

RED HAT Migration Toolkit for Applications

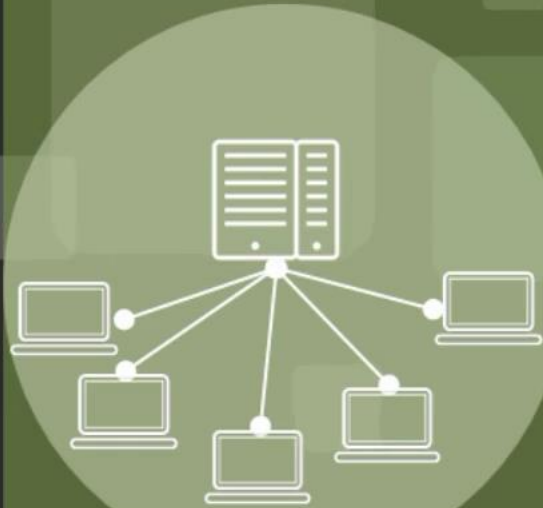
1960

1990

2000

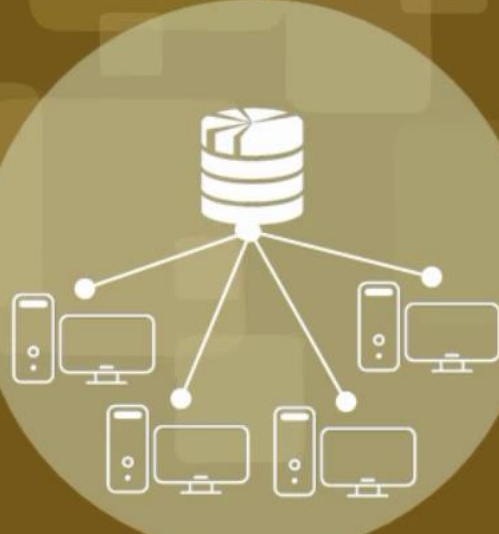
2010

Mainframe



Mainframe
Vendor

Client/Server



2 Tier
Unix (Vendor)
Scale Up

Web



3 Tier
Linux
Scale out

Cloud



IOT/Big Data
Mobile
Elastic Scaling

Development Process



WATERFALL



AGILE



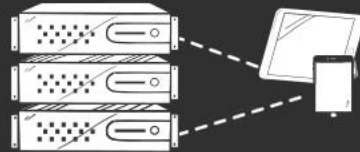
DEVOPS



Application Architecture



MONOLITHIC



N-TIER



MICROSERVICES



Deployment & Packaging



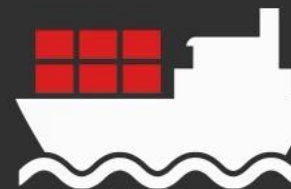
PHYSICAL SERVERS



VIRTUAL SERVERS



CONTAINERS



Application Infrastructure



DATA CENTER



HOSTED



CLOUD



3 티어 아키텍처 – 2000년대

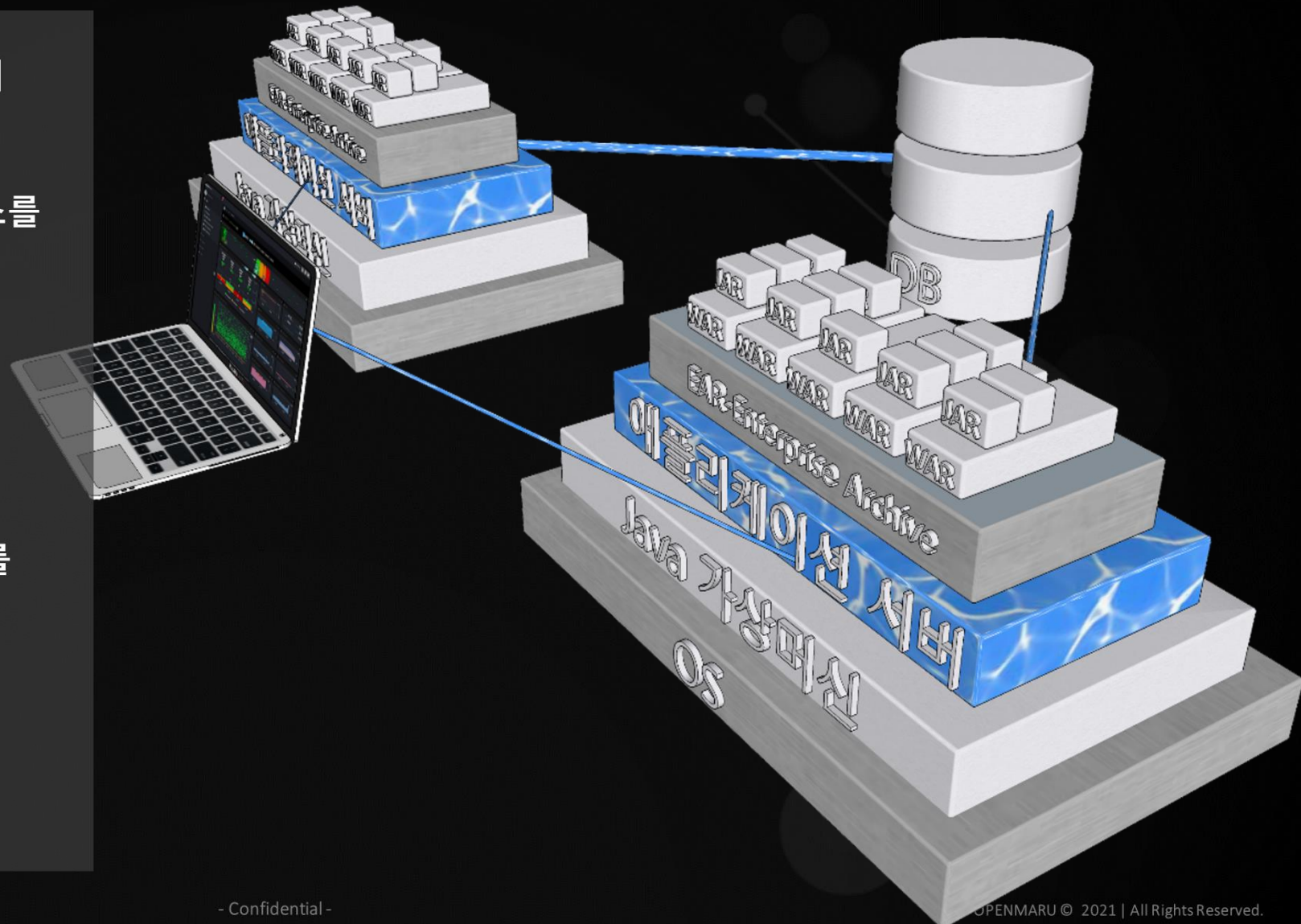
클라이언트는 주로 웹 브라우저를
사용하며, 서블릿이나 JSP 를 사용하여
서버 구현

H/W와 N/W자원을 효율적으로
사용하기 위하여 App Server 인스턴스를
최소화하고 세션정보도 최소화

데이터 내용과 중요도에 관계없이
주요 저장소로 RDBMS를 사용

아키텍처 이슈

1. 인스턴스가 적기 때문에 장애의
영향이 큼
2. Hot Deploy 등을 사용하여 서버를
중지하지 않고 운영하는 구조
3. 많은 데이터가 RDBMS에 저장되기
때문에 RDBMS 튜닝에 의존
4. 확장하는 것이 거의 불가능



4티어 아키텍처 – 2010년 이후

멀티 디바이스와 옴니 채널 대응을 위한 REST가 주류

H/W와 N/W자원이 저렴해져 서버 인스턴스를 여러 대 구성

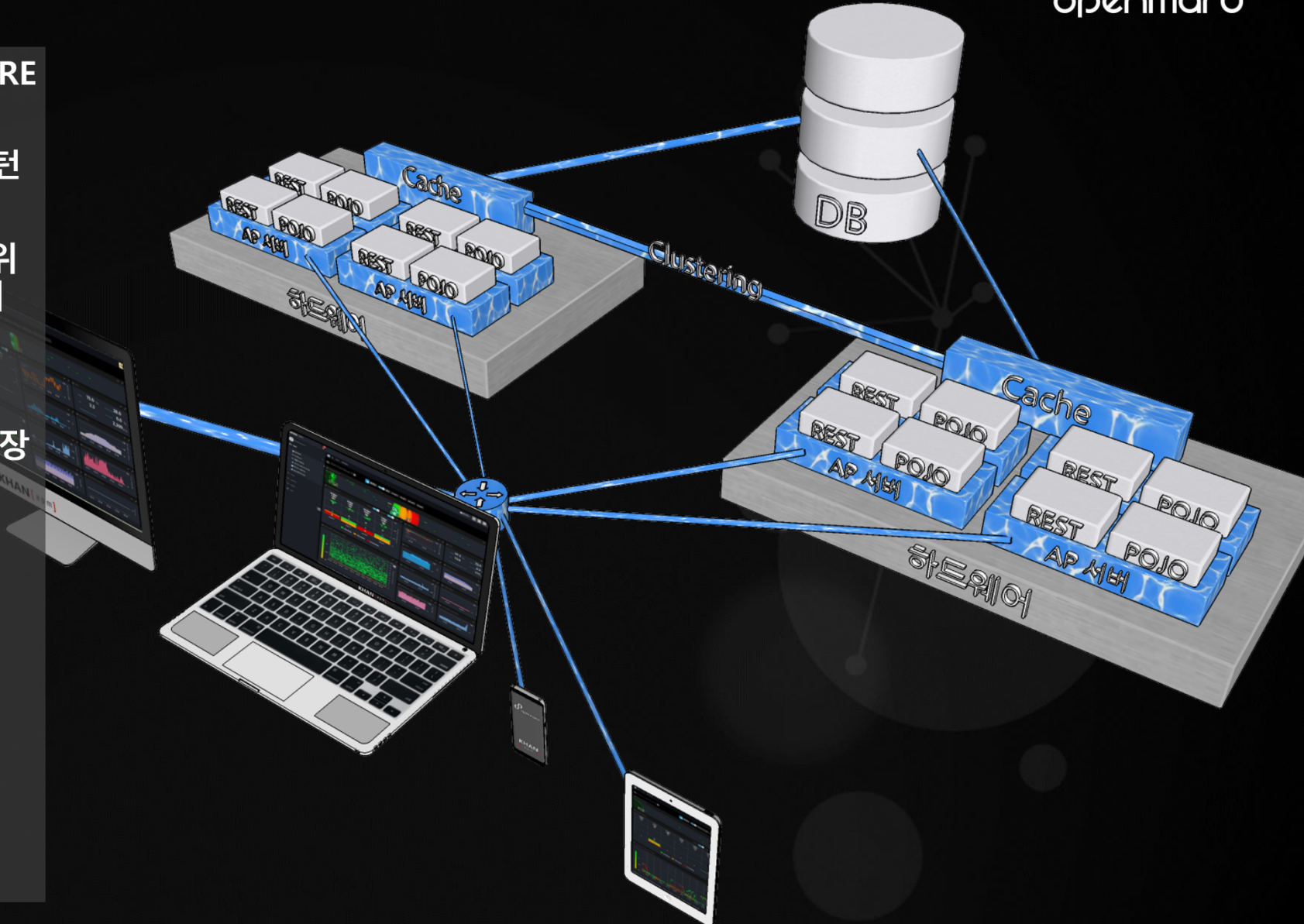
1. 인스턴스가 많기 때문에 장애 범위가 작고, 서버 장애는 서비스 중지가 아님

