



# 준비된 애플리케이션 환경 Java Cloud Service

Java Cloud Service는 자바 기반 애플리케이션을 위해 “플랫폼”을 제공하는 클라우드 서비스이며, 오라클이 제공하는 PaaS(Platform as a Service) 클라우드 중에서 기반이 되는 서비스이기도 하다. 이 섹션에서는 Java Cloud Service의 개요와 활용법 및 이를 통해서 변화되는 인프라 구축 과정 및 관리 운영 상의 이점이 무엇인지에 대하여 소개하고자 한다.



### 이미남 부장

한국오라클

Middleware본부 Technology & Product팀  
(mee-nam.lee@oracle.com)

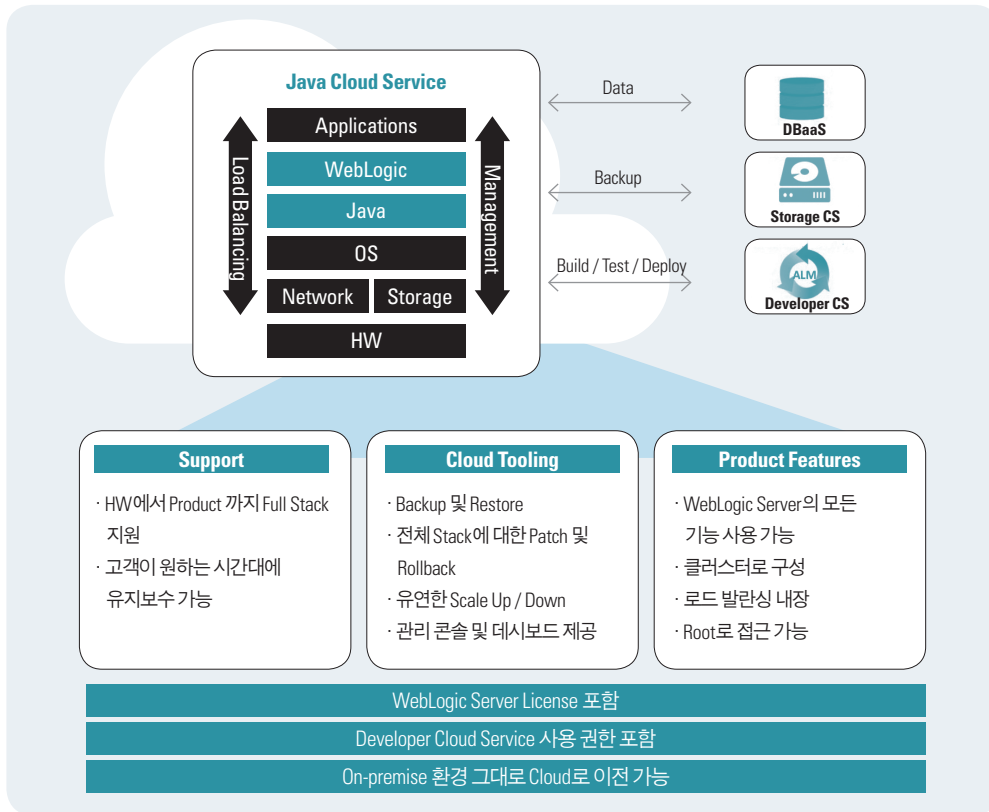


**M** 현재 한국오라클 Middleware본부 Technology & Product팀의 이미남 부장은 Principal Sales Consultant로 재직중이다.

이미남 부장은 현대정보기술, BEA Systems Korea에 재직하였으며 다양한 프로젝트를 수행해왔다.

## 1 Java Cloud Service가 무엇이며 어디에 어떻게 활용할 수 있을 것인가

Java Cloud Service는 오라클이 제공하고 있는 IaaS(Infrastructure as a Service), PaaS (Platform as a Service), SaaS(Software as a Service) 중 PaaS에 속하는 서비스 중 하나이다. PaaS가 애플리케이션을 위한 플랫폼을 서비스로 제공하는 형태이기 때문에 당연히 Java Cloud Service로 애플리케이션을 위한 플랫폼을 제공하지만 그 이름에서부터 알 수 있듯이 자바 기반의 애플리케이션을 위해 만들어진 플랫폼 서비스이다.



