

Domain 안내서

JEUS 9

TMAXSOFT

저작권 공지

Copyright 2025. TmaxSoft Co., Ltd. All Rights Reserved.

회사 정보

(주)티맥스소프트

주소 : 경기도 성남시 분당구 황새울로258번길 29, 티맥스수내타워 8-9층

기술 서비스 센터: 1544-8629

홈페이지: <https://www.tmaxsoft.com>

제한된 권리

이 소프트웨어(JEUS®) 사용설명서와 프로그램은 저작권법과 국제 조약에 의해 보호됩니다. 사용설명서와 프로그램은 TmaxSoft Co., Ltd.와의 사용권 계약 하에서만 사용할 수 있으며, 사용설명서는 사용권 계약의 범위 내에서만 배포 또는 복제할 수 있습니다. 이 사용설명서의 전부 또는 일부를 TmaxSoft의 사전 서면 동의 없이 전자, 기계, 녹음 등의 수단으로 전송, 복제, 배포하거나 2차적 저작물을 작성할 수 없습니다.

이 소프트웨어 사용설명서와 프로그램의 사용권 계약은 어떠한 경우에도 사용설명서 및 프로그램과 관련된 지적 재산권(등록 여부를 불문)을 양도하는 것으로 해석되지 않으며, 브랜드나 로고, 상표 등을 사용할 권한을 부여하지 않습니다. 사용설명서는 오로지 정보 제공만을 목적으로 하며, 이로 인한 계약상의 직접적 또는 간접적 책임을 지지 않습니다. 또한 사용설명서 상의 내용이 법적 또는 상업적인 특정 조건을 만족시킬 것을 보장하지 않습니다. 사용설명서는 제품의 업그레이드나 수정에 따라 예고 없이 변경될 수 있으며, 내용상의 오류가 없음을 보장하지 않습니다.

상표 공지

JEUS®는 TmaxSoft Co., Ltd.의 등록 상표입니다.

Java, Solaris는 Oracle Corporation 및 그 자회사, 관계회사의 등록 상표입니다.

Microsoft, Windows, Windows NT는 Microsoft Corporation의 등록 상표 또는 상표입니다.

HP-UX는 Hewlett Packard Enterprise Company의 등록 상표입니다.

AIX는 International Business Machines Corporation의 등록 상표입니다.

UNIX는 X/Open Company, Ltd.의 등록 상표입니다.

Linux는 Linus Torvalds의 등록 상표입니다.

Noto는 Google Inc.의 상표입니다. Noto 글꼴은 오픈 소스입니다. 모든 Noto 글꼴은 SIL Open Font License, 버전 1.1에 따라 게시됩니다. (<https://www.google.com/get/noto/>)

기타 모든 제품들과 회사 이름은 각각 해당 소유주의 상호, 상표, 또는 등록 상표입니다.

본 사용설명서에 기재된 회사, 시스템, 제품 이름 등에 반드시 상표 표시(™, ®)를 하지는 않습니다.

오픈 소스 소프트웨어 공지

본 제품의 일부 파일 또는 모듈은 다음의 라이선스를 준수합니다.: APACHE2.0, CDDL1.0, EDL1.0, OPEN SYMPHONY SOFTWARE1.1, TRILEAD-SSH2, Bouncy Castle, BSD, MIT, SIL OPEN FONT1.1

관련 상세 정보는 제품의 다음 디렉터리에 기재된 사항을 참고하시기 바랍니다. :

`${INSTALL_PATH}/license/oss_licenses`

유지 보수

구분	지원항목	서비스 내용
제품지원	패치 & 업그레이드	무상 패치 서비스 제공
		메이저 버전 업그레이드 시 할인 혜택
		웹 지원을 통한 패치 내역 제공
기술 지원 - 기본 서비스	장애 지원	장애 발생 시 원인 분석 및 조치 Service Desk팀 → 기술팀 → R&D의 3단계 장애 분석 및 조치
	일상 지원(온라인 지원)	E-mail, 전화, 원격, 웹 사이트 등 온라인 자원을 통한 질의 응답 서비스
	고객 맞춤 지원(방문 지원)	고객의 요청으로 수행하는 방문 지원 서비스
기술 지원 - 옵션 서비스	예방 지원	정기 점검을 통한 시스템 운영현황 보고 및 장애 예방 ◦ 관리자 또는 운영자의 요구사항 수렴 ◦ 운영 현황(시스템, 엔진 운영) 보고서 제공 ◦ 필요 시 시스템 개선 권장 사항 보고
유지 보수 비용 및 기간	계약 시 별도 협의	계약 시 EOL/EOS 문서 제공

안내서 이력

제품 버전	안내서 버전	발행일	비고
JEUS 9	3.1.2	2025-01-03	-
JEUS 9	3.1.1	2024-09-27	-

목차

1. 소개	1
1.1. 기본 개념	1
1.2. JEUS와 도메인 관계	1
1.3. 구성요소	2
1.3.1. Master Server(MASTER)	3
1.3.2. Managed Server(MS)	4
1.4. 운영모드와 개발모드	5
1.5. 도메인 구성 제약 사항	5
2. 도메인 생성	7
2.1. 도메인 생성	7
2.1.1. 기본 도메인 생성	7
2.1.2. Virtual Multicast를 사용하는 도메인 생성	9
2.2. 도메인 구성	11
2.2.1. 콘솔 툴 사용	12
2.3. 디렉터리 구성	13
2.3.1. JEUS_HOME 디렉터리	13
2.3.2. 도메인 디렉터리	15
3. 도메인 설정변경	17
3.1. 개요	17
3.2. 설정변경	17
3.3. 설정변경 적용	17
3.3.1. 동적 반영이 가능한 설정	17
3.3.2. 서버의 재시작이 필요한 설정	18
3.4. 설정 로테이션	19
3.5. 도메인 공통 설정변경	19
3.6. 설정변경 예제	22
3.6.1. System Thread Pool 개수 변경	22
3.6.2. 서버의 JVM 설정변경	22
4. 서버 Life Cycle	24
4.1. 서버 시작 준비	24
4.2. 서버 시작	24
4.2.1. Master Server(MASTER) 시작	24
4.2.2. Managed Server(MS) 시작	25
4.3. 서버 종료	26
4.3.1. Master Server(MASTER) 종료	26
4.3.2. Managed Server(MS) 종료	27
4.4. 서버 Life Cycle 상태 확인	28
5. JEUS 클러스터링	31
5.1. 개요	31

5.2. 클러스터와 도메인 관계	31
5.3. 클러스터 기능.....	32
5.4. 클러스터 생성.....	33
5.5. 클러스터 설정.....	33
5.5.1. 서버 공통 설정	33
5.5.2. 클러스터 설정	33
5.6. 클러스터 변경.....	34
5.6.1. 클러스터에 서버 추가	34
5.6.2. 클러스터에서 서버 삭제	35
5.7. 클러스터 삭제.....	36
6. 서버 장애.....	38
6.1. Master Server(MASTER) 비정상 종료	38
6.1.1. 비정상 종료될 때 기능 제한.....	38
6.1.2. 도메인 백업을 이용한 장애 극복	38
6.1.3. INDEPENDENT 모드의 Managed Server(MS)	40
6.2. Managed Server(MS)의 비정상 종료.....	40
7. Security 관리	41
7.1. 계정 관리	41
7.2. 패스워드 암호화 저장	41
7.2.1. 콘솔 툴 사용	42
7.2.2. 암호화 툴을 통한 직접 설정.....	42
7.3. 비밀 Key 파일 관리	43
7.3.1. 비밀 Key 파일의 생성과 관리	43
7.3.2. 비밀 Key 파일 보호 옵션과 서버 기동 방법.....	43
7.4. Keystore와 Truststore 관리	44
7.5. SSL이 설정된 도메인에서의 실행 스크립트 관리.....	45

1. 소개

본 장에서는 도메인의 개념과 구성 요소에 대해 설명한다.

1.1. 기본 개념

도메인(Domain)은 관련 있는 서버들의 그룹으로 기본 관리 단위이다. 하나의 도메인은 비즈니스에 따라 여러 개의 서버와 클러스터로 구성된다.

각 도메인마다 서비스의 종류, 서버의 위치, 보안 정책들이 모두 다를 수 있다. 이에 따라 각 도메인은 하나의 설정 파일을 가지고, 여러 도메인 사이에 설정을 공유할 수 없다. 즉, 하나의 도메인 내에서만 보안 설정, 필요한 애플리케이션 및 리소스 등이 공유되는 점을 고려하여 도메인을 구성해야 한다.

도메인을 구성할 때 고려해야 하는 사항은 다음과 같다.

- 해당 도메인에 속한 서버들의 역할

비즈니스 관계가 높은 업무들을 하나의 도메인으로 구성한다.

- 해당 도메인에 속한 서버들의 물리적 위치

안정적인 서비스를 위해 같은 업무를 수행하는 서버의 물리적 위치를 적절하게 분배한다.

- 해당 도메인에 속한 서버들에게 가해지는 부하량

수행하는 업무의 부하량에 따라 적당량의 서버를 구성한다.

1.2. JEUS와 도메인 관계

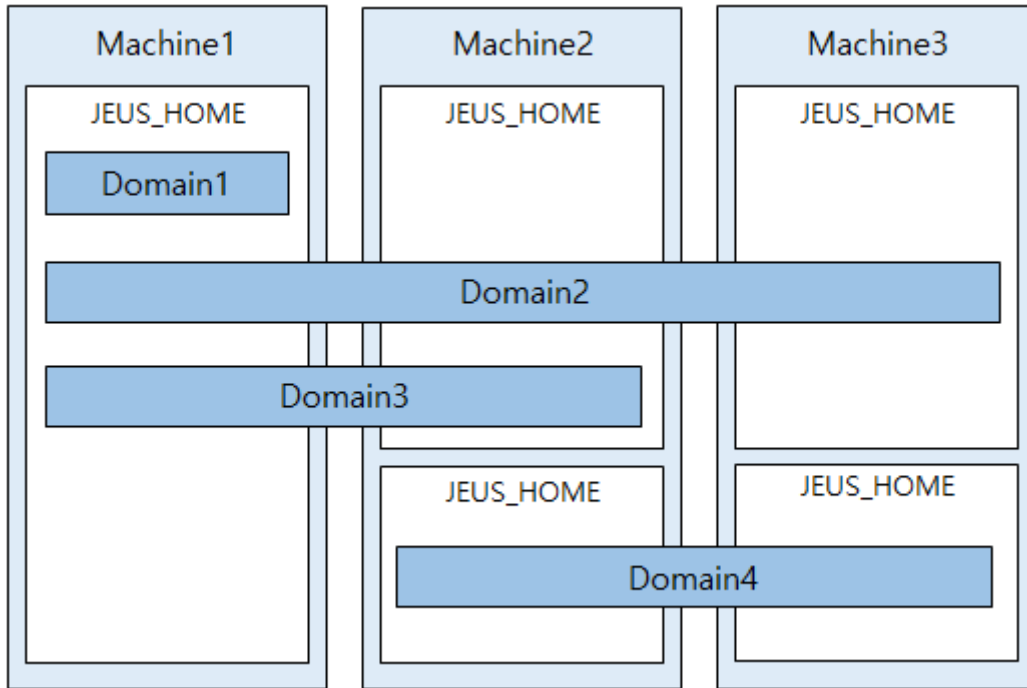
하나의 머신에는 하나의 JEUS(JEUS_HOME)가 필요하고, 하나의 JEUS(JEUS_HOME)에는 한 개 이상의 도메인이 존재할 수 있다. 또한 하나의 도메인은 한 개의 JEUS(JEUS_HOME) 내에 구성될 수도 있고, 여러 개의 머신 즉 여러 개의 JEUS(JEUS_HOME)에 걸쳐 구성될 수도 있다.



