****

**삼성카드 - JBoss Standalone모드 운영 메뉴얼**



**Focus Area:** **삼성카드 JBoss 운영 메뉴얼**

**Product/Process:** JBoss EWS 3.0.3, JBoss EAP 7.0.4



**Prepared By:**

|  |  |
| --- | --- |
| **Document Owner(s)** | **Project/Organization Role** |
| Jun-Young Son | JBoss Technical Assistance |
|  |  |
|  |  |

**Project Status Report Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Change Description** |
| 3.0 | 2017/08/08 | Jun-Young Son | Document Update |
|  |  |  |  |
|  |  |  |  |

**TABLE OF CONTENTS**

[1. 본 문서의 목적 6](#_Toc484595056)

[1.1 JBOSS ENTERPRISE MIDDLEWARE 정보 7](#_Toc484595057)

[1.2 주의 사항 7](#_Toc484595058)

[2 Red Hat JBoss EAP 7 8](#_Toc484595059)

[2.1 Java Virtual Machine 호환성 8](#_Toc484595060)

[2.2 인증된 구성 환경 8](#_Toc484595061)

[2.3 JBoss EAP 7 의 표준 지원 현황 11](#_Toc484595062)

[2.3.1 Java Enterprise Edition (EE) 지원 현황 11](#_Toc484595063)

[2.3.2 Web Services Standards / APIs 12](#_Toc484595064)

[2.3.3 Additional Standards / APIs 13](#_Toc484595065)

[2.3.4 JBoss 컴포넌트 버전 정보 13](#_Toc484595066)

[3 서버 설치 정보 16](#_Toc484595067)

[3.1 서버 시스템 정보 16](#_Toc484595068)

[3.2 WEB /WAS구성 16](#_Toc484595069)

[3.2.1 WEB Server 16](#_Toc484595070)

[3.2.2 WAS Server 17](#_Toc484595071)

[3.3 웹 서버 인스턴스 구성 18](#_Toc484595072)

[3.4 WAS 서버 인스턴스 구성 23](#_Toc484595073)

[3.4.1 인스턴스 구성 위치 23](#_Toc484595074)

[3.4.2 인스턴스 구성파일 23](#_Toc484595075)

[4 Jboss Enterprise Application Platform – standalone 모드 표준구성 24](#_Toc484595076)

[4.1 표준 구성도 24](#_Toc484595077)

[4.2 Directory Sturcture 24](#_Toc484595078)

[4.2.1 JBoss Domain 모드 설치 디렉토리 구조 24](#_Toc484595079)

[4.3 JBoss EAP 엔진 설치 25](#_Toc484595080)

[4.3.1 JBoss EAP 다운로드 25](#_Toc484595081)

[4.3.2 JBoss EAP 설치 26](#_Toc484595082)

[4.3.3 JBoss EAP 엔진 디렉토리 구조 27](#_Toc484595083)

[4.4 JBoss standalone설치 및 설정 28](#_Toc484595084)

[4.4.1 standalone 디렉터리 구조 28](#_Toc484595085)

[4.4.2 configuration 파일 29](#_Toc484595086)

[4.4.3 profile 의 이해 29](#_Toc484595087)

[4.4.4 MSC방식의 이해 30](#_Toc484595088)

[4.4.5 관리자 계정 생성 및 추가 30](#_Toc484595089)

[5 JBoss 인스턴스 생성 32](#_Toc484595090)

[5.1 인스턴스 생성 32](#_Toc484595091)

[5.1.1 Standalone template 복사 32](#_Toc484595092)

[5.1.2 실행 스크립트의 구성 32](#_Toc484595093)

[5.1.3 실행 스크립트의 구성 32](#_Toc484595094)

[5.1.4 관리 스크립트의 구성 44](#_Toc484595095)

[6 Jboss Enterprise Application Server + 웹서버 연동 47](#_Toc484595096)

[6.1 Directory Sturcture 47](#_Toc484595097)

[6.1.1 JBoss EWS (웹서버) 설치 홈 디렉토리 47](#_Toc484595098)

[6.1.2 JBoss EWS 디렉토리 구조 47](#_Toc484595099)

[6.1.3 JBoss EWS 연동 설정 파일 48](#_Toc484595100)

[6.2 JBoss EWS엔진의 설치 49](#_Toc484595101)

[6.2.1 JBoss EWS 다운로드 49](#_Toc484595102)

[6.2.2 JBoss EWS 설치 50](#_Toc484595103)

[6.2.3 mod\_jk 다운로드 50](#_Toc484595104)

[6.3 JBoss EWS + EAP 연동 설정 51](#_Toc484595105)

[6.3.1 mod\_jk.conf 파일 작성 51](#_Toc484595106)

[6.3.2 workers.properties 파일 작성 52](#_Toc484595107)

[6.3.3 uriworkermap.properties 파일 작성 53](#_Toc484595108)

[6.4 Webapp session replication 을 위한 web.xml 수정 54](#_Toc484595109)

[7 Jboss DataSource 설정 56](#_Toc484595110)

[7.1 JDBC 드라이버 추가 및 등록 56](#_Toc484595111)

[7.2 JDBC DataSource 설정시 JNDI 이름 지정 58](#_Toc484595112)

[7.3 Datasource 기본 설정 항목의 설명 61](#_Toc484595113)

[8 JBoss EWS 모니터링 가이드 65](#_Toc484595114)

[8.1 모니터링 항목 65](#_Toc484595115)

[8.2 모니터링 가이드 66](#_Toc484595116)

[8.2.1 System Resource 체크 66](#_Toc484595117)

[8.2.2 Process Resource 체크 68](#_Toc484595118)

[8.2.3 Server Status 69](#_Toc484595119)

[8.2.4 Execute Status – 실행 상태 72](#_Toc484595120)

[8.2.5 로그 모니터링 73](#_Toc484595121)

[9 JBoss EAP 모니터링 가이드 75](#_Toc484595122)

[9.1 모니터링 항목 75](#_Toc484595123)

[9.2 모니터링 가이드 76](#_Toc484595124)

[9.2.1 System Resource 체크 76](#_Toc484595125)

[9.2.2 Process Resource 체크 78](#_Toc484595126)

[9.2.3 Server Status 80](#_Toc484595127)

[9.2.4 Execute Status – 실행 상태 82](#_Toc484595128)

[9.2.5 JDBC DataSource 상태 모니터링 88](#_Toc484595129)

[9.2.6 로그 모니터링 93](#_Toc484595130)

# 1. 본 문서의 목적

본 문서의 목적은 JBoss 운영에 필요한 내용을 담은 가이드 문서입니다. 본 문서에는 JBoss EWS, EAP 설치 현황 및 설치 및 환경 구성에 대한 내용이 포함되어 있으며, 기타 모니터링을 위한 정보 등을 모두 포함하고 있습니다.

## JBOSS ENTERPRISE MIDDLEWARE 정보

JBoss Enterprise Middleware 는 표준 기반의 오픈 소스이며 Java 어플리케이션 개발, 어플리케이션 및 서비스 호스팅, 컨텐츠 집적 및 프레젠테이션, 데이터 통합, 서비스 통합 및 조정을 위한 포괄적인 엔드 투 엔드 엔터프라이즈 미들웨어 포트폴리오입니다.

JBoss Enterprise Middleware 에는 전체 어플리케이션 수명 주기의 솔루션이 포함됩니다. 혁신적인 Web 2.0 개발 도구 및 프레임워크부터 SOA 기반 어플리케이션 및 서비스 배포, 완전한 미들웨어 관리 및 모니터링에 이르기까지 다양한 수명 주기의 솔루션이 여기에 해당됩니다.

JBoss Enterprise Middleware 는 서브스크립션을 통해 사용할 수 있고 여기에는 인증 소프트웨어, 업계 선두의 지원, 업데이트 및 패치, 설명서, 다년간의 유지 보수 정책 등이 포함됩니다.

https://access.redhat.com/ 에서 자세한 내용을 확인하십시오.

## 주의 사항

본 문서에서 설명하고 있는 JBoss EAP 7 를 구성하고 관리하는 내용은 JBoss EAP 7 에 대한 매뉴얼을 기초로 하여 필요한 간단한 지식을 정리하였으므로 정확하게 상세한 내용은 JBoss EAP 매뉴얼을 참조합니다.

<https://access.redhat.com/knowledge/docs/JBoss_Enterprise_Application_Platform>

또한 본 문서의 내용은 참고용이므로 어떠한 법적 책임이 없습니다.



# Red Hat JBoss EAP 7

## Java Virtual Machine 호환성

다음 URL 에서 JBoss EAP 7.x 버전에서 인증된 운영환경에 대한 정보를 확인 할 수 있습니다.

<https://access.redhat.com/articles/2026253>

Red Hat 은 아래와 같은 구성환경에 대한 호환과 인증 모두를 지원합니다.

|  |  |
| --- | --- |
| **Java Virtual Machine** | **Version** |
| Oracle JDK | 1.8 |
| OpenJDK | 1.8 |
| IBM JDK | 1.8 |

## 인증된 구성 환경

다음과 같은 OS, 칩아키텍처 그리고 JVM 구성 환경에 대하여 JBoss Enterprise Application Platform 6 에 대한 완벽한 지원을 인증합니다

|  |  |  |
| --- | --- | --- |
| **Operating System** | **Chip Architecture** | **Java Virtual Machine(s)** |
| Red Hat Enterprise Linux 7 (latest update) | x86, x86\_64 | Oracle JDK 1.8.x  IBM JDK 1.8.x  OpenJDK 1.8.x |
| Red Hat Enterprise Linux 6 (latest update) | x86, x86\_64 | Oracle JDK 1.8.x  IBM JDK 1.8.x  OpenJDK 1.8.x |
| Microsoft Windows 2012 Server R2 | x86\_64 | Oracle JDK 1.8.x  Azul Zulu 1.8.x |
| Microsoft Windows 2008 Server R2 | x86\_64 | Oracle JDK 1.8.x |
| Oracle Solaris 10 | x86, x86\_64, SPARC 64 | Oracle JDK 1.8.x |
| Oracle Solaris 11 | x86, x86\_64, SPARC 64 | Oracle JDK 1.8.x |
| HP-UX 11v3 | Itanium 64 | HP JDK 1.8.x |

JBoss Enterprise Application Platform 7 는 다음과 같은 데이터베이스와 데이터베이스 드라이버에 대해서 지원을 인증합니다.

|  |  |  |
| --- | --- | --- |
| **Database** | **Version** | **Database Driver** |
| IBM DB2 | 9.7 | IBM DB2 JDBC Universal Driver Architecture 4 |
| 10.5 | IBM DB2 JDBC Universal Driver Architecture 4 |
| Oracle | 11g R2 | Oracle JDBC Driver v12 |
| 11g R2 RAC | Oracle JDBC Driver v12 |
| 12c | Oracle JDBC Driver v12 |
| 12c RAC | Oracle JDBC Driver v12 |
| MySQL | 5.7 | MySQL Connector/J 5 |
| 5.5 | MySQL Connector/J 5 |
| MariaDB | 10.1 | MariaDB Connector/J 1 |
| Microsoft | SQL Server 2014 SP1 | Microsoft SQL Server JDBC Driver 4 |
| SQL Server 2012 SP3 | Microsoft SQL Server JDBC Driver 4 |
| PostgreSQL | 9.4 | JDBC4 Postgresql Driver, Version 9 |
| 9.3 | JDBC4 Postgresql Driver, Version 9 |
| Enterprise DB Postgres Plus Advanced Server | 9.3 | Postgres Plus Advanced Server Driver 9.3 |
| 9.4 | Postgres Plus Advanced Server Driver 9.4 |
| Sybase | ASE 15.7 | Sybase jConnector JDBC Driver v7 |

JBoss Enterprise Application Platform 7 는 다음과 같은 웹서버와 커넥터에 대해서 지원을 인증합니다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Server** | **Version** | **OS** | **Connectors** | **Notes** |
| JBoss EAP 7.0  (Undertow Web Proxy) | 7.0 | Red Hat Enterprise Linux 7(x86,x86\_64)  Red Hat Enterprise Linux 6(x86, x86\_64)  Solaris 11(x86, x86\_64, SPARC64)  Solaris 10(x86, x86\_64, SPARC64)  Windows Server 2012 R2(x86\_64)  Windows Server 2008 R2(x86\_64)  HP-UX 11v3 | N/A | Protocols:  AJP/HTTP/HTTPS/WebSocket  Natives are not required |
| Apache Web(httpd) Server from Red Hat JBoss core Service0 (JBCS)  Apache HTTP Server and Red Hat JBoss Web Server 3.0.3 | 2.4.6 | Red Hat Enterprise Linux 7 (x86\_64)  Red Hat Enterprise Linux 6 (x86\_64)  Solaris 11(x86, x86\_64, SPARC64)  Solaris 10(x86, x86\_64, SPARC64)  Windows Server 2012 R2(x86\_64)  Windows Server 2008 R2(x86\_64)  HP-UX 11v3 | mod\_cluster 1.3.1 Final  mod\_jk 1.2.40(from JBCS) | Protocol(mod\_cluster):  AJP/HTTP/HTTPS/WebSocket  Protocol(mod\_jk):  AJP |
| Microsoft IIS | 7.5 | Windows Server 2008 R2(x86\_64) | Isapi\_redirect 1.2.40(from JBCS) | Protocol:  AJP |
| Microsoft IIS | 8.5 | Windows Server 2012 R2(x86\_64) | Isapi\_redirect 1.2.40(from JBCS) | Protocol:  AJP |
| Oracle iPlanet Web Server | 7.0.15 | Solaris 11(x86\_64)  Solaris 10(x86\_64)  Solaris 11(SPARC64)  Solaris 10(SPARC64) | Nsapi\_redirector 1.2.40(from JBCS) | Protocol:  AJP |

## JBoss EAP 7 의 표준 지원 현황

아래의 URL 에서 최신 표준 지원 현황에 대한 확인이 가능하다.

[https://access.redhat.com/knowledge/ko/articles/299283](https://access.redhat.com/knowledge/ko/articles/2992831)

JBoss EAP (JBoss Enterprise Application Platform) 릴리스에서 다음과 같은 업계 표준이 지원됩니다.

### Java Enterprise Edition (EE) 지원 현황

JBoss EAP 릴리즈 에 해당 하는 Java EE 표준 지원 현황은 다음과 같다.

|  |  |
| --- | --- |
| **기술/사양** | **JBoss EAP 7** |
| Java EE[JSR-151,244,316] | 7.0 |
| JAX-RPC(Java API for XML-based RPC) [JSR-101] | 1.1 |
| SAAJ(SOAP with Attachments API for Java) [JSR-067] | 1.3 |
| Java Servlet [JSR-154, 315] | 3.1 |
| JSF(JavaServer Faces) [JSR-252, 314] | 2.2 |
| JSP(JavaServer Pages & Expression Language) [JSR-245] | 2.3 |
| JTA(Java Transaction API) [JSR-907] | 1.2 |
| JACC(Java Authorization Contract for Containers) [JSR-115] | 1.3 |
| EJB(Enterprise JavaBeans) with Interceptors 1.1 [JSR-153, 220, 318] | 3.2, 3.1, 3.0, 2.1, 2.0 |
| Java EE Connector Architecture [JSR-112, 322] | 1.7 |
| J2EE Management [JSR-077] | 1.1 |
| Enterprise Web Services [JSR-109] | 1.3 |
| Java EE Application Deployment [JSR-088] | 1.2 |
| JavaMail [JSR-919] | 1.5 |
| JMS(Java Message Service) [JSR-914] | 2.0 |
| JPA(Java Persistence) [JSR-220, 317] | 2.1 |
| Web Services Metadata for the Java Platform [JSR-181] | 2.1 |
| JAX-WS(Java API for XML Web Services) [JSR-224] | 2.2 |
| Debugging Support for Other Languages [JSR-045] | 1.0 |
| JSTL(Standard Tag Library for JavaServer Pages) [JSR-052] | 1.2 |
| Common Annotations for the Java Platform [JSR-250] | 1.2 |
| JAXB(Java Architecture for XML Binding) [JSR-222] | 2.2 |
| JAX-RS(Java API for RESTful Web Services) [JSR-311] | 2.0 |
| JAXR(Java API for XML Registries) [JSR-093] | 1.0 |
| CDI(Contexts and Dependency Injection for Java) [JSR-299] | 1.1 |
| Bean Validation [JSR-303] | 1.1 |

### Web Services Standards / APIs

JBoss EAP 에서 지원하는 웹서비스 표준은 다음과 같습니다.

|  |  |
| --- | --- |
| **기술/사양** | **JBoss EAP 7.0** |
| SOAP(Simple Object Access Protocol) | 1.1, 12 |
| SOAP MTOM(Message Transmission Optimization Mechanism) | 예 |
| XOP(XML-Binary Optimized Packaging) | 예 |
| WSDL(Web Services Description Language) | 1.1 |
| WS-I Basic Profile | 1.1, 1.2, 2.0 |
| WS-Addressing | 1.0 |
| WS-Reliable Messaging | 1.0 |
| WS-Security | 1.1 |
| UDDI(Universal Description Discovery and Integration) | 2.0, 3.0 |
| JAX-WSA(Java API for XML Web Services Addressing) [JSR-261] | 1.0 |
| WS-Atomic Transactions | 1.1 |
| Fast Infoset(FI - ISO/IEC 24824-1, ITU-T Rec. X.891) | 예 |
| WS-Business Activity | 1.1 |
| WS-Coordination | 1.1 |
| WS-Security Policy | 1.3 |
| WS-Policy | 1.5 |
| WS-Policy Attachment | 1.5 |
| WS-I Attachments Profile | 1.0 |
| WS-Eventing | - |
| WS-Trust | 1.3 |
| SAML(Security Assertion Markup Language) | 2.0 |
| XACML | 2.0 |

### Additional Standards / APIs

JBoss EAP 에서 지원하는 추가적인 표준 지원은 다음과 같다.

|  |  |
| --- | --- |
| **기술/사양** | **JBoss EAP 7.0** |
| JTS(Java Transaction Service) | 1.0 |
| JDBC [JSR-054] | [4] |
| CORBA | 2.3.1 |
| Streaming API for XML Processing(StAX) [JSR-173] | [4] |
| JAF(JavaBeans Activation Framework) [JSR-925] | [4] |
| JMX(Java Management Extensions) [JSR-255] | [4] |
| JDBC(Java Database Connectivity) [JSR-221] | [4] |
| JAXP(Java API for XML Processing) [JSR-205] | [4] |
| SPNEGO/Kerberos | 예 |

### JBoss 컴포넌트 버전 정보

JBoss EAP 에 포함된 컴포넌트에 대한 버전 정보는 아래의 URL 에서 확인 가능하다.