〈그림1〉 Java Cloud Service 개요

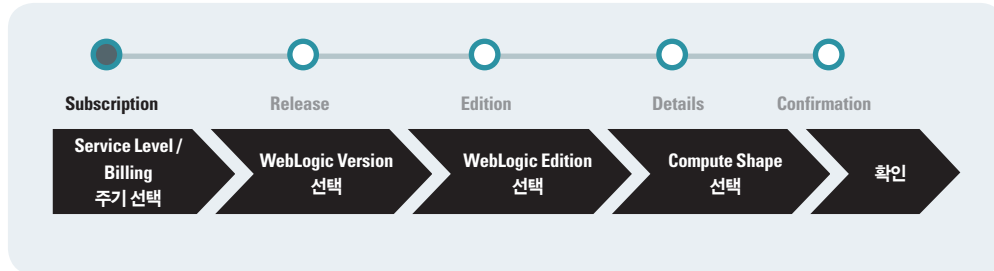
## 2 즉시 사용 가능한 준비된 WebLogic 환경

Java Cloud Service는 IaaS 기반에 Java와 WebLogic Server를 올려 순수 자바 애플리케이션뿐만 아니라 Java EE 애플리케이션을 위한 런타임 환경을 제공한다. Java Cloud Service내의 WebLogic Server는 즉시 사용 가능 하도록 클러스터 환경 및 로드발란서 기능이 모두 구성되어 있기 때문에, 별도의 추가적인 작업 없이 바로 여러분의 애플리케이션을 배포한 후 서비스 하면 된다.

On-premise와 동일한 애플리케이션 런타임, 동일 아키텍처를 사용하기 때문에, 애플리케이션 개발 시에 개발된 애플리케이션이 어디에 배포될 것인가를 염두에 둘 필요가 없다. 운영환경이 결정되고 나서 개발된 애플리케이션을 아무런 수정 없이 배포하기만 하면 된다. 운영 환경 선택이 자유로워지고, 클라우드로의 이전이 단지 선택 사항 중 하나로 단순화 되는 것이다.

### 3 서비스 환경 프로비전

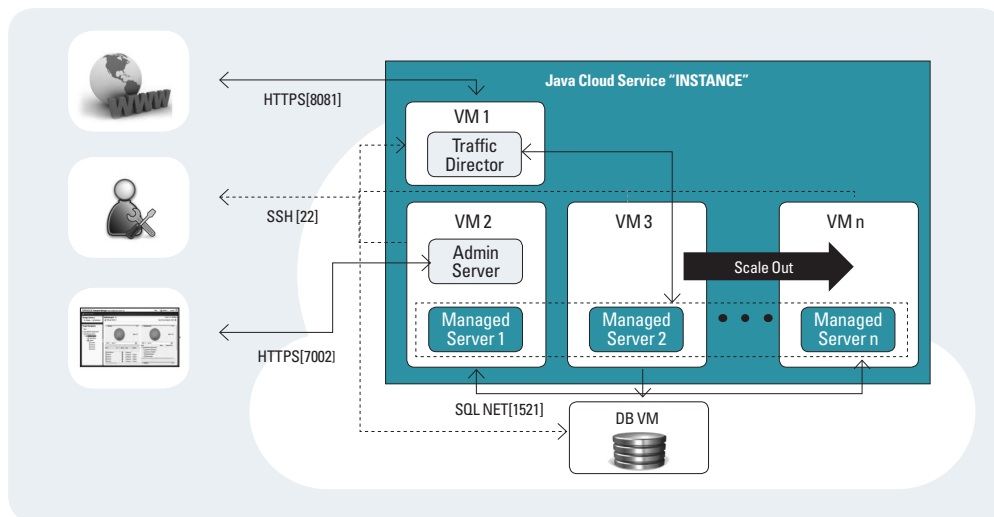
Java Cloud Service 환경을 생성하기 위해서는 다음과 같은 프로비전 단계를 거친다. 아래 표에서 보듯이 단지 “5번의 클릭”을 거쳐 서비스 환경이 생성된다. 생성 과정에서 용도에 맞는 웹로직 서버 버전 및 에디션, 그리고 사용할 자원의 크기를 선택할 수 있다.



〈그림2〉 Java Cloud Service 프로비전 단계

WebLogic Edition 선택 단계에서 “Enterprise Edition with Coherence”를 선택하게 되면 Coherence Data Tier가 포함된 인스턴스가 생성 되는데, 이를 통해 애플리케이션과 데이터를 분리 하여 데이터 계층을 별도로 확장하거나, 백엔드 계층에 대한 부하 경감 용도로 활용할 수 있다.

