: function(e, t) {
    var n = t || [];
    return null != e && (M(Object(e)) ? x.merge(n, "string"
),
isArray: function(e, t, n) {
    var r;
    if (t) {
        if (n) return m.call(t, e, n);
        for (r = t.length, r = r ? 0 > n ? Math.max(0, r + n) : n; n in t && t[n] === e) return n;
    }
}

```

## 01. 웹 개발 진행 과정

# 웹 개발 진행 과정

## ■ 웹 개발 진행 과정

1. 기획 및 설계
2. 개발 및 구현
3. 테스트 및 배포

# 1단계. 기획 및 설계

- 개발하고자 하는 웹 애플리케이션의 요구사항을 분석
  - 필요한 기능과 데이터 결정
- DBMS 선택
- 기술 스택 (개발 언어 및 프레임워크) 선택
- 서버 아키텍처, 데이터베이스 스키마 설계로 구조 정의

## 2단계. 개발 및 구현

- 선택한 DBMS로 데이터베이스 구축
- 선택한 기술 스택으로 비즈니스 로직 개발 및 API 설계 (API 문서화)
- 사용자의 보안과 인증, 권한 부여 등 시스템 구현

## 3단계. 테스트 및 배포

- 여러 테스트를 거쳐 서버 성능 및 쿼리 최적화
- 실제 운영 환경에 배포
- 사용자들이 접근하고 사용할 수 있도록 유지보수를 진행
  - 지속적으로 버그 수정 및 신규 기능 추가



# 배포 관련 용어

## ■ 컴파일

- Compile
- 개발 언어를 컴퓨터가 이해할 수 있는 언어로 변환하는 과정

## ■ 빌드

- Build
- 컴파일 된 코드를 실행할 수 있는 상태의 파일(jar 또는 war) 만드는 과정
- Maven, Gradle 같은 툴을 사용

## ■ 배포

- Deploy
- 빌드된 실행 파일을 사용자가 사용할 수 있도록 서버에 배치하는 과정
- 서버 운영체제는 주로 Linux이므로 Linux의 학습이 필요함

```

each: function(e, t, n) {
    var r, i = 0,
        o = e.length,
        a = M(e);
    if (n) {
        if (a) {
            for (; o > i; i++)
                if (r = t.apply(e[i], n), r === !1) break;
        } else
            for (i in e)
                if (r = t.apply(e[i], n), r === !1) break;
    } else if (a) {