[https://access.redhat.com/knowledge/ko/articles/299223](https://access.redhat.com/knowledge/ko/articles/2992231)

|  |  |
| --- | --- |
| **Component** | **EAP 7.0.0** |
| Wildfly AS | 2.1.2.Final |
| WELD | 2.3.3.Final |
| Rest Easy | 3.0.16.Final |
| mod\_cluster | 1.3.2.Final |
| Picketlink Federation | 2.5.5.SP1 |
| JBoss DMR | 1.3.0.Final |
| JBoss Metadata | 10.0.0.Final |
| JBoss XNIO | 3.3.6.GA |
| PicketBox | 4.9.6..Final |
| PicketBox Commons | 1.0.0.Final |
| Hibernate Core | 5.0.9.Final |
| Hibernate Infinispan (2LC) | 5.0.9.Final |
| Hibernate Envers | 5.0.9.Final |
| Hibernate Validator | 5.2.4.Final |
| Hibernate JPA 2.1 API | 1.0.0.Final |
| Infinispan | 8.1.2.Final |
| IronJacamar (JCA 1.6) | 1.3.3.Final |
| JGroups | 3.6.8.Final |
| JBoss Transactions(Narayana) | 5.2.16.Final |
| JBoss EJB Client | 2.1.4.Final |
| JBoss EJB3 EXT API | 2.2.0.Final |
| JBoss WS API | 1.0.3.Final |
| JBoss WS SPI | 3.1.0.Final |
| JBoss WS Common | 3.1.1.Final |
| JBoss WS Common Tools | 1.2.2.Final |
| JBoss WS-CXF | 5.1.3.SP1 |
| JBoss VFS | 3.2.11.Final |
| JBoss Remoting3 | 4.0.18.Final |
| JBoss Marshalling | 1.4.10.Final |
| JBoss SASL | 1.0.5.Final |
| JSF2 | 2.2.12 |
| JBoss Security Negotiation | 3.0.2.Final |
| JBoss Management Console(HAL) | 2.8.24.Final |
| Netty(HornetQ and Inifnispan에 포함) | 4.0.32.Final |
| JBoss Logging | 3.3.0.Final |
| Apache CXF | 3.1.4 |
| Apache CXF XJC-Utils | 3.0.5 |
| Apache WSS4J(WS용) | 2.1.4 |
| Xalan | 2.7.1 |
| JAXB | 2.2.11 |
| JBeret | 1.2.0 |

# 서버 설치 정보

최초 요청받은 시스템 설치 관련 사항은 다음과 같습니다.

* httpd (JBoss EWS), AP (JBoss EAP) 설치 및 설정

## 서버 시스템 정보

운영 시스템에 대한 정보는 다음과 같습니다.

|  |  |  |  |
| --- | --- | --- | --- |
| **시스템 분류** | **IP** | **계정 정보** | **엔진 홈디렉터리** |
| Web Server 엔진   * JBoss EWS 3.0.3 (apache 2.4.6) | xx.xxx.xx.xx | jws3 | /web/jws3/httpd |
| Was Server 엔진   * JBoss EAP 7.0.4 | xx.xxx.xx.xx | jboss7 | /was/jboss7/jboss-eap-7.0 |

## WEB /WAS구성

### WEB Server

|  |  |
| --- | --- |
| **서버** | **(WEB01)** |
| 응용관리계정 | jws3 ( /web/jws3 ) |
| 기동 | 도메인별 기동  /web/jws3/domains/{도메인명}/{인스턴스명}/bin/start.sh |
| 정지 | /web/jws3/domains/{도메인명}/{인스턴스명}/bin/stop.sh |
| 프로세스확인 | /web/jws3/domains/{도메인명}/{인스턴스명}/bin/pcheck.sh |
| WEB HTTP Port | 설치결과서 참고  3.\_(통합)미들웨어JBoss\_설치결과서(최종)\_v0.7\_20170726.xlsx |
| WEB HTTPS Port | 설치결과서 참고  3.\_(통합)미들웨어JBoss\_설치결과서(최종)\_v0.7\_20170726.xlsx |
| logs 위치 | /logs/jws3 밑에 도메인별로 위치함 |

### WAS Server

|  |  |
| --- | --- |
| **서버** | **(WAS01)** |
| 엔진관리계정 | Jboss7 ( /was/jboss7 ) |
| domain home 위치 | /was/jboss7/domains |
| 기동 | 도메인별 기동  /was/jboss7/domains/{도메인명}/{인스턴스명}/bin/start.sh  (MSC 방식으로 기동 하기 때문에 start 시에 CPU가 높을 수 있음) |
| 정지 | /was/jboss7/domains/{도메인명}/{인스턴스명}/bin/shutdown.sh |
| 비상정지 | /was/jboss7/domains/{도메인명}/{인스턴스명}/bin/kill.sh |
| Log 위치 | /log/jboss7 밑에 도메인별로 위치함 |
| Config Filie | 도메인별로 위치함  /was/jboss7/domains/{도메인명}/{인스턴스명}/bin  /was/jboss7/domains/{도메인명}/{인스턴스명}/configuration |
| JVM Heap Size | Min: 4096MB Max: 4096MB MaxPerm : 256MB  /was/jboss7/domains/{도메인명}/{인스턴스명}/bin/env.sh |

## 웹 서버 인스턴스 구성

#### Apache httpd.conf ( Web Servrer 기본 구성 파일 )

/web/jws3/domains/{도메인명}/{인스턴스명}/conf/httpd.conf파일에 아래의 내용을 추가합니다.

|  |
| --- |
| <IfModule remoteip\_module>  RemoteIPHeader X-Client-IP  RemoteIPInternalProxy 127.0.0.1  </IfModule>  Include conf/mod\_jk.conf  Include conf/vhosts.conf |

#### mod\_jk.conf ( # JBoss 연결을 위한 기본 설정 파일 )

/web/jws3/domains/{도메인명}/{인스턴스명}/conf/mod\_jk.conf파일에 아래의 내용을 추가합니다

|  |
| --- |
| # Load mod\_jk module  # Update this path to match your modules location  LoadModule jk\_module /web/jws3/httpd/modules/mod\_jk.so  # Where to find workers.properties  # Update this path to match your conf directory location (put workers.properties next to httpd.conf)  JkWorkersFile /web/jws3/domains/**{도메인명}/{인스턴스명}**/conf/workers.properties  # Where to put jk logs  # Update this path to match your logs directory location (put mod\_jk.log next to access\_log)  #JkLogFile /log/jws3/**{도메인명}/{인스턴스명}**/mod\_jk.log  #JkShmFile /log/jws3/**{도메인명}/{인스턴스명}**/mod\_jk.shm  JkLogFile "|/web/jws3/httpd/sbin/rotatelogs /log/jws3/**{도메인명}/{인스턴스명}**/mod\_jk.log.%Y%m%d 86400"  JkShmFile /log/jws3/**{도메인명}/{인스턴스명}**/mod\_jk.shm  # Set the jk log level [debug/error/info]  JkLogLevel error  # Select the log format  JkLogStampFormat "[%a %b %d %H:%M:%S %Y] "  # JkOptions indicate to send SSL KEY SIZE,  JkOptions +ForwardKeySize +ForwardURICompatUnparsed -ForwardDirectories +ForwardPhysicalAddress +ForwardURIProxy  # JkRequestLogFormat set the request format  JKRequestLogFormat "%w %R %V %T %U %s"  # You can use external file for mount points  # It will be checked for updates each 60 secondes  # The format of the file is : /url=worker  # /examples/\*=loadbalancer  JkMountFile /web/jws3/domains/**{도메인명}/{인스턴스명}**/conf/uriworkermap.properties |

#### workers.properties

/web/jws3/domains/{도메인명}/{인스턴스명}/conf/workers.properties는 뒷단의 JBoss instance 정보 및 connection 설정을 위한 설정파일 입니다.

|  |
| --- |
| # Defind list of workers that will be used  # for mapping requests  # The configuration directives are valid  # for the mod\_jk version 1.2.37 and later  worker.list=wlb\_SV11,wlb\_SV12,wlb\_SV13,wlb\_SV14,jkstatus  worker.jkstatus.type=status  ##################################  # default configuration template #  ##################################  worker.template.type=ajp13  worker.template.lbfactor=1  worker.template.maintain=30  worker.template.socket\_keepalive=True  worker.template.socket\_connect\_timeout=3000  worker.template.socket\_timeout=60  worker.template.ping\_mode=A  worker.template.ping\_timeout=5000  #worker.template.reply\_timeout=30000  worker.template.connection\_pool\_timeout=30  worker.template.connection\_pool\_size=32  worker.template.retries=2  worker.template.recovery\_options=7  #=========================================  # **worker SV11**  #=========================================  # Define standalone worker  worker.P-SV11-O-F11.reference=worker.template  worker.P-SV11-O-F11.host=ap1.p.flb.samsungcard.biz  worker.P-SV11-O-F11.port=11040  worker.P-SV11-O-F12.reference=worker.template  worker.P-SV11-O-F12.host=ap1.p.flb.samsungcard.biz  worker.P-SV11-O-F12.port=11041  worker.P-SV11-O-F21.reference=worker.template  worker.P-SV11-O-F21.host=ap2.p.flb.samsungcard.biz  worker.P-SV11-O-F21.port=11040  worker.P-SV11-O-F22.reference=worker.template  worker.P-SV11-O-F22.host=ap2.p.flb.samsungcard.biz  worker.P-SV11-O-F22.port=11041  # Load-balancing behaviour  worker.wlb\_SV11.type=lb  worker.wlb\_SV11.method=S  worker.wlb\_SV11.sticky\_session=T  worker.wlb\_SV11.session\_cookie=TEZUJSESSIONID  worker.wlb\_SV11.balance\_workers=**P-SV11-O-F11,P-SV11-O-F21,P-SV11-O-F12,P-SV11-O-F22**  #=========================================  # **worker SV12**  #=========================================  # Define standalone worker  worker.P-SV12-O-S11.reference=worker.template  worker.P-SV12-O-S11.host=ap1.p.flb.samsungcard.biz  worker.P-SV12-O-S11.port=11058  worker.P-SV12-O-S21.reference=worker.template  worker.P-SV12-O-S21.host=ap2.p.flb.samsungcard.biz  worker.P-SV12-O-S21.port=11058  # Load-balancing behaviour  worker.wlb\_SV12.type=lb  worker.wlb\_SV12.method=S  worker.wlb\_SV12.sticky\_session=1  worker.wlb\_SV12.session\_cookie=TEZCJSESSIONID  worker.wlb\_SV12.balance\_workers=**P-SV12-O-S11,P-SV12-O-S21**  #=========================================  # **worker SV13**  #=========================================  # Define standalone worker  worker.P-SV13-O-F11.reference=worker.template  worker.P-SV13-O-F11.host=ap1.p.flb.samsungcard.biz  worker.P-SV13-O-F11.port=11060  worker.P-SV13-O-F12.reference=worker.template  worker.P-SV13-O-F12.host=ap1.p.flb.samsungcard.biz  worker.P-SV13-O-F12.port=11061  worker.P-SV13-O-F21.reference=worker.template  worker.P-SV13-O-F21.host=ap2.p.flb.samsungcard.biz  worker.P-SV13-O-F21.port=11060  worker.P-SV13-O-F22.reference=worker.template  worker.P-SV13-O-F22.host=ap2.p.flb.samsungcard.biz  worker.P-SV13-O-F22.port=11061  # Load-balancing behaviour  worker.wlb\_SV13.type=lb  worker.wlb\_SV13.method=S  worker.wlb\_SV13.sticky\_session=1  worker.wlb\_SV13.session\_cookie=TEZMUJSESSIONID  worker.wlb\_SV13.balance\_workers=**P-SV13-O-F11,** **P-SV13-O-F12,P-SV13-O-F21, P-SV13-O-F22**  #=========================================  # **worker SV14**  #=========================================  # Define standalone worker  worker.P-SV14-O-S11.reference=worker.template  worker.P-SV14-O-S11.host=ap1.p.flb.samsungcard.biz  worker.P-SV14-O-S11.port=11078  worker.P-SV14-O-S21.reference=worker.template  worker.P-SV14-O-S21.host=ap2.p.flb.samsungcard.biz  worker.P-SV14-O-S21.port=11078  # Load-balancing behaviour  worker.wlb\_SV14.type=lb  worker.wlb\_SV14.method=S  worker.wlb\_SV14.sticky\_session=1  worker.wlb\_SV14.session\_cookie=TEZAJSESSIONID  worker.wlb\_SV14.balance\_workers=**P-SV14-O-S11,P-SV14-O-S21** |

#### uriworkermap.properties 파일 작성

/web/jws3/domains/{도메인명}/{인스턴스명}/conf/uriworkermap\_{도메인코드}.properties는 뒷단의 JBoss에게 sv11, sv12, sv13, sv14로 분기하기 위한 URL Pattern 을 설정하는 파일입니다.

|  |
| --- |
| uriworkermap.properties |
| # uri base로 특정 url 패턴이 오게 되면 wlb worker를 통해 뒷단의 JBoss EAP 와 연동이 되게 한다.  # /servlet 와 그 하위를 wlb worker 로 매핑  /version/\*=wlb\_SV11 # context-root 가 version 모든 것을 worker loadbalancer 로 포워딩  /test/\*=wlb\_SV11 # context-root 가 test 모든 것을 worker loadbalancer 로 포워딩  # /img 아래의 \*.jpg 는 웹서버에 있으므로 **매핑에서 제외**  !/img/\*.jpg=wlb\_F11  # 확장자 base로 특정 확장자가 오게 되면 wlb worker를 통해 뒷단의 JBoss EAP 와 연동이 되게 한다.  # \*.jsp 와 \*.do 는 wlb\_SV11 worker 로 매핑  /\*.jsp=wlb\_F11  /\*.do=wlb\_F11 |

/web/jws3/domains/{도메인명}/{인스턴스명}/conf 파일 하위에 아래와 같이 정의된다.

uriworkermap\_sv11.properties: SV11 balance\_workers로 포워딩

uriworkermap\_sv12.properties: SV12 balance\_workers로 포워딩

uriworkermap\_sv13.properties: SV13 balance\_workers로 포워딩

uriworkermap\_sv14.properties: SV13 balance\_workers로 포워딩

ex)

|  |
| --- |
| ######################  /test/\*=wlb\_SV11  /version/\*=wlb\_SV11  /servlet/\*=wlb\_SV11  /ezwel-image/\*=wlb\_SV11  /\*.jsp=wlb\_SV11  /\*.do=wlb\_SV11  /\*.ez=wlb\_SV11  /jkstatus\*=jkstatus |

## WAS 서버 인스턴스 구성

### 인스턴스 구성 위치

Jboss Server의 도메인별로 구성된 디렉토리 아래의 configuration 디렉토리에 설치되어 있습니다

**/was/jboss7/domains/{도메인명}/{인스턴스명}/configuration**

### 인스턴스 구성파일

실제 인스턴스 구성 파일 : ***standalone-ha.xml***

해당 파일은 각도메인 디렉토리 밑의 서버 Instance 디렉토리 밑에 위치합니다

**/was/jboss7/domains/{도메인명}/{인스턴스명}/configuration**

#### Datasource 설정

데이터베이스에 연결하는 방법으로 RDBMS와 연결을 인터페이스를 사용해 연결할 수 있도록 하는 방식입니다.(7.3 Datasource 기본 설정 항목의 설명 참조)

|  |
| --- |
| Standalone-ha.xml |
| <datasource jta="true" jndi-name="java:/dataSourceMain" pool-name="dataSourceMain" enabled="true"  use-java-context="true" use-ccm="true" statistics-enabled="true">  <connection-url>  jdbc:oracle:thin:@(description=(address\_list=(load\_balance=off)(failover=on  )(address=(protocol=tcp)(host=db-v1.p.flb.samsungcard.biz)(port=1579))  (address=(protocol=tcp)(host=db-v2.p.flb.samsungcard.biz)(port=1579)))  (connect\_data=(service\_name=PFLB)(failover\_mode=(type=select)(method=basic))))  </connection-url>  <driver>jdbc.oracle</driver>  <new-connection-sql>select 1 from dual</new-connection-sql>  <pool>  <min-pool-size>20</min-pool-size>  <max-pool-size>20</max-pool-size>  <prefill>true</prefill>  </pool>  <security>  <security-domain>encrypted-ds-dataSourceMain</security-domain>  </security>  <validation>  <valid-connection-checker class-name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleValidConnectionChecker"/>  <check-valid-connection-sql>select 1 from dual</check-valid-connection-sql>  <validate-on-match>false</validate-on-match>  <background-validation>true</background-validation>  <background-validation-millis>600000</background-validation-millis>  <stale-connection-checker class-name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleStaleConnectionChecker"/>  <exception-sorter class-name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleExceptionSorter"/>  </validation>  <timeout>  <blocking-timeout-millis>5000</blocking-timeout-millis>  <idle-timeout-minutes>10</idle-timeout-minutes>  <allocation-retry>1</allocation-retry>  <allocation-retry-wait-millis>1000</allocation-retry-wait-millis>  </timeout>  <statement>  <track-statements>false</track-statements>  <prepared-statement-cache-size>1</prepared-statement-cache-size>  <share-prepared-statements>false</share-prepared-statements>  </statement>  </datasource> |

#### Global-module 설정

AS-IS에서 CLASSPATH를 사용해 사용했던 라이브러리를 JBOSS에서는 모듈로 등록 하여 사용합니다.