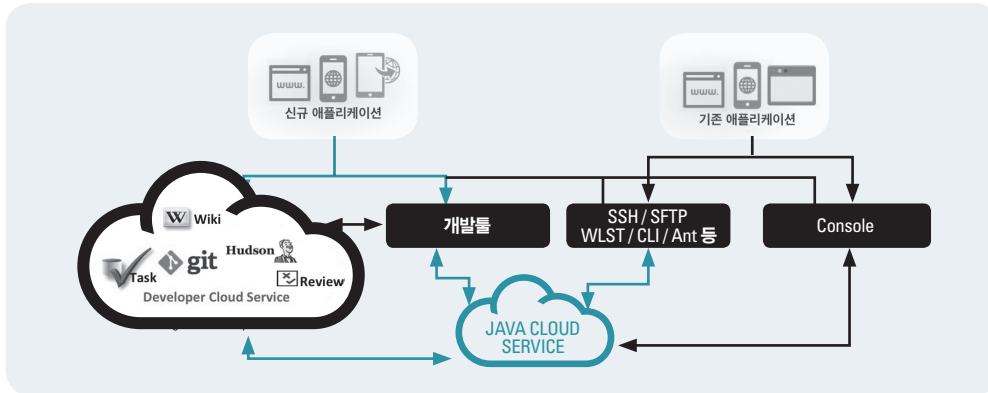
프로비전이 되고 나면 선택한 클러스터 크기, 사용 자원의 단위 및 로드발란서 사용 여부에 따른 웹로직 도메인 환경이 구성된다. 웹로직의 각 Managed Server는 각기 독립된 가상 머신에 구성되고 클러스터의 구성원이 된다. 각 가상 머신은 SSH를 통해 접속되고 웹로직 관리 콘솔 및 트래픽 디렉터(로드 발란서) 콘솔도 HTTPS를 통해 접속 되게 열려있다. 로드발란서 사용 옵션을 선택하였다면 서비스 요청은 로드발란서를 통해 각 Managed Server로 전달된다. Managed Server는 동적으로 Scale Out 또는 Scale In 될 수 있다.



〈그림3〉 Java Cloud Service 구성도

#### 4 애플리케이션 배포

Java Cloud Service는 신규 애플리케이션을 서비스 하거나 기존 On-premise 애플리케이션의 클라우드 전환 용도로 사용될 수 있다. 기존 애플리케이션은 On-premise 환경에서 사용하던 웹로직 콘솔이나 WLST(WebLogic Scripting Tool), CLI(Command Line Interface) 등 기존 경험 그대로 배포할 수 있도록 모든 툴들이 열려 있다. SSH 및 SFTP를 통한 접근도 가능하기 때문에 기 구축된 배포 환경의 대상만 변경함으로써 기존 환경을 그대로 이용할 수도 있다.



〈그림4〉 Java Cloud Service 배포 옵션

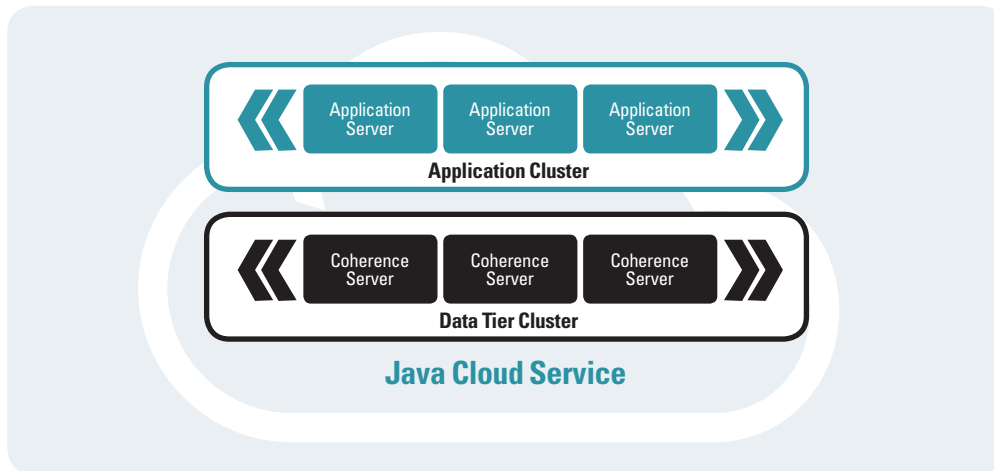
신규 애플리케이션 개발자들을 위해서는 더 쉽고 편하게 Java Cloud Service에 배포할 수 있도록, Java Cloud Service 플러그인이 포함된 개발 툴이 제공한다. 이를 통해 Java Cloud Service 사용자들에게 “무상으로 제공되는” 팀 개발 플랫폼인 Developer Cloud Service에 접속하여 애플리케이션 프로젝트를 다운받고 Task를 할당 받으며, 개발 소스를 컨트롤 하고 자동 빌드하여 Java Cloud Service로 배포할 수 있게 된다.