한 개의 머신에는 일반적으로 한 개의 JEUS를 설치하기 때문에 JEUS_HOME도 한 개이다. 그러나 하나의 머신에 여러 개의 JEUS를 설치하여 여러 개의 JEUS_HOME이 존재하는 것도 가능하다.

예를 들어 인사 관리 업무 도메인과 주문 관리 도메인을 각 업무의 특성에 따라 머신 3대에 구성할 수 있다. 인사 관리 업무의 경우에는 하나의 머신에서만 서비스하도록 Machine1에 Domain1를 구성하고, 주문 관리 업무의 경우에는 고객의 주문이 장비 결함 등으로 인해 중단되는 경우를 방지하기 위해 3대의 머신에 걸쳐 Domain2를 구성할 수 있다.

다음은 JEUS 도메인의 관계를 나타낸 그림이다.



JEUS와 도메인의 관계

1.3. 구성요소

도메인은 Master Server(MASTER), Managed Server(MS), Cluster로 구성되어 있다.

- **Master Server(MASTER)**

도메인 내에는 Master Server(이하 MASTER)라는 특별한 하나의 서버가 항상 존재한다. MASTER는 도메인 내 서버들 간의 설정과 도메인 내의 모든 애플리케이션 및 리소스를 중앙에서 관리하고, 서버들을 제어 및 모니터링하는 관리 톨과 통신한다.

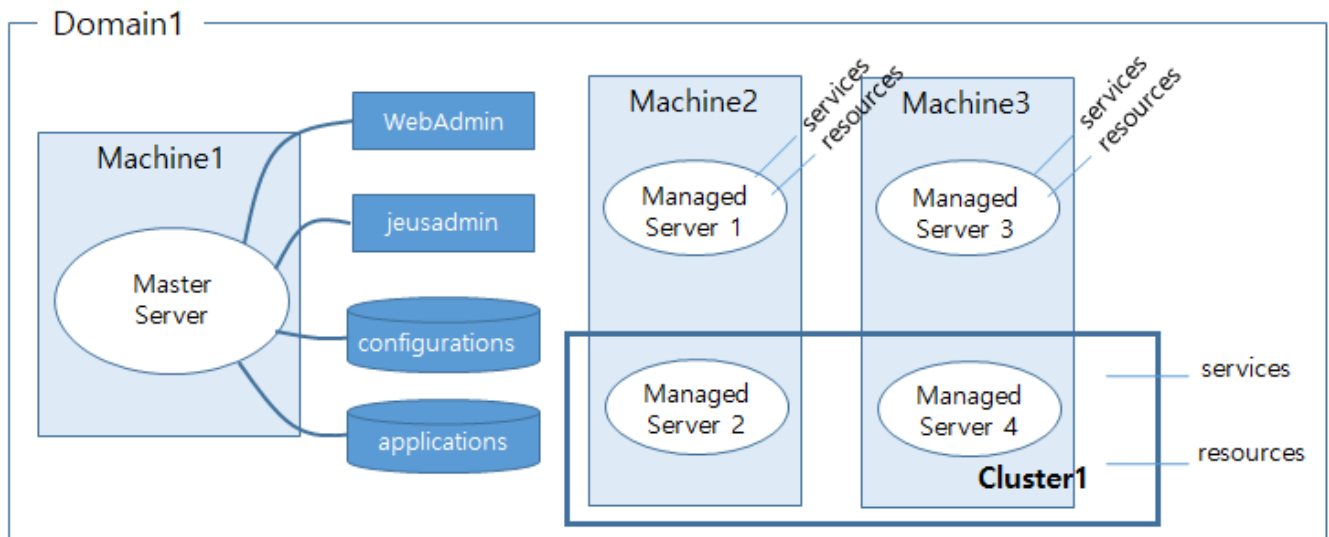
- **Managed Server(MS)**

실제 서비스를 담당하는 Managed Server(이하 MS)가 도메인 내에는 1개 이상 존재한다. 같은 서비스를 하는 여러 MS를 하나의 클러스터로 구성하거나 서비스별 단독 MS로 도메인을 구성할 수 있다.

- **Cluster**

도메인 내에는 0개 이상의 클러스터들과 1개 이상의 클러스터에 포함되지 않는 서버들이 존재한다.

클러스터는 확장성과 안정성을 위해 동일한 서비스를 수행하는 서버들의 집합을 의미한다. 서버가 클러스터의 일원이 되면 같은 애플리케이션이 deploy되고 같은 리소스를 사용하게 된다. 클러스터에 대한 자세한 내용은 [JEUS 클러스터링](#)을 참고한다.



도메인의 구성 요소

Machine1에는 MASTER가 위치한다. MASTER는 관리 툴과 통신하여 사용자의 명령을 받아 처리한다.

Machine2와 Machine3에는 각각 MS가 2개씩 존재한다. Managed Server2와 Managed Server4는 Cluster1이라는 하나의 클러스터를 구성하여 애플리케이션과 리소스를 공유하고 같은 서비스를 수행한다. Cluster1에 포함되지 않는 단독 서버인 Managed Server1과 Managed Server3은 각각 애플리케이션과 리소스를 갖는다.



가장 작은 도메인은 하나의 서버로 구성된 도메인으로 하나의 서버가 MASTER와 MS 역할을 하게 된다. 이는 개발 단계에서의 가벼운 테스트를 위한 도메인으로만 사용하고, 운영 환경에서는 관리와 서비스를 분리하여 MASTER는 MS를 관리하는 목적으로만 사용하기를 권장한다.

1.3.1. Master Server(MASTER)

MASTER는 모든 도메인에 반드시 하나씩 존재해야 하는 서버로 MS들을 중앙에서 관리 및 모니터링한다.

다음은 MASTER의 주요 기능이다.

- 도메인 전체의 설정을 관리한다.

- 도메인 내의 모든 MS가 다른 머신에 존재하더라도 동일한 설정으로 구동할 수 있다.

MASTER가 먼저 구동되어야 MS가 MASTER로부터 설정을 받아 구동될 수 있다. MS가 구동될 때 MASTER의 부재로 동기화되지 않은 설정은 MASTER가 구동될 때 동기화된다.

- 변경된 설정 사항은 도메인 내의 모든 MS에 반영된다.

MASTER는 변경된 설정을 검증하여 다른 서버들로 분배하여 도메인 내의 설정을 동기화한다. 그렇기 때문에 설정 변경은 반드시 MASTER를 통해서 진행되어야 하고, MASTER가 운영 중일 때만 가능하다.

MASTER의 설정 관리에 대한 자세한 설명은 [도메인 설정변경](#)을 참고한다.

- **도메인 전체의 애플리케이션을 관리한다.**

- 도메인 내의 모든 애플리케이션의 상태를 관리한다.

MASTER에서는 도메인에 존재하는 모든 애플리케이션의 상태를 관리하고, 도메인에서 애플리케이션의 상태가 하나의 상태로 유지될 수 있도록 한다. MS의 애플리케이션 상태가 MASTER에서 관리하고 있는 애플리케이션의 상태와 동일하게 유지될 수 있도록 한다.

- 도메인 내의 모든 애플리케이션의 파일을 관리한다.

MASTER에서는 MS가 구동될 때와 MS에서 애플리케이션이 deploy되는 시점에 파일을 동기화한다. MS가 구동될 때 MASTER의 부재로 애플리케이션 파일을 동기화하지 못했다면 MASTER가 구동될 때 MS에서 deploy에 실패한 애플리케이션에 대해 다시 동기화한다.

- 도메인에 속한 서버나 클러스터에 애플리케이션을 deploy하는 작업을 관리한다.

도메인에서 애플리케이션을 서비스하기 위해서는 Install Application 명령어를 통해 서비스할 애플리케이션을 MASTER에 설치하거나, MASTER에서 알 수 있는 애플리케이션 저장소(repository)에 위치해야 한다.

MS에서 애플리케이션을 서비스하기 위해서는 MASTER에 서비스 대상을 주고 애플리케이션 deploy 명령어를 실행해야 한다. deploy 명령어뿐만 아니라 애플리케이션 조회나 제어에 관련된 모든 명령어는 MASTER를 통해서 수행한다. 만약 MASTER가 종료된 상태라면 도메인에서 서비스되는 애플리케이션에 대해 어떤 명령어도 실행할 수 없다.

MS가 클러스터에 속해 있다면 클러스터 타겟(Cluster Target)으로만 deploy될 수 있다. MASTER에서는 클러스터에 서버가 추가된 경우 해당 서버에서도 이 애플리케이션을 서비스될 수 있도록 deploy의 성공을 보장한다.

MASTER에서의 애플리케이션 관리에 대한 자세한 설명은 JEUS Applications & Deployment 안내서의 [애플리케이션 관리](#)를 참고한다.

- **도메인 전체를 모니터링 및 제어할 수 있다.**

관리 툴을 통해 도메인 내의 모든 서버, 서비스, 애플리케이션 및 리소스를 한 번에 모니터링 및 제어할 수 있다.

1.3.2. Managed Server(MS)

MS에는 필요한 리소스가 위치하고, 애플리케이션들이 deploy된다.

MS는 MASTER에 의해 다음과 같이 동작한다.

- **설정 및 애플리케이션 동기화**

도메인의 전체 설정과 애플리케이션은 MASTER가 관리하기 때문에 MASTER로 연결하여 설정 정보와 해당 MS에 deploy되어야 할 애플리케이션을 전달받는다. 이때 성능을 위하여 MASTER로 받아온 설정 정보와 애플리케이션은 read-only 상태로 로컬에 cache된다. MASTER와 연결된 시점에 로컬의 Cache와 일치하지 않는 파일만을 받아와 Cache를 업데이트하여 MASTER와 동기화한다. 즉, MS에 존재하는 설정이나 애플리케이션을 변경해도 반영되지 않고, MASTER를 통해 변경되어야 모든 서버에 반영된다.

- 설정의 동기화

설정 동기화는 MS가 구동될 때 MS가 MASTER와 연결이 끊어졌다가 재연결된 경우 사용자가 톨을 통해서 명시적으로 도메인 설정을 변경했을 때 일어난다.

- 애플리케이션의 동기화

애플리케이션의 동기화는 파일뿐 아니라 애플리케이션의 상태도 함께 전달되기 때문에 MASTER의 애플리케이션 상태가 MS에서도 유지된다. 애플리케이션의 동기화는 MS가 구동될 때 MASTER와 연결이 끊어졌다가 재연결된 경우 일어난다. 애플리케이션 동기화와 관련된 자세한 내용은 [애플리케이션 동기화](#)를 참고한다.

- 클러스터 구성

부하 분산과 장애 극복을 위해 여러 MS를 묶어 하나의 클러스터를 구성할 수 있다. 클러스터 내의 모든 서버에 같은 리소스와 애플리케이션이 위치하는 것을 권장한다. 클러스터에 대한 자세한 내용은 [JEUS 클러스터링](#)을 참고한다.

1.4. 운영모드와 개발모드

도메인을 운영하는 방법은 다음 중 한 가지로 설정할 수 있다. 모드를 설정하기 위해서는 도메인 설정 파일(domain.xml)을 직접 수정해야 한다.

- 운영모드(Production mode)

실제 운영환경에 적합한 모드로, 웹 애플리케이션의 auto-reload나 hot-swap과 같은 기능은 제공하지 않는다. domain.xml의 production-mode 항목을 true로 설정한다.

- 개발모드(Development mode)

domain.xml의 production-mode 항목을 false로 설정하면 개발모드로 동작한다.