* 라이브러리 디렉토리

$JBOSS\_HOME/modules.ext/com/softforum/xdbe/main/conf/xdsp\_java.jar

* 라이브러리 모듈 정의

$JBOSS\_HOME/modules.ext/com/softforum/xdbe/main/module.xml

|  |
| --- |
| module.xml |
| <module xmlns="urn:jboss:module:1.1" name="com.softforum.xdbe">  <resources>  <resource-root path="./conf"/>  <resource-root path="./conf/xdsp\_java.jar"/>  <!-- Insert resources here -->  </resources>  </module> |

Module을 설정 한 후 standalone-ha.xml파일에 module을 정의한다.

|  |
| --- |
| Standalone-ha.xml |
| <global-modules>  <module name="org.bouncycastle" slot="main"/>  <module name="com.softforum.xdbe" slot="main"/>  </global-modules> |

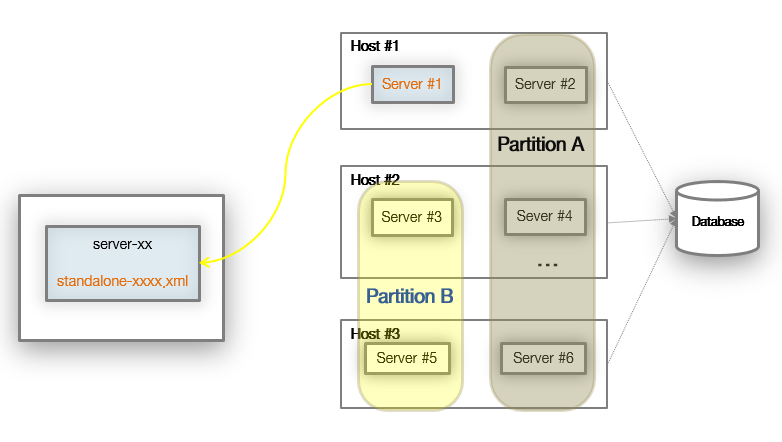
#### Ezwel-image handlers 설정

AS-IS에서 ezwel-image 심볼릭 싱크를 사용하여 request요청시 이미지 파일을 찾는 방법이였고 이와 같은 방법은 인스턴스 구동시에 1분정도 걸립니다. JBOSS에서는 인스턴스 구동 시간을 줄이기 위해 WAS서버 내에 Handler를 등록하여 인스턴스 구동 시간은 줄이고 정상적으로 이미지를 찾을 수 있도록 설정하였습니다.

|  |
| --- |
| Standalone-ha.xml |
| <handlers>  <file name="ezwel-image" path="/sv1\_attach/ezwel-image"/>  </handlers> |

# Jboss Enterprise Application Platform – standalone 모드 표준구성

## 표준 구성도



## Directory Sturcture

### JBoss Domain 모드 설치 디렉토리 구조

JBoss domain 모드 설치를 위한 디렉토리구조는 다음과 같습니다.

|  |  |  |
| --- | --- | --- |
| **Directory** | **설명** | **NamingRule** |
| /was/jboss7 | Jboss7계정 HOME | 설치 계정 |
| /was/jboss7/jboss-eap-7.0 | JBoss 엔진HOME | ${계정홈}/**${솔루션명+설치버전}** |
| /was/jboss7/domains | Domain HOME | ${계정홈}/**domains** |
| /was/jboss7/domains/{Domain\_Name}/{Server\_Name} | 각 Server\_HOME | ${계정홈}/domains**/${Domain\_Name}/{Server\_Name}** |

## JBoss EAP 엔진 설치

JBoss는 GUI 기반의 설치(X 필요) 또는 zip 형태의 압축을 푸는 것으로 설치를 수행합니다. 일반적으로 압축 형태의 설치 파일을 다운받아 설치하는 것이 속도도 빠릅니다.

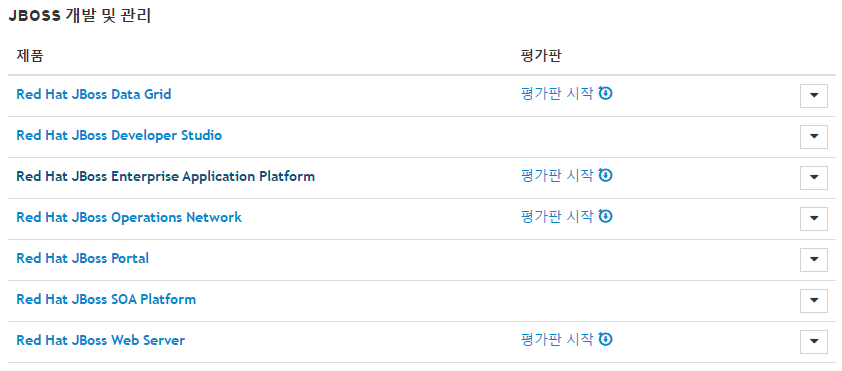
소프트웨어를 다운받기 위해서는 레드햇 서브스크립션이 필요하며, 아래의 URL에서 다운받으실 수 있습니다.

* 다운로드 URL: <http://access.redhat.com>

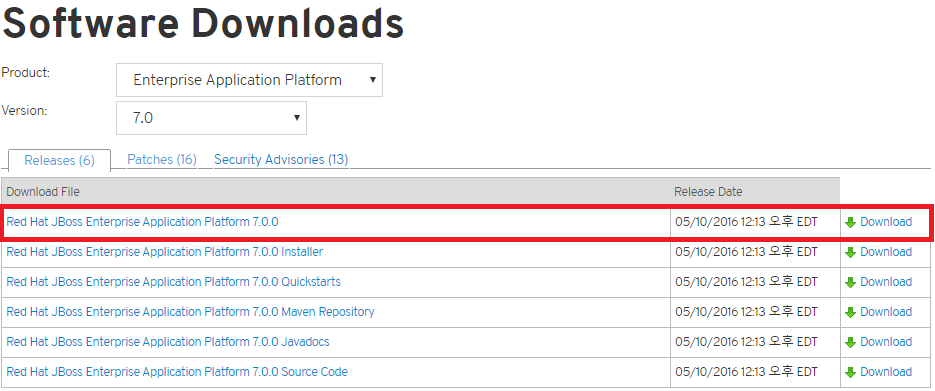
### JBoss EAP 다운로드

로그인을 수행하면 하단에 다운로드 링크가 있으며, 필요한 JBoss 링크를 통해 들어갑니다.



중간에 위치 한 Red Hat JBoss Enterprise Application Platform이동

Application Platform은 다양한 버전을 포함하고 있으며 최신 버전으로 각 항목에 대한 다운로드 링크를 눌러 소프트웨어를 다운받습니다.



### JBoss EAP 설치

설치 방법은 앞서 설명했던 것과 같이 서버에 업로드를 수행한 후 압축을 풀면 설치가 됩니다.

일반적으로 엔진과 애플리케이션 위치로 구성을 시키고 있으며, 엔진은 /opt/was 하위로 설정을 잡습니다.

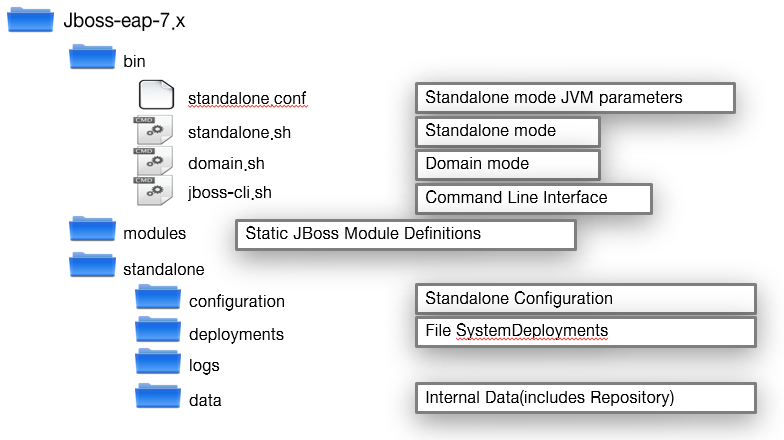
JBoss설치 파일을 /opt/was 로 업로드 후 아래의 명령어를 통하여 설치를 진행합니다.

|  |
| --- |
| $ unzip jboss-eap-7.0.0.zip  $ rm -f jboss-eap-7.0.0.zip |

NOTE. 엔진 설치는 파일 업로드 시간 + 압축 해제 시간이 전부입니다.

### JBoss EAP 엔진 디렉토리 구조

JBoss의 디렉토리 구조는 다음과 같습니다.



인스턴스 생성을 위한 기본 teamplate 정보는 standalone 및 domain 폴더에 위치하고 있습니다.

## JBoss standalone설치 및 설정

JBoss 의 각 서비스별 인스턴스 생성을 위하여 JBoss Engine 폴더의 standalone 폴더를 복사하여 생성을 진행합니다.

### standalone 디렉터리 구조

각 도메인 디렉토리 하위 정보는 다음과 같이 구성됩니다.

/was/jboss7/domains/{도메인명}/{인스턴스명}

|  |  |
| --- | --- |
| **디렉터리** | **설명** |
| bin | 인스턴스 실행, 중지, 강제종료, 로그 보기, 스레드 덤프 생성 등 운영 필요한 스크립트들이 보관되어 있다.   * 별도의 디렉토리 생성 및 shell script 생성 필요 |
| configuration | JBoss 메인구성 파일 보관 폴더 |
| data | JBoss 실행시 생성되는 파일들이 보관되는 디렉토리이다. |
| deployments | 사용안함 |
| Lib | 사용안함 |
| logs | 사용안함  /log/jboss7 디렉토리를 사용하여 log를 적재함 |
| tmp | 서비스 tmp 폴더 |

### configuration 파일

domain 설정에 필요한 configuration 폴더내의 필수 설정 파일의 내용을 확인합니다.

위치 : /was/jboss7/domains/{도메인명}/{인스턴스명}/configuration/standalone-ha.xml

ex) /was/jboss7/domains/P-MS71-O/P-MS71-O-F11/configuration

|  |  |
| --- | --- |
| **파일명** | **설명** |
| standalone-ha.xml | 인스턴스에 대한 설정 파일   * profile 설정 * deployments * datasource * logging |

### profile 의 이해

standalone.xml 파일내에 server 내의 실제 인스턴스 역할을 하는 서버그룹에서 사용될 Service 의 속성을 정의 해 놓은 profile 설정입니다.

|  |  |  |
| --- | --- | --- |
| **profile** | **내용** | **파일명** |
| default | 가장 기본이 되는 profile 이며 기본 web 구성으로 되어 있습니다.   * Basic Web Application Server | standalone.xml |
| ha | 기본 default 설정에 ha (high availability) 의 session Clustering 설정이 추가되어 있습니다.   * default + JGroups ( for Clustering ) | standalone-ha.xml |
| full | 기본 default 설정에 JMS 서비스가 추가 되어 있습니다.   * default + JMS ( Java Messaging Service ) | standalone-full.xml |
| full-ha | EAP 전체 서비스 기능을 사용할 수 있습니다.   * default + JMS ( Java Messaging Service ) + JGroups ( for Clustering ) | standalone-full-ha.xml |

### MSC방식의 이해

계층형 클래스로더 방식에서는 상위 계층을 검색하기 때문에 시간이 많이 소요되고 애플리케이션에서 사용하지 않는 클래스도 로드 됩니다. 계층형 방식은 메모리 사용량이 많아지고 시작 시간이 오래 걸리지만 반면 MSC방식(module service container)은 모듈의 의존하는 클래스만 검색하기 때문에 클래스 검색이 빠르고 필요한 모듈을 필요할 때에 로드 되기 때문에 실행 속도가 빠르며 메모리 사용량이 낮습니다. 하지만 최초 기동 시에 CPU 사용량이 높을 수도 있습니다.

### 관리자 계정 생성 및 추가

해당 JBoss 서비스의 관리를 위하여 서비스의 관리자 계정 생성 합니다.

생성방법은 다음과 같습니다.

/was/jboss7/domains/{도메인명}/{인스턴스명}/bin/add-user.sh 를 실행합니다.

ⓐ - Management/Application User 선택 (default = a)

ⓑ - 관리영역 (default = ManagementRealm)

ⓒ - 등록할 관리자 ID

ⓓ - 등록할 관리자 PassWord

|  |
| --- |
| [jboss@ticoap01 ~]$ ./add-user.sh  ================================================  What type of user do you wish to add?  a) Management User (mgmt-users.properties)  b) Application User (application-users.properties)  **(a): a**  Enter the details of the new user to add.  Realm (ManagementRealm) :  **Username : jboaadmin**  **Password : !tadmin123**  **Re-enter Password : !admin123**  About to add user 'test' for realm 'ManagementRealm'  **Is this correct yes/no? yes**  Added user 'test' to file '/app/jboss/servers/standalone\_ha\_11/configuration/mgmt-users.properties'  Added user 'test' to file '/app/jboss/jboss-eap-6.1/domain/configuration/mgmt-users.properties'  Is this new user going to be used for one AS process to connect to another AS process?  e.g. for a slave host controller connecting to the master or for a Remoting connection for server to server EJB calls.  **yes/no? yes**  To represent the user add the following to the server-identities definition <secret value="IXRlc3QxMjM=" /> |

# JBoss 인스턴스 생성

## 인스턴스 생성

Jboss standalone인스턴스 설치는 Naming Rule에 따라 **/was/jboss7/{업무도메인}/{업무서버명}** 라는 디렉토리를 사용하도록 합니다.

### Standalone template 복사

기본 Jboss engine 폴더의 standalone template 폴더를 복사합니다.

eX)

cp –fR /was/jboss7/jboss-eap-7.0/standalone /was/jboss7/domains/{도메인명}/{인스턴스명}

### 실행 스크립트의 구성

생성된 인스턴스를 구동하기 위한 스크립트 파일을 생성합니다.

### 실행 스크립트의 구성

#### env.sh

모든 설정이 env.sh를 통해서 이루어지며 JVM 옵션 등은 env.sh파일만 조작하시면 됩니다.(JVM옵션, JNDI 포트 포함)