#### 5 데이터 그리드 계층 사용

프로비전 과정에서 언급한 대로 WebLogic Edition 선택 단계에서 “Enterprise Edition with Coherence”를 선택하게 되면 Coherence Data Tier가 포함된 인스턴스가 생성된다. 이 데이터 계층은 별도의 가상 머신에 생성되어 애플리케이션 계층과 별도로 확장 및 축소가 가능하다.

데이터 그리드를 포함한 Java Cloud Service 인스턴스를 통해 데이터베이스나 패키지 애플리케이션, 타 클라우드 애플리케이션 등의 백엔드 서비스 응답 및 데이터를 캐싱하여 백엔드 계층에 대한 부하 경감용으로 활용하거나, 애플리케이션에서 생성되는 세션 정보를 분리된 데이터 그리드 내에 저장함으로써 애플리케이션 계층의 메모리 사용률을 낮춰 궁극적으로 더 많은 사용자 요청을 처리하도록 구성하는데 활용할 수 있다.

Coherence Data Tier는 Java Cloud Service 인스턴스 도메인 내에 캐시를 사용하도록 설정된 웹로직 Managed Server로 만들어지며 웹로직 콘솔을 통해 관리된다.



〈그림5〉 데이터 그리드가 포함된 Java Cloud Service

## 6 관리 및 운영

Java Cloud Service는 인스턴스 환경을 구성하는 가상 환경, 스토리지 볼륨, 네트워크 환경, 웹로직 도메인과 클러스터 등을 제어하고 관리하는 클라우드 포털을 제공한다. Java Cloud Service 관리자는 인스턴스 시스템 환경에 직접 접근하지 않고도 웹 인터페이스를 통해 제공되는 관리 기능을 통해 서비스 상태를 모니터링 하거나 시스템을 백업 및 복구, 패치 하는 등의 관리 작업을 간편히 수행할 수 있다.



### ·모니터링

클라우드 포털을 통해 서비스 사용 현황 및 서비스 응답시간, 서비스 상태 등을 모니터링 하고, 각 웹로직 서버 별, 애플리케이션 별 상세 내용은 웹로직 콘솔이나 퓨전 미들웨어 콘솔을 통해서 모니터링 하도록 하고 있다.



### ·백업 및 복구

정기적으로 주 1회 전체 백업과 매일 증분 백업이 이루어지고 있으며, On-demand 방식으로도 백업을 수행할 수 있다. 백업된 데이터는 7일 동안은 가상 머신에 마운트 되어 있는 블록 스토리지에 저장되며, 이후에는 서비스 인스턴스에 연결된 스토리지 클라우드 서비스 컨테이너로 옮겨져 보관된다. Java Cloud Service의 백업은 바이너리, 구성정보 등을 포함한 전체 백업 및 도메인 구성만 백업하는 증분 백업으로 나누어 수행 할 수 있으며, 백업 시 인스턴스와 연관된 데이터베이스와 함께 백업 할 수 있다는 것이 장점이다.



### ·패치 및 롤백

Java Cloud Service를 구성하는 각 계층에 속한 패치를 자동 권고해주고, 클라우드 포털을 통해 패치를 확인하고 적용할 수 있도록 한다. 패치 적용 시 서비스 다운타임이 발생하지 않도록 롤링으로 패치를 반영하며, 반영된 패치는 유사시 롤백 할 수 있도록 하고 있다.



### · 확장 및 축소

부하에 따른 노드 추가 및 제거 또한 클라우드 포털을 통해 이루어지며, 노드 추가/제거 시 웹로직 클러스터 및 로드밸런서 구성에 자동 반영이 된다. 확장 및 축소는 애플리케이션 계층과 데이터 그리드 계층이 독립적으로 이루어진다.

## 6 왜 Java Cloud Service여야 하는가?

### 배포 선택권

On-premise와 동일한 아키텍처, 동일한 소프트웨어들로 구성되어 있기 때문에, 개발 시점에 배포 위치를 고려한 설계 및 개발이 되어질 필요가 없다. 애플리케이션 코드의 변경 없이 클라우드로의 배포가 가능하기 때문에 배포는 이제 단지 운영 이관 시점 선택의 문제일 뿐이다.