세션 정보는 기본 캐시 서버에 저장

데이터 내용 및 중요도에 따라 정보의 저장 위치를 선택

확장 가능한 아키텍처

관리 오버헤드 발생



Digital Transformation

웹시스템 운영 현실

Linux vs. Unix

- Unix 와 Linux는 기술적 차이를 가리기 힘든 상황
- 최근에는 벤더 종속성 없는 기술 저변과 오픈 소스 기반으로 시스템 구축이 대세
- Linux Base 플랫폼이 웹시스템을 주도하는 플랫폼으로 전환하는 추세

비교 항목	Unix	X86 (Linux)
신기술 적용 용이성	<ul style="list-style-type: none"> • 해당 벤더의 H/W, S/W 상의 UNIX만 지원 • 과거의 주요 웹시스템은 Unix에서 운영 	<ul style="list-style-type: none"> • 여러 벤더의 다양한 H/W 플랫폼을 폭넓게 지원 • X86은 오픈소스 S/W 로 구성하는 경향
웹시스템 기술 트렌드	<ul style="list-style-type: none"> • 최근에는 Unix 기반의 기술이 이전 보다 많이 뒤쳐짐 	<ul style="list-style-type: none"> • 웹시스템은 Unix 보다 X86(Linux)로 구성하는 경향 • 최신 IT 기술들은 오픈소스 기반으로 발전
벤더 의존성	<ul style="list-style-type: none"> • Unix 기반의 H/W와 S/W 가 특정 벤더에 종속적인 구성 	<ul style="list-style-type: none"> • 특정 업체가 전체 시장을 컨트롤 하지 않고 상호 경쟁하는 상황 • Server, OS, DBMS 등 다양한 제품과 기술지원 채널 선택이 가능
비용	<ul style="list-style-type: none"> • 벤더를 통한 새로운 CPU와 서버 교체 주기가 2~4 년 소요 • 성능과 비용을 위한 신기술 적용이 제한적이며 고비용 구조 	<ul style="list-style-type: none"> • 신기술 발표 및 적용 주기가 1~2년 이내 • 다양한 성능 및 비용 절감을 위한 신기술 적용이 용이하며 저비용 구조

Unix2Linux – 리눅스 기반 시스템 구축 원칙 수립

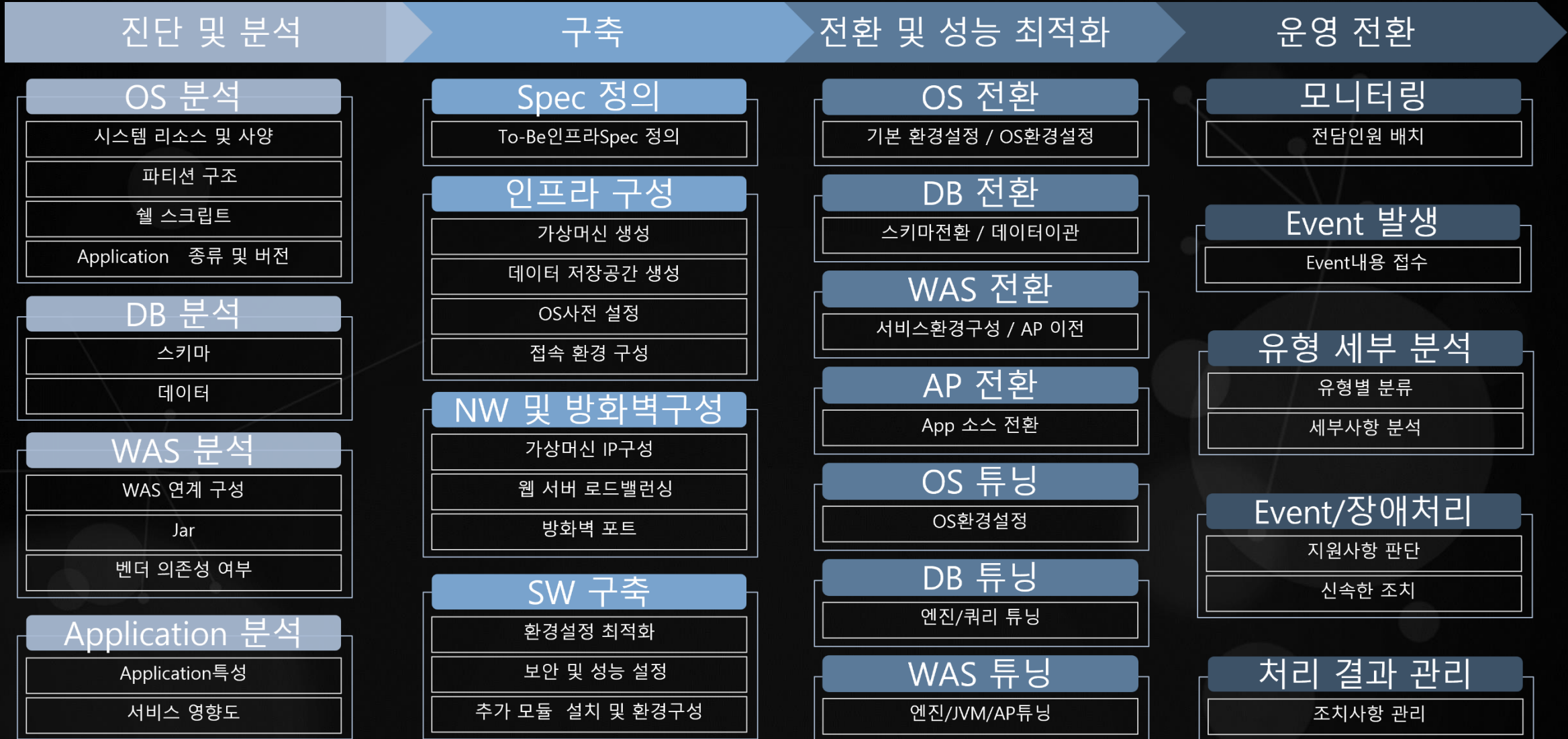


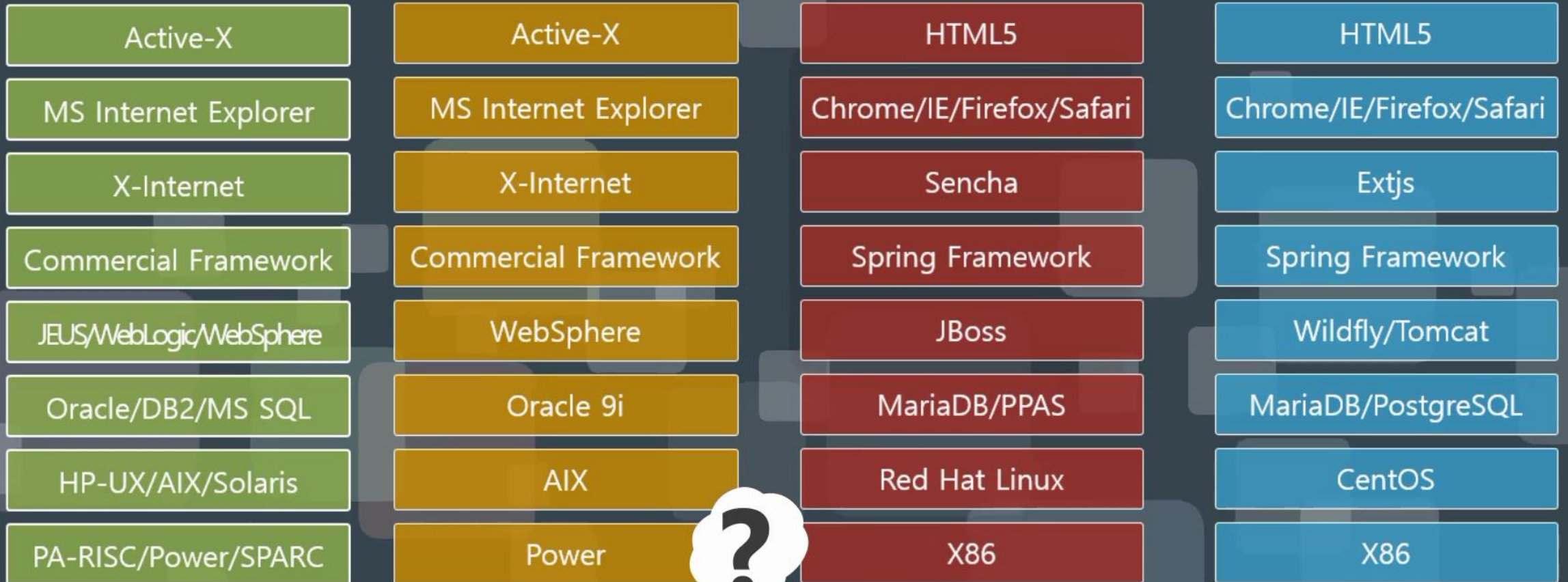
기존 Unix 환경에서의 운영상 이슈 사항 도출

- Unix High End 장비는 벤더 종속적으로 전문 Engineer에 의한 고비용 유지보수 발생
- OS가 장비의 CPU Chip에 의존적으로 새로운 장비 도입 시 반드시 개발 장비와 상용 장비 일치 작업 필요
- Unix 가상화를 위한 VM 기반 서버 집적화가 가능한 High End 장비는 매우 비쌈(동일한 장비의 Pool 형성 필요)