|  |
| --- |
| **env.sh** |
| #!/bin/sh  #######################################  # JBoss Default Setting #  #######################################  export DATE=`date +%Y%m%d%H%M%S`  export JBOSS\_HOME=/was/jboss7/jboss-eap-7.0  export DOMAIN\_BASE=/was/jboss7/domains/도메인명명  export SERVER\_NAME=인스턴스명  export SERVER\_HOME=$DOMAIN\_BASE/$SERVER\_NAME  #######################################  # Configration File #  #######################################  export CONFIG\_FILE=standalone-ha.xml  export HOST\_NAME=ap1.p.flb.samsungcard.biz  export NODE\_NAME=$SERVER\_NAME  export PORT\_OFFSET=인스턴스별 구분 포트 OFFSET  export LOG\_HOME=/log/jboss7/도메인명/인스턴스명  export JBOSS\_USER=jboss7  #######################################  # Bind Address #  #######################################  export BIND\_ADDR=$HOST\_NAME  export MULTICAST\_ADDR=230.71.30.31  export JMS\_MULTICAST\_ADDR=231.72.30.31  export MODCLUSTER\_MULTICAST\_ADDR=224.73.30.31  export MGMT\_ADDR=$BIND\_ADDR  export CONTROLLER\_IP=$MGMT\_ADDR  let CONTROLLER\_PORT=9990+$PORT\_OFFSET  export CONTROLLER\_PORT  #######################################  # port usage #  #######################################  #=============================  #ex) export HTTP\_CUSTOM\_PORT=8080  #ex) export AJP\_CUSTOM\_PORT=8309  #=============================  export HTTP\_DEFAULT\_PORT=8080  export AJP\_DEFAULT\_PORT=8009  if [ "x$HTTP\_CUSTOM\_PORT" = "x" ]; then  HTTP\_PORT=$HTTP\_DEFAULT\_PORT  AJP\_PORT=$AJP\_DEFAULT\_PORT  let HTTP\_FINAL\_PORT=$HTTP\_PORT+$PORT\_OFFSET  let AJP\_FINAL\_PORT=$AJP\_PORT+$PORT\_OFFSET  export HTTP\_FINAL\_PORT  export AJP\_FINAL\_PORT  else  if [ "x$AJP\_CUSTOM\_PORT" = "x" ]; then  AJP\_PORT=$AJP\_DEFAULT\_PORT  let HTTP\_PORT=$HTTP\_CUSTOM\_PORT-$PORT\_OFFSET  let AJP\_FINAL\_PORT=$AJP\_PORT+$PORT\_OFFSET  export HTTP\_FINAL\_PORT=$HTTP\_CUSTOM\_PORT  export AJP\_FINAL\_PORT  else  let HTTP\_PORT=$HTTP\_CUSTOM\_PORT-$PORT\_OFFSET  let AJP\_PORT=$AJP\_CUSTOM\_PORT-$PORT\_OFFSET  export HTTP\_FINAL\_PORT=$HTTP\_CUSTOM\_PORT  export AJP\_FINAL\_PORT=$AJP\_CUSTOM\_PORT  fi  fi  export HTTP\_PORT  export AJP\_PORT  export JGROUP\_TCP\_PORT=7600  let JGROUP\_TCP\_PORT=7600+$PORT\_OFFSET  export JGROUP\_TCP\_PORT  export JMXREMOTE\_TCP\_PORT=5999  let JMXREMOTE\_TCP\_PORT=5999+$PORT\_OFFSET  export JMXREMOTE\_TCP\_PORT  export JBOSS\_JGROUPS\_STACK=tcp  export JBOSS\_JGROUPS\_TCPPING\_INITIAL\_HOSTS=ap1.p.flb.samsungcard.biz[7600],ap2.p.flb.samsungcard.biz[7600],ap1.p.flb.samsungcard.biz[7601],ap2.p.flb.samsungcard.biz[7601]  export JBOSS\_JGROUPS\_TCPPING\_NUM\_INITIAL\_MEMBERS=2  export LAUNCH\_JBOSS\_IN\_BACKGROUND=true  ######################################################  ##### Logging Directory Setup  ######################################################  export LOG\_DIR=${LOG\_HOME}/log  export GC\_LOG\_DIR=${LOG\_HOME}/gclog  export HEAP\_DUMP\_DIR=${LOG\_HOME}/heaplog  export LOG\_LEVEL=INFO  ##### JBoss System module and User module directory #####  export JBOSS\_MODULEPATH=$JBOSS\_HOME/modules:$JBOSS\_HOME/modules.ext  # JVM Options : Server  export JAVA\_OPTS="-server $JAVA\_OPTS"  # JVM Options : Memory  export JAVA\_OPTS="$JAVA\_OPTS -Xms4096m -Xmx4096m"  export JAVA\_OPTS="$JAVA\_OPTS -XX:MetaspaceSize=256m -XX:MaxMetaspaceSize=256m"  export JAVA\_OPTS="$JAVA\_OPTS -XX:+UseG1GC"  export JAVA\_OPTS="$JAVA\_OPTS -XX:+UnlockDiagnosticVMOptions"  export JAVA\_OPTS="$JAVA\_OPTS -XX:+G1SummarizeConcMark"  export JAVA\_OPTS="$JAVA\_OPTS -XX:InitiatingHeapOccupancyPercent=35"  export JAVA\_OPTS="$JAVA\_OPTS -XX:ConcGCThreads=2"  export JAVA\_OPTS="$JAVA\_OPTS -XX:+UseLargePagesInMetaspace"  export JAVA\_OPTS="$JAVA\_OPTS -verbose:gc"  export JAVA\_OPTS="$JAVA\_OPTS -Xloggc:$GC\_LOG\_DIR/gc\_$DATE.log"  export JAVA\_OPTS="$JAVA\_OPTS -XX:+PrintGCDetails"  export JAVA\_OPTS="$JAVA\_OPTS -XX:+PrintGCTimeStamps"  export JAVA\_OPTS="$JAVA\_OPTS -XX:+PrintAdaptiveSizePolicy"  export JAVA\_OPTS="$JAVA\_OPTS -XX:+ExplicitGCInvokesConcurrent"  export JAVA\_OPTS="$JAVA\_OPTS -XX:+DisableExplicitGC"  export JAVA\_OPTS="$JAVA\_OPTS -XX:+HeapDumpOnOutOfMemoryError"  export JAVA\_OPTS="$JAVA\_OPTS -XX:HeapDumpPath=$HEAP\_DUMP\_DIR/$SERVER\_NAME.hprof.$DATE"  export JAVA\_OPTS="$JAVA\_OPTS -Djava.net.preferIPv4Stack=true"  export JAVA\_OPTS="$JAVA\_OPTS -Dorg.jboss.resolver.warning=true"  export JAVA\_OPTS="$JAVA\_OPTS -Dsun.rmi.dgc.client.gcInterval=3600000 "  export JAVA\_OPTS="$JAVA\_OPTS -Dsun.rmi.dgc.server.gcInterval=3600000"  export JAVA\_OPTS="$JAVA\_OPTS -Djava.awt.headless=true"  export JAVA\_OPTS="$JAVA\_OPTS -Djboss.modules.system.pkgs=org.jboss.byteman,scouter,org.jboss.logmanager,jennifer"  # Linux Large Page Setting  #for darwin  export JBOSS\_BASE\_DIR="$DOMAIN\_BASE/$SERVER\_HOME"  export JAVA\_OPTS="$JAVA\_OPTS -Djboss.server.base.dir=$DOMAIN\_BASE/$SERVER\_NAME"  export JAVA\_OPTS="$JAVA\_OPTS -Djboss.server.log.dir=$LOG\_DIR"  export JAVA\_OPTS="$JAVA\_OPTS -Djboss.socket.binding.port-offset=$PORT\_OFFSET"  export JAVA\_OPTS="$JAVA\_OPTS -Djboss.node.name=$NODE\_NAME"  export JAVA\_OPTS="$JAVA\_OPTS -Djboss.server.base.dir=$DOMAIN\_BASE/$SERVER\_NAME"  export JAVA\_OPTS="$JAVA\_OPTS -Djboss.bind.address=$BIND\_ADDR"  export JAVA\_OPTS="$JAVA\_OPTS -Djboss.bind.address.private=$BIND\_ADDR"  export JAVA\_OPTS="$JAVA\_OPTS -Djboss.bind.address.management=$MGMT\_ADDR"  export JAVA\_OPTS="$JAVA\_OPTS -Djboss.socket.binding.port-offset=$PORT\_OFFSET"  export JAVA\_OPTS="$JAVA\_OPTS -Djboss.default.jgroups.stack=$JBOSS\_JGROUPS\_STACK"  export JAVA\_OPTS="$JAVA\_OPTS -Djboss.default.jgroups.stack.tcp.tcpping.initial\_hosts=$JBOSS\_JGROUPS\_TCPPING\_INITIAL\_HOSTS"  export JAVA\_OPTS="$JAVA\_OPTS -Djboss.default.jgroups.stack.tcp.tcpping.num\_initial\_members=$JBOSS\_JGROUPS\_TCPPING\_NUM\_INITIAL\_MEMBERS"  export JAVA\_OPTS="$JAVA\_OPTS -Djgroups.bind\_addr=${BIND\_ADDR}"  export JAVA\_OPTS="$JAVA\_OPTS -Djgroups.tcp.bind\_port=${JGROUP\_TCP\_PORT}"  export JAVA\_OPTS="$JAVA\_OPTS -Djboss.default.multicast.address=$MULTICAST\_ADDR"  export JAVA\_OPTS="$JAVA\_OPTS -Djboss.messaging.group.address=$JMS\_MULTICAST\_ADDR"  export JAVA\_OPTS="$JAVA\_OPTS -Djboss.modcluster.multicast.address=$MODCLUSTER\_MULTICAST\_ADDR"  export JAVA\_OPTS="$JAVA\_OPTS -Djboss.http.port=$HTTP\_PORT"  export JAVA\_OPTS="$JAVA\_OPTS -Djboss.http.final.port=$HTTP\_FINAL\_PORT"  export JAVA\_OPTS="$JAVA\_OPTS -Djboss.ajp.port=$AJP\_PORT"  export JAVA\_OPTS="$JAVA\_OPTS -Djboss.ajp.final.port=$AJP\_FINAL\_PORT"  export JAVA\_OPTS="$JAVA\_OPTS -Xbootclasspath/p:/was/jboss7/jboss-eap-7.0/modules/system/layers/base/org/jboss/logmanager/main/jboss-logmanager-2.0.3.Final-redhat-1.jar"  export JAVA\_OPTS="$JAVA\_OPTS -Djava.util.logging.manager=org.jboss.logmanager.LogManager"  export JAVA\_OPTS="$JAVA\_OPTS -Dserver.mode=prod"  export JAVA\_OPTS="$JAVA\_OPTS -Dezwel.server.type=real"  # JVM crash  export JAVA\_OPTS="$JAVA\_OPTS -Dsun.zip.disableMemoryMapping=true"  # Use log4j in application  export JAVA\_OPTS=" $JAVA\_OPTS -Dorg.jboss.as.logging.per-deployment=false"  # JENNIFER SETTING  export JAVA\_OPTS="$JAVA\_OPTS -Djennifer.config=/was/jennifer/agent.java/conf/301.conf"  export JAVA\_OPTS="$JAVA\_OPTS -javaagent:/was/jennifer/agent.java/jennifer.jar"  #######################################  # XecureWeb LIB PATH #  #######################################  export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:/product/xecureweb\_ver7/lib/java/jni  #######################################  # D'Amo LIB PATH #  #######################################  export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:/product/dbenc/64  echo "================================================"  echo "JAVA\_HOME=$JAVA\_HOME"  echo "JBOSS\_HOME=$JBOSS\_HOME"  echo "DOMAIN\_BASE=$DOMAIN\_BASE"  echo "SERVER\_NAME=$SERVER\_NAME"  echo "CONFIG\_FILE=$CONFIG\_FILE"  echo "BIND\_ADDR=$BIND\_ADDR"  echo "PORT\_OFFSET=$PORT\_OFFSET"  echo "MULTICAST\_ADDR=$MULTICAST\_ADDR"  echo "CONTROLLER=$CONTROLLER\_IP:$CONTROLLER\_PORT"  echo "HTTP\_FINAL\_PORT=$HTTP\_FINAL\_PORT"  echo "AJP\_FINAL\_PORT=$AJP\_FINAL\_PORT"  echo "================================================" |

#### start.sh

start.sh는 인스턴스를 구동시킬 때 사용하며 로그파일의 백업 옵션등을 설정할 수 있습니다.

(MSC 방식으로 기동 하기 때문에 start 시에 CPU가 높을 수 있음)

|  |
| --- |
| **start.sh** |
| #!/bin/sh  DATE=`date +%Y%m%d%H%M%S`  . ./env.sh  PID=`ps -ef | grep java | grep "=$SERVER\_NAME " | awk '{print $2}'`  echo $PID  if [ e$PID != "e" ]  then  echo "JBoss SERVER - $SERVER\_NAME is already RUNNING..."  exit;  fi  UNAME=`id -u -n`  if [ e$UNAME != "e$JBOSS\_USER" ]  then  echo "Use $JBOSS\_USER account to start JBoss SERVER - $SERVER\_NAME..."  exit;  fi  if [ ! -d "${LOG\_DIR}" ];  then  mkdir -p ${LOG\_DIR}  fi  if [ ! -d "${GC\_LOG\_DIR}" ];  then  mkdir -p ${GC\_LOG\_DIR}  fi  if [ ! -d "${HEAP\_DUMP\_DIR}" ];  then  mkdir -p ${HEAP\_DUMP\_DIR}  fi  mv ${LOG\_DIR}/$SERVER\_NAME.out ${LOG\_DIR}/$SERVER\_NAME.out.$DATE  nohup $JBOSS\_HOME/bin/standalone.sh -DSERVER=$SERVER\_NAME -P=$DOMAIN\_BASE/$SERVER\_NAME/bin/env.properties -c $CONFIG\_FILE >> ${LOG\_DIR}/$SERVER\_NAME.out &  if [ e$1 = "enotail" ]  then  echo "Starting... $SERVER\_NAME"  exit;  fi |

#### shutdown.sh

shutdown.sh 파일은 인스턴스를 종료할 때 사용합니다.

|  |
| --- |
| **shutdown.sh** |
| #!/bin/sh  . ./env.sh  unset JAVA\_OPTS  FIRST\_PID=`ps -ef | grep java | grep "=$SERVER\_NAME " | awk '{print $2}'`  echo "First Process ID = $FIRST\_PID"  $JBOSS\_HOME/bin/jboss-cli.sh --connect --controller=$CONTROLLER\_IP:$CONTROLLER\_PORT --command=:shutdown --timeout=5000  sleep 5;  LAST\_PID=`ps -ef | grep java | grep "=$SERVER\_NAME " | awk '{print $2}'`  echo "LAST Process ID = $LAST\_PID"  if [ e$LAST\_PID != "e" ]  then  if [ "$LAST\_PID" = "$FIRST\_PID" ]  then  echo "JBoss SERVER - $SERVER\_NAME is Stopping..."  kill -9 $FIRST\_PID;  fi  fi |

#### kill.sh

kill.sh 파일은 kill -9 명령으로 java process를 즉시 종료할 때 사용합니다.

Linux OS의 경우는 다음의 script를 이용합니다.

|  |
| --- |
| **kill.sh** |
| #!/bin/sh  . ./env.sh  ps -ef | grep java | grep "SERVER=$SERVER\_NAME " | awk {'print "kill -9 " $2'} | sh -x |

### 관리 스크립트의 구성

#### dump.sh

dump.sh 파일은 인스턴스가 느려지거나, JVM의 현재 상황을 보려고 Thread Dump를 얻으려고 할 때 사용하는 스크립트입니다.

|  |
| --- |
| **dump.sh** |
| #!/bin/sh  . ./env.sh  for count in 1 2 3 4 5; do  echo "Thread Dump : $count"  for i in `ps -ef | grep java | grep "SERVER=$SERVER\_NAME " | awk '{print $2}'`;do  echo "+kill -3 $i"  kill -3 $i  echo "sleep 1 sec"  sleep 1  done  echo "done"  sleep 3  done |

#### status.sh

status.sh 스크립트는 프로세스가 동작 중인지 체크하기 위한 스크립트입니다.

|  |
| --- |
| **status.sh** |
| #!/bin/sh  . ./env.sh  ps -ef | grep java | grep "SERVER=$SERVER\_NAME " |

#### tail.sh

tail.sh 는 nohup로그를 보기 위한 스크립트입니다.

|  |
| --- |
| **tail.sh** |
| #!/bin/sh  . ./env.sh  tail -f $LOG\_DIR/server.log |

#### jboss-cli.sh

jboss-clil.sh 는 Jboss Command Line Interface 접속을 위한 스크립트입니다.

|  |
| --- |
| **jboss-cli.sh** |
| #!/bin/sh  . ./env.sh  unset JAVA\_OPTS  export JAVA\_OPTS="-Djava.awt.headless=false $JAVA\_OPTS"  $JBOSS\_HOME/bin/jboss-cli.sh --controller=$CONTROLLER\_IP:$CONTROLLER\_PORT --connect $@ |

# Jboss Enterprise Application Server + 웹서버 연동

Apache HTTP(Application Platform 7.x Apache HTTP Server for RHEL 7 x86\_64) + mod\_jk.so(Application Platform 7.x Webserver Connector Natives for RHEL 7 x86\_64) + JBoss EAP 7.x 환경에서 JBoss 와 Apache HTTP 서버 간의 세션 클러스터링 및 로드 밸런싱 환경 구성을 위하여 다음의 설정을 진행합니다.