1. 애플리케이션 변경이 빈번하게 일어날 수 있는 개발단계에서는 개발모드로, 실 운영계에서는 운영모드로 설정해서 사용할 것을 권장한다.
2. 설정을 변경하고 난 뒤에는 도메인에 운영 중인 서버를 모두 내리고 재기동해 주어야 한다.

1.5. 도메인 구성 제약 사항

도메인을 구성할 때 다음과 같은 제약 사항이 존재한다.

- 하나의 도메인에 반드시 하나의 MASTER가 존재해야 한다.
- 도메인 사이에는 설정 정보와 리소스를 공유할 수 없다.
- 도메인 내의 모든 서버는 같은 버전을 사용해야 한다.
- 도메인 내의 모든 서버와 클러스터의 이름은 각각 유일해야 한다.

- 설치한 하나의 JEUS 내에 모든 도메인 이름은 유일해야 한다.

2. 도메인 생성

본 장에서는 도메인을 생성할 때 필요한 조건과 JEUS가 제공하는 툴(jeusadmin)을 사용하여 도메인 생성 방법, 도메인 생성 후 디렉터리 구조에 대해 설명한다.

2.1. 도메인 생성

도메인을 생성하면 도메인 디렉터리가 생성되고 그 안에 도메인에 속한 서버들을 시작하고 종료할 수 있는 스크립트와 기본적인 설정 파일들이 생성된다. 각 도메인의 최상위 디렉터리는 DOMAIN_HOME이라고 하고, 다음의 경로에 위치한다.

```
JEUS_HOME/domains/<domain_name>
```

도메인을 처음 생성하면 MASTER만 포함된 도메인이 생성되며 이후 MS를 추가하여 원하는 도메인을 구성한다. 추가한 MS를 원하는 머신에서 시작하면 해당 머신에 필요한 도메인 정보가 MASTER를 통해 전달된다. 즉, MASTER는 도메인 생성을 통해 도메인 디렉터리를 생성하고, MS는 처음 시작할 때 MASTER를 통해 도메인 디렉터리를 생성한다.

Ant Task나 jeusadmin을 통해 도메인을 생성할 수 있다. 이때 다음의 2가지 조건이 필요하다.

- 도메인 구성에 필요한 네트워크 설정

JEUS는 기본적으로 도메인 내의 서버들 사이에 Multicast로 서로의 상태를 공유하게 되므로 IP 주소와 Port가 필요하다. 도메인 구성이 서브넷을 벗어나서 Multicast로 정보를 공유할 수 없는 경우에는 TCP로 Multicast를 대체한다. 이를 JEUS에서는 Virtual Multicast라고 하며, 도메인을 생성할 때 Virtual Multicast를 사용하겠다는 것을 명시해 줄 수 있다.

이때에는 서버의 기본 Listen IP 주소, Port 번호를 이용하여 서버들 사이에 정보를 공유하게 된다. 따라서 서브넷을 벗어난 도메인 구성에서는 반드시 모든 서버가 Listen IP 주소, Port 번호를 명시해야 한다. 또한 Multicast를 사용하지 못한다는 것도 명시해야 한다.

- MASTER의 기본 설정

하나의 도메인에는 하나의 MASTER가 반드시 포함되기 때문에 도메인을 생성할 때 MASTER의 기본 설정을 명시해야 한다. 동적 변경이 가능한 서버 설정들은 도메인 생성 후 변경하면 되지만, 그렇지 않은 설정들은 도메인 생성할 때 설정해야 한다. MASTER의 기본 Listen IP 주소와 Port 번호 그외 MASTER를 띄울 JVM의 정보를 설정한다.

2.1.1. 기본 도메인 생성

본 절에서는 기본 도메인을 생성하는 방법에 대해 설명한다. 기본 도메인을 생성할 때에는 Virtual Multicast가 동작하지 않음에 유의한다.

Ant Task와 jeusadmin 명령에 사용할 속성들은 공통적으로 다음과 같은 기본값을 갖는다.

속성	기본값
MASTER 이름	adminServer
Listen IP 주소	0.0.0.0 / 0:0:0:0:0:0:0
Listen Port 번호	9736
사용자 이름	administrator
Multicast IP 주소	230.30.1.1 / FF01:0:0:0:0:0:1
Multicast Port 번호	3030

• Ant Task를 사용한 도메인 생성

Ant Task에서 설정의 기본값은 JEUS_HOME/setup/domain-config-template.properties에 정의되어 있다. 기본값은 사용자가 별도로 설정하지 않아도 자동으로 값이 설정되지만 아래의 생성 예제에서는 설정 방법 확인을 위해 표시한다.

```
JEUS_HOME/setup$ ant create-domain -Ddomain=jeus_domain
-Dservername=adminServer
-Djeus.address=0.0.0.0
-Djeus.port=9736
-Dtransport.address=230.30.1.1
-Dtransport.port=3030
```

• jeusadmin을 사용한 도메인 생성

jeusadmin에서 설정 기본값은 JEUS_HOME/setup/domain-config-template.properties에 정의되어 있다. domain-config-template.properties에 jeus.password 설정이 있다면 설정을 따르고 없다면 아래 예제와 같이 password를 입력 받는다.

```
offline>create-domain -domain jeus_domain -mastername adminServer -taddress 230.30.1.1 -tport 3030
Enter the password for [administrator]: *****
Confirm the password: *****
Do you want to encrypt the password? (y/n): y
=====
The domain [jeus_domain] was created successfully.
+-----+-----+
| Property | Value |
+-----+-----+
| Domain name | jeus_domain |
| Master Name | adminServer |
| Master Base Listen Port | 9736 |
| Master Base Listen Address | 0.0.0.0 |
| HTTP Port | 8088 |
| User Name | administrator |
| Node Name | node1 |
| Password | {base}amV1cw== |
| JVM Option | -Xmx256m -XX:MaxMetaspaceSize=128m |
| Production Mode | true |
| Transport Type | HYBRID |
+-----+-----+
```

2.1.2. Virtual Multicast를 사용하는 도메인 생성

본 절에서는 IP Multicast를 사용할 수 있는 환경인지 테스트하는 방법, 그리고 IP Multicast를 사용할 수 없는 환경이나 도메인이 서브넷을 넘어서는 범위에 걸쳐서 구성되어있을 경우에 Virtual Multicast를 사용하는 도메인의 생성 방법에 대해 설명한다.

IP Multicast를 사용할 수 있는 환경인지 테스트하기 위해서는 mcastSender와 mcastReceiver라는 두 개의 테스트 스크립트를 실행하면 된다. 두 개의 스크립트는 JEUS_HOME/bin에 위치하고 있으며 다음과 같이 각각 다른 머신에서 실행하면 된다.

```
JEUS_HOME/bin$ mcastReceiver -addr 224.0.0.1 -port 3030
Socket == 0.0.0.0/0.0.0.0 : 3030, bound to /fe80:0:0:0:2a37:37ff:fe1e:fc4%5
Socket == 0.0.0.0/0.0.0.0 : 3030, bound to /192.168.0.26
Socket == 0.0.0.0/0.0.0.0 : 3030, bound to /fe80:0:0:0:3e07:54ff:fe6f:95e%4
Socket == 0.0.0.0/0.0.0.0 : 3030, bound to /61.77.153.207
Socket == 0.0.0.0/0.0.0.0 : 3030, bound to /fe80:0:0:0:0:0:0:1%1
Socket == 0.0.0.0/0.0.0.0 : 3030, bound to /0:0:0:0:0:0:0:1
Socket == 0.0.0.0/0.0.0.0 : 3030, bound to /127.0.0.1
```

```
JEUS_HOME/bin$ mcastSender -addr 224.0.0.1 -port 3030
1 == 0.0.0.0/0.0.0.0 : 3030, bound to /fe80:0:0:0:2a37:37ff:fe1e:fc4%5
2 == 0.0.0.0/0.0.0.0 : 3030, bound to /192.168.0.26
3 == 0.0.0.0/0.0.0.0 : 3030, bound to /fe80:0:0:0:3e07:54ff:fe6f:95e%4
4 == 0.0.0.0/0.0.0.0 : 3030, bound to /61.77.153.207
5 == 0.0.0.0/0.0.0.0 : 3030, bound to /fe80:0:0:0:0:0:0:1%1
6 == 0.0.0.0/0.0.0.0 : 3030, bound to /0:0:0:0:0:0:0:1
7 == 0.0.0.0/0.0.0.0 : 3030, bound to /127.0.0.1
```

이 테스트는 머신의 모든 네트워크 인터페이스를 통해서 IP Multicast 패킷을 송수신하려고 시도한다. 최초에 나오는 주소들은 모든 네트워크 인터페이스를 나타내는 것이다.

이어서 Sender쪽에 나타난 프롬프트에 임의의 메시지를 입력하면 Receiver쪽에 수신된다. 만일 수신이 되지 않는다면 IP Multicast가 제대로 동작하지 않는 환경이기 때문에 아래에 나오는 Virtual Multicast를 설정해야 한다. 위의 mcastSender에서 메시지 테스트의 실행 예는 다음과 같다.

```
> test
>
```

```
test [sender = 192.168.0.26 : 5555], [bind_addr=/127.0.0.1]
test [sender = 192.168.0.26 : 5555], [bind_addr=/61.77.153.207]
test [sender = 192.168.0.26 : 5555], [bind_addr=/fe80:0:0:0:3e07:54ff:fe6f:95e%4]
test [sender = 192.168.0.26 : 5555], [bind_addr=/fe80:0:0:0:3e07:54ff:fe6f:95e%4]
test [sender = 192.168.0.26 : 5555], [bind_addr=/fe80:0:0:0:2a37:37ff:fe1e:fc4%5]
test [sender = 192.168.0.26 : 5555], [bind_addr=/61.77.153.207]
test [sender = 192.168.0.26 : 5555], [bind_addr=/192.168.0.26]
```

```
test [sender = 192.168.0.26 : 5555], [bind_addr=/192.168.0.26]
test [sender = 192.168.0.26 : 5555], [bind_addr=/fe80:0:0:0:3e07:54ff:fe6f:95e%4]
test [sender = 192.168.0.26 : 5555], [bind_addr=/61.77.153.207]
test [sender = 192.168.0.26 : 5555], [bind_addr=/fe80:0:0:0:2a37:37ff:fe1e:fc4%5]
test [sender = 192.168.0.26 : 5555], [bind_addr=/61.77.153.207]
test [sender = 192.168.0.26 : 5555], [bind_addr=/61.77.153.207]
test [sender = 192.168.0.26 : 5555], [bind_addr=/fe80:0:0:0:2a37:37ff:fe1e:fc4%5]
test [sender = 192.168.0.26 : 5555], [bind_addr=/61.77.153.207]
test [sender = 192.168.0.26 : 5555], [bind_addr=/fe80:0:0:0:3e07:54ff:fe6f:95e%4]
test [sender = 192.168.0.26 : 5555], [bind_addr=/fe80:0:0:0:3e07:54ff:fe6f:95e%4]
test [sender = 192.168.0.26 : 5555], [bind_addr=/192.168.0.26]
test [sender = 192.168.0.26 : 5555], [bind_addr=/fe80:0:0:0:3e07:54ff:fe6f:95e%4]
test [sender = 192.168.0.26 : 5555], [bind_addr=/fe80:0:0:0:3e07:54ff:fe6f:95e%4]
test [sender = 192.168.0.26 : 5555], [bind_addr=/192.168.0.26]
test [sender = 192.168.0.26 : 5555], [bind_addr=/fe80:0:0:0:3e07:54ff:fe6f:95e%4]
test [sender = 192.168.0.26 : 5555], [bind_addr=/61.77.153.207]
test [sender = 192.168.0.26 : 5555], [bind_addr=/fe80:0:0:0:2a37:37ff:fe1e:fc4%5]
test [sender = 192.168.0.26 : 5555], [bind_addr=/192.168.0.26]
test [sender = 192.168.0.26 : 5555], [bind_addr=/fe80:0:0:0:2a37:37ff:fe1e:fc4%5]
test [sender = 192.168.0.26 : 5555], [bind_addr=/192.168.0.26]
test [sender = 192.168.0.26 : 5555], [bind_addr=/fe80:0:0:0:2a37:37ff:fe1e:fc4%5]
test [sender = 192.168.0.26 : 5555], [bind_addr=/192.168.0.26]
test [sender = 192.168.0.26 : 5555], [bind_addr=/fe80:0:0:0:2a37:37ff:fe1e:fc4%5]
test [sender = 192.168.0.26 : 5555], [bind_addr=/fe80:0:0:0:2a37:37ff:fe1e:fc4%5]
```

하나의 메시지가 다수의 네트워크 인터페이스를 통해서 송신되고, 또한 다수의 네트워크 인터페이스를 통해서 수신되기 때문에 하나 이상의 메시지가 수신될 수 있다. Sender쪽의 프롬프트에 quit 또는 exit를 입력하면 종료되며, Receiver는 <Ctrl> + C를 이용하여 종료해야 한다.