이슈 해결이 가능한 U2L 전환의 원칙 수립

HW 측면	<ul style="list-style-type: none"> • 시장에서 싸고 쉽게 구할 수 있는 HW 활용(ex. x86) • 고비용의 서버 용량 증설 방안 대신 저비용 서버의 수량 증설 기반 용량 확대 추진(Scale Out)
SW 측면	<ul style="list-style-type: none"> • 오픈 OS 기반으로 특정 벤더 종속 이슈 탈피 • Open Source SW 구현으로 Echo 시스템 확대 및 Linux 호환성 제공 • DB는 동일 버전 전환 기준으로 버전 Upgrade 데이터량에 따른 전환 일정 별도 산정
O&M 측면	<ul style="list-style-type: none"> • 비용효율성 및 오픈형 HW/SW 도입으로 유지보수가 용이한 운영 환경 수립 • 향후 가상화/클라우드/컨테이너 운영 환경을 대비한 자동화된 운영 방안 내제화 • 기존 Unix 일괄 전환 보다 신규 서비스 중심 Unix 도입 억제 및 Linux x86 장비 도입 유도





독점
소프트웨어

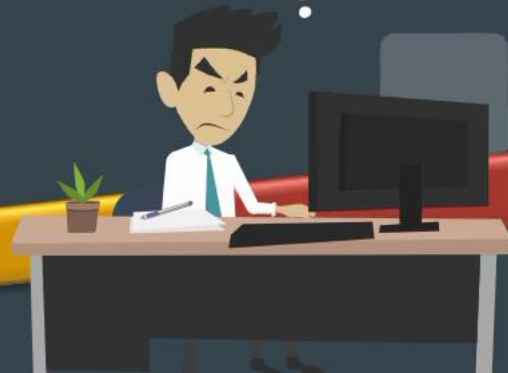
현재 구성

유상
오픈소스

무상
오픈소스



물리 환경



Private/Public
Cloud



벤더 종속적 시스템

액티브 X

WAS

UNIX

벤더 H/W

물리서버

개방형 표준 시스템

HTML5

 **Red Hat**
JBoss Enterprise
Application Platform

 **Red Hat**
Enterprise
Linux

X86

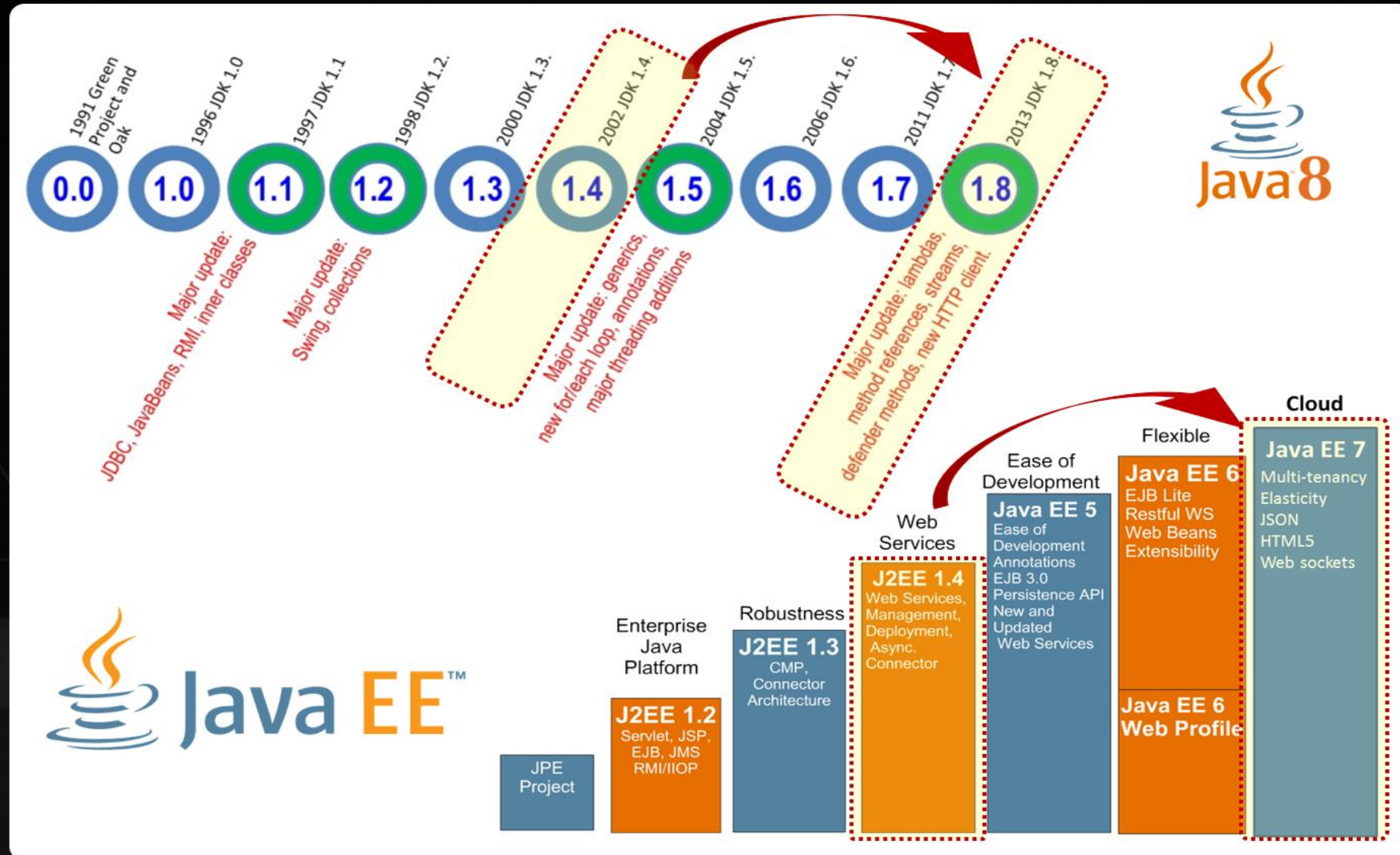
가상화/클라우드/컨테이너



Digital Transformation

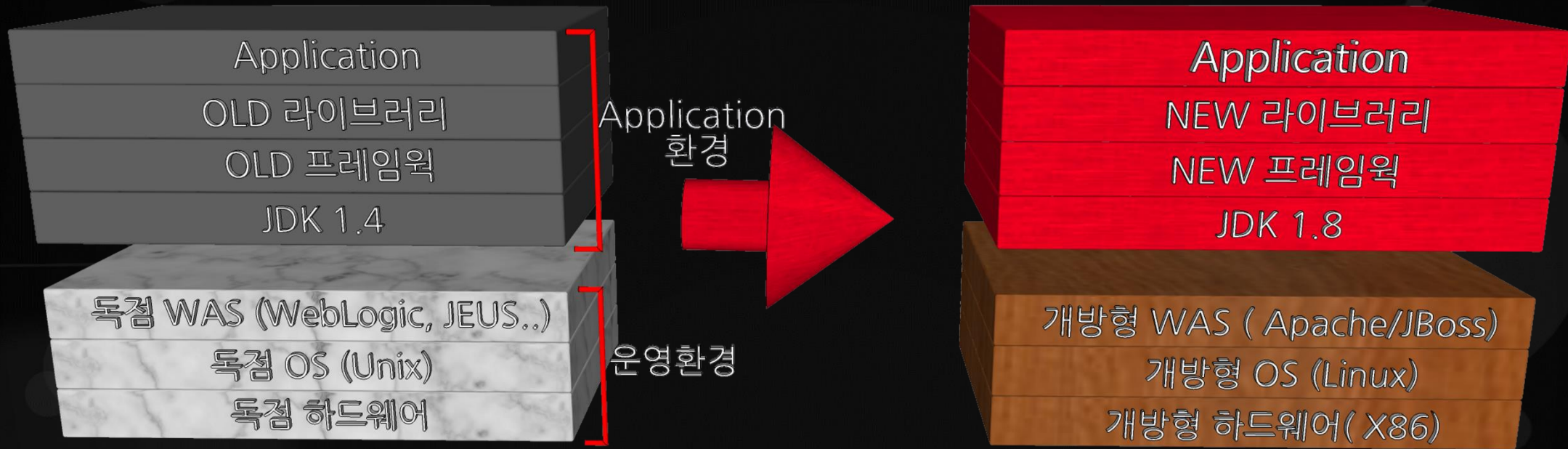
Red Hat Migration Toolkit for Cloud Native Application

JDK 및 WAS 업그레이드



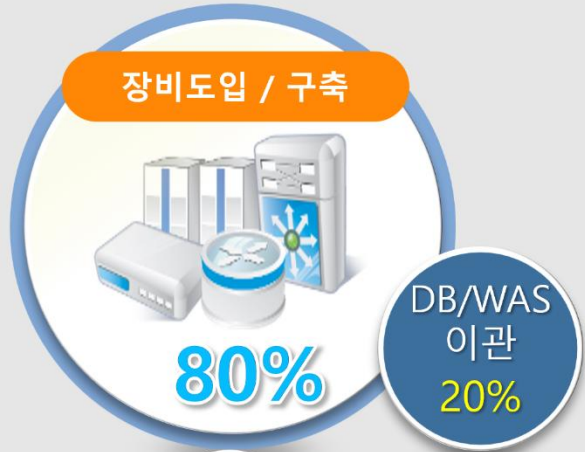
웹시스템 Unix To Linux 전환 개요

- 운영환경과 Application 환경을 분리하여 진행
- Application 환경 전환은 Java, 프레임워크, 라이브러리 업그레이드 및 개발환경 등을 전환
- 운영환경 전환은 하드웨어, OS 및 WAS 전환