## Directory Sturcture

### JBoss EWS (웹서버) 설치 홈 디렉토리

JBoss 연동 JBoss-EWS (Apache) 설치를 위한 디렉토리구조는 다음과 같습니다.

|  |  |  |
| --- | --- | --- |
| **Directory** | **설명** | **내용** |
| /web/jws3/httpd | HTTPD\_HOME | JBoss EWS 웹서버 설치 홈 디렉토리 |

### JBoss EWS 디렉토리 구조

JBoss EWS 가 설치된 디렉토리구조는 다음과 같습니다.

|  |  |
| --- | --- |
| **Directory** | **설명** |
| ${ HTTPD \_HOME}/conf | Apache의 메인 설정 파일이 보관되어 있는 디렉토리 |
| ${HTTPD\_HOME}/conf.d | 추가 모듈 설정이 보관되는 디렉토리 JBoss연동을 위한 설정 파일 mod\_jk.conf등을 배치 \*.conf 파일이 모두 로딩된다. |
| ${HTTPD\_HOME}/modules | Apache 동적 모듈 보관 디렉토리 Apache – JBoss 연동을 위한 mod\_jk.so파일 |
| ${HTTPD\_HOME}/logs | Apache 발생 로그 저장 디렉토리 |
| ${HTTPD\_HOME}/run | Apache 실행시 관련 정보(PID등) 저장 디렉토리 |
| ${HTTPD\_HOME}/sbin | Apache 인스턴스 실행 및 운영 스크립트 보관 디렉토리 |

### JBoss EWS 연동 설정 파일

아래의 표에서 JBoss Http Session Clustering 구성에 필요한 파일과 위치를 확인 할 수 있다.

|  |  |  |  |
| --- | --- | --- | --- |
| 구분 | 파일명 | 위치 | 설명 |
| Apache | httpd.conf | ${HTTPD\_HOME}/conf | RHEL 6 이상일 경우에는 추가할 내용 없음 |
| mod\_jk.so | ${HTTPD\_HOME}/modules | 다운로드 후 $HTTPD\_HOME/modules 디렉터리로 복사 |
| mod\_jk.conf | ${HTTPD\_HOME}/conf.d | mod\_jk 모듈을 로드하고 worker 파일 및 uri mapping 파일을 설정 |
| workers.properties | ${HTTPD\_HOME}/conf.d | JBoss 인스턴스별로 worker 를 설정 |
| uriworkermap.properties | ${HTTPD\_HOME}/conf.d | URL 과 Worker 맵핑 |
| JBoss | standalone\*.xml | /was/jboss7/domains/{도메인명}/{업무서버영}/configuration/standalone\*.xml | jvmRoute 의 이름과 maxThreads 수 조정 |
| web.xml | $Application/WEB-INF/ | web.xml 파일에<distributable/>를 추가 |

## JBoss EWS엔진의 설치

### JBoss EWS 다운로드

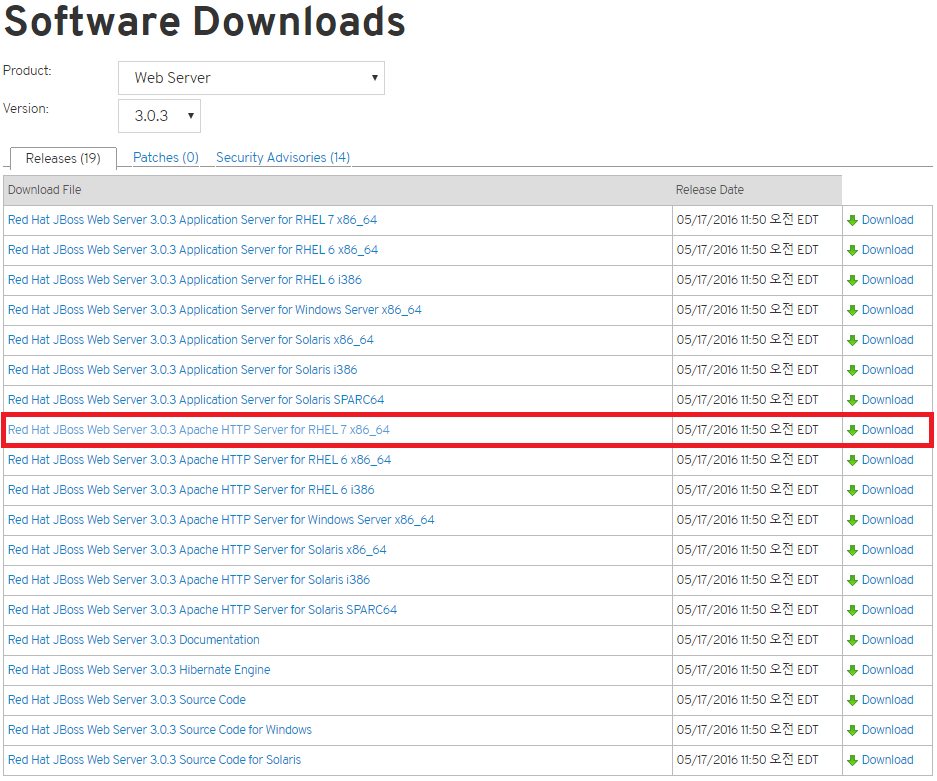
고객께서 직접 설치시 아래의 URL을 통해 다운로드를 받으실 수 있습니다.

<http://access.redhat.com>

다운로드🡪JBOSS Web Server 선택합니다.



필요한 OS에 맞는 EWS를 선택하여 다운로드를 받습니다.



### JBoss EWS 설치

#### 압축 해제

설치 방법은 앞서 설명했던 것과 같이 서버에 업로드를 수행한 후 압축을 풀면 설치가 됩니다.

서버에 업로드 한 후 설치할 엔진 디렉토리 위치에 압축을 해제합니다.

|  |
| --- |
| $ unzip -zxvf jboss-ews-httpd-3.0.3-RHEL7-x86\_64.zip |

JBoss EWS는 컴파일되는 버전으로 설치될 경우 공식적으로 기술지원이 되지 않습니다.

#### HTTPD\_HOME 설정

|  |
| --- |
| $ unzip jboss-ews-httpd-3.0.3-RHEL7-86\_64.zip  $ cd jboss-ews-3.0/httpd  **$ sh .postinstall** |

### mod\_jk 다운로드

https://access.redhat.com/jbossnetwork/restricted/listSoftware.html?downloadType=distributions&product=appplatform&version=6.4 에서 “Application Platform 6.x Webserver Connector Natives for RHEL 7 x86\_64”를 다운로드 합니다.

다운로드된 “jboss-eap-native-webserver-connectors-6.x-RHEL7-x86\_64.zip” 파일의

“/module/native/lib64/httpd/modules/mod\_jk.so” 파일을 “/web/jboss/jboss-ews-3.0/httpd/modules” 디렉토리에 복사합니다.

|  |
| --- |
| $ cp /module/native/lib64/httpd/modules/mod\_jk.so . |

## JBoss EWS + EAP 연동 설정

JBoss 의 각 서비스별 인스턴스와 JBoss EWS (웹서버) 와의 연동 설정을 위하여 다음의 설정 파일들을 생성합니다.

### mod\_jk.conf 파일 작성

“/web/jws3/domains/{도메인명}/{인스턴스명}/conf” 디렉토리에서 “mod\_jk.conf” 파일을 다음과 같은 내용으로 작성합니다.

|  |
| --- |
| **mod\_jk.conf** |
| # Load mod\_jk module  # Update this path to match your modules location  LoadModule jk\_module /web/jws3/httpd/modules/mod\_jk.so  # Where to find workers.properties  # Update this path to match your conf directory location (put workers.properties next to httpd.conf)  JkWorkersFile /web/jws3/domains/**{도메인명}/{인스턴스명}**/conf/workers.properties  # Where to put jk logs  # Update this path to match your logs directory location (put mod\_jk.log next to access\_log)  JkLogFile "|/web/jws3/httpd/sbin/rotatelogs /log/jws3/**{도메인명}/{인스턴스명}**/mod\_jk.log.%Y%m%d 86400"  JkShmFile /log/jws3/**{도메인명}/{인스턴스명}**/mod\_jk.shm  # Set the jk log level [debug/error/info]  JkLogLevel error  # Select the log format  JkLogStampFormat "[%a %b %d %H:%M:%S %Y] "  # JkOptions indicate to send SSL KEY SIZE,  JkOptions +ForwardKeySize +ForwardURICompatUnparsed -ForwardDirectories +ForwardPhysicalAddress +ForwardURIProxy  # JkRequestLogFormat set the request format  JKRequestLogFormat "%w %R %V %T %U %s"  # You can use external file for mount points  # It will be checked for updates each 60 secondes  # The format of the file is : /url=worker  # /examples/\*=loadbalancer  JkMountFile /web/jws3/domains/**{도메인명}/{인스턴스명}**/conf/uriworkermap.properties |

### workers.properties 파일 작성

“/web/jws3/domains/{도메인명}/{인스턴스명}/conf” 디렉토리에서 “workers.properties” 파일을 다음과 같은 내용으로 작성합니다.

|  |
| --- |
| workers.properties |
| # Defind list of workers that will be used  # for mapping requests  # The configuration directives are valid  # for the mod\_jk version 1.2.37 and later  worker.list=wlb\_F11,jkstatus  worker.jkstatus.type=status  # default configuration template #  worker.template.type=ajp13  worker.template.lbfactor=1  worker.template.maintain=30  worker.template.socket\_keepalive=True  worker.template.socket\_connect\_timeout=3000  #worker.template.socket\_timeout=60  worker.template.ping\_mode=A  worker.template.ping\_timeout=5000  #worker.template.reply\_timeout=45000  worker.template.connection\_pool\_timeout=30  worker.template.connection\_pool\_size=15  #worker.template.retries=2  worker.template.recovery\_options=7  # worker F11  # Define standalone worker  worker.worker명.reference=worker.template  worker.worker명.host=40.\*\*.\*\*.\*\*\*  worker.worker명.port=**AJP포트**  # Load-balancing behaviour  worker.wlb\_F11.type=lb  worker.wlb\_F11.sticky\_session=1  worker.wlb\_F11.session\_cookie=design  worker.wlb\_F11.balance\_workers= **worker명** |

### uriworkermap.properties 파일 작성

“/web/jws3/domains/{도메인명}/{인스턴스명}/conf” 디렉토리에서 “uriworkermap\_sv11.properties”, “uriworkermap\_sv12.properties” , “uriworkermap\_sv13.properties” , “uriworkermap\_sv14.properties” 파일을 다음과 같은 내용으로 작성합니다.

|  |
| --- |
| uriworkermap.properties |
| # uri base로 특정 url 패턴이 오게 되면 wlb worker를 통해 뒷단의 JBoss EAP 와 연동이 되게 한다.  /test/\*=wlb\_F11  /version/\*=wlb\_F11  /servlet/\*=wlb\_F11  /\*\*/servlet/\*=wlb\_F11  **/ezwel-image/\*=wlb\_SV11**  /fileload/\*=wlb\_F11  /\*\*/fileload/\*=wlb\_F11  /weblogic/\*=wlb\_F11  /\*\*/weblogic/\*=wlb\_F11  /testload/\*=wlb\_F11  /\*\*/testload/\*=wlb\_F11  /\*.jsp=wlb\_F11  /\*.do=wlb\_F11  /jkstatus\*=jkstatus |

## Webapp session replication 을 위한 web.xml 수정

Web Application 의 HTTP 세션 복제를 설정하려면 web.xml 파일에 <distributable/>를 추가한다.

|  |
| --- |
| Session Clustering 위하여 web.xml 수정 |
| <?xml version="1.0" encoding="UTF-8"?>  <web-app id="WebApp\_9" version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app\_2\_4.xsd">  <display-name>Ezwel-user-front</display-name>  <!-- session clustering configuration -->  <distributable />  <context-param>  <param-name>contextConfigLocation</param-name>  <param-value>classpath:config/application-context.xml</param-value>  </context-param>  <filter>  <filter-name>struts-prepare</filter-name>  <filter-class>org.apache.struts2.dispatcher.ng.filter.StrutsPrepareFilter</filter-class>  </filter>  <filter-mapping>  <filter-name>struts-prepare</filter-name>  <url-pattern>\*.ez</url-pattern>  </filter-mapping>  <filter>  <filter-name>sitemesh</filter-name>  <filter-class>com.opensymphony.sitemesh.webapp.SiteMeshFilter</filter-class>  </filter>  <filter-mapping>  <filter-name>sitemesh</filter-name>  <url-pattern>\*.ez</url-pattern>  <dispatcher>REQUEST</dispatcher>  <dispatcher>FORWARD</dispatcher>  <dispatcher>INCLUDE</dispatcher>  </filter-mapping>  <filter>  <filter-name>struts-execute</filter-name>  <filter-class>org.apache.struts2.dispatcher.ng.filter.StrutsExecuteFilter</filter-class>  </filter>  <filter-mapping>  <filter-name>struts-execute</filter-name>  <url-pattern>\*.ez</url-pattern>  </filter-mapping>  <listener>  <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>  </listener>  <session-config>  <session-timeout>30</session-timeout>  </session-config>  <welcome-file-list>  <welcome-file>index.jsp</welcome-file>  </welcome-file-list>  <error-page>  <error-code>503</error-code>  <location>/view/commons/errors/error\_503.jsp</location>  </error-page>  <error-page>  <error-code>500</error-code>  <location>/view/commons/errors/error\_500.jsp</location>  </error-page>  <error-page>  <error-code>404</error-code>  <location>/view/commons/errors/error\_404.jsp</location>  </error-page>  <error-page>  <error-code>403</error-code>  <location>/view/commons/errors/error\_404.jsp</location>  </error-page>  <error-page>  <error-code>401</error-code>  <location>/view/commons/errors/error\_404.jsp</location>  </error-page>  </web-app> |

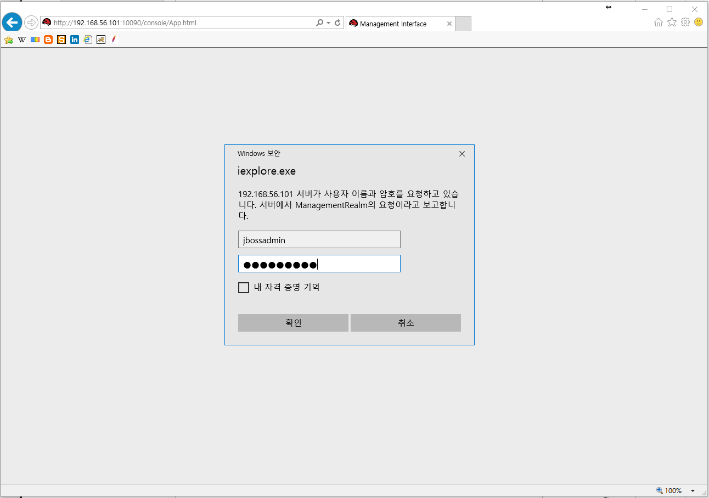
# Jboss DataSource 설정

상세한 내용은 “JBoss EAP 7.0 Development Guide”를 참조한다.

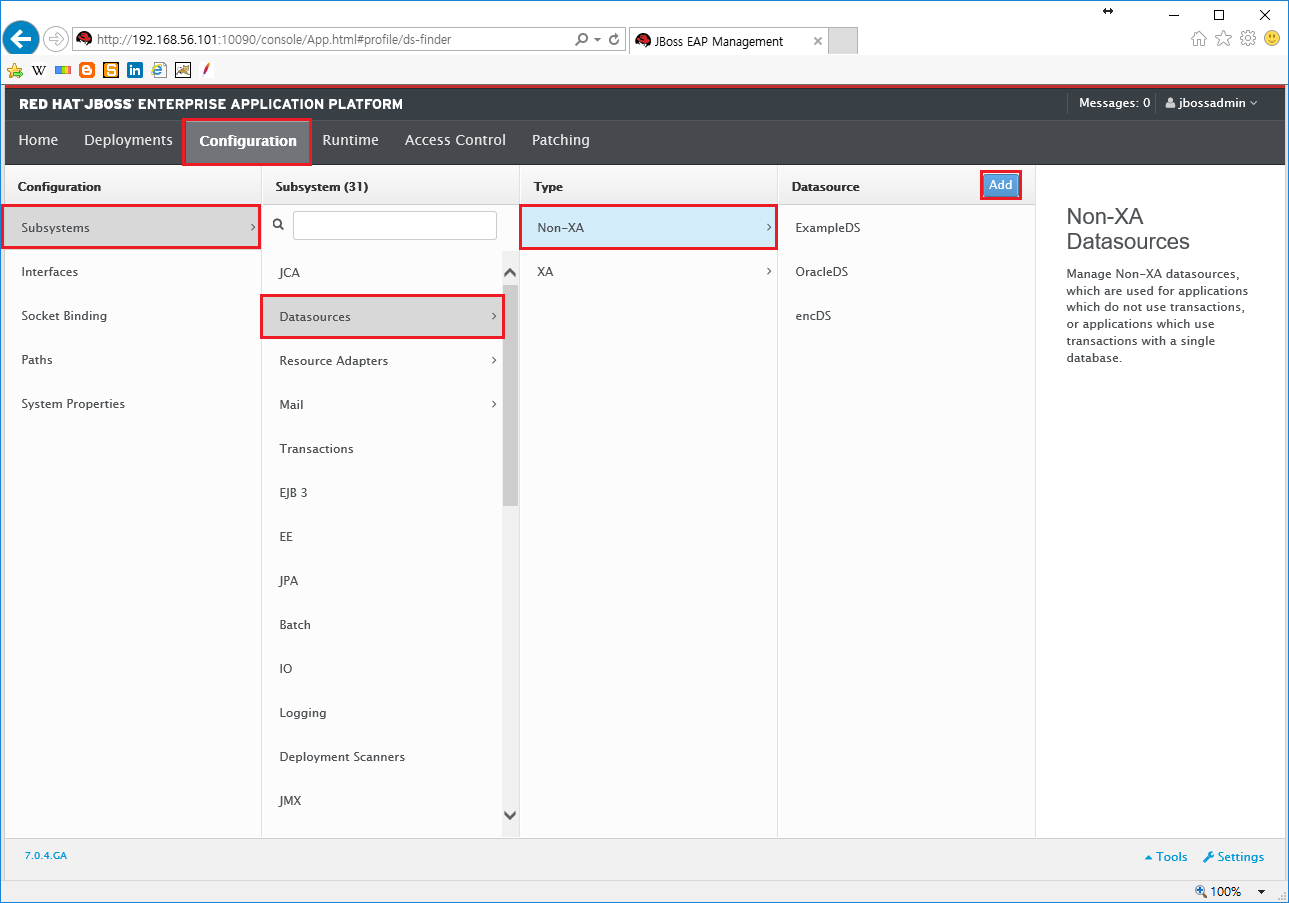
<https://access.redhat.com/knowledge/docs/en-US/JBoss_Enterprise_Application_Platform/6/html-single/Development_Guide/index.html#sect-Transactional_Datasource_Configuration>

## JDBC 드라이버 추가 및 등록

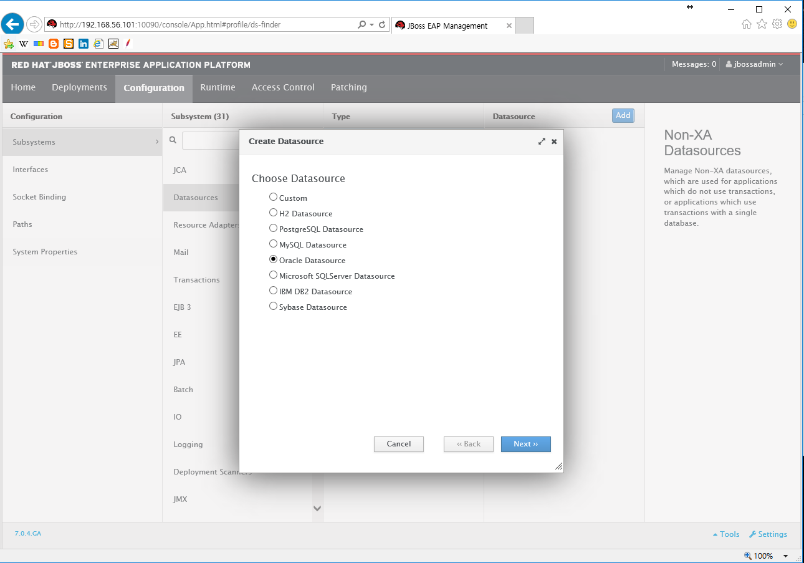
1. <http://40.xxx.xxx.xxx:xxx> admin console 접속
2. Add-user.sh 로 만든 관리자 계정으로 로그인