각각의 도움말을 보려면 인자로 -help를 입력하거나 인자를 주지 않으면 된다.

다음은 Virtual Multicast를 설정하는 방법이다. 속성의 기본값은 [기본 도메인 생성](#)과 동일하고, Virtual Multicast에는 MESH와 TREE의 두가지 type이 존재한다.

• Ant Task를 사용한 도메인 생성 (MESH 방식)

```
JEUS_HOME/setup$ ant create-domain
-Ddomain=jeus_domain
-Djeus.address=192.168.34.1
-Dtransport.type=MESH
```

• jeusadmin을 사용한 도메인 생성 (MESH 방식)

```
offline>create-domain -domain jeus_domain -baseaddress 192.168.34.1 -ttype MESH
Enter the password for [administrator]: *****
Confirm the password: *****
Do you want to encrypt the password? (y/n): y
=====
The domain [jeus_domain] was created successfully.
```

Property	Value
Domain name	jeus_domain
Master Name	adminServer
Master Base Listen Port	9736
Master Base Listen Address	192.168.34.1

HTTP Port	8088	
User Name	administrator	
Node Name	node1	
Password	{base}amV1cw==	
JVM Option	-Xmx256m -XX:MaxMetaspaceSize=128m	
Production Mode	true	
Transport Type	MESH	
+-----+-----+		
=====		

• jeusadmin을 사용한 도메인 생성 (TREE 방식)

```
offline>create-domain -domain jeus_domain -baseaddress 192.168.34.1 -ttype TREE -taddress
192.168.34.1 -tport 9736
Enter the password for [administrator]: *****
Confirm the password: *****
Do you want to encrypt the password? (y/n): y
=====
The domain [jeus_domain] was created successfully.
+-----+-----+
| Property | Value |
+-----+-----+
| Domain name | jeus_domain |
| Master Name | adminServer |
| Master Base Listen Port | 9736 |
| Master Base Listen Address | 192.168.34.1 |
| HTTP Port | 8088 |
| User Name | administrator |
| Node Name | node1 |
| Password | {base}amV1cw== |
| JVM Option | -Xmx256m -XX:MaxMetaspaceSize=128m |
| Production Mode | true |
| Transport Type | TREE |
| Transport address | 192.168.34.1 |
| Transport port | 9736 |
+-----+-----+
=====
```

2.2. 도메인 구성

본 절에서는 [도메인 생성](#)에서 생성한 도메인에 서버와 클러스터를 추가하여 도메인을 구성하는 방법에 대해 설명한다.

서버와 클러스터를 추가하는 것 또한 설정을 변경하는 것이기 때문에 추가하기 전에 다음과 같이 MASTER를 구동한다. 설정 변경에 대한 자세한 내용은 [도메인 설정변경](#)을 참고한다.

```
DOMAIN_HOME/bin$ startMasterServer -u jeus -p jeus
```


2.2.1. 콘솔 툴 사용

도메인에 서버와 클러스터를 추가할 때 다음과 같은 순서로 추가 및 설정한다.

1. [Managed Server\(MS\) 추가](#)
2. [클러스터 추가](#)

Managed Server(MS) 추가

MS를 추가할 때에는 동적으로 변경하지 못하는 설정들을 신중히 고려하여 입력해야 한다.

[도메인 생성](#)에서 설명한 도메인을 생성할 때 MASTER의 기본 설정과 마찬가지로 MS를 추가할 때의 기본 설정은 동적 반영이 안 되는 설정이다. 같은 머신에 있는 다른 서버들과 충돌되지 않는 Port 번호를 선정해야 하고, 외부에서 이 서버로 접속할 IP 주소와 서버의 JVM에게 전달할 옵션을 설정해야 한다. 서브넷을 벗어난 도메인의 경우라면 기본값 '0:0:0:0'을 사용하지 않고 하나의 IP 주소를 지정한다. 도메인 내에서 서버 이름은 유일한 값이어야 하므로 등록된 서버 이름을 확인한 후 충돌이 발생하지 않게 설정해야 한다. 본 예제에서 생성할 서버는 같은 클러스터를 구성할 서버들이므로 클러스터 내에서 동일해야 하는 설정들은 각 서버에서 별도로 설정하지 않아도 된다.

다음은 jeusadmin을 사용하여 MS를 추가하는 방법에 대한 설명이다.

add-server 명령어를 사용하여 'server1'이라는 이름의 MS를 추가한다. 동일한 방법으로 'server2'도 추가한다. add-server 명령어에 대한 자세한 사용법은 JEUS Reference 안내서의 "add-server"를 참고한다.

```
[MASTER]domain1.adminServer>add-server server1 -node node1 -addr 192.168.34.1 -port 9836
Successfully performed the ADD operation for server (server1).
NOTICE : base-addr [192.168.34.1] base-port [9836] http-port [8088]
Check the results using "list-servers or add-server"
```

클러스터 추가

클러스터를 추가할 때에는 클러스터 내의 모든 서버가 같은 서비스를 하기 위한 설정과 클러스터의 기능인 부하 분산(Load Balance)과 장애 극복(Failover)을 위한 설정을 해야한다. 클러스터 설정에 대한 자세한 설명은 [클러스터 설정](#)을 참고한다.

다음은 클러스터를 추가하는 과정에 대한 설명이다.

1. **add-cluster** 명령어를 사용하여 'cluster1'이라는 이름의 'server1'과 'server2'로 구성된 클러스터를 추가한다. add-cluster 명령어에 대한 자세한 사용법은 JEUS Reference 안내서의 [add-cluster](#)를 참고한다.

```
[MASTER]domain1.adminServer> add-cluster cluster1 -servers server1,server2
Successfully performed the ADD operation for cluster (cluster1).
Check the results using "list-clusters or add-cluster"
```

2. 클러스터의 추가까지 완료되면 다음과 같이 **serverinfo** 명령어를 사용하여 도메인의 구성 결과를 확인할 수 있다.

```
[MASTER]domain1.adminServer>serverinfo
```

Information about Domain (domain1)

Server	Status	Node Name	PID	Cluster	Latest Start Time / Shutdown Time	Need to Restart	Listen Ports	Running Engines
adminServer(*)	RUNNING (00:01:11)	N/A	5360	N/A	2016-08-23 (화) 오후 03:53:28 KST	false	base-192.168.34.3:9736 http-server-0.0.0.0:8088 jms-internal-0.0.0.0:9741	jms, ejb, web
server1	RUNNING (00:00:03)	node1	6424	cluster1	2016-08-23 (화) 오후 03:54:37 KST	false	base-192.168.34.1:9836	jms, ejb, web
server2	RUNNING (00:00:03)	node2	3012	cluster1	2016-08-23 (화) 오후 03:54:36 KST	false	base-192.168.34.2:9936	jms, ejb, web

2.3. 디렉터리 구성

기본적으로 JEUS를 설치하면 JEUS_HOME 디렉터리와 JEUS_HOME 디렉터리 하위에 jeus_domain이라는 도메인 디렉터리가 생성된다. 본 절에서는 user home의 path가 /home/user1이고 domain이 domain1이라고 설치했다고 가정했을 때의 구성에 대해 설명한다.

2.3.1. JEUS_HOME 디렉터리

JEUS를 설치했을 때의 전체 디렉터리 구조는 다음과 같다. JEUS_HOME은 JEUS의 최상위 디렉터리로 실제 디렉터리 이름과 위치는 설치할 때 설정할 수 있다.

```
/home/jeus/JEUS_HOME
```

JEUS_HOME의 디렉터리 구조는 다음과 같다.

```
{JEUS_HOME}
|--bin
|   |--[01]startMasterServer
|   |--[01]startManagedServer
|   |--[01]stopserver
|   |--[01]jeusadmin
|--derby
|--docs
|--domains
|   |--domain1
|--lib
```

```
|--license  
|--setup  
|--templates
```

*** Legend**

- [01]: binary or executable file
- [X] : XML document
- [J] : JAR file
- [T] : Text file
- [C] : Class file
- [V] : java source file
- [DD] : deployment descriptor

bin

서버의 시작, 종료 스크립트인 startMasterServer, startManagedServer, stopServer와 JEUS 콘솔 툴(jeusadmin)과 같은 실행 파일들이 위치한다.

derby

샘플 애플리케이션이나 테스트에서 쉽게 사용할 수 있도록 Apache Derby를 포함시킨다.

docs

JEUS에서 제공하는 API에 대한 Javadoc이 존재한다.

domains

하위에 도메인별로 DOMAIN_HOME이 존재한다.

lib

JEUS를 기동하는 데 필요한 라이브러리가 존재한다. shared 디렉터리를 제외한 나머지 디렉터리들은 사용자가 접근할 필요가 없다.

하위 디렉터리	설명
shared	<p>shared 디렉터리에는 애플리케이션에서 사용하는 라이브러리가 존재한다.</p> <p>shared 디렉터리에 있는 라이브러리를 사용하기 위해서는 libraries.xml에 라이브러리의 정보를 추가해야 한다. 그리고 해당 라이브러리를 사용할 애플리케이션의 JEUS Deployment Descriptor(DD)에서 해당 라이브러리에 대한 레퍼런스 정보를 지정해야 한다.</p> <p>shared 라이브러리에 대한 자세한 설명은 JEUS Applications & Deployment 안내서의 공유 라이브러리를 참고한다.</p>

license

JEUS 라이선스 파일이 위치한다. 라이선스 파일은 JEUS가 실행되기 위해서 반드시 필요한 파일이다.

setup

JEUS 설치 후 JEUS를 사용할 수 있도록 환경을 구축하기 위해 필요한 파일들이 위치한다.

templates

각종 설정 정보나 환경 정보 등의 template 파일이 위치한다.

2.3.2. 도메인 디렉터리

JEUS_HOME/domains/domain1이 domain1의 DOMAIN_HOME 디렉터리가 된다.

도메인 디렉터리의 구조는 다음과 같다.

```
domain1
|--.applications
|--.deploymentplans
|--.libraries
|--bin
|--config
|--lib
|   |--application
|--servers
```

.applications

해당 도메인에서 관리하는 애플리케이션 파일이 존재한다. JEUS가 사용하는 디렉터리로 사용자의 접근을 제한한다. install-application이나 uninstall-application 명령어를 통해서만 추가 및 삭제가 가능하다. 각 명령어에 대한 설명은 JEUS Reference 안내서의 [install-application](#), [uninstall-application](#)을 참고한다.