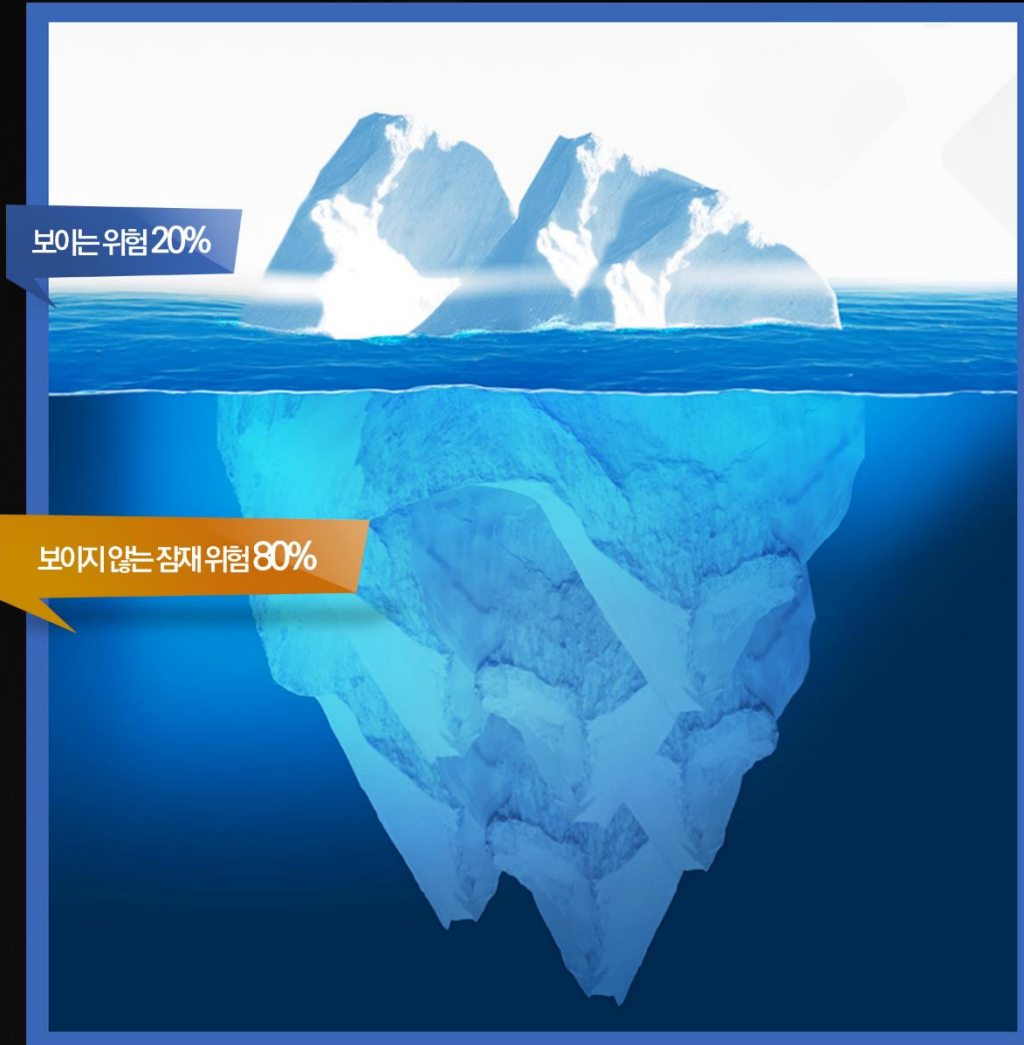
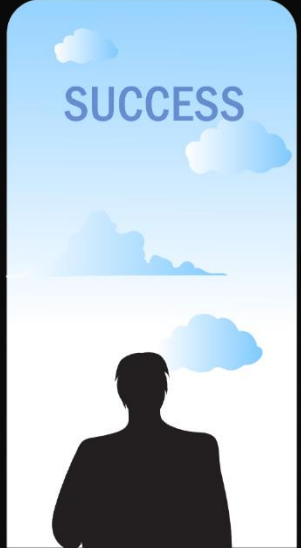
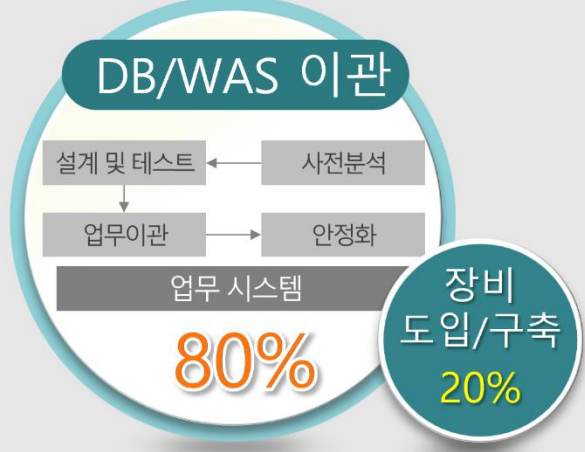
마이그레이션 프로젝트의 핵심은?

단순히
사업규모를
고려해 보면



VS

중요도와
난이도를
고려해 보면



Red Hat Migration Toolkit for Application

- RHMTA(Red Hat Migration Toolkit for Application)은 오픈소스 커뮤니티인 Windup의 Red Hat 컨설턴트 팀에 의해 개발
- CLI, Web Console, Eclipse-plugin 3가지 방식을 지원
- Java 애플리케이션을 분석하고 Java Code, JSP, XML들에 대하여 수정이 필요한 부분을 HTML 형식으로 Report 출력



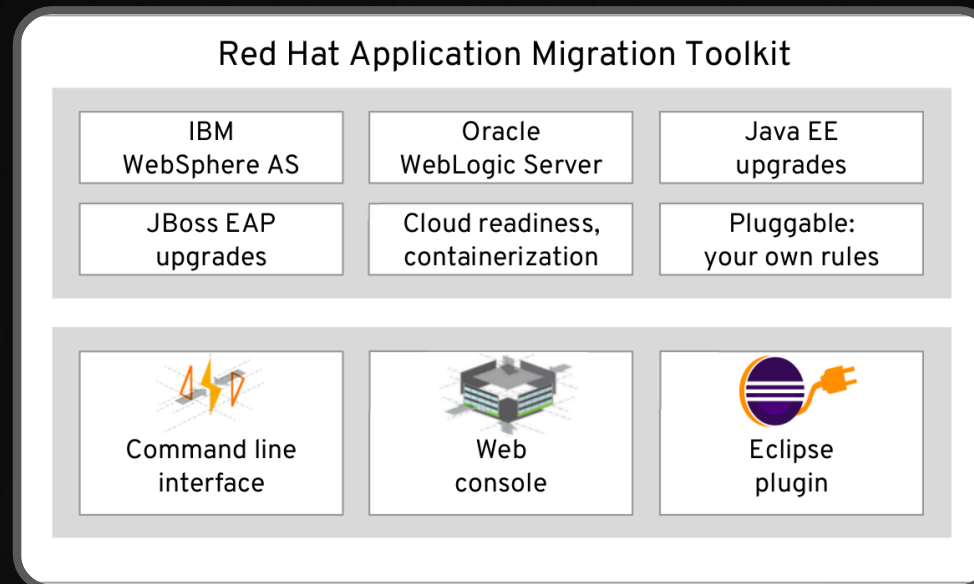
Red Hat
Application
Migration Toolkit



RHMTA is a tool to simplify **application migrations**.
Running from the command line,
the tool **reads EAR, WAR and JAR files**.
and **produces an HTML report** detailing the inner workings
of the Java application **to simplify migration efforts**.

RHMTA – Tools

- CLI
 - Command Line을 사용하여 마이그레이션이 필요한 소스를 분석이 가능
- Web Console
 - WEB 기반으로 되어있으며 Web Console의 통하여 분석이 필요한 소스들을 관리가 가능
 - 각각 프로젝트별로 소스를 분리 할 수 있으며 여러 개발자들이 동시에 관리 및 분석 가능
- Eclipse-Plugin
 - Eclipse와 JBoss Developer Studio에서 사용 가능한 플러그인 을 제공하여 소스 개발 중 변경 해야 할 이슈 부분을 IDE에서 바로 확인 가능한 것이 장점



RHMTA – CLI



- CLI는 아래 링크에서 다운로드 받을 수 있으며, Linux, Window 두 플랫폼에서 다 실행이 가능
- Download 링크 : <https://developers.redhat.com/download-manager/file/4.0.0/migrationtoolkit-RHMTA-cli-4.0.0.offline.zip>

1. 압축 해제
#unzip migrationtoolkit-RHMTA-cli-4.0.0.offline.zip
2. 디렉토리 이동
#cd RHMTA-cli-4.0.0.Final
3. 스크립트 실행
#./bin/RHMTA-cli --input ~/egovframework-all-in-one.war --output ~/test --source spring -target eap:7

• 실행 화면

```
root@localhost rhamt-cli-4.0.0.Final# ls
addons          cache           lib             rhamt-cli-version.txt  wls_medrec.zip
bash-completion ECLIPSE_LICENSE.txt logging.properties  rules
bin            ignore         README.txt       samples

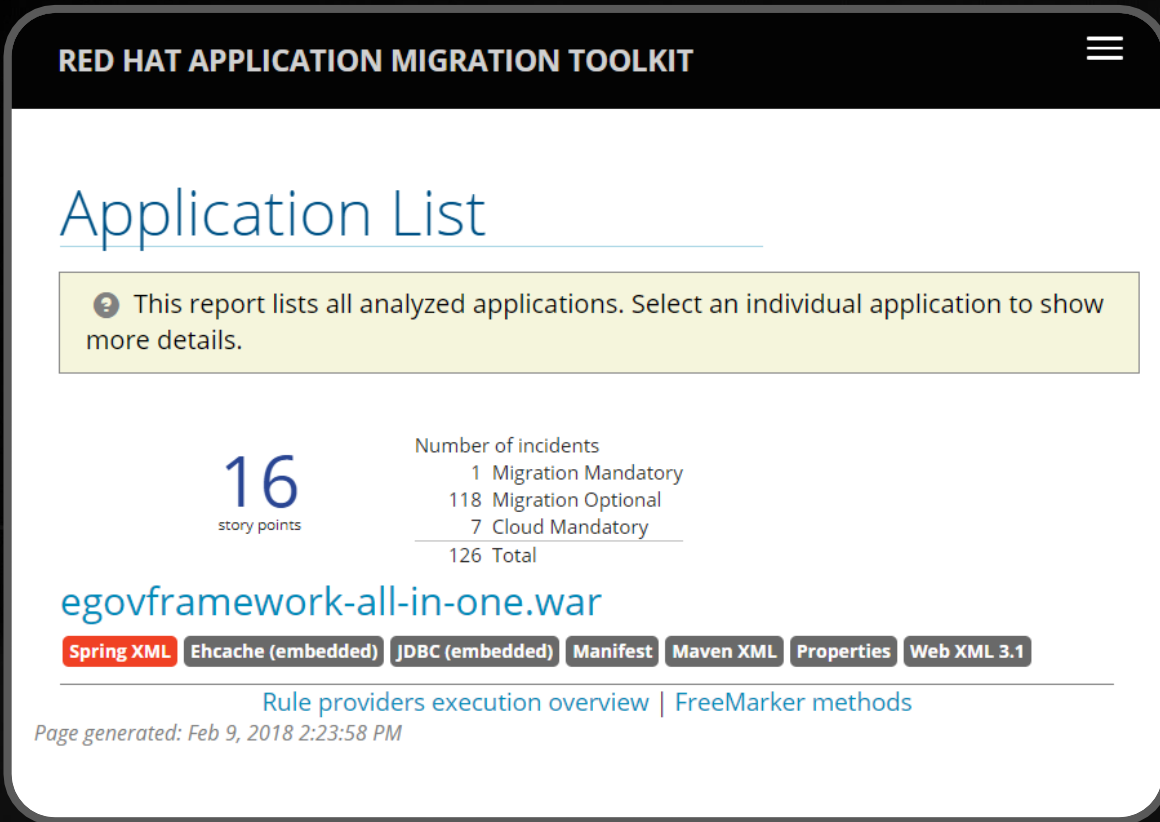
[root@localhost rhamt-cli-4.0.0.Final]# ./bin/rhamt-cli --input ~/egovframework-all-in-one.war --output ~/test --source spring -target eap:7
Using RHAMT at /root/rhamt-cli-4.0.0.Final
Warning: JAVA_HOME environment variable is not set.
> Red Hat Application Migration Toolkit (RHAMT) CLI, version 4.0.0.Final.
These input applications or directories are large:
  /root/egovframework-all-in-one.war
Processing may take a very long time. Please consult the RHAMT User Guide for performance tips. Would you like to continue? [Y,n] Y
WARNING: No packages were set in --packages. This will cause all .jar files to be decompiled and can possibly take a long time. Check the RHAMT User Guide for performance tips.
Overwrite all contents of "/root/test" (anything already in the directory will be deleted)? [y,N] Y

Input Application:/root/egovframework-all-in-one.war
Output Path:/root/test

Using user rules dir: /root/.rhamt/rules
Using user rules dir: /root/rhamt-cli-4.0.0.Final/rules
2018-02-19 16:13 [0/234] Executing RHAMT
2018-02-19 16:13 [1/234] InitializationPhase - LoadIssueCategoriesRuleProvider - LoadIssueCategoriesRuleProvider.attachToGraph
2018-02-19 16:13 [2/234] InitializationPhase - RegisterApiPackagesInTypeInterestFactoryRuleProvider - RegisterApiPackagesInTypeInterestFactoryRuleProvider_1
2018-02-19 16:13 [3/234] InitializationPhase - RegisterApiPackagesInTypeInterestFactoryRuleProvider - RegisterApiPackagesInTypeInterestFactoryRuleProvider_2
2018-02-19 16:13 [4/234] InitializationPhase - CopyJavaConfigToGraphRuleProvider - CopyJavaConfigToGraphRuleProvider_1
2018-02-19 16:13 [5/234] InitializationPhase - IgnoredArchivesConfigLoadingRuleProvider - IgnoredArchivesConfigLoadingRuleProvider_1
2018-02-19 16:13 [6/234] InitializationPhase - ArchiveIdentificationConfigLoadingRuleProvider - ArchiveIdentificationConfigLoadingRuleProvider_1
2018-02-19 16:13 [7/234] InitializationPhase - ArchiveIdentificationConfigLoadingRuleProvider - ArchiveIdentificationConfigLoadingRuleProvider_1
```