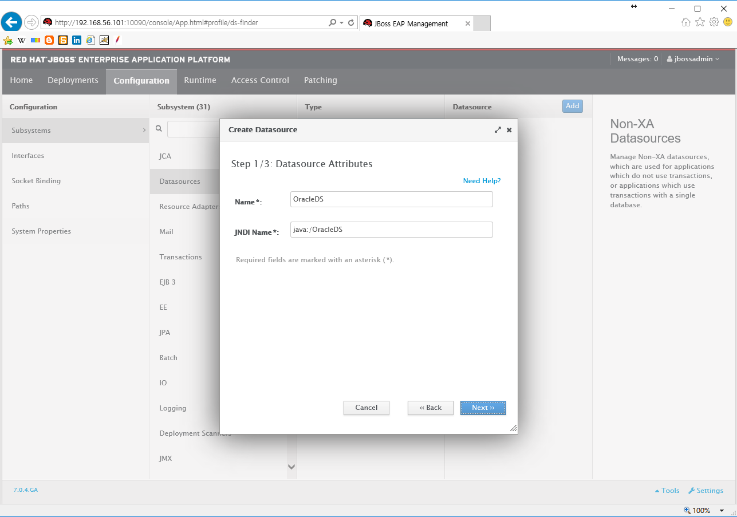
1. Configuration 🡺 Subsystems 🡺 Datasource 🡺 Non-XA 🡺 add 클릭



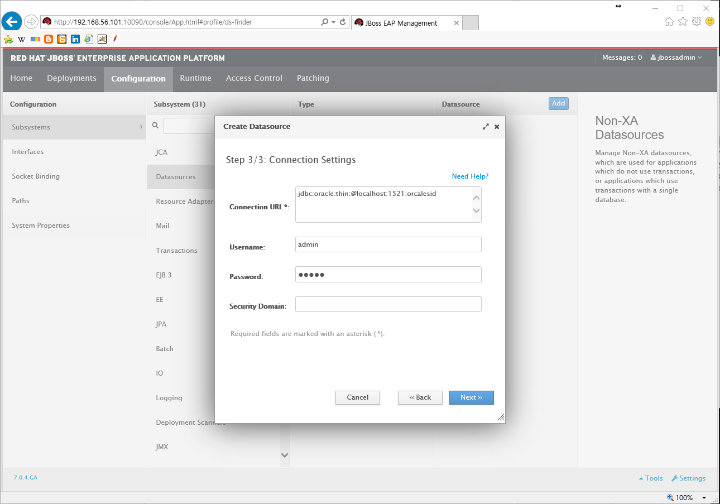
1. JDBC 드라이버 선택
   1. Choose Datasource



* 1. Datasource name, JNDI name 등록



* 1. Detected Driver 선택 후 module.ext 등록된 driver 선택 후 connection 정보 설정



* 1. 설정 정보 Summary 확인 후 Finish 버튼 클릭

## JDBC DataSource 설정시 JNDI 이름 지정

JBoss 에서 JDBC DataSource 설정시에 <jndi-name>DataSourceName</jndi-name>에 이름을 지정하고 이를 application 에서 JNDI 를 통해서 참조하게 되는데, WebLogic 의 경우에는 작성된 DataSourceName 으로 바로 참조할 수 있지만, JBoss 의 경우엔 java:DataSourceName 와 같이 앞에 java: 를 붙여야 참조할 수 있습니다.

WebLogic 과 같이 java: 없이 참조하려면 CLI 를 이용하여 Datasource 의 use-java-context 를 false 로 설정하면 됩니다.

|  |
| --- |
| **CLI 를 이용한 use-java-context 설정** |
| $ ./jboss-cli.sh  ## use-java-context 설정  [standalone@\*\*\*.\*\*\*.\*\*\*.\*\*\*:\*\*\*\* /] /subsystem=datasources/data-source=OracleDS:write-attribute(name=use-javacontext, value=false)  {  "outcome" => "success",  "response-headers" => {  "operation-requires-reload" => true,  "process-state" => "reload-required"  }  }  ## 확인  [standalone@\*\*\*.\*\*\*.\*\*\*.\*\*\*:\*\*\*\*/] /subsystem=datasources/data-source=OracleDS:read-resource(recursive=true)  {  "outcome" => "success",  "result" => {  "allocation-retry" => undefined,  "allocation-retry-wait-millis" => undefined,  "allow-multiple-users" => undefined,  "background-validation" => false,  "background-validation-millis" => undefined,  "blocking-timeout-wait-millis" => undefined,  "check-valid-connection-sql" => undefined,  "connection-properties" => undefined,  "connection-url" => "jdbc:oracle:thin:@localhost:1521:XE",  "datasource-class" => undefined,  "driver-class" => "oracle.jdbc.OracleDriver",  "driver-name" => "ojdbc6.jar",  "enabled" => true,  "exception-sorter-class-name" => undefined,  "exception-sorter-properties" => undefined,  "flush-strategy" => undefined,  "idle-timeout-minutes" => undefined,  "jndi-name" => "java:jboss/datasources/OracleDS",  "jta" => false,  "max-pool-size" => undefined,  "min-pool-size" => undefined,  "new-connection-sql" => undefined,  "password" => "admin",  "pool-prefill" => undefined,  "pool-use-strict-min" => undefined,  "prepared-statements-cache-size" => undefined,  "query-timeout" => undefined,  "reauth-plugin-class-name" => undefined,  "reauth-plugin-properties" => undefined,  "security-domain" => undefined,  "set-tx-query-timeout" => false,  "share-prepared-statements" => false,  "spy" => false,  "stale-connection-checker-class-name" => undefined,  "stale-connection-checker-properties" => undefined,  "track-statements" => "NOWARN",  "transaction-isolation" => undefined,  "url-delimiter" => undefined,  "url-selector-strategy-class-name" => undefined,  "use-ccm" => false,  "use-fast-fail" => false,  "use-java-context" => false,  "use-try-lock" => undefined,  "user-name" => "admin",  "valid-connection-checker-class-name" => undefined,  "valid-connection-checker-properties" => undefined,  "validate-on-match" => false,  "statistics" => {  "jdbc" => undefined,  "pool" => undefined  }  },  "response-headers" => {"process-state" => "reload-required"}  } |

## ****Datasource 기본 설정 항목****의 설명

1. **<driver-class>**

java.sql.Driver 인터페이스를 구현한 드라이버 클래스를 <driver-class>에 설정합니다.

|  |  |
| --- | --- |
| Oracle 12c | oracle.jdbc.driver.OracleDriver |
| PPAS | com.edb.Driver |

1. **<connection-url>**

DB접속 URL을 명시한다.

|  |  |
| --- | --- |
| Oracle thin | jdbc:oracle:thin:@host:port:sid |
| Oracle thin (description) | jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=host)(PORT=port))(CONNECT\_DATA=(SERVICE\_NAME=sid))) |
| Oracle oci | jdbc:oracle:oci:@sid |
| Oracle thin | jdbc:oracle:thin:@host:port:sid |
| PAS | jdbc:edb:@host:port/titn |

1. **<jndi-name>과 <use-java-context>**

JNDI에서 데이터소스를 찾을 때 사용할 이름으로 jdbc/xxx 형식을 사용하는 것을 권장합니다.

주의할 점은 데이터소스는 JNDI에 바인딩될 때 java: 네임스페이스로 바인딩된다는 점이다.

이때문에 Context.lookup()시에 java:/jdbc/xxx와 같이 명시해야 데이터소스를 찾을 수 있다.

만약 WebLogic에서 처럼 global하게 바인딩하려면 <use-java-context>false</use-java-context>로 설정하면 된다.

이 때는 Context.lookup()시에 jdbc/xxx로 명시한다.

1. **<user-name>과 <password>**

DB 접속 계정과 패스워드를 명시한다.

1. **ConnectionPool 설정**
   1. **<min-pool-size>**

Pool에 유지할 최소 connection 개수로 기본값은 0 이다.

기본적으로 최초 getConnection() 호출전까지 물리적인 db connection은 만들어지지 않는다**.**

* 1. **<max-pool-size>**

최대 connection 개수를 설정한다. 기본값은 20 이다.

* 1. **<prefill>true</prefill>**

**서버 기동시에 <min-pool-size>만큼 connection을 만들고자 할 때 true로 설정 한다. Default는 false 입니다.**

1. **Connection property**

**JDBC Driver 자체에서 제공되는 attribute properties설정을 추가해주는 것으로 아래는 MySQL JDBC 에서 권고하는 설정을 Jboss 내의 설정에서 추가해주는 설정입니다.**

* 1. **<connection-property name="characterEncoding">UTF-8</connection-property>**

**DB 연결후 character set을 지정한다. Default는 UTF-8 이다.**

* 1. **<connection-property name="autoReconnect">true</connection-property>**

**DB 연결 실패시 재 시도후 자동 연결을 하기 위한 설정이다.**

* 1. **<connection-property name="failOverReadOnly">true</connection-property>**

**DB 연결 에러 발생시 Fail-Over 된 시점에서 DB의 Write, Read 관련 설정 default는 false이며, false시에는 DB Write 도 가능하다.**

* 1. **<connection-property name="roundRobinLoadBalance">false</connection-property>**

**Multi Connection Pool 연결 시 loadBalancing 설정을 roundRobin 으로 할지 아니면, 한 ConnectionPool만 바라 볼지를 결정 default는 true**

* 1. **<connection-property name="connectTimeout">2000</connection-property>**

**DB의 접근 connection Time out 설정 Connection Pool 에서 connection을 얻기 위한 대기시간.DB 절체를 위해서 필요한 설정이며, 단위는 ms 입니다.**

**위의 설정이면 Connection을 얻기 위해 2초간 대기한 후 실패시 다른 DB를 바라보게 된다.**

* 1. **<connection-property name="defaultFetchSize">1000</connection-property>**

**DB의 연결 fetch 할당을 default 로 얼마나 잡을지를 결정한다.**

* 1. **<connection-property name="tcpSndBuf">65534</connection-property>**

**tcp send buffer 를 설정하여, data 이동 delay 가 발생하는지 check 합니다.**

* 1. **<connection-property name="tcpRcvBuf">65534</connection-property>**

**tcp receive buffer 를 설정하여, data 이동 delay 가 발생하는지 check 합니다.**

* 1. **<connection-property name="tcpNoDelay">true</connection-property>**

**TCP 연결설정에 delay time 을 없애기 위한 설정이다.**

1. **각종 Timeout 설정**
2. **<blocking-timeout-millis>**

**모든 connection이 사용중일 때 getConnection()에서 connection이 반환될 때까지 기다릴 시간으로 ms 단위이다.**

1. **<idle-timeout-minute>**

**pool에서 일정 시간동안 사용되지 않고 있는 connection을 닫도록 설정할 수 있는데 그 시간을 분 단위로 설정한다.**

1. **<query-timeout>**

**query에 대한 응답을 기다리며 대기할 수 있는 시간을 초 단위로 설정한다.**

1. **<set-tx-query-timeout>true</set-tx-query-timeout>**

**Transaction 이 남이 있을 때까지 qury timeout 을 유예하고 싶을 경우 적용하며, 트랜잭션 timeout까지 남아 있는 시간을 query timeout으로 자동으로 설정한다.**

1. **Connection 에러 처리 설정**
2. **<check-valid-connection-sql>**

**getConnection()시 pool에서 어플리케이션에 connection을 반환하기 전에 connection 상태를 체크하는데 사용할 SQL을 명시한다.**

**ex) SELECT 1 FROM DUAL**

1. **<valid-connection-checker-class-name>**

**getConnection()시 pool에서 어플리케이션에 connection을 반환하기 전에 connection 상태를 체크하는 DB 벤더별 클래스를 명시한다. MySQL에 대한 구현체가 제공된다**

|  |  |
| --- | --- |
| Oracle | org.jboss.resource.adapter.jdbc.vendor.OracleValidConnectionChecker |
| PAS | org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLValidConnectionChecker |

1. **<exception-sorter-class-name>**

**SQLException 발생시 connection error를 검출하는데 사용할 클래스를 명시한다. 예외가 Connection 문제로 발생한 경우 해당 connection을 close하게 된다.**

|  |  |
| --- | --- |
| Oracle | org.jboss.resource.adapter.jdbc.vendor.OracleExceptionSorter |
| PAS | org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLExceptionSorter |

1. **<background-validation>**

**connection 상태를 매번 getConnection()시마다 체크하지 않고 일정주기로 백그라운드에서 체크할지 여부를 설정한다. 기본값은 false이다.**

1. **<background-validation-minutes>**

**connection 상태를 백그라운드로 체크하도록 설정한 경우 그 주기를 분 단위로 설정한다. 기본값은 10분 이다.**

# JBoss EWS 모니터링 가이드

## 모니터링 항목

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **대항목** | **중항목** | **세부항목** | **설명** | **모니터링 결과** |
| **OS** | System Resource Usage | CPU 사용량 | CPU 전체 사용량 | 양호 |
| Memory 사용량 | Memory 전체 사용량 | 양호 |
| **운영환경** | Process Resource Usage | Process CPU 과부하 사용 | 프로세스 CPU 사용률 | 양호 |
| Disk Usage | 설치 디스크의 사용량 | 양호 |
| **서버 상태** | Server Status | Server Status | Server Life Cycle | 양호 |
| access log size 확인 | 파일 사이즈 2G 이내 | 양호 |
| **실행 상태** | Execute Status | Name Virtual Host 연결 상태 | 접속 테스트 | 정상 |
| SSL Virtual Host 연결 상태 | 접속 테스트 | 정상 |
| WAS 연결 정상 여부 | 접속테스트 | 정상 |
| **로그 분석** | ErrorLog | Critical Message | error 로그 확인 | 양호 |
| Error Message | error 로그 확인 | 양호 |

## 모니터링 가이드

### System Resource 체크

#### cpu 사용량

사용서비스 되어지는 CPU의 사용량 체크를 위하여 다음의 명령어를 통해 모니터링을 진행합니다.

vmstat 1 100

항목 설명

us : user cpu usage

sy : system cpu usage

id : cpu idle usage

정상 여부 확인 : id 60% 이상일 경우 양호, id 30% 미만인 경우 위험

|  |
| --- |
| **$ vmstat 1 100**  procs -----------memory---------- ---swap-- -----io---- --system-- **-----cpu-----**  r b swpd free buff cache si so bi bo in cs us sy id wa st  0 0 853424 69016 4648 29128 0 0 13 8 8 4 0 0 99 0 0  0 0 853424 69008 4648 29128 0 0 0 0 113 242 0 0 100 0 0  0 0 853424 69008 4648 29128 0 0 0 0 75 211 0 0 100 0 0  0 0 853424 69008 4648 29128 0 0 0 0 106 254 0 0 100 0 0  0 0 853424 69008 4648 29128 0 0 0 0 73 208 0 0 100 0 0  0 0 853424 69008 4648 29128 0 0 0 0 118 263 0 0 100 0 0  0 0 853424 69008 4648 29128 0 0 0 0 78 203 0 0 100 0 0  0 0 853424 69008 4648 29128 0 0 0 0 95 244 0 0 100 0 0  0 0 853424 69008 4648 29128 0 0 0 0 89 219 0 0 100 0 0  0 0 853424 69008 4648 29128 0 0 0 0 85 237 0 0 100 0 0  0 0 853424 69008 4648 29128 0 0 0 0 93 221 0 0 100 0 0  0 0 853424 69008 4648 29128 0 0 0 0 109 244 1 0 100 0 0 |

#### Memory 사용량

사용서비스 되어지는 Memory 사용량 체크를 위하여 다음의 명령어를 통해 모니터링을 진행합니다.

vmstat 1 100

항목 설명

total : 전체 메모리 용량

used : 메모리 사용량

free : 여유 메모리 공간

shared : 공유메모리 사용량

buffers : 버퍼 메모리 공간

cached : 캐쉬메모리 사용량

정상 여부 확인 : free 최소 여유공간이 전체 메모리 대비 최소 10% 이상일 경우

|  |
| --- |
| **$ free -m**  total used free shared buffers cached  Mem: 1877 1810 66 0 0 30  -/+ buffers/cache: 1779 98  Swap: 4031 831 3200 |

### Process Resource 체크

#### 프로세스 과부하 cpu 여부 확인

서비스 되어지는 process의 CPU사용량이 과부하 인지 여부를 체크합니다.

top

항목 설명

top 명령을 통해서 해당 프로세스의 현재 CPU 정보를 확인합니다.