.deploymentplans

해당 도메인에서 관리하는 Deployment Plan이 존재한다. JEUS가 사용하는 디렉터리로 사용자의 접근을 제한한다. install-deployment-plan과 uninstall-deployment-plan 명령어를 통해 Deployment Plan의 추가와 삭제가 가능하다. 각 명령어에 대한 설명은 JEUS Reference 안내서의 [install-deployment-plan](#)과 [uninstall-deployment-plan](#)을 참고한다.

.libraries

해당 도메인에 설치한 라이브러리 파일이 존재한다. JEUS가 사용하는 디렉터리로 사용자의 접근을 제한한다. install-library 와 uninstall-library 명령어를 통해 라이브러리의 추가와 삭제가 가능하다. 각 명령어에 대한 설명은 JEUS Reference 안내서의 [install-library](#)와 [uninstall-library](#)를 참고한다.

bin

해당 도메인에 속한 MASTER와 MS의 시작 및 종료 스크립트가 위치한다.

JEUS_HOME/bin의 startMasterServer, startManagedServer, stopServer와 동일한 기능을 수행하나 도메인 이름을 설정할 필요가 없다.

config

도메인의 설정 파일인 domain.xml이 변경된 경우 이전 이력을 위해 존재하는 백업 파일들이 위치한다. 도메인 설정에 대한 자세한 설명은 [도메인 설정변경](#)을 참고한다.

- **security**

구분	설명
SYSTEM_DO MAIN	도메인 단위로 적용되는 보안 도메인 파일인 accounts.xml, policies.xml이 존재하며, 각 XML 파일은 jeusadmin을 통해 동적 설정 변경이 가능하다. 보안 도메인 설정에 대한 자세한 설명은 JEUS Security 안내서의 보안 도메인 설정 를 참고한다.
security.key	대칭 Key 암호화 알고리즘에 대한 Key를 저장하는 파일로 JEUS_HOME/bin/encryption을 수행하면 생성된다. security.key 파일에 대한 자세한 설명은 JEUS Security 안내서의 패스워드 보안 설정 을 참고한다.
policy	Java Permission 설정 파일이다. JEUS의 보안 시스템과는 별도로 JavaSE Security Manager에서 사용된다.

◦ servlet

구분	설명
webcommon.xml	도메인 내 서버 웹 엔진의 모든 웹 모듈에 적용되는 공통 설정 파일이다. 이 설정에 대한 자세한 설명은 JEUS Web Engine 안내서의 디렉터리 구조 를 참고한다.

lib/application

도메인 전체에 적용할 애플리케이션 라이브러리를 위치시키는 디렉터리다.

SERVER_HOME에 존재하는 애플리케이션 라이브러리와 충돌이 발생할 경우 SERVER_HOME/lib/application이 우선되고 경고 메시지가 남는다. lib/application 디렉터리에 대한 자세한 설명은 JEUS Applications & Deployment 안내서의 [lib/application 디렉터리](#)를 참고한다.

servers

디렉터리 하위에 SERVER_HOME 디렉터리가 서버 이름으로 생성된다. SERVER_HOME 디렉터리 구조에 대한 자세한 설명은 JEUS Server 안내서의 [서버 디렉터리 구조](#)를 참고한다.

3. 도메인 설정변경

본 장에서는 도메인 설정 관리에 관련한 배경 지식을 설명하고, 실제로 설정을 변경하는 방법에 대해 설명한다.

3.1. 개요

도메인에 대한 설정은 각 도메인의 config 디렉터리(DOMAIN_HOME/config)에 존재한다. 설정 파일인 domain.xml에는 도메인에 속한 서버 및 클러스터에 대한 설정과 리소스 및 서비스에 대한 설정, deploy된 애플리케이션의 정보 등이 포함된다.

동적 설정 반영은 서버 운영 중에 설정을 변경하면 서버를 재부팅하지 않아도 실시간으로 반영이 되는 기능을 의미한다. 변경된 설정은 MASTER가 검증하여 MS로 분배하기 때문에 사용자가 직접 XML을 편집하여 변경한 경우는 동적 반영이 되지 않고, 관리 도구를 사용하여 변경한 설정에 대해서만 동적 반영된다.

MASTER를 통한 동적 설정변경을 위해서는 리소스 이름 jeus.domain.<domain-name>, 리소스 액션 DynamicConfiguration 권한이 필요하다. 특히, 보안 설정변경의 경우 사용자는 자신의 동적 설정변경 권한을 삭제할 수 없다. 이는 사용자가 언제나 자신이 설정한 부분을 다시 원래대로 복구할 수 있도록 보장하기 위한 것이다.



JEUS 시스템에서 특정 사용자에게 특정 리소스 권한을 부여하는 방법은 JEUS Security 안내서의 [보안 시스템 정책 설정](#)을 참고한다.

3.2. 설정변경

서버 운영에 필요한 설정은 서버가 속한 도메인마다 하나의 파일로 관리됩니다.

도메인 설정 파일은 MASTER에서 관리 및 유지되고, 성능을 위해 서버가 존재하는 여러 머신에 분배되어 read-only 상태로 로컬 Cache에 존재한다. JEUS는 MASTER를 통해 동적 설정 반영을 지원한다.

3.3. 설정변경 적용

설정 내용이 변경되면 서버에 적용해야 하는데 설정은 동적 반영이 가능한 설정과 서버의 재시작이 필요한 설정으로 구분된다.

3.3.1. 동적 반영이 가능한 설정

동적 반영이 가능한 설정은 변경 후 서버의 재시작 없이 즉시 서버에 반영된다.

콘솔 툴 사용

콘솔 툴(jeusadmin)에서 동적 변경이 가능한 항목은 명령어 파라미터 설명에 **[Dynamic]**이 표시되어 있다.

jeusadmin의 각 명령어에서 **lock, save, activate**가 자동으로 수행되므로 **[Dynamic]** 항목에 설정값을 입력하여

해당 명령어를 수행하면 변경 사항이 즉시 반영된다.

```
[MASTER]domain1.adminServer>help modify-server
...
OPTIONS
[-removeLogdir,--removeLogHomeDirectory]
    remove the log directory which has all the log files created by this
    server

[-node,--nodeName <node-name>]
    name of the node this server is located

<server-name>
    the name of server you want to modify configuration of

[-jvm,--jvmOptions <jvm-options>]
    jvm configurations applied to this server jvm

[-logdir,--logHomeDirectory <server-log-home-directory>]
    the log directory which has all the log files created by this server

[-a,--actionOnResourceLeak <action-on-resource-leak>]
    strategy when resource leak is detected. it must be one of
    AutoClose, NoAction, Warning

[-l,--logStdoutToRawFormat <print-stdout-to-raw-format>]
[Dynamic] whether stdout is printed in row format or in JEUS Logger
    format.

[-m,--mejb <enable-MEJB>]
[Dynamic] whether using MEJB or not .true, false

[-c,--classFtp <enable-class-ftp-service>]
[Dynamic] whether using class ftp service or not .true, false

[-f,--forceLock]
    Acquires the configuration lock forcibly.

[-detail]
    show detail results
```



전체 설정에 대한 동적 변경 요소의 자세한 내용은 JEUS Reference 안내서의 [jeusadmin](#)을 참고한다.

3.3.2. 서버의 재시작이 필요한 설정

서버의 재시작이 필요한 설정과 실시간 동적 변경 반영이 가능한 설정이 함께 변경된 경우는 **activate** 명령어를 수행하면 후자만 실시간으로 반영된다. 서버의 재시작이 필요한 설정들은 반드시 해당 서버가 재시작되어야 반영된다. 본 절에서는 변경된 설정 중 반영되지 않은 설정 정보를 확인하는 방법에 대해서 설명한다.

콘솔 툴 사용

콘솔 툴(jeusadmin)에서 **server-info** 명령어를 수행하면 다음과 같이 서버 목록 및 상태 정보가 조회된다. 각 서버의 정보에서 '**Need to Restart**' 항목이 'true'로 출력된 경우에는 해당 서버의 재시작이 필요하다.

```
[MASTER]domain1.adminServer>server-info
```

Information about Domain (domain1)

Server	Status	Node Name	PID	Cluster	Latest Start Time / Shutdown Time	Need to Restart	Listen Ports	Running Engines
adminServer(*)	RUNNING (00:25:52)	N/A	5360	N/A	2016-08-23 (화) 오후 03:53:28 KST	false	base-192.168. .34.3:9736 http-server- 0.0.0.0:8088 jms-internal- 0.0.0.0:9741	jms, ejb, web
server1	RUNNING (00:16:15)	node1	6424	N/A	2016-08-23 (화) 오후 04:03:05 KST	true	base-192.168. .34.1:9836	jms, ejb, web
server2	RUNNING (00:16:15)	node2	3012	N/A	2016-08-23 (화) 오후 04:03:05 KST	true	base-192.168. .34.2:9936	jms, ejb, web

3.4. 설정 로테이션

설정을 변경하면 기존 설정이 백업되는데 이것을 **설정 로테이션**이라고 한다.

현재 운영 중인 설정은 항상 다음의 경로에 있는 파일들이다.

```
DOMAIN_HOME/config/domain.xml  
DOMAIN_HOME/config/security/SECURITY_DOMAIN_NAME/policies.xml  
DOMAIN_HOME/config/security/SECURITY_DOMAIN_NAME/accounts.xml
```

동적 변경이 됐을 경우 또는 서버의 재시작이 필요한 경우 모두 포함하여 domain.xml에 대한 변경 사항이 있으면 이전 domain.xml에 대한 백업 파일이 DOMAIN_HOME/config/.history 폴더 안에 'domain_yyyymmdd_hhmmss.xml'의 형식으로 저장된다. jeus.server.configfile.backup.size 프로퍼티를 통해 백업 파일 개수를 지정할 수 있다. (기본값: 50)

3.5. 도메인 공통 설정변경

다음은 도메인 공통 설정에 대한 예시이다.


```

<domain>
  ...
  <production-mode>true</production-mode>
  <id>427655071</id>
  <master-server-name>adminServer</master-server-name>
  <domain-log-home>${JEUS_HOME}/domain1/logs</domain-log-home>
  <enable-json-command>false</enable-json-command>
  <enable-to-resynchronize-applications>false</enable-to-resynchronize-applications>
  <application-repositories>
    ...
  </application-repositories>
  <system-clustering-framework>
    <transport-type>HYBRID</transport-type>
    <transport-address>224.0.0.1</transport-address>
    <transport-port>12488</transport-port>
    <leader-discovery-timeout>3000</leader-discovery-timeout>
    <failure-detection-timeout>3000</failure-detection-timeout>
    <max-thread-pool-size>10</max-thread-pool-size>
    <min-thread-pool-size>0</min-thread-pool-size>
  </system-clustering-framework>
  <domain-backup>
    <backup-on-boot>false</backup-on-boot>
  </domain-backup>
  ...
</domain>

```

다음은 설정 항목에 대한 설명이다.