        for (; o > i; i++)
            if (r = t.call(e[i], e[i]), r === !1) break;
    } else
        for (i in e)
            if (r = t.call(e[i], e[i]), r === !1) break;
    return e;
},
trim: b && !b.call("\uffff\u00a0") ? function(e) {
    return null == e ? "" : b.call(e);
} : function(e) {
    return null == e ? "" : (e + "").replace(C, "");
},
makeArray: function(e, t) {
    var n = t || [];
    return null != e && (M(Object(e)) ? x.merge(n, "string"
    ),
inArray: function(e, t, n) {
    var r;
    if (t) {
        if (n) return m.call(t, e, n);
        for (r = t.length, r = r ? 0 > n ? Math.max(0, r + n) : n; r < t.length; r++)
            if (n in t && t[n] === e) return n;
    }
}

```

## 02. OnPremise와 Cloud

# 서버 운영 방식

## ■ On-Premise

- 고전적인 서버 운영 방식
- 모든 장비(하드웨어, 소프트웨어, 네트워크)를 직접 운영하는 방식

## ■ Cloud

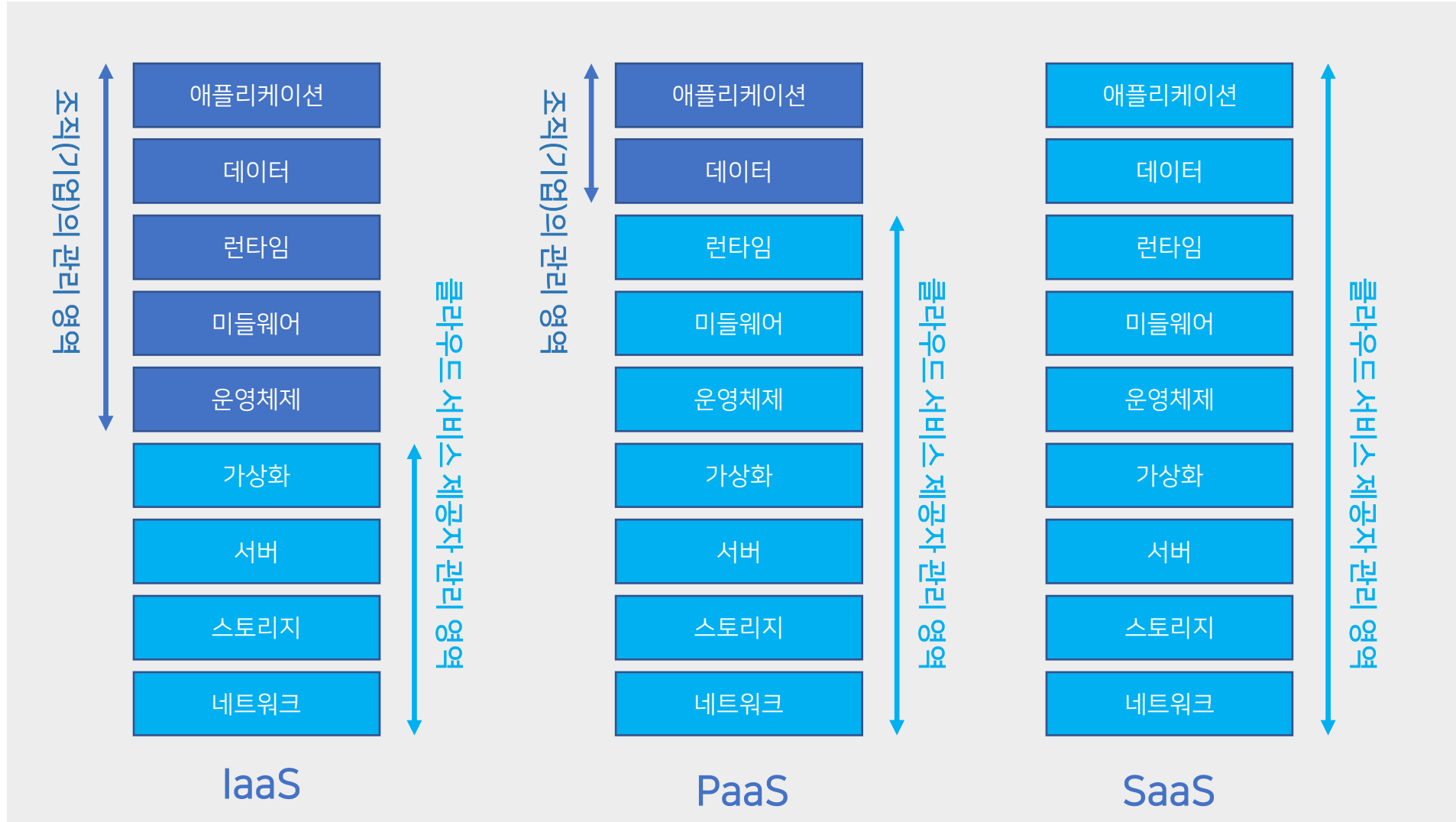
- 시스템 운영에 필요한 장비를 임대하는 방식
- 클라우드 서비스를 지원하는 업체를 이용함 (AWS, NPC, OCI 등)

# 서버 운영 방식의 차이

## ■ On-Premise VS Cloud

구분	장점	단점
ON-PREMISE	<ul style="list-style-type: none"><li>- 커스터 마이징이 쉬움</li><li>- 보안(기밀성)이 좋음</li></ul>	<ul style="list-style-type: none"><li>- 초기비용 및 관리비용 부담</li><li>- 인프라 증설이 쉽지 않음</li></ul>
CLOUD	<ul style="list-style-type: none"><li>- 초기 도입 비용이 없음</li><li>- 서비스를 빠르게 제공</li><li>- 인프라 증설이 자유로움 (이용료만내면 됨)</li></ul>	<ul style="list-style-type: none"><li>- 커스터 마이징이 어려움</li><li>- 보안(기밀성)이 취약</li></ul>