정상 여부 확인 : 프로세스당 CPU 사용이 많은지 여부를 확인합니다.

|  |
| --- |
| **$ top**  top - 03:14:41 up 14 days, 12:59, 4 users, load average: 0.19, 0.13, 0.09  Tasks: 336 total, 1 running, 335 sleeping, 0 stopped, 0 zombie  Cpu(s): 3.7%us, 3.5%sy, 0.0%ni, 92.1%id, 0.7%wa, 0.0%hi, 0.0%si, 0.0%st  Mem: 16271536k total, 15962368k used, 309168k free, 697304k buffers  Swap: 16777208k total, 529588k used, 16247620k free, 6697584k cached  PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND  28336 jboss 20 0 4003m 798m 15m S 1.0 5.0 20:21.52 /opt/java/jdk1.6.0\_45/bin/java -D[Standalone] -XX:+UseCompr  1732 jboss 20 0 6051m 115m 11m S 0.0 0.7 1:29.62 java -server -cp lib/\* org.ngrinder.NGrinderAgentStarter --  30949 jboss 20 0 188m 9632 6520 S 0.0 0.1 0:00.00 /app/jboss/jboss-ews-2.0.1/httpd/sbin/httpd.worker -f /opt/we  31319 jboss 20 0 915m 7096 1216 S 0.0 0.0 0:00.00 /app/jboss/jboss-ews-2.0.1/httpd/sbin/httpd.worker -f /opt/we |

#### Disk Usage

서비스 되어지는 Disk 사용량을 파악하여 여유 공간을 확인합니다.

df -h

정상 여부 확인 : 서비스 사용 영역 및 root 파일 시스템의 사용율이 80% 인 경우 위험으로 간주합니다.

|  |
| --- |
| **$ df -h**  Filesystem Size Used Avail Use% Mounted on  /dev/mapper/vg\_rhevmtest-lv\_root  26G 23G 1.9G 93% /  tmpfs 939M 72K 939M 1% /dev/shm  /dev/vda1 485M 38M 422M 9% /boot |

### Server Status

#### Service Process 정상 여부 확인

서비스 되어지는 JBoss EWS 의 현재 서버 상태에 대해서 모니터링을 합니다.

ps –ef | grep ${processName}

ex) ps –ef | grep httpd

항목 설명

ps 명령을 통해서 해당 프로세스의 현재 동작 여부를 확인합니다.

정상 여부 확인 : 프로세스 목록에 해당 내용이 있어야만 합니다.

|  |
| --- |
| **$ ps -ef | grep httpd**  root 3935 1 0 11:35 ? 00:00:00 /web/jws3/httpd/sbin/httpd -f /web/jws3/domains/T-MS71-O/T-MS71-O-F11/conf/httpd.conf -k start  nobody 3938 3935 0 11:35 ? 00:00:00 /web/jws3/httpd/sbin/httpd -f /web/jws3/domains/T-MS71-O/T-MS71-O-F11/conf/httpd.conf -k start  nobody 3939 3935 0 11:35 ? 00:00:00 /web/jws3/httpd/sbin/httpd -f /web/jws3/domains/T-MS71-O/T-MS71-O-F11/conf/httpd.conf -k start |

#### Port Listen 정상 여부 확인

netstat –an | grep ${servicePort}

ex) netstat –an | grep 8000

항목 설명

netstat 명령을 통해서 해당 프로세스의 현재 Network 동작 여부를 확인합니다.

정상 여부 확인 : Port 목록에 해당 서비스에서 사용되는 포트가 LISTEN 항목으로 되어 있어야 합니다.

|  |
| --- |
| **$ netstat -an | grep 8000**  tcp 0 0 :::8000 :::\* LISTEN |

#### access 로그 파일 사이즈 체크

서비스 되어지는 해당 process 의 access 로그 및 기타 error 로그등의 사용량을 체크합니다.

du –csh \*log

정상 여부 확인 : 서비스 사용 로그 파일이 최대 2G 이내여야 합니다.

|  |
| --- |
| **$ du -csh \***  4.0K access\_log  8.0K error\_log  0 mod\_jk.log  16K mod\_jk.shm.4930  0 mod\_jk.shm.4930.lock  28K total |

### Execute Status – 실행 상태

#### Name Virtual Host 연결 상태

NameVirtualHost 설정을 통해서 정상적인 서비스가 이루어지는지 모니터링을 합니다.

curl http

ex) curl http://127.0.0.1:8251/index.html

항목 설명

curl 명령을 통해서 http 서비스 요청에 대한 처리 응답이 정상적인지 확인합니다.

정상 여부 확인 : 페이지 요청에 따른 응답 페이지 확인이 가능하면 정상입니다.

|  |
| --- |
| **$ ping ticoap01**  PING DollyWas1 (192.168.0.221) 56(84) bytes of data.  64 bytes from DollyWas1 (192.168.0.221): icmp\_seq=1 ttl=64 time=0.010 ms  64 bytes from DollyWas1 (192.168.0.221): icmp\_seq=2 ttl=64 time=0.015 ms  $ curl http://127.0.0.1:8251/index.html  <meta http-equiv="Refresh" content="0; URL=/index.jsp"> |

#### WAS 정상 동작 여부

netstat –an | grep ${servicePort}

ex) netstat –an | grep 8000

항목 설명

netstat 명령을 통해서 해당 프로세스 와 WAS 단의 AJP 서비스 포트가 ESTABLISHED 상태 혹은 TIME\_WAIT 상태등이 정상인지 확인한다.

정상 여부 확인 : Port 목록에 해당 서비스에서 사용되는 포트가 다수의 ESTABLISHED 항목으로 되어 있어야 합니다.

|  |
| --- |
| **$ netstat -an | grep 8109**  tcp 0 0 192.168.0.221:8109 0.0.0.0:\* LISTEN  tcp 0 0 192.168.0.221:16153 192.168.0.221:8109 ESTABLISHED  tcp 0 0 192.168.0.221:16657 192.168.0.221:8109 ESTABLISHED  tcp 0 0 192.168.0.221:16659 192.168.0.221:8109 TIME\_WAIT  tcp 0 0 192.168.0.221:16663 192.168.0.221:8109 CLOSE\_WAIT |

### 로그 모니터링

#### Critical Message

서비스 Error 로그 중 서비스에 문제가 될 소지가 있는 Message 여부를 모니터링 확인합니다.

egrep “critical|error” error.log

항목 설명

grep 명령을 통해서 해당 로그 파일 내에 비정상적인 로그 부분을 확인합니다.

정상 여부 확인 : 특이한 에러 로그 등이 없다면 정상입니다.

|  |
| --- |
| $ egrep "critical|error" error\_stand\_log.140401  [Tue Apr 01 14:26:02 2014] [error] [client 192.168.0.172] client denied by server configuration: session.jsp  [Tue Apr 01 14:26:02 2014] [error] [client 192.168.0.172] client denied by server configuration: session.jsp  [Tue Apr 01 14:26:02 2014] [error] [client 192.168.0.172] client denied by server configuration: session.jsp |

# JBoss EAP 모니터링 가이드

## 모니터링 항목

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **대항목** | **중항목** | **세부항목** | **설명** | **모니터링 결과** |
| **OS** | System Resource Usage | CPU 사용량 | CPU 전체 사용량 | 양호 |
| Memory 사용량 | Memory 전체 사용량 | 양호 |
| **운영환경 서버 상태** | Process Resource Usage | Process CPU 과부하 사용 | 프로세스 CPU 사용률 | 양호 |
| Disk Usage | 설치 디스크의 사용량 | 양호 |
| Server Status | Server Status | Server Life Cycle | 양호 |
| **실행 상태** | Execute Status | Thread status | 실행 쓰레드 및  idle thread status | 양호 |
| Garbage Collection | Full GC Duration | Full GC 처리 상태 | 양호 |
| Heap Memory | Heap Memory Init | 힙메모리 초기값 | 양호 |
| Heap Memory Max | 힙메모리 최대값 설정 | 양호 |
| Perm Area Max Size | MaxPermSize 의 설정 | 양호 |
| **JDBC 커넥션풀** | Configuration | JDBC url | jdbc connection meta data URL | 양호 |
| JDBC driver name | jdbc driver class name | 양호 |
| Monitorming | Connections | 사용중인 커넥션 개수 | 양호 |
| **로그 분석** | ServerLog | Critical Message | 부팅 Critical 에러 | 양호 |
| Too many open files | 파일 오픈 개수 초과 | 양호 |
| OutOfMemoryError | 힙메모리 부족 | 양호 |

## 모니터링 가이드

### System Resource 체크

#### cpu 사용량

사용서비스 되어지는 CPU의 사용량 체크를 위하여 다음의 명령어를 통해 모니터링을 진행합니다.

vmstat 1 100

항목 설명

us : user cpu usage

sy : system cpu usage

id : cpu idle usage

정상 여부 확인 : id 60% 이상일 경우 양호, id 30% 미만인 경우 위험

|  |
| --- |
| **$ vmstat 1 100**  procs -----------memory---------- ---swap-- -----io---- --system-- **-----cpu-----**  r b swpd free buff cache si so bi bo in cs us sy id wa st  0 0 853424 69016 4648 29128 0 0 13 8 8 4 0 0 99 0 0  0 0 853424 69008 4648 29128 0 0 0 0 113 242 0 0 100 0 0  0 0 853424 69008 4648 29128 0 0 0 0 75 211 0 0 100 0 0  0 0 853424 69008 4648 29128 0 0 0 0 106 254 0 0 100 0 0  0 0 853424 69008 4648 29128 0 0 0 0 73 208 0 0 100 0 0  0 0 853424 69008 4648 29128 0 0 0 0 118 263 0 0 100 0 0  0 0 853424 69008 4648 29128 0 0 0 0 78 203 0 0 100 0 0  0 0 853424 69008 4648 29128 0 0 0 0 95 244 0 0 100 0 0  0 0 853424 69008 4648 29128 0 0 0 0 89 219 0 0 100 0 0  0 0 853424 69008 4648 29128 0 0 0 0 85 237 0 0 100 0 0  0 0 853424 69008 4648 29128 0 0 0 0 93 221 0 0 100 0 0  0 0 853424 69008 4648 29128 0 0 0 0 109 244 1 0 100 0 0 |

#### Memory 사용량

사용서비스 되어지는 Memory 사용량 체크를 위하여 다음의 명령어를 통해 모니터링을 진행합니다.

vmstat 1 100

항목 설명

total : 전체 메모리 용량

used : 메모리 사용량

free : 여유 메모리 공간

shared : 공유메모리 사용량

buffers : 버퍼 메모리 공간

cached : 캐쉬메모리 사용량

정상 여부 확인 : free 최소 여유공간이 전체 메모리 대비 최소 10% 이상일 경우

|  |
| --- |
| **$ free -m**  total used free shared buffers cached  Mem: 1877 1810 66 0 0 30  -/+ buffers/cache: 1779 98  Swap: 4031 831 3200 |

### Process Resource 체크

#### 프로세스 과부하 cpu 여부 확인

서비스 되어지는 process의 CPU사용량이 과부하 인지 여부를 체크합니다.

top

항목 설명

top 명령을 통해서 해당 프로세스의 현재 CPU 정보를 확인합니다.

정상 여부 확인 : 프로세스당 CPU 사용이 많은지 여부를 확인합니다.

|  |
| --- |
| **$ top**  top - 03:14:41 up 14 days, 12:59, 4 users, load average: 0.19, 0.13, 0.09  Tasks: 336 total, 1 running, 335 sleeping, 0 stopped, 0 zombie  Cpu(s): 3.7%us, 3.5%sy, 0.0%ni, 92.1%id, 0.7%wa, 0.0%hi, 0.0%si, 0.0%st  Mem: 16271536k total, 15962368k used, 309168k free, 697304k buffers  Swap: 16777208k total, 529588k used, 16247620k free, 6697584k cached  PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND  12037 jboss7 20 0 4275552 876160 22404 S 0.3 0.8 0:22.28 /usr/jdk/instances/jdk1.8.0\_121/bin/java -D[Standalone] -server -Xms1024m -Xmx1024m -XX:MetaspaceSize=256m -XX:MaxMetaspaceSize=256m -XX:+UseG1GC -XX:+UnlockDiagnostic+  1732 jboss 20 0 6051m 115m 11m S 0.0 0.7 1:29.62 java -server -cp lib/\* org.ngrinder.NGrinderAgentStarter --  30949 jboss 20 0 188m 9632 6520 S 0.0 0.1 0:00.00 /app/jboss/jboss-ews-2.0.1/httpd/sbin/httpd.worker -f /opt/we  31319 jboss 20 0 915m 7096 1216 S 0.0 0.0 0:00.00 /app/jboss/jboss-ews-2.0.1/httpd/sbin/httpd.worker -f /opt/we |

#### Disk Usage

서비스 되어지는 Disk 사용량을 파악하여 여유 공간을 확인합니다.

df -h

정상 여부 확인 : 서비스 사용 영역 및 root 파일 시스템의 사용율이 80% 인 경우 위험으로 간주합니다.

|  |
| --- |
| **$ df -h**  Filesystem Size Used Avail Use% Mounted on  /dev/mapper/ol-root 50G 9.9G 41G 20% /  devtmpfs 52G 0 52G 0% /dev  tmpfs 52G 172K 52G 1% /dev/shm |

### Server Status

#### Service Process 정상 여부 확인

서비스 되어지는 JBoss EAP 의 현재 서버 상태에 대해서 모니터링을 합니다.

ps –ef | grep ${processName}

ex) ps –ef | grep java | grep T-MS71-O-F11

항목 설명

ps 명령을 통해서 해당 프로세스의 현재 동작 여부를 확인합니다.

정상 여부 확인 : 프로세스 목록에 해당 내용이 있어야만 합니다.

|  |
| --- |
| **$ ps –ef | grep java | grep T-MS71-O-F11**  jboss7 7787 7526 0 00:01:33 /usr/jdk/instances/jdk1.8.0\_121//bin/java -D[Standalone] -server -Xms1024m -Xmx1024m -XX:MetaspaceSize=256m -XX:MaxMetaspaceSize=256m -XX:+UseG1GC -XX:+UnlockDiagnosticVMOptions -XX:+G1SummarizeConcMark -XX:InitiatingHeapOccupancyPercent=35 -XX:ConcGCThreads=2 -XX:+UseLargePagesInMetaspace -verbose:gc -Xloggc:/log/jboss7/T-MS71-O/T-MS71-O-F11/gclog/gc\_20170530172716.log -Xloggc:/log/jboss7/T-MS71-O/T-MS71-O-F11/gclog/gc.log -XX:+PrintGCDetails -XX:+PrintGCTimeStamps -XX:+PrintAdaptiveSizePolicy -XX:+ExplicitGCInvokesConcurrent -XX:+DisableExplicitGC -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/log/jboss7/T-MS71-O/T-MS71-O-F11/heaplog/T-MS71-O-F11.hprof.20170530172716 -Djava.net.preferIPv4Stack=true -Dorg.jboss.resolver.warning=true -Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.awt.headless=true -Djboss.modules.system.pkgs=org.jboss.byteman,scouter,org.jboss.logmanager -Djboss.server.base.dir=/was/jboss7/domains/T-MS71-O/T-MS71-O-F11 -Djboss.server.log.dir=/log/jboss7/T-MS71-O/T-MS71-O-F11/log -Djboss.socket.binding.port-offset=4431 -Djboss.node.name=T-MS71-O-F11 -Djboss.server.base.dir=/was/jboss7/domains/T-MS71-O/T-MS71-O-F11 -Djboss.bind.address=0.0.0.0 -Djboss.bind.address.management=40.10.22.156 -Djboss.socket.binding.port-offset=4431 -Djboss.default.jgroups.stack=udp -Djboss.default.jgroups.stack.tcp.tcpping.initial\_hosts=192.168.0.143[12031],192.168.0.144[12031] -Djboss.default.jgroups.stack.tcp.tcpping.num\_initial\_members=2 -Djgroups.bind\_addr=0.0.0.0 -Djgroups.tcp.bind\_port=12031 -Djboss.default.multicast.address=230.1.44.31 -Djboss.messaging.group.address=231.2.44.31 -Djboss.modcluster.multicast.address=224.3.44.31 -Djboss.http.port=8080 -Djboss.http.final.port=12511 -Djboss.ajp.port=8009 -Djboss.ajp.final.port=12440 -Xbootclasspath/p:/was/jboss7/jboss-eap-7.0/modules/system/layers/base/org/jboss/logmanager/main/jboss-logmanager-2.0.3.Final-redhat-1.jar -Djava.util.logging.manager=org.jboss.logmanager.LogManager -Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=10430 -Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.authenticate=false -Dserver.mode=local -Dorg.jboss.as.logging.per-deployment=false -Dorg.jboss.boot.log.file=/log/jboss7/T-MS71-O/T-MS71-O-F11/log/server.log -Dlogging.configuration=file:/was/jboss7/domains/T-MS71-O/T-MS71-O-F11/configuration/logging.properties -jar /was/jboss7/jboss-eap-7.0/jboss-modules.jar -mp /was/jboss7/jboss-eap-7.0/modules:/was/jboss7/jboss-eap-7.0/modules.ext org.jboss.as.standalone -Djboss.home.dir=/was/jboss7/jboss-eap-7.0 -Djboss.server.base.dir=/was/jboss7/domains/T-MS71-O/T-MS71-O-F11 -DSERVER=T-MS71-O-F11 -P=/was/jboss7/domains/T-MS71-O/T-MS71-O-F11/bin/env.properties -c standalone-ha.xml |

#### Port Listen 정상 여부 확인

netstat –an | grep ${servicePort}

ex) netstat –an | grep 8000

항목 설명

netstat 명령을 통해서 해당 프로세스의 현재 Network 동작 여부를 확인합니다.