항목	설명
Jeus Master Server Name	MASTER의 ID에 해당하는 값으로 도메인 구성한 후에는 변경할 수 없다.
Id	도메인의 ID에 해당하는 값으로 도메인 구성한 후에는 변경할 수 없다.
Domain Log Home	도메인 내의 서버들이 공통으로 사용할 log의 폴더명으로 도메인 구성한 후에는 변경할 수 없다.
Production Mode	실제 운영환경에 적합한 모드로, 웹 애플리케이션의 auto-reload나 hot-swap과 같은 기능은 제공하지 않는다.
Enable Json Command	Json Command 사용 여부를 선택하는 항목으로 기본적으로 꺼져있다. 선택적으로 비활성화할 수 있고 필요한 경우 다시 체크하여 활성화할 수 있다.
Enable To Resynchronize Applications	MS가 INDEPENDENT 모드에서 DEPENDENT 모드로 변경된 경우 MS에서 MASTER와 애플리케이션을 동기화할 것인지에 대한 설정이다. (기본값: false, 동기화하지 않음)
Application Repositories	서비스할 애플리케이션의 위치를 설정하는 항목으로 동적으로 추가 및 삭제할 수 있다. 이에 대한 자세한 설명은 JEUS Applications & Deployment 안내서의 애플리케이션 저장소 추가/삭제/조회 를 참고한다.

항목	설명
System Clustering Framework	<p>서버들 간의 내부 통신 및 상태 확인에 사용되는 네트워크 설정으로써 도메인 전체에 걸쳐서 적용되므로 변경할 경우에는 도메인 전체를 다시 시작해야 한다.</p> <p>JEUS에서는 서버 간에 서로의 상태를 공유하기 위해 기본적으로 Multicast를 사용한다. 자세한 내용은 도메인 생성을 참고한다.</p> <ul style="list-style-type: none"> ◦ transport : 타 서버와 통신하는 방식을 정하는 항목이다. DUMMY, HYBRID, MESH, TREE를 선택할 수 있다. <ul style="list-style-type: none"> ◦ DUMMY : 타 서버와 일체의 통신을 수행하지 않는 방식이다. ◦ HYBRID : IP Multicast를 사용하는 방식이다. ◦ MESH, TREE : 도메인의 구성이 서브넷을 벗어나있거나 IP Multicast가 허용되지 않는 환경에서 TCP/IP를 이용하여 Multicast를 흉내내는 방식이다. MESH는 전체 서버가 Full-mesh 형태로 연결되는 방식이며, TREE는 Master서버를 root로 하고 Managed Server들을 leaf로 하는 tree의 형태로 연결되는 방식을 의미한다. ◦ transport-address, transport-port : 선택한 transport의 종류에 따라서 그 동작에 필요한 주소 및 포트값을 설정하는 항목이다. <p>아래의 항목들은 대부분의 경우에 특별히 설정할 필요는 없으나, 도메인의 규모에 따라서 적절한 값으로 조절해야 한다.</p> <ul style="list-style-type: none"> ◦ failure-detection-timeout : 서버를 장애로 판단하는데 소요하는 시간을 설정하는 항목이다. 짧게 설정하면 장애가 민감하게 감지되나 오감지의 위험성이 생긴다. HYBRID 모드일 경우에는 IP Multicast를 사용하기 때문에 설정된 것보다 장애가 감지되기까지 더 긴 시간이 소요될 수 있다. ◦ leader-discovery-timeout : 최초로 기동하는 서버가 이미 기동되어있는 서버 그룹을 찾는데 소요하는 시간을 설정하는 항목이다. ◦ max-thread-pool-size : SCF가 사용하는 Thread pool의 최대 크기를 설정하는 항목이다. ◦ min-thread-pool-size : SCF가 사용하는 Thread pool의 최소 크기를 설정하는 항목이다.
Domain Backup	<p>JEUS Master Server를 기동할 때 백업 파일을 만들지에 대한 설정이다.</p> <p>백업 파일의 생성 여부와 디렉터리를 설정할 수 있다. 기본적으로 도메인 백업을 생성하지 않도록 설정은 꺼져있다. 해당 기능에 대한 자세한 설명은 도메인 백업을 이용한 장애 극복을 참고한다.</p>



'Enable Json Command', 'Enable To Resynchronize Application', 'Application Repositories', 'Domain Backup'을 제외한 도메인 설정 항목은 동적 변경이 불가능하기 때문에 설정을 반영하기 위해서는 도메인 전체를 재시작해야 한다. 도메인 설정은 개별 서버의 설정과는 달리 도메인 구성에 관한 설정이다. 따라서 도메인을 구성할 때 **'Jeus Master Server Name', 'Id', 'Domain Log Home', 'Production Mode', 'System Clustering Framework'** 영역의 항목들은 신중하게 설정해야 한다.

3.6. 설정변경 예제

본 절에서는 콘솔 툴(jeusadmin)을 사용하여 System Thread Pool의 개수와 서버의 JVM 설정을 변경하는 방법에 대해 설명한다.

3.6.1. System Thread Pool 개수 변경

본 절에서는 콘솔 툴(jeusadmin)을 사용하여 System Thread Pool 개수를 변경하는 방법에 대해 설명한다.

콘솔 툴 사용

콘솔 툴(jeusadmin)의 **modify-system-thread-pool** 명령어를 실행하면 System Thread Pool 개수를 변경할 수 있다. modify-system-thread-pool 명령어에 대한 자세한 내용은 JEUS Reference 안내서의 "modify-system-thread-pool"을 참고한다.

```
[MASTER]domain1.adminServer>modify-system-thread-pool server1 -max 120
Successfully performed the MODIFY operation for the system thread pool of the server (server1), but
all changes were non-dynamic. They will be applied after restarting.
Check the results using "modify-system-thread-pool server1 or show-system-thread-pool server1".
```

3.6.2. 서버의 JVM 설정변경

본 절에서는 콘솔 툴(jeusadmin)을 사용하여 서버의 JVM 설정을 변경하는 방법에 대해 설명한다.

콘솔 툴 사용

콘솔 툴(jeusadmin)을 사용하여 서버의 JVM 설정을 추가하려면 다음과 같이 **add-jvm-option** 명령어를 실행한다. add-jvm-option 명령어는 사용자가 입력한 옵션을 jvm-option list와 비교하여 같은 것이 없으면 추가한다.

```
[MASTER]domain1.adminServer>add-jvm-option -server server1 -opt "-Xmx256m -XX:MaxPermSize=128m"
Successfully performed the ADD operation for JVM configuration for the server(server1)..
Check the results using "list-jvm-options or add-jvm-option".
```

콘솔 툴(jeusadmin)을 사용하여 서버의 JVM 설정을 변경하려면 다음과 같이 **modify-jvm-option** 명령어를 실행한다. modify-jvm-option 명령어는 사용자가 입력한 옵션을 jvm-option list와 비교하여 같은 것을 수정한다.

```
[MASTER]domain1.adminServer>modify-jvm-option -server server1 -old "-Xmx256m -XX:MaxPermSize=128m"
-new "-Xmx512m -XX:MaxPermSize=128m"
Successfully performed the MODIFY operation for JVM configuration for the server(server1)..
Check the results using "list-jvm-options or modify-jvm-option".
```

콘솔 툴(jeusadmin)을 사용하여 서버의 JVM 설정을 삭제하려면 다음과 같이 **remove-jvm-option** 명령어를 실행한다. remove-jvm-option 명령어는 사용자가 입력한 option을 jvm-option list와 비교하여 같은 것을

삭제한다.

```
[MASTER]domain1.adminServer>remove-jvm-option -server server1 -opt "-Xmx256m -XX:MaxPermSize=128m"
Successfully performed the REMOVE operation for JVM configuration for the server(server1)..
Check the results using "list-jvm-options or remove-jvm-option".
```



JVM 설정을 추가하는 경우에는 한번에 하나의 옵션을 추가해서 jvm-option 항목 하나에 하나의 옵션만 추가할 수 있도록 하고, 변경 및 삭제할 경우에는 **list-jvm-options** 명령어를 통해 설정된 옵션을 주어 수정하는 것을 권장한다.

4. 서버 Life Cycle

본 장에서는 서버의 시작과 종료, 서버 Life Cycle을 확인하는 방법에 대해 설명한다.

4.1. 서버 시작 준비

서버를 시작하기 전에 준비되어야 하는 몇 가지 사항은 다음과 같다.

- JEUS가 설치되고, 도메인 및 서버가 생성되어 있어야 한다. 또한 서버를 시작 및 종료할 수 있는 사용자명과 패스워드가 생성되어 있어야 한다. 이러한 사전 작업이 되어 있지 않다면 [JEUS 설치 및 시작하기 안내서](#)를 참고하여 환경을 구축한다.
- 서버에 적용하려는 JVM 설정이 XML에 설정되었는지 확인한다.

도메인을 생성할 때 설정하지 않았다면, JVM 설정 변경은 재시작이 필요한 설정이기 때문에 MASTER를 시작하여 설정을 변경한 후 MASTER를 재시작하여 반영한다.

JVM 설정이 누락된 MS는 XML 편집하거나 콘솔 툴을 사용해서 JVM 설정을 추가한 후에 서버를 시작한다. 변경 방법은 [서버의 JVM 설정변경](#)을 참고한다.

- 적합한 라이선스를 가지고 있는지 확인한다.

4.2. 서버 시작

본 절에서는 Master Server(MASTER), Managed Server(MS)를 시작하는 방법에 대해 설명한다.

4.2.1. Master Server(MASTER) 시작

MASTER를 시작하는 스크립트는 **startMasterServer**로 MASTER의 도메인 이름, JEUS를 시작할 수 있는 권한을 가진 사용자명과 패스워드가 필요하다.

JEUS_HOME/bin 하위에서 실행할 경우에는 이 모든 값이 필요하지만, MASTER의 DOMAIN_HOME 하위에서 실행할 때에는 도메인 이름을 생략할 수 있다.

위치에 따라 다음과 같이 실행한다.

- JEUS_HOME/bin

```
JEUS_HOME/bin$ startMasterServer -domain <domain_name> -u <user_name> -p <password>
```

- DOMAIN_HOME/bin

```
DOMAIN_HOME/bin$ startMasterServer -u <user_name> -p <password>
```

- SERVER_HOME/bin

```
SERVER_HOME/bin$ startMasterServer -u <user_name> -p <password>
```

기본적으로는 MASTER가 RUNNING 상태가 된 이후에는 파일로만 서버의 로그(SERVER_HOME/logs/JeusServer.log)가 남게 된다. 서버의 로그를 콘솔 툴 화면에서 모니터링하려면 추가적으로 -verbose 옵션을 설정한다. 이 경우 서버 프로세스 외에도 Launcher 프로세스가 함께 운영된다. Launcher 프로세스는 서버 프로세스를 운영하고 서버의 로그를 콘솔 툴에 출력한다.

다음과 같이 **jps** 명령어로 확인하면 LauncherBootStrapper와 MasterServerBootstrapper를 확인할 수 있다.

```
jps -l
399092 jeus.server.LauncherBootstrapper
399352 jeus.server.admin.MasterServerBootstrapper
409880 sun.tools.jps.Jps
```

서버 시작에 실패한 경우에는 SERVER_HOME/logs/JeusLauncher.log에서 실패 이유를 확인할 수 있다.



1. jps는 Java 프로세스를 조회하는 툴이며, JAVA_HOME/bin에 위치한다.
2. Launcher에 대한 자세한 설명은 JEUS Server 안내서의 [Launcher](#)를 참고한다.

4.2.2. Managed Server(MS) 시작

MS가 위치하는 머신에 설치된 스크립트를 이용하여 MS를 시작할 수 있다. 도메인 내에 여러 서버가 여러 머신에 분포한다면 각 머신에 접속하여 MS가 위치하는 머신에서 스크립트를 수행해야 한다. 스크립트 이름은 **startManagedServer**이고, MASTER와 설정을 동기화하기 위해 **masterurl** 옵션을 추가로 설정해야 한다.

스크립트로 MS 시작

MASTER를 통하지 않고 MS가 위치하는 머신에 설치된 스크립트를 이용하여 MS를 시작할 수 있다.