- Report 화면

- Report 화면에서는 전체적인 summary를 출력



RED HAT APPLICATION MIGRATION TOOLKIT

Application List

This report lists all analyzed applications. Select an individual application to show more details.

16 story points

Number of incidents

- 1 Migration Mandatory
- 118 Migration Optional
- 7 Cloud Mandatory
- 126 Total

egovframework-all-in-one.war

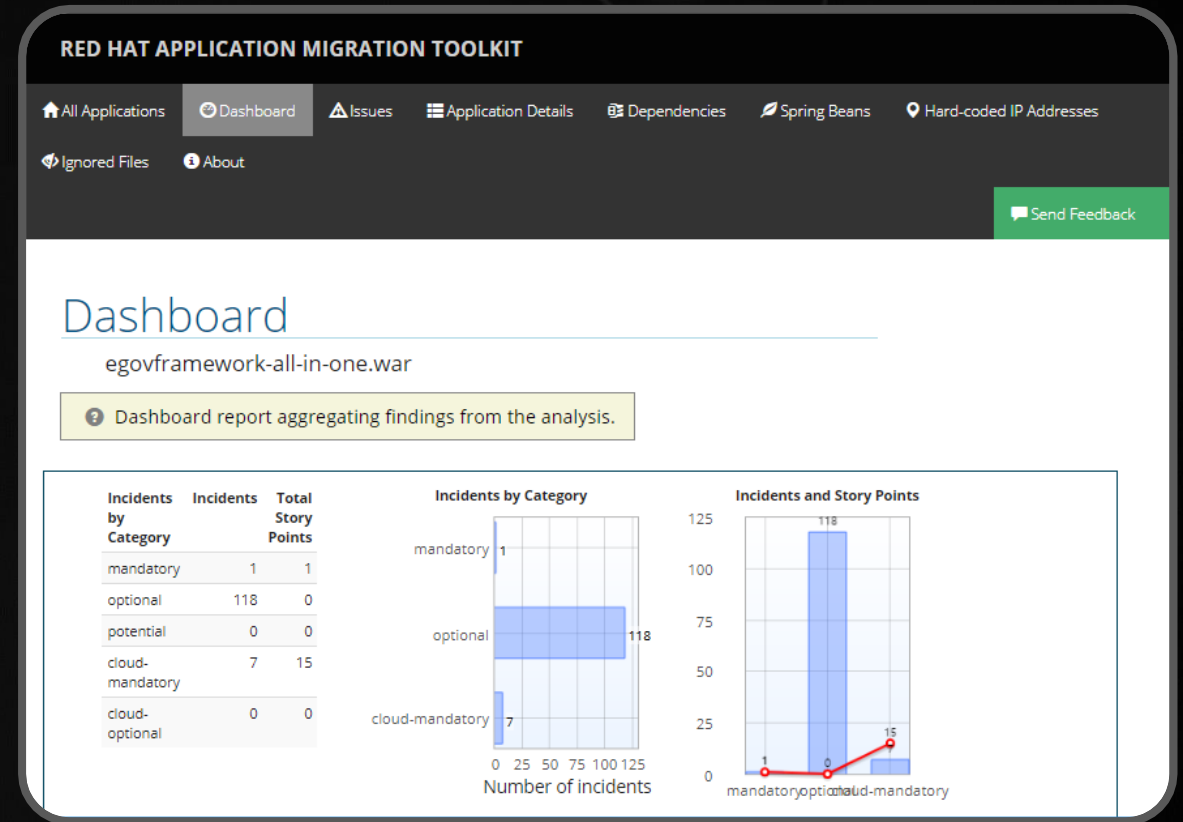
Spring XML Ehcache (embedded) JDBC (embedded) Manifest Maven XML Properties Web XML 3.1

Rule providers execution overview | FreeMarker methods

Page generated: Feb 9, 2018 2:23:58 PM

- 분석된 화면

- 애플리케이션을 선택 하면, 위와 같이 전체 적인 Menu가 나오며 변경 해야 할 가이드 내용을 확인 가능



RED HAT APPLICATION MIGRATION TOOLKIT

All Applications Dashboard Issues Application Details Dependencies Spring Beans Hard-coded IP Addresses Ignored Files About

Send Feedback

Dashboard

egovframework-all-in-one.war

Dashboard report aggregating findings from the analysis.

Incidents by Category	Incidents	Total Story Points
mandatory	1	1
optional	118	0
potential	0	0
cloud-mandatory	7	15
cloud-optional	0	0

Incidents by Category

Incidents and Story Points

RHMTA – WEB Console

- WEB Console는 아래 링크에서 다운로드 받을수 있으며, Linux, Window 두 플랫폼에서 다 실행이 가능
- Download 링크: <https://developers.redhat.com/download-manager/file/4.0.0/migrationtoolkit-RHMTA-web-distribution-4.0.0.with-authentication.zip>

1. 압축 해제

```
#unzip migrationtoolkit-RHMTA-web-distribution-4.0.0.with-authentication.zip
```