# Cloud 서비스



## ■ IaaS

- Infrastructure as a Service
- IT 서비스를 제공하기 위한 주요 인프라 자원을 공유 자원 형태로 관리하고 이를 나눠서 제공
- 서버, 하드웨어, 네트워크, 스토리지 등을 서비스로 제공
- 물리적인 기반 장치를 가상화해서 제공, 하드웨어만 제공
- 서비스 이용자는 해당 인프라 위에 자원(Resource)을 할당 받아 OS와 미들웨어(DB, WAS)를 설치하여 서비스를 이용할 수 있음
- AWS EC2, Microsoft Azure 등 가상 머신
- 블록 스토리지, 파일 스토리지, 오브젝트 스토리지 같은 각종 스토리지 서비스

## ■ PaaS

- Platform as a Service
- 애플리케이션을 실행하는데 필요한 플랫폼을 서비스로 제공
- 물리적 장비에 추가로 OS, 미들웨어, 개발도구 등을 제공
- 비즈니스 분석/인텔리전스, 데이터베이스, 사물인터넷(IoT), 인공지능(AI) 서비스 제공
- AWS Elastic Beanstalk, Google 앱엔진 등

## ■ SaaS

- Software as a Service
- 소프트웨어를 설치 없이 사용할 수 있는 애플리케이션을 서비스로 제공
- 사용자는 소프트웨어를 직접 구매해 설치할 필요가 없음
- 소프트웨어의 업데이트 작업은 클라우드 사업자가 처리
- 메일 서비스, Dropbox, Google Docs 등



# Cloud 기반 기술

## ■ 클라우드 컴퓨팅 아키텍처

- 물리적 시스템, 가상화 계층, 프로비저닝 계층, 서비스 관리체계, 서비스 계층의 5계층으로 구분



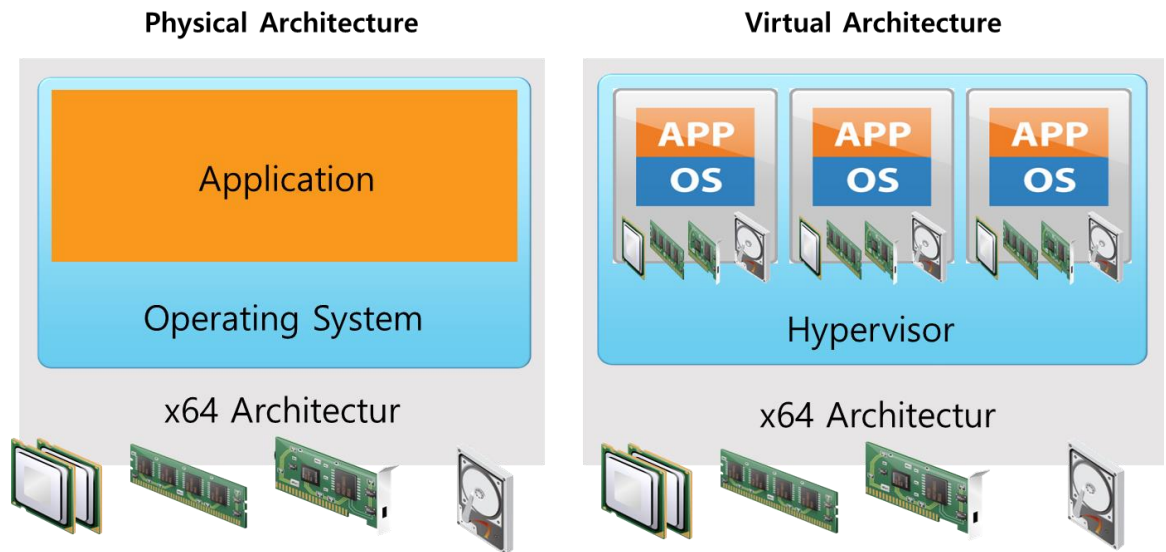
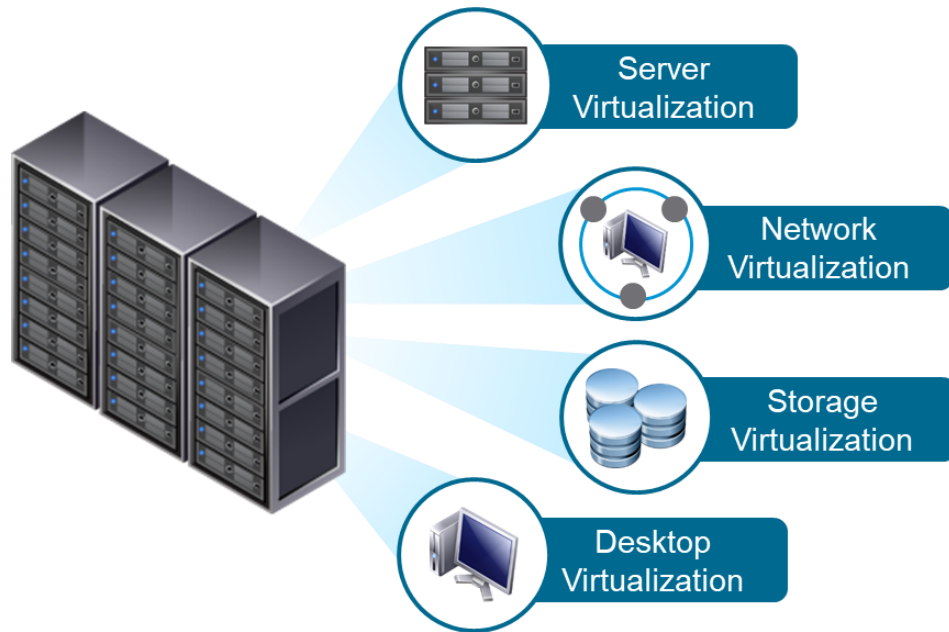
- 가상화 : 물리적 구성은 은폐하고, 가상환경을 여러 개 만들어, 개별 가상 머신이 자원을 갖추고 있게 만드는 기술
- 프로비저닝 : IT인프라 자원을 사용자의 니즈에 맞게 할당/배치/배포해서 시스템을 사용할 수 있도록 하는 기술
- 하이퍼바이저 : 물리서버 위에 존재하는 가상화 계층으로서, OS구동을 위한 HW환경을 가상으로 제공
- 분산 파일 시스템 : 막대한 양의 데이터를 저장 및 관리하기 위해 수많은 서버에 나누어 저장하는 파일 시스템

출처: <http://www.cloud.or.kr/>

# Cloud 기반 기술

## ■ 하이퍼바이저를 이용한 가상화

- 서버, 네트워크, 저장장치 같은 물리적인 것들을 소프트웨어 기반으로 표현하는 프로세서
- 비즈니스 규모에 대한 효율성과 민첩성을 높이면서 IT 비용을 줄이는 가장 효과적인 방법



출처: VMware