도메인 내에 여러 서버가 여러 머신에 분포한다면 각 머신에 접속하여 MS가 위치하는 머신에서 스크립트를 수행해야 한다. 스크립트 이름은 **startManagedServer**이고, MASTER와 설정을 동기화하기 위해 **masterurl** 옵션을 추가로 설정해야 한다.



masterurl을 설정하지 않으면 INDEPENDENT 모드로 MS가 시작되고, 시작 후 MASTER가 MS를 찾으면 다시 MASTER의 관리를 받게 된다. MASTER가 시작하지 않은 경우 INDEPENDENT 모드로 MS를 시작하기 위해 masterurl을 생략할 수 있으나, 그 외 경우는 명시하기를 권장한다. INDEPENDENT 모드에 대한 설명은 [INDEPENDENT 모드의 Managed Server\(MS\)](#)를 참고한다.

startManagedServer 스크립트는 위치에 따라 다음과 같이 실행한다.

- JEUS_HOME/bin

```
JEUS_HOME/bin$ startManagedServer -masterurl <master_ip:master_baseport>  
-domain <domain_name> -server <server_name> -u <user_name> -p <password>
```

- DOMAIN_HOME/bin

```
DOMAIN_HOME/bin$ startManagedServer -masterurl <master_ip:master_baseport>  
-server <server_name> -u <user_name> -p <password>
```

- SERVER_HOME/bin

```
SERVER_HOME/bin$ startManagedServer -masterurl <master_ip:master_baseport> -u <user_name> -p  
<password>
```

4.3. 서버 종료

본 절에서는 Master Server(MASTER)와 Managed Server(MS)를 종료하는 방법에 대해 설명한다.

4.3.1. Master Server(MASTER) 종료

MASTER의 종료는 콘솔 툴(jeusadmin)이나 스크립트를 사용한다.

콘솔 툴로 MASTER 종료

MS를 모두 종료한 후 MASTER를 종료할 수 있다([Managed Server\(MS\) 종료](#) 참고). local-shutdown 명령어를 다음과 같이 실행한다.

```
[MASTER]domain1.adminServer>local-shutdown
```

스크립트로 MASTER 종료

MASTER가 위치하는 머신에 설치된 스크립트를 이용하여 종료할 수 있다. **stopServer** 스크립트는 위치에 따라 다음과 같이 실행한다.

- JEUS_HOME/bin

```
JEUS_HOME/bin$ stopServer -host <server_ip:server_baseport> -u <user_name> -p <password>
```

- DOMAIN_HOME/bin

```
DOMAIN_HOME/bin$ stopServer -host <server_ip:server_baseport> -u <user_name> -p <password>
```

- SERVER_HOME/bin

```
SERVER_HOME/bin$ stopServer -host <server_ip:server_baseport> -u <user_name> -p <password>
```

4.3.2. Managed Server(MS) 종료

MASTER를 통하는 방법과 스크립트를 사용하는 방법이 있다.

MASTER를 통해 MS 종료

MASTER를 통해 MS를 종료할 경우에는 콘솔 툴(jeusadmin)을 사용한다.

콘솔 툴(jeusadmin)을 사용하여 다음과 같이 **stop-server** 명령어를 통해 MASTER가 운영 중인 경우에 MS를 종료할 수 있다. stop-server 명령어에 대한 자세한 사용법은 JEUS Reference 안내서의 [stop-server](#)를 참고한다.

```
[MASTER]domain1.adminServer>stop-server <server_list>
```

스크립트로 MS 종료

MS가 위치한 머신에 설치된 스크립트를 이용하여 MS를 종료할 수 있다. 즉, 도메인 내에 여러 서버가 여러 머신에 분포한다면 각 머신에 접속하여 MS가 위치하는 머신에서 스크립트를 수행해야 한다.

stopServer 스크립트는 위치에 따라 다음과 같이 실행한다.

- JEUS_HOME/bin

```
JEUS_HOME/bin$ stopServer -host <server_ip:server_baseport> -u <user_name> -p <password>
```

- DOMAIN_HOME/bin

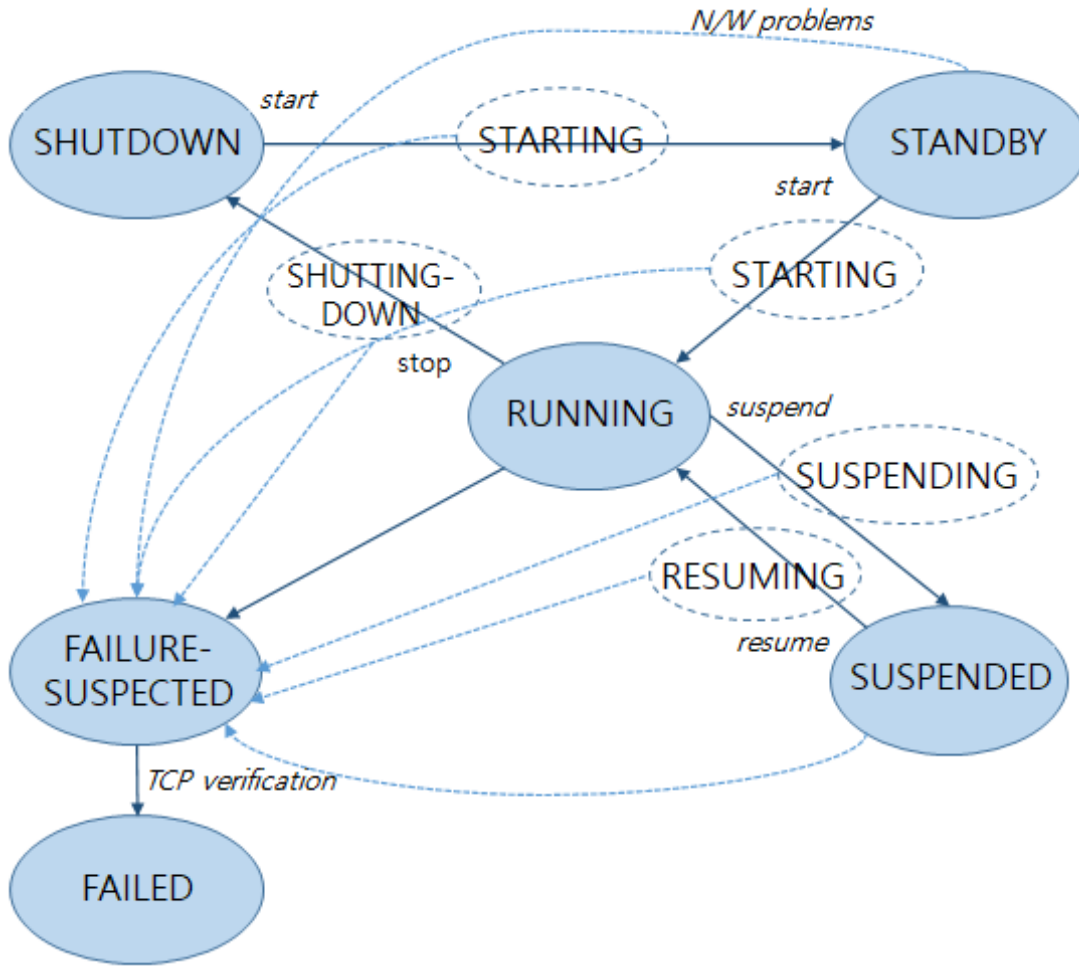
```
DOMAIN_HOME/bin$ stopServer -host <server_ip:server_baseport> -u <user_name> -p <password>
```

- SERVER_HOME/bin

```
SERVER_HOME/bin$ stopServer -host <server_ip:server_baseport> -u <user_name> -p <password>
```


4.4. 서버 Life Cycle 상태 확인

MS의 Life Cycle은 MASTER의 판단에 의해 여러 가지 상태로 구분된다.



MASTER의 판단에 의한 서버의 Life Cycle

각 상태에 대한 설명은 다음과 같다.

상태	설명
SHUTDOWN	시작되지 않은 상태 또는 정상 종료된 상태를 나타낸다.
STARTING	서버를 시작하는 중으로 아직 서비스는 할 수 없는 상태를 나타낸다.
STANDBY	시작 중에 등록된 애플리케이션을 Distribute하던 중 실패한 것이 있어 RUNNING 상태로 이동하지 못하고 머무르는 상태를 나타낸다. 이 상태에서 문제의 모듈을 수정하여 다시 시작을 하거나, 그 모듈을 무시하고 서비스를 하기 위해서는 force 옵션을 주고 start 명령어를 수행한다.
RUNNING	서버 시작이 완료되어 서비스를 할 수 있는 상태를 나타낸다.
SHUTTING_DOWN	서버가 종료 중인 상태를 나타낸다.
SUSPENDING	RUNNING 상태에서 모든 deploy된 애플리케이션의 서비스를 중단하기 위해 Suspend를 수행 중인 상태를 나타낸다.

상태	설명
RESUMING	SUSPENDED 상태에서 다시 모든 정지된(Stopped) 애플리케이션의 서비스를 재가하기 위해 Resume을 수행 중인 상태를 나타낸다.
SUSPENDED	관리자가 의도적으로 모든 애플리케이션을 정지하도록 Suspend를 수행한 상태를 나타낸다. 이 상태에서는 새롭게 애플리케이션을 추가하고 싶은 경우에도 해당 애플리케이션을 Distribute만 가능하고 시작할 수는 상태이다.
FAILURE_SUSPECTED	MASTER가 판단하기로 MS와 연결되지 않아 해당 MS의 비정상 종료를 의심하는 상태를 나타낸다.
FAILED	MASTER가 판단하기로 MS와 일정 시간동안 계속 연결되지 않아 해당 MS의 비정상 종료를 확신한 상태를 나타낸다.



MASTER가 MS의 상태에 대해서 판단할 때는 SCF(System Clustering Framework)의 기능을 이용한다.

SCF는 도메인을 구성하는 여러 서버들간에 서로의 상태를 모니터링하거나 서로간의 메시지를 담당하는 모듈로써 네트워크를 통해서 원격에 존재하는 다른 서버의 상태를 파악하도록 디자인되어 있다. 네트워크를 통하기 때문에 원격 서버의 부하등의 원인으로 통신이 지연되거나 할 경우 장애로 오감지 될 수 있으며 이 경우 실제 장애가 아니기 때문에 원인이 해소되면 다시 정상 상태로 인식된다.

SCF는 기본적으로 Multicast를 사용하지만, 이를 지원하지 않는 환경이거나 다른 필요나 요구사항이 있을 경우 Virtual Multicast 모드를 사용하도록 설정할 수도 있다. Virtual Multicast는 Multicast를 대신하여 도메인을 구성하는 각 서버들간에 Socket 연결을 맺어서 통신하는 방식이다.

다음의 3가지 방법으로 서버의 Life Cycle 상태를 확인할 수 있다.

• MBean 사용

J2EEDomain#getServerState(String serverName)로도 서버의 상태를 모니터링할 수 있다. MBean의 사용 방법에 대한 자세한 내용은 [JEUS JMX 안내서](#)를 참고한다.