2. 디렉토리 이동

```
#cd RHMTA-web-distribution-4.0.0.Final
```

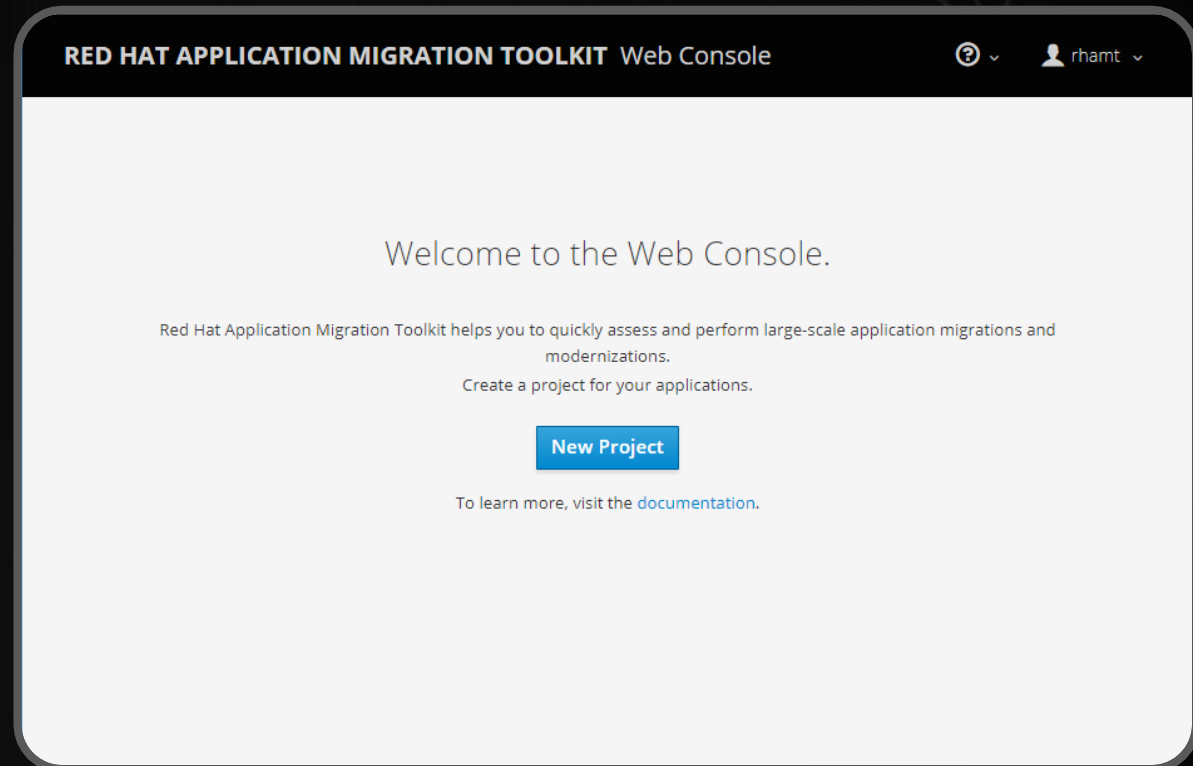
3. 스크립트 실행

```
#./run_RHMTA.sh
```

4. WEB Console 접속

```
http://localhost:8080/RHMTA-web
```

• 실행 화면

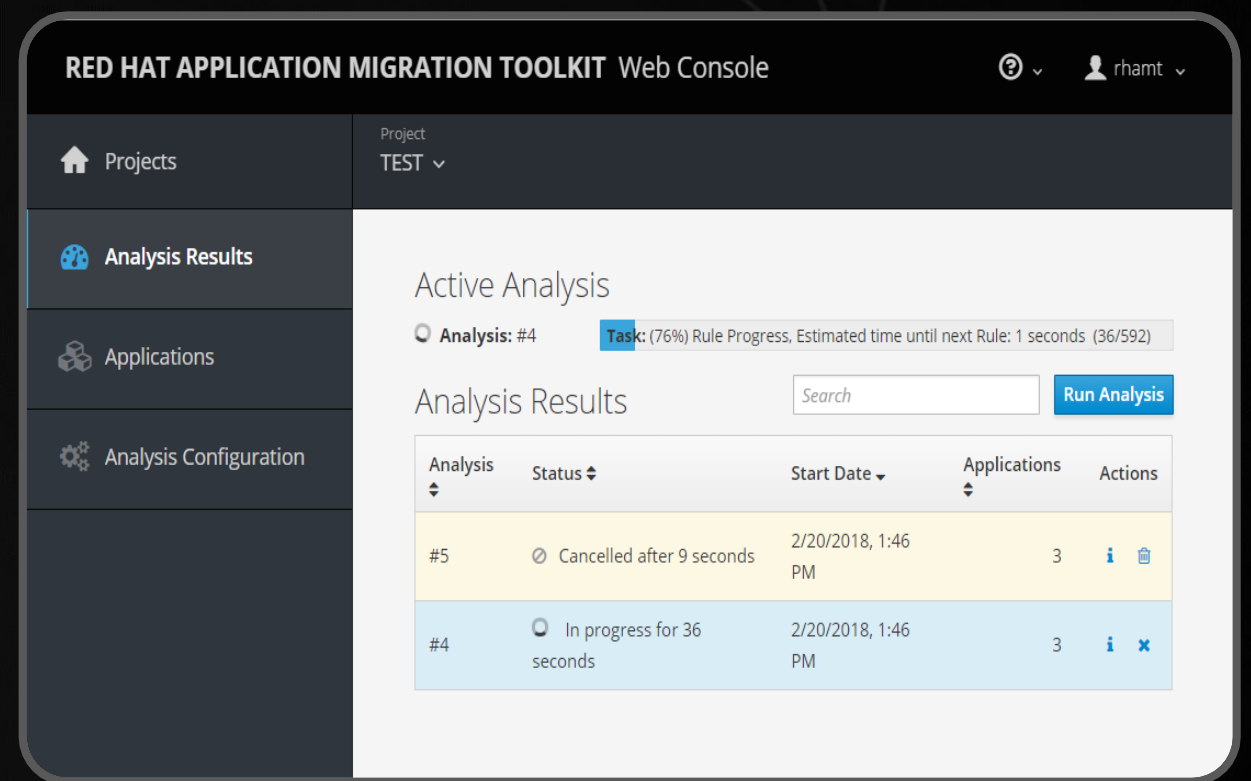
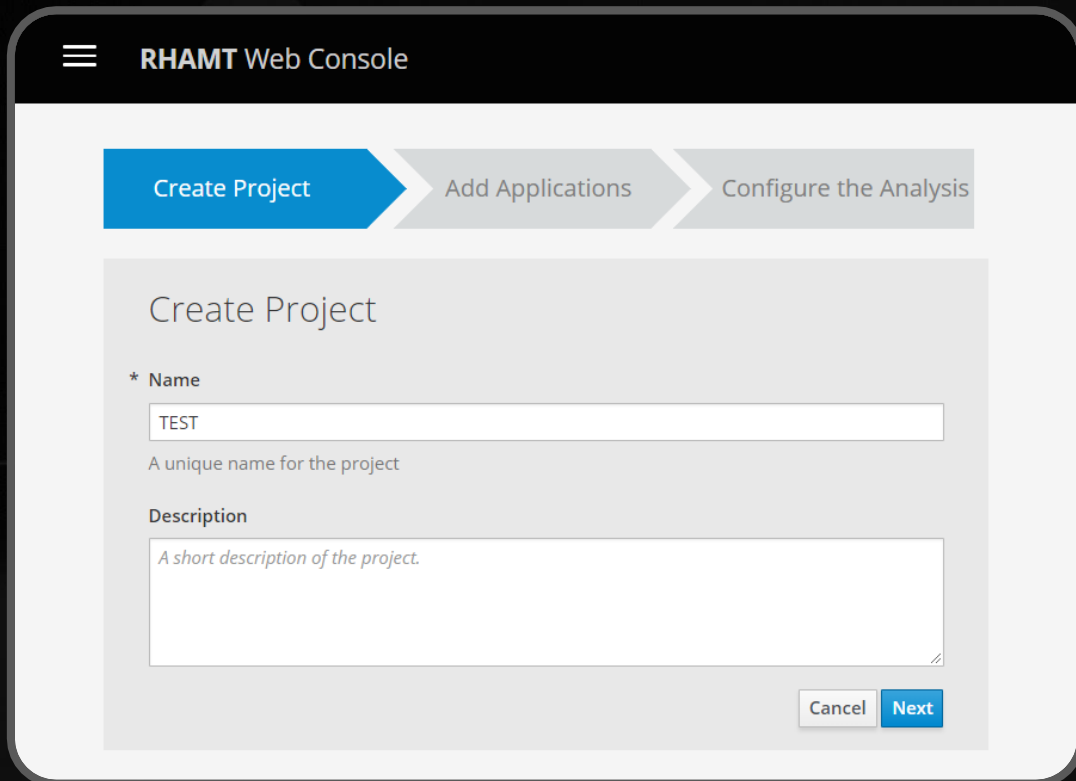


RHMTA – WEB Console



- Project 생성
 - Project를 생성하여 해당 프로젝트에서 등록된 애플리케이션을 관리가 가능

- 애플리케이션 분석
 - Project에 등록된 애플리케이션은 Run Analysis를 클릭하면 바로 분석을 진행



RHMTA – WEB Console

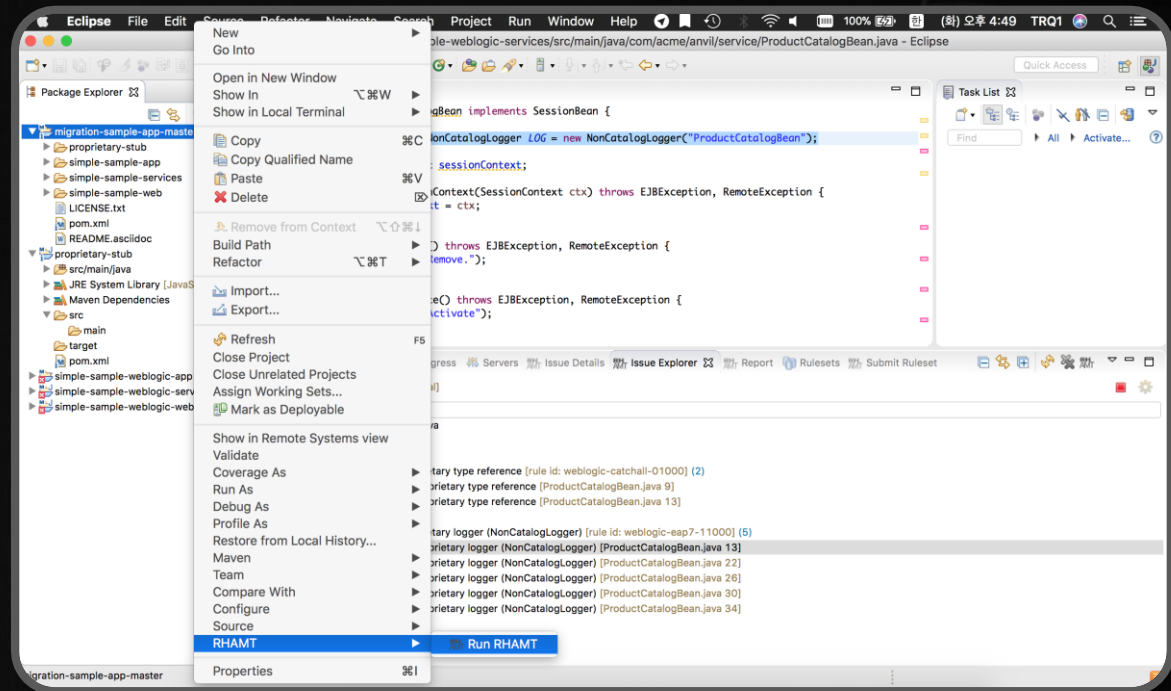
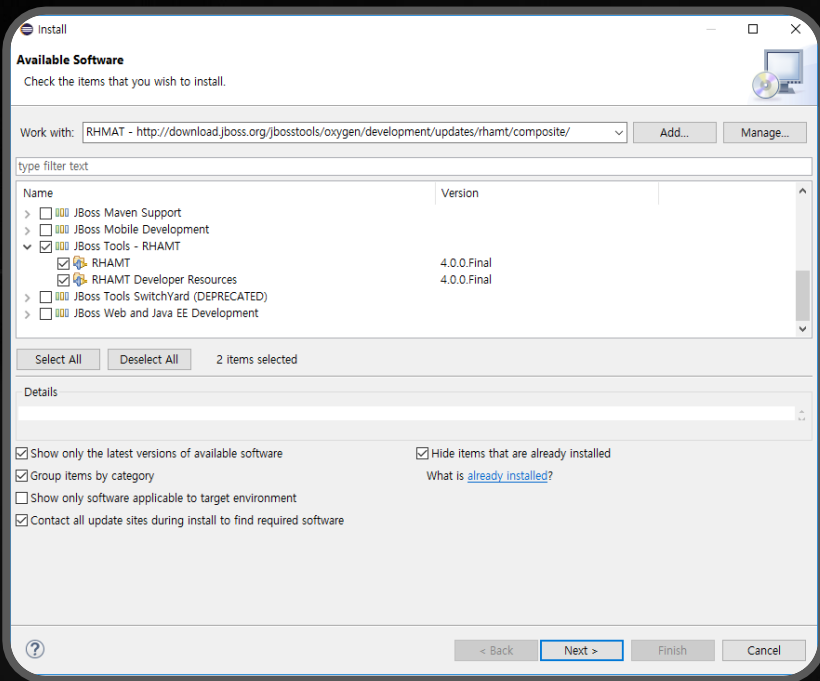