정상 여부 확인 : Port 목록에 해당 서비스에서 사용되는 포트가 LISTEN 항목으로 되어 있어야 합니다.

|  |
| --- |
| **$ netstat -anp | grep 12511**  **tcp 0 0 0.0.0.0:12511 0.0.0.0:\* LISTEN 7787/java** |

#### 로그 파일 사이즈 체크

서비스 되어지는 해당 process 의 server 로그 및 기타 gc, nohup로그등의 사용량을 체크합니다.

du –csh \*log

정상 여부 확인 : 서비스 사용 로그 파일이 최대 2G 이내여야 합니다.

|  |
| --- |
| **$ du -csh \***  **1.4M T-MS71-O-F11.out** |

### Execute Status – 실행 상태

#### 서버 실행 및 idle thread 동작 상태 확인

execute thread가 정상적으로 동작하고 있는지 모니터링을 합니다.

JBoss CLI Tool

ex) ./jboss-cli.sh

/subsystem=jca/workmanager=default/long-running-threads=default:read-resource

항목 설명

각 EAP 서버 홈의 실행 폴더에 있는 JBoss Command Line Interface (jboss-cli.sh) 를 통해서 서버의 동작 thread의 설정 및 정보를 확인합니다.

|  |
| --- |
| **$ ./jboss-cli.sh**  **/subsystem=jca/workmanager=default/long-running-threads=default:read-resource**  ================================================  JAVA\_HOME=/usr/jdk/instances/jdk1.8.0\_121  JBOSS\_HOME=/was/jboss7/jboss-eap-7.0  DOMAIN\_BASE=/was/jboss7/domains/T-MS71-O  SERVER\_NAME=T-MS71-O-F11  CONFIG\_FILE=standalone-ha.xml  BIND\_ADDR=0.0.0.0  PORT\_OFFSET=4431  MULTICAST\_ADDR=230.1.44.31  CONTROLLER=40.10.22.156:14421  HTTP\_FINAL\_PORT=12511  AJP\_FINAL\_PORT=12440  {  "outcome" => "success",  "result" => {  "allow-core-timeout" => false,  "core-threads" => 100,  "handoff-executor" => undefined,  "keepalive-time" => {  "time" => 30L,  "unit" => "SECONDS"  },  "max-threads" => 100,  "name" => "default",  "queue-length" => 100,  "thread-factory" => undefined  }  }  **$ ./jboss-cli.sh**  **/subsystem=jca/workmanager=default/short-running-threads=default:read-resource**  {  "outcome" => "success",  "result" => {  "allow-core-timeout" => false,  "core-threads" => 200,  "handoff-executor" => undefined,  "keepalive-time" => {  "time" => 3L,  "unit" => "SECONDS"  },  "max-threads" => 200,  "name" => "default",  "queue-length" => 100,  "thread-factory" => undefined  }  } |

#### Garbage Collection 로그를 통한 서비스 정상 여부

서버 기동시 verbosegc 옵션을 통하여 EAP 서버의 gc 모니터링을 통한 서비스 정상 여부를 확인합니다.

tail

ex) tail -f gc.log.20170523113119

항목 설명

tail 명령을 통해서 현재 verbosegc 로그를 확인하여 지속적인 Full GC 여부를 모니터링 합니다.

정상 여부 확인 : 2~5 지속적인 Full GC 로그가 보인다면 비정상으로 판단

**단 GC와 GC 사이의 간격이 넓다면 정상으로 간주한다.**

|  |
| --- |
| **$ tail -f gc.log.20170523113119**  182.278: [GC [PSYoungGen: 195658K->12765K(270336K)] 405054K->234073K(969728K), 0.0317180 secs] [Times: user=0.06 sys=0.01, real=0.04 secs]  194.332: [GC [PSYoungGen: 201693K->12224K(268288K)] 423001K->245396K(967680K), 0.0428970 secs] [Times: user=0.07 sys=0.01, real=0.04 secs]  352.075: [GC [PSYoungGen: 257408K->15776K(302080K)] 628028K->402092K(1001472K), 0.0556460 secs] [Times: user=0.10 sys=0.00, real=0.06 secs]  498.763: [GC [PSYoungGen: 75588K->288K(299520K)] 461905K->402244K(998912K), 0.0359840 secs] [Times: user=0.04 sys=0.01, real=0.04 secs]  **498.799:** [Full GC [PSYoungGen: 288K->0K(299520K)] [ParOldGen: 401956K->347983K(699392K)] 402244K->347983K(998912K) [PSPermGen: 72985K->70633K(143872K)], 1.4463740 secs] [Times: user=2.71 sys=0.00, real=1.44 secs]  508.691: [GC [PSYoungGen: 3035K->64K(302080K)] 351018K->348047K(1001472K), 0.0066010 secs] [Times: user=0.01 sys=0.00, real=0.00 secs]  508.697: [Full GC [PSYoungGen: 64K->0K(302080K)] [ParOldGen: 347983K->340430K(699392K)] 348047K->340430K(1001472K) [PSPermGen: 70636K->70636K(162304K)], 0.5912480 secs] [Times: user=1.13 sys=0.00, real=0.59 secs]  509.497: [GC [PSYoungGen: 1345K->64K(302080K)] 341775K->340494K(1001472K), 0.0111840 secs] [Times: user=0.01 sys=0.00, real=0.01 secs]  509.509: [Full GC [PSYoungGen: 64K->0K(302080K)] [ParOldGen: 340430K->340427K(699392K)] 340494K->340427K(1001472K) [PSPermGen: 70637K->70637K(156160K)], 0.8215780 secs] [Times: user=1.40 sys=0.00, real=0.83 secs]  510.331: [GC [PSYoungGen: 189K->96K(303616K)] 340616K->340523K(1003008K), 0.0101970 secs] [Times: user=0.01 sys=0.00, real=0.01 secs]  510.341: [Full GC [PSYoungGen: 96K->0K(303616K)] [ParOldGen: 340427K->340463K(699392K)] 340523K->340463K(1003008K) [PSPermGen: 70637K->70637K(148480K)], 0. |

#### Heap Memory 설정 확인

서비스 되어지는 JBoss EAP 의 Heap Memory 설정 상태를 모니터링을 합니다.

ps –ef | grep ${processName}

ex) ps –ef | grep java | grep T-MS71-O-F11

항목 설명

ps 명령을 통해서 해당 프로세스의 현재 동작 여부를 확인합니다.

정상 여부 확인 : 프로세스 목록 내용에서 Heap Memory 설정값이 적정값 ( min:max 1G <= 2G) 으로 설정이 되어 있는지 확인합니다.

|  |
| --- |
| **$ ps -ef|grep java | grep T-MS71-O-F11**  jboss7 7787 7526 0 00:01:37 /usr/jdk/instances/jdk1.8.0\_121//bin/java -D[Standalone] -server -Xms1024m -Xmx1024m -XX:MetaspaceSize=256m -XX:MaxMetaspaceSize=256m -XX:+UseG1GC -XX:+UnlockDiagnosticVMOptions -XX:+G1SummarizeConcMark -XX:InitiatingHeapOccupancyPercent=35 -XX:ConcGCThreads=2 -XX:+UseLargePagesInMetaspace -verbose:gc -Xloggc:/log/jboss7/T-MS71-O/T-MS71-O-F11/gclog/gc\_20170530172716.log -Xloggc:/log/jboss7/T-MS71-O/T-MS71-O-F11/gclog/gc.log -XX:+PrintGCDetails -XX:+PrintGCTimeStamps -XX:+PrintAdaptiveSizePolicy -XX:+ExplicitGCInvokesConcurrent -XX:+DisableExplicitGC -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/log/jboss7/T-MS71-O/T-MS71-O-F11/heaplog/T-MS71-O-F11.hprof.20170530172716 -Djava.net.preferIPv4Stack=true -Dorg.jboss.resolver.warning=true -Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.awt.headless=true -Djboss.modules.system.pkgs=org.jboss.byteman,scouter,org.jboss.logmanager -Djboss.server.base.dir=/was/jboss7/domains/T-MS71-O/T-MS71-O-F11 -Djboss.server.log.dir=/log/jboss7/T-MS71-O/T-MS71-O-F11/log -Djboss.socket.binding.port-offset=4431 -Djboss.node.name=T-MS71-O-F11 -Djboss.server.base.dir=/was/jboss7/domains/T-MS71-O/T-MS71-O-F11 -Djboss.bind.address=0.0.0.0 -Djboss.bind.address.management=40.10.22.156 -Djboss.socket.binding.port-offset=4431 -Djboss.default.jgroups.stack=udp -Djboss.default.jgroups.stack.tcp.tcpping.initial\_hosts=192.168.0.143[12031],192.168.0.144[12031] -Djboss.default.jgroups.stack.tcp.tcpping.num\_initial\_members=2 -Djgroups.bind\_addr=0.0.0.0 -Djgroups.tcp.bind\_port=12031 -Djboss.default.multicast.address=230.1.44.31 -Djboss.messaging.group.address=231.2.44.31 -Djboss.modcluster.multicast.address=224.3.44.31 -Djboss.http.port=8080 -Djboss.http.final.port=12511 -Djboss.ajp.port=8009 -Djboss.ajp.final.port=12440 -Xbootclasspath/p:/was/jboss7/jboss-eap-7.0/modules/system/layers/base/org/jboss/logmanager/main/jboss-logmanager-2.0.3.Final-redhat-1.jar -Djava.util.logging.manager=org.jboss.logmanager.LogManager -Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=10430 -Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.authenticate=false -Dserver.mode=local -Dorg.jboss.as.logging.per-deployment=false -Dorg.jboss.boot.log.file=/log/jboss7/T-MS71-O/T-MS71-O-F11/log/server.log -Dlogging.configuration=file:/was/jboss7/domains/T-MS71-O/T-MS71-O-F11/configuration/logging.properties -jar /was/jboss7/jboss-eap-7.0/jboss-modules.jar -mp /was/jboss7/jboss-eap-7.0/modules:/was/jboss7/jboss-eap-7.0/modules.ext org.jboss.as.standalone -Djboss.home.dir=/was/jboss7/jboss-eap-7.0 -Djboss.server.base.dir=/was/jboss7/domains/T-MS71-O/T-MS71-O-F11 -DSERVER=T-MS71-O-F11 -P=/was/jboss7/domains/T-MS71-O/T-MS71-O-F11/bin/env.properties -c standalone-ha.xml |

### JDBC DataSource 상태 모니터링

#### Data Source 설정 모니터링

connection 설정값이 적정하게 설정되어 있는지 모니터링을 합니다.

JBoss CLI Tool

**ex) ./jboss-cli.sh**

**/subsystem=datasources/data-source=ExampleDS:read-resource**

항목 설명

각 EAP 서버 홈의 실행 폴더에 있는 JBoss Command Line Interface (jboss-cli.sh) 를 통해서 서버의 Datasource JNDI 값을 입력 각각의 Datasource 설정을 확인합니다.

|  |
| --- |
| **$ ./jboss-cli.sh**  **/subsystem=datasources/data-source=ExampleDS:read-resource**  {  "outcome" => "success",  "result" => {  "allocation-retry" => undefined,  "allocation-retry-wait-millis" => undefined,  "allow-multiple-users" => false,  "background-validation" => undefined,  "background-validation-millis" => undefined,  "blocking-timeout-wait-millis" => undefined,  "capacity-decrementer-class" => undefined,  "capacity-decrementer-properties" => undefined,  "capacity-incrementer-class" => undefined,  "capacity-incrementer-properties" => undefined,  "check-valid-connection-sql" => undefined,  "connectable" => false,  "connection-listener-class" => undefined,  "connection-listener-property" => undefined,  "connection-url" => "jdbc:h2:mem:test;DB\_CLOSE\_DELAY=-1;DB\_CLOSE\_ON\_EXIT=FALSE",  "datasource-class" => undefined,  "driver-class" => undefined,  "driver-name" => "h2",  "enabled" => true,  "enlistment-trace" => true,  "exception-sorter-class-name" => undefined,  "exception-sorter-properties" => undefined,  "flush-strategy" => undefined,  "idle-timeout-minutes" => undefined,  "initial-pool-size" => undefined,  "jndi-name" => "java:jboss/datasources/ExampleDS",  "jta" => true,  "max-pool-size" => undefined,  "mcp" => "org.jboss.jca.core.connectionmanager.pool.mcp.SemaphoreConcurrentLinkedDequeManagedConnectionPool",  "min-pool-size" => undefined,  "new-connection-sql" => undefined,  "password" => "sa",  "pool-fair" => undefined,  "pool-prefill" => undefined,  "pool-use-strict-min" => undefined,  "prepared-statements-cache-size" => undefined,  "query-timeout" => undefined,  "reauth-plugin-class-name" => undefined,  "reauth-plugin-properties" => undefined,  "security-domain" => undefined,  "set-tx-query-timeout" => false,  "share-prepared-statements" => false,  "spy" => false,  "stale-connection-checker-class-name" => undefined,  "stale-connection-checker-properties" => undefined,  "statistics-enabled" => false,  "track-statements" => "NOWARN",  "tracking" => false,  "transaction-isolation" => undefined,  "url-delimiter" => undefined,  "url-selector-strategy-class-name" => undefined,  "use-ccm" => true,  "use-fast-fail" => false,  "use-java-context" => true,  "use-try-lock" => undefined,  "user-name" => "sa",  "valid-connection-checker-class-name" => undefined,  "valid-connection-checker-properties" => undefined,  "validate-on-match" => undefined,  "connection-properties" => undefined,  "statistics" => {  "pool" => undefined,  "jdbc" => undefined  }  }  } |

#### Connection Pool 설정 모니터링

connection 설정값이 적정하게 설정되어 있는지 모니터링을 합니다.

JBoss CLI Tool

ex) **./jboss-cli.sh**

**/subsystem=datasources/data-source=ExampleDS/statistics=pool:read-resource\(include-runtime=true\)**

항목 설명

각 EAP 서버 홈의 실행 폴더에 있는 JBoss Command Line Interface (jboss-cli.sh) 를 통해서 서버의 Datasource JNDI 값을 입력 각각의 JDBC Connection현재 통계값을 확인합니다.

|  |
| --- |
| **$ ./jboss-cli.sh**  **/subsystem=datasources/data-source=ExampleDS/statistics=pool:read-resource\(include-runtime=true\)**  {  "outcome" => "success",  "result" => {  "ActiveCount" => "0",  "AvailableCount" => "20",  "AverageBlockingTime" => "0",  "AverageCreationTime" => "0",  "CreatedCount" => "0",  "DestroyedCount" => "0",  "InUseCount" => "0",  "MaxCreationTime" => "0",  "MaxUsedCount" => "0",  "MaxWaitCount" => "0",  "MaxWaitTime" => "0",  "TimedOut" => "0",  "TotalBlockingTime" => "0",  "TotalCreationTime" => "0"  }  } |

### 로그 모니터링

#### Critical Message

서버 로그 중 서비스에 문제가 될 소지가 있는 Message 여부를 모니터링 확인합니다.

egrep “Critical|ERROR|OutOfMemory” server.log

항목 설명

grep 명령을 통해서 해당 로그 파일 내에 비정상적인 로그 부분을 확인합니다.

정상 여부 확인 : 특이한 에러 로그 등이 없다면 정상입니다.

|  |
| --- |
| ^[[0m23:05:25,169 INFO [org.jboss.modules] (main) JBoss Modules version 1.3.0.Final-redhat-2  ^[[0m^[[0m23:05:25,457 INFO [org.jboss.msc] (main) JBoss MSC version 1.0.4.GA-redhat-1  ^[[0m^[[0m23:05:25,557 INFO [org.jboss.as] (MSC service thread 1-2) JBAS015899: JBoss EAP 7.0.4.GA (AS 7.3.0.Final-redhat-14) starting  ^[[0m^[[0m23:05:26,532 INFO [org.jboss.as.server.deployment.scanner] (DeploymentScanner-threads - 1) JBAS015003: Found spring.war in deployment directory. To trigger deployment create a file called spring.war.dodeploy  ^[[0m^[[0m23:05:26,537 INFO [org.jboss.as.server.deployment.scanner] (DeploymentScanner-threads - 1) JBAS015003: Found test.war in deployment directory. To trigger deployment create a file called test.war.dodeploy  ^[[0m^[[0m23:05:26,583 INFO [org.xnio] (MSC service thread 1-1) XNIO Version 3.0.7.GA-redhat-1  ^[[0m^[[0m23:05:26,593 INFO [org.xnio.nio] (MSC service thread 1-1) XNIO NIO Implementation Version 3.0.7.GA-redhat-1  ^[[0m^[[0m23:05:26,596 INFO [org.jboss.as.server] (Controller Boot Thread) JBAS015888: Creating http management service using socket-binding (management-http)  ^[[0m^[[0m23:05:26,600 INFO [org.jboss.remoting] (MSC service thread 1-1) JBoss Remoting version 3.2.18.GA-redhat-1  ^[[0m^[[0m23:05:26,647 INFO [org.jboss.as.clustering.infinispan] (ServerService Thread Pool -- 29) JBAS010280: Activating Infinispan subsystem.  ^[[0m^[[0m23:05:26,694 INFO [org.jboss.as.connector.logging] (MSC service thread 1-1) JBAS010408: Starting JCA Subsystem (IronJacamar 1.0.23.Final-redhat-1)  ^[[0m^[[0m23:05:26,709 INFO [org.jboss.as.security] (ServerService Thread Pool -- 42) JBAS013171: Activating Security Subsystem  E486: Pattern not found: ERROR 2,1 Top  ^[[0m^[[0m23:05:26,694 INFO [org.jboss.as.connector.logging] (MSC service thread 1-1) JBAS010408: Starting JCA Subsystem (IronJacamar 1.0.23.Final-redhat- |