### 표준 기반의 소프트웨어 기반

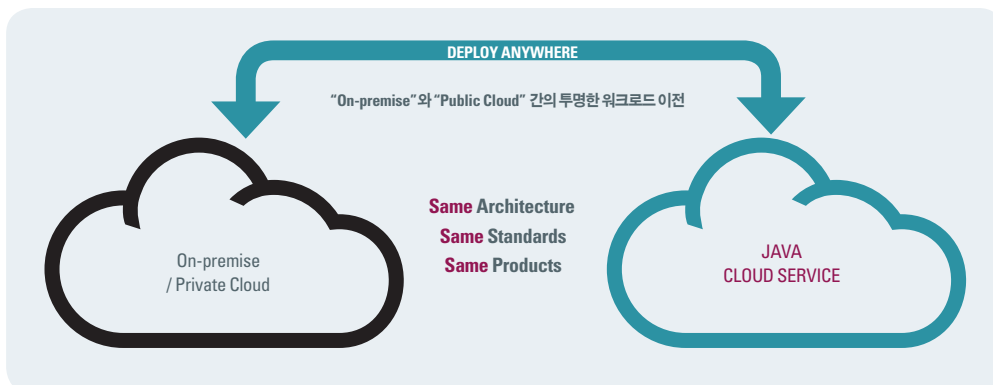
표준 기반의 소프트웨어로 구성되어 있으며, 기반의 플랫폼 인프라로써 마켓 #1 미들웨어와 #1데이터 베이스를 사용한다. 이런 상용 소프트웨어의 모든 기능이 제약 없이 클라우드 상에서 이용 가능하다.

### 완벽한 접근 제어와 관리

루트 권한으로 가상 머신에 접근이 가능하며, 패치 및 백업, 업그레이드 등의 관리 작업과 네트워크 설정에 이르기까지의 모든 제어와 관리 기능을 허용하기 때문에 On-premise와 거의 유사한 제어권을 확보할 수 있다.

### SaaS와의 통합

오라클의 클라우드는 IaaS, PaaS, SaaS가 완벽하게 통합되어 있기 때문에, 어떤 클라우드 서비스를 사용하더라도 확장이 용이하고, SaaS 클라우드를 커스터마이징하고자 할 때 동일 기술, 동일 아키텍처를 사용하는 Java Cloud Service를 활용하여 구현을 단순화 시킬 수 있다.



<그림6> 배포 선택권을 제공하는 Java Cloud Service

## 무상 Developer Cloud Service 제공

팀 개발 플랫폼인 Developer Cloud Service의 사용 권한이 부여된다. 이를 통해 Task 관리, 소스 코드 버전 관리, 빌드 자동화, 위키 등의 팀 개발 환경을 무상으로 사용할 수 있게 된다.

## 7 활용하기

Java Cloud Service는 웹 기반의 플랫폼이 필요한 모든 애플리케이션에서 활용 될 수 있다. 기존 On-premise 업무를 클라우드로 이관 하거나 클라우드로 확장하여 하이브리드 형태로 구성하는데 활용 될 수 있다. 또한 새롭게 시작하는 비즈니스의 초기 고정 투자비를 낮추거나 전세계 서비스를 목적으로 바로 클라우드로 시작되는 업무 등에 사용될 수도 있다. 오라클 클라우드는 운영 작업 부하를 안정적으로 서비스 할 수 있게 설계 되어 있지만, 아직까지 클라우드 상에서의 서비스 운영이 새로운 도전이라고 생각된다면 클라우드 운영 및 관리 경험 축적을 위해서라도 개발, 테스트 먼저 시작해 보는 것도 한 방법이 될 것이다.

## 8 경험해 보기



오라클은 Java Cloud Service의 무료 시험 서비스를 제공하고 있다. 오라클 클라우드 홈페이지([cloud.oracle.com](https://cloud.oracle.com))에서 계정 등록을 하고 시험 서비스를 요청하면 30일간 이용할 수 있는 무료 Java Cloud Service를 획득하게 된다. 이 시험판 Java Cloud Service에는 Oracle Database Cloud Service와 Oracle Storage Cloud Service를 포함하고 있기 때문에, 애플리케이션 배포 및 관리 및 운영의 모든 기능을 경험해 볼 수 있을 것이다.

### 유용한 자료

오라클 클라우드 홈페이지 <https://cloud.oracle.com>

오라클 클라우드 블로그 <https://blogs.oracle.com/cloud>

Java Cloud Service Get Started: [http://docs.oracle.com/cloud/latest/jcs\\_gs/index.html](http://docs.oracle.com/cloud/latest/jcs_gs/index.html)