- 분석 설정
 - 어느 애플리케이션을 분석 할지, 어느 JBoss 버전으로 마이그레이션할지 설정이 가능

- Rules 설정
 - 직접 마이그레이션에 대한 Rule을 정의 하고 싶으면 Rules Configuration에서 설정이 가능

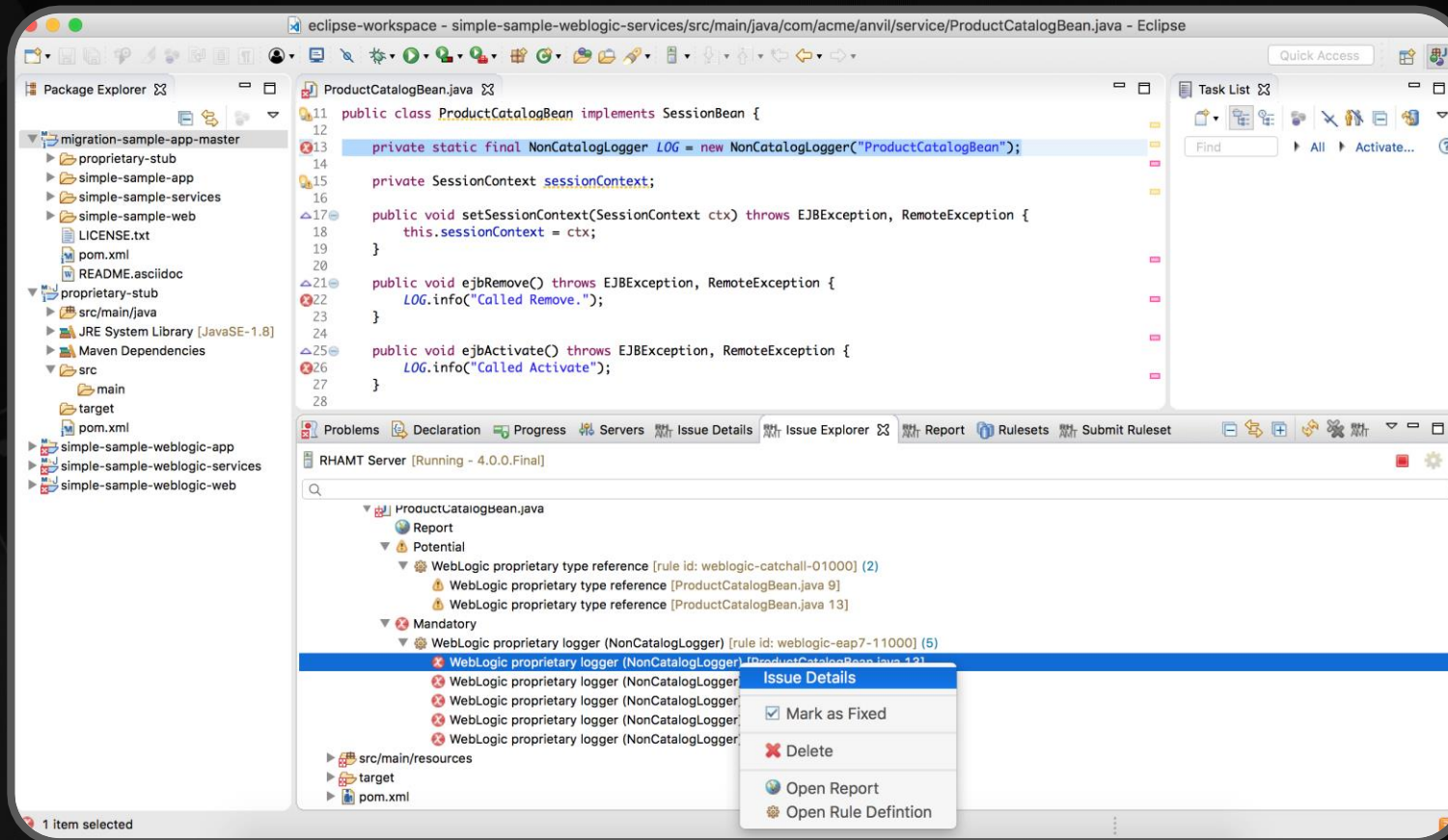
RHMTA – Eclipse-Plugin

- Eclipse-plugin은 Eclipse에서 Plugin을 설치 하여 사용
- Plugin 링크 : <http://download.jboss.org/jbosstools/oxygen/development/updates/RHMTA/composite/>
- 설치 화면
 - Plugin 설치하는 위에 링크를 등록하면 자동적으로 설치
- RHMTA RUN
 - 원하는 애플리케이션을 선택 후 분석하기 위해 RHMTA를 실행



RHMTA – Eclipse-Plugin

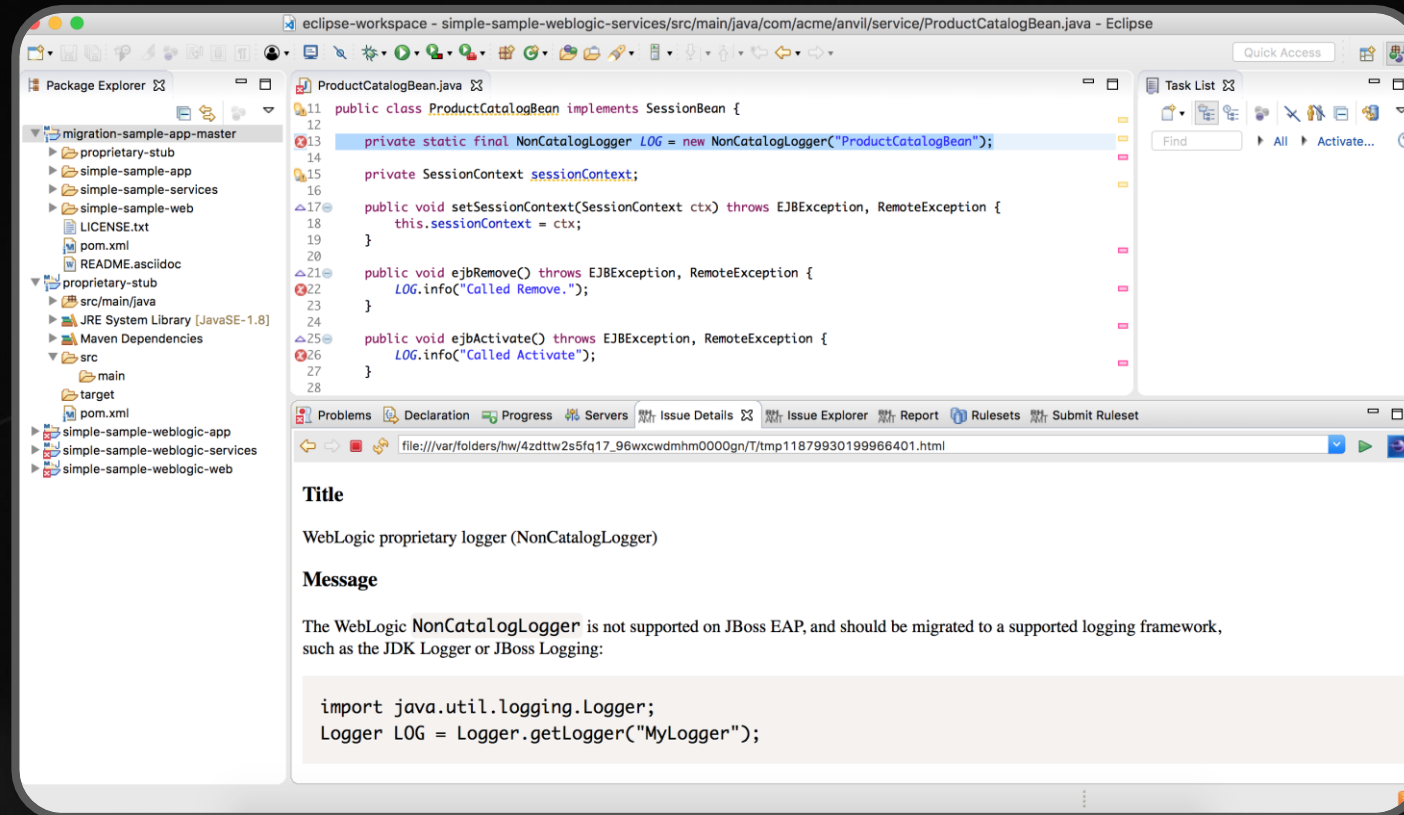
- Issue Explorer
 - Issue Explorer에서는 해당 애플리케이션에서 발생한 Issue 리스트를 제공하며, 해당 이슈에서 더 자세한 내용도 확인 가능



RHMTA – Eclipse-Plugin

- Issue Details

- Issue Details에서는 해당 애플리케이션의 Code나 Config 파일에서 변경 해야 할 부분은 자세히 가이드



RHMTA – 기능

RHMTA에서 식별 대상

1. 특정 Application Server에 종속적인 어플리케이션 코드
2. Java 코드 중 더 이상 사용할 수 없는 코드 (Deprecated Java code)
3. 비표준 -JMS 메시징 코드
4. 웹서비스 식별
5. EJB 버전 (2 / 3) 식별
6. 하이버네이트, 스프링 , 스트럿츠 등에 대한 업그레이드 여부
7. 잘못된 XML 코드

문제가 되는 어플리케이션 코드에 대한 가이드

Call of JNDI lookup

This method lookups an object using a JNDI String. During the migration process, some entity JNDI bindings may change. Ensure that the JNDI Name does not need to change for JBoss EAP.

For Example:

```
(ConnectionFactory)initialContext.lookup("weblogic.jms.ConnectionFactory");
```

should become:

```
(ConnectionFactory)initialContext.lookup("/ConnectionFactory");
```

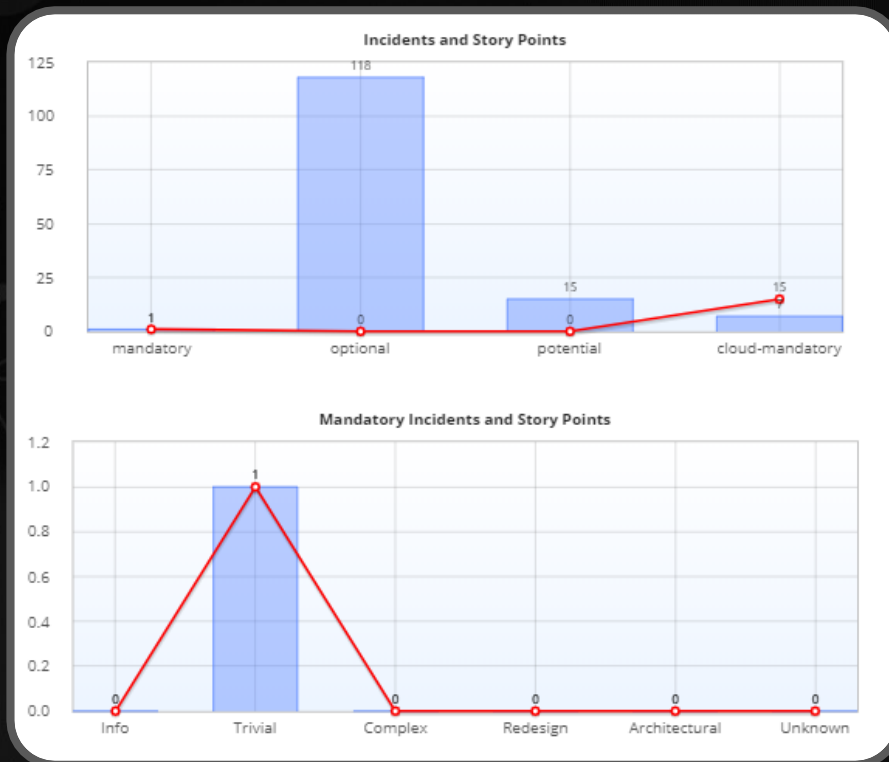
• [JBoss JNDI Reference](#)

RHMTA – 마이그레이션 점검 샘플

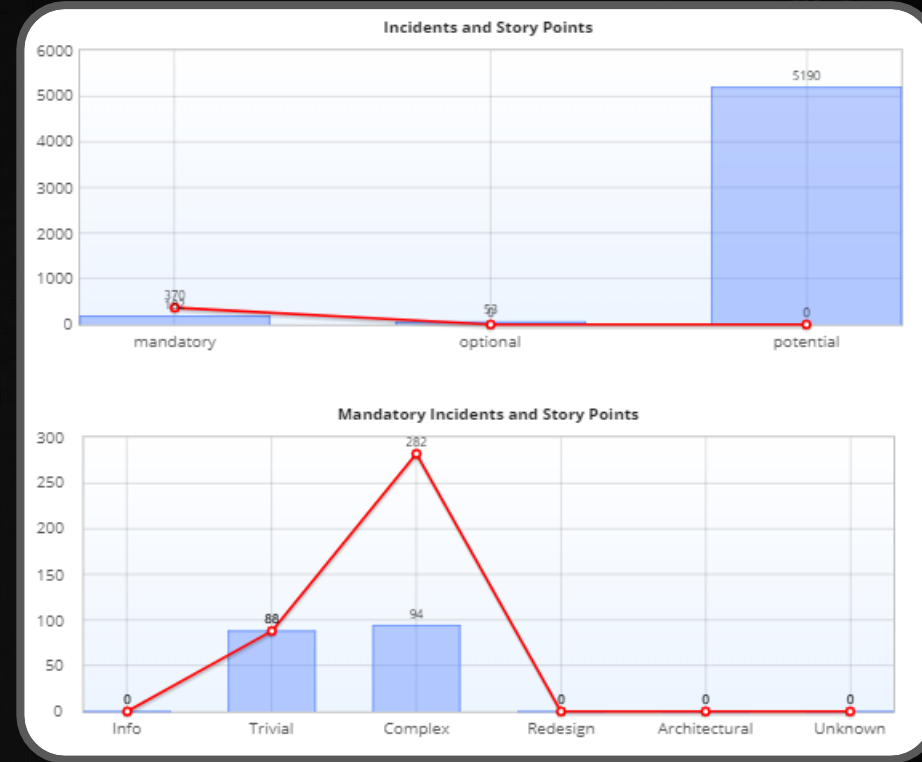
- 스토리 포인트란

- 특정 Application Server에 종속적인 어플리케이션 코드
- 애자일 프로젝트에서 사용자 스토리나 기능 또는 어떤 작업의 규모를 표현하기 위하여 사용되는 단위
- RHMTA 의 경우 스토리 포인트 1은 기술 숙련도에 따라 1시간 ~ 3시간으로 산정이 가능

- 전자정부프레임워크



- 웹로직 MedRec 샘플



RHMTA – 보고서 내용

- 변경 사항에 대한 상세한 내용

Issue Detail: WebLogic web application descriptor (weblogic.xml)
Show Rule

The Oracle WebLogic Server deployment descriptor file (`weblogic.xml`) provides functionality that is not included in the standard Java EE specification. While there is no direct mapping of these descriptor elements, many of these features may be configured in the application deployment or JBoss server configuration files.

For more information on how to configure JBoss EAP, please refer to the Configuration Guide for JBoss Enterprise Application Platform 7.

- [Migrate Oracle WebLogic configuration files to JBoss EAP](#)
- [Map weblogic.xml configurations to JBoss EAP](#)
- [Configuration Guide for JBoss EAP 7](#)

- 변경이 필수 또는 선택적인지 여부

Migration Mandatory				
Issue by Category	Incidents Found	Story Points per Incident	Level of Effort	Total Story Points
Call of JNDI lookup	1	1	Trivial change or 1-1 library swap	1
	1			1

Migration Optional				
Issue by Category	Incidents Found	Story Points per Incident	Level of Effort	Total Story Points
Maven POM (pom.xml)	69	0	Info	0
Dynamic class instantiation	45	0	Info	0
Embedded library - JDBC	3	0	Info	0
Web XML	1	0	Info	0
	118			0

RHMTA – 보고서 내용

- 변경이 복잡하거나 쉬운지 여부(Level of Effort)

Migration Mandatory				
Issue by Category	Incidents Found	Story Points per Incident	Level of Effort	Total Story Points
WebLogic proprietary logger (NonCatalogLogger)	19	1	Trivial change or 1-1 library swap	19
Call of JNDI lookup	5	1	Trivial change or 1-1 library swap	5
WebLogic T3 JNDI binding	3	3	Complex change with documented solution	9
WebLogic T3ServicesDef usage	3	3	Complex change with documented solution	9
JMX MBean object name (javax.management.ObjectName)	3	1	Trivial change or 1-1 library swap	3
WebLogic TransactionManager usage	2	1	Trivial change or 1-1 library swap	2
WebLogic InitialContextFactory	2	3	Complex change with documented solution	6

- 마이그레이션에 필요한 변경을 위한 팁 및 정보에 대한 링크

Issue Detail: JSF embedded library Show Rule

The application has a JSF library embedded. Red Hat JBoss EAP includes JSF as a module with a version that has been tested and is supported by Red Hat. There are two options for using the JSF library:

1. Keep it embedded as it is now. This approach is low effort but the application will not use a tested and supported library.
2. Switch to use the JSF library in the EAP module. This will require effort to remove the embedded library and configure the application to use the module's library, but then the application will rely on a tested and supported version of the JSF library.

In the links below there are instructions to enable alternative versions for both EAP 6 and 7.

- [Red Hat JBoss EAP: Component Details](#)
- [Red Hat JBoss EAP 6: JavaServer Faces \(JSF\) Code Changes](#)
- [How to use JSF 1.2 with EAP 6](#)
- [Red Hat JBoss EAP 7: JavaServer Faces \(JSF\) Code Changes](#)
- [How to use JSF 1.2 with EAP 7?](#)

CASE 1: WebLogic web application descriptor (weblogic.xml)

- WebLogic Web application descriptor(weblogic.xml)은 JBoss web application descriptor(jboss-web.xml)과 다르기 때문에 반드시 규격대로 변경을 해야 함
- WebLogic 에서 JBoss 으로 마이그레이션 작업 시 반드시 해야 함

Issue Detail: WebLogic web application descriptor (weblogic.xml)

[Show Rule](#)

The Oracle WebLogic Server deployment descriptor file (`weblogic.xml`) provides functionality that is not included in the standard Java EE specification. While there is no direct mapping of these descriptor elements, many of these features may be configured in the application deployment or JBoss server configuration files.

For more information on how to configure JBoss EAP, please refer to the Configuration Guide for JBoss Enterprise Application Platform 7.

- [Migrate Oracle WebLogic configuration files to JBoss EAP](#)
- [Map weblogic.xml configurations to JBoss EAP](#)
- [Configuration Guide for JBoss EAP 7](#)

Application Performance Management

Red Hat MTA Demo

CASE 1: WebLogic web application descriptor (weblogic.xml)

- weblogic.xml (변경 전)

```
<?xml version='1.0' encoding='UTF-8'?>
<weblogic-web-app xmlns="http://www.bea.com/ns/weblogic/90" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <security-role-assignment>
    <role-name>WebRunAsRole</role-name>
    <principal-name>Admin</principal-name>
  </security-role-assignment>
  <context-root>jee-example-web</context-root>
</weblogic-web-app>
```


CASE 1: WebLogic web application descriptor (weblogic.xml)

- **jboss-web.xml (변경 후)**

```
<?xml version='1.0' encoding='UTF-8'?>
<jboss-web>
  <security-domain>java:/jboss-web-policy</security-domain>
  <context-root>jee-example-web</context-root>
</jboss-web>
```

- **standalone.xml (security role 추가)**

```
<security-domains>
  <security-domain name="jboss-web-policy" cache-type="default">
    <authorization>
      <policy-module code="Delegating" flag="required"/>
    </authorization>
  </security-domain>
</security-domains>
```

- **참조 링크 : <https://access.redhat.com/articles/1327803>**

CASE 3: Call of JNDI Lookup

- JAVA EE 플랫폼에 정의된 JNDI context
 - Java:comp – 최근 구성요소에 대한 범위
 - Java:module – 최근 모듈에 대한 범위
 - Java:app – 최근 애플리케이션에 대한 범위
 - Java:global – 애플리케이션 서버에 대한 범위
- JBoss에서 제공하는 2개의 global namespace
 - Java:jboss/
 - Java:/

CASE 4: WebLogic T3 JNDI binding

- **JNDI Binding (변경 전)**

```
Properties environment = new Properties();  
environment.put("java.naming.factory.initial", "weblogic.jndi.WLInitialContextFactory");  
environment.put("java.naming.provider.url", "t3://localhost:7001");  
Context context = new InitialContext(environment);
```

- **JNDI Binding (변경 후)**

```
Properties environment = new Properties();  
environment.put("java.naming.factory.initial", "org.jboss.naming.remote.client.InitialContextFactory ");  
environment.put("java.naming.provider.url", "remote://localhost:4447 ");  
Context context = new InitialContext(environment);
```

- **참조 링크 : https://access.redhat.com/documentation/en-us/red_hat_jboss_enterprise_application_platform/6.4/html/development_guide/configuring_a_remote_jndi_client**



openmaru