• 콘솔 툴(jeusadmin) 사용

다음과 같이 콘솔 툴에서 **serverinfo** 명령어를 사용하여 서버의 Life Cycle 상태를 확인할 수 있다.

```
[MASTER]domain1.adminServer>serverinfo
Information about Domain (domain1)
=====
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Server | Status | Node | PID | Clu | Latest | Need | Listen | Running |
|         |        | Name |     | ster | Start Time | to | Ports | Engines |
|         |        |      |     |     | / Shutdown | Restart |      |         |
|         |        |      |     |     | Time      |       |      |         |
+-----+-----+-----+-----+-----+-----+-----+-----+
| adminS | RUNNIN | nod | 902 | N/A | 2022-07-19 | false | base-0.0. | jms, |
| rver   | G(00:42 | e1 | 88 |   | (화) 오후 |   | 0.0:9736 | web, ejb |
| (*)   | :15)   |    |    |    | 12:55:20 KST |   | http-serv |    |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

							er-0.0.0.0		
							:8088		
+-----+									
server1	RUNNIN	N/A	928	clu	2022-07-19	false	base-0.0.	jms,	
	G(00:00		05	ster1	(화) 오후		0.0:9836	web, ejb	
	:08)				01:37:27 KST		http-serv		
							er-0.0.0.0		
							:8188		
+-----+									
server2	SHUTDO	N/A	N/A	clu	2022-07-19	N/A	N/A	N/A	
	WN			ster1	(화) 오후				
					12:55:20 KST				
+-----+									
server3	SHUTDO	nod	N/A	N/A	2022-07-19	N/A	N/A	N/A	
	WN	e1			(화) 오후				
					12:55:20 KST				
+-----+									
=====									
=====									



serverinfo 명령어에 대한 자세한 내용은 JEUS Reference 안내서의 "server-info"를 참고한다.

5. JEUS 클러스터링

본 장에서는 도메인 내의 클러스터에 대해 정의하고 클러스터의 구성과 도메인과의 관계에 대해 설명한다. 또한 클러스터를 생성, 설정, 변경 및 삭제하는 방법에 대해 설명한다.

5.1. 개요

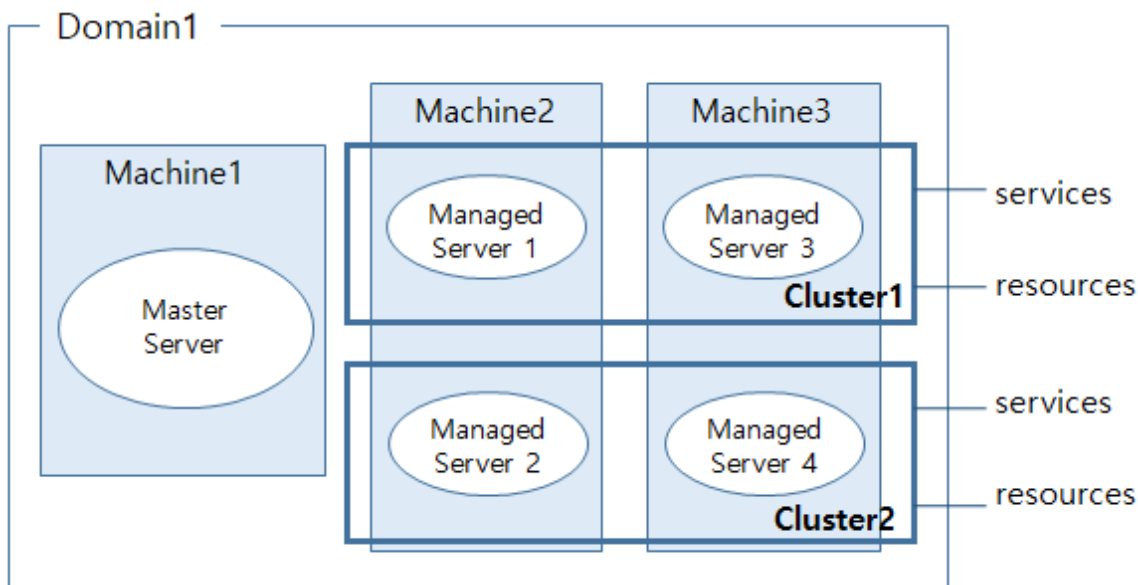
클러스터는 서비스의 확장성을 위한 부하 분산(Load Balancing)과 안정성을 위한 장애 극복(Failover) 기능을 제공하기 위해 동일한 서비스를 수행하는 여러 개 서버들의 집합이다.

같은 서비스를 수행하는 여러 개의 서버가 부하를 나눠서 처리하여 빠른 서비스가 가능하며, 부하의 정도에 따라 클러스터에 동적으로 서버를 추가 또는 삭제할 수 있어 부하 조절이 가능하다. 또한 몇 대의 서버가 서비스를 수행하지 못하는 상황에서도 같은 서비스를 하는 나머지 서버들이 서비스를 처리해주므로 사용자에게 서비스 중단이 노출되지 않는다. 즉, 많은 부하를 무리 없이 소화하고 서비스 불가능 상태가 되지 않는 하나의 서버라고 할 수 있다.

5.2. 클러스터와 도메인 관계

도메인은 관련된 서비스를 수행하는 서버 또는 클러스터들의 집합이다. 클러스터는 같은 서비스를 수행하는 여러 개 서버의 집합으로 하나의 도메인에 여러 개 존재할 수 있다. 클러스터를 구성하는 서버들은 한 머신에 모두 존재할 수도 있고, 각각 다른 머신에 존재할 수도 있다. 일반적으로 머신에 문제가 발생하여 서비스가 안될 경우를 고려하여 다른 머신에 위치한 서버들로 클러스터를 구성한다.

하나의 마스터 서버는 단 하나의 도메인을 관리하므로, 도메인 내에 여러 개의 클러스터가 있다면 각 클러스터도 동일한 마스터 서버의 관리를 받는다.



클러스터와 도메인과의 관계

5.3. 클러스터 기능

클러스터는 도메인 내에서 같은 서비스를 수행하는 MS들로 구성된다. 보통 여러 머신에 분산되어 있다. 이렇게 구성된 클러스터는 다음과 같은 기능을 갖는다.

- **부하 분산(Load Balancing)**

과부하로 인한 서비스 중단을 방지하고 응답 속도를 향상하기 위해 여러 서버 간에 수행해야 할 서비스를 분배하는 기능이다.

동일한 서비스를 수행할 수 있는 환경을 구성해야 하고, 서로의 위치와 상태를 알고 있어야 한다.

- **장애 극복(Failover)**

하나의 서버가 비정상 종료된 경우에 그 서버가 수행하고 있던 서비스를 다른 서버가 대신 수행하는 것을 의미한다.

비정상 종료한 서버 대신 서비스를 수행하는 서버는 비정상 종료된 서버와 같은 서비스를 수행할 수 있어야 하고, 비정상 종료된 서버를 감지할 수 있어야 한다. 또한, 비정상 종료된 서버의 서비스 진행 상태를 알아야 한다.

JEUS에서는 이러한 기능을 위해 클러스터 내의 모든 서버에 같은 애플리케이션을 deploy하고, 같은 리소스를 사용하고, 같은 서비스를 등록할 것을 권장한다. Multicast를 사용하여 클러스터 내의 서버들 간에 서로의 위치와 상태를 공유하고, 클러스터 가능한 서비스들의 장애 극복을 위해 서비스의 진행 상태 정보를 서버들 간에 공유한다. 또한 리소스의 효율을 높일 수 있도록 부하에 따라 클러스터에 서버의 추가 및 삭제가 가능하다.

모든 서비스에 대해 클러스터링이 가능한 것은 아니다. 클러스터링이 가능한 서비스는 다음과 같다.

서비스	설명
Servlets /Jsp	웹 엔진은 웹 서버에 의한 웹 애플리케이션 클러스터링과 HTTP 세션 클러스터링을 지원한다. 웹 서버에 의한 웹 애플리케이션 클러스터링에 대한 자세한 내용은 JEUS Web Engine 안내서의 부하 분산을 위한 웹 서버 설정 을 참고하고, HTTP 세션 클러스터링에 대한 자세한 내용은 JEUS 세션 관리 안내서의 분산 세션 서버 를 참고한다.
EJB	EJB 엔진에서는 세션, Message Driven, Entity Bean과 Timer Service에서 클러스터링을 지원한다. EJB 엔진에서 제공하는 클러스터링에 대한 자세한 내용은 JEUS EJB 안내서의 EJB 클러스터링 을 참고한다.
JMS	JMS에서는 Connection Factory와 Destination, Durable Subscriber의 클러스터링을 지원한다. JMS에서 제공하는 클러스터링에 대한 자세한 내용은 JEUS MQ 안내서의 JEUS MQ 클러스터링 을 참고한다.

클러스터가 가능한 서비스들에 한해서 클러스터를 구성할 때에는 다음의 제약사항을 고려해야 한다.

- 클러스터로 구성된 모든 서버는 같은 버전의 JEUS를 사용해야 한다.
- 각 서버는 단 하나의 클러스터에만 포함되어야 한다.
- 도메인 사이에는 어떤 리소스도 공유가 불가능하기 때문에 클러스터로 구성된 모든 서버는 같은 도메인에 포함되어야 한다.

5.4. 클러스터 생성

일반적으로 도메인을 구성할 때 서비스의 종류와 중요도 등을 고려하여 클러스터를 생성한다. 클러스터 생성 방법은 [도메인 구성](#)의 "[클러스터 추가](#)" 부분을 참고한다.

5.5. 클러스터 설정

클러스터 설정은 크게 서버 공통 설정과 클러스터 설정으로 구분된다.

5.5.1. 서버 공통 설정

서버 공통 설정은 클러스터 내의 모든 서버가 동일한 서비스를 수행하기 위한 설정으로 클러스터뿐만 아니라 단독 서버를 생성할 경우에도 필요한 설정이다.



서버 공통 설정에 대한 자세한 내용은 JEUS Server 안내서의 [JEUS 설정](#)을 참고한다.

5.5.2. 클러스터 설정

클러스터 설정은 클러스터의 기능인 부하 분산(Load Balance)과 장애 극복(Failover)을 위한 설정으로 클러스터 통신, 세션 서버, Timer Service, JMS 리소스에 대해 설정하고, 데이터소스를 등록 및 제거할 수 있다.



클러스터 설정 중 서버의 재시작이 필요한 항목을 변경한 경우에는 클러스터 전체를 재시작하는 것을 권장한다.

• 클러스터 세션 서버 설정

클러스터링 환경을 전제로 운영되는 분산식 세션 서버는 부하 분산과 장애 극복의 2가지 기능을 모두 가지고 있다.

내부적으로 분산식 세션 서버를 사용할지에 대한 설정은 별도로 존재하지 않으며 클러스터링에 참여하면, 클러스터링에 참여하는 서버들 간에 자동적으로 분산식 세션 서버가 운영된다.

분산식 세션 서버에 대한 자세한 설명은 JEUS 세션 관리 안내서의 [분산 세션 서버](#)를 참고하고, 세부 설정 및 각 항목들에 대한 설명은 해당 안내서의 [분산 세션 서버 설정](#)을 참고한다.

• 클러스터 Timer Service 설정

클러스터 환경에서 Timer Service를 설정하고 싶은 경우에 사용한다. Timer Service 설정에 대한 자세한 내용은 JEUS EJB 안내서의 [EJB Timer Service](#)를 참고한다.

• 클러스터 JMS 리소스 설정

클러스터 내의 Destination과 Durable Subscriber를 설정하는 경우에 사용한다. JMS 리소스 설정에 대한

자세한 내용은 JEUS MQ 안내서의 [서버 설정](#)을 참고한다.

• 데이터소스 등록 및 제거

클러스터에 등록된 데이터소스는 클러스터에 속한 모든 서버에서 사용될 수 있으며, 데이터소스 등록은 동적으로 처리가 가능하다. 또한, 클러스터에 등록된 데이터소스의 제거 역시 동적으로 처리가 가능하다. 클러스터로부터 제거된 데이터소스는 클러스터에 속한 서버들에서 더 이상 사용이 불가능하다.

클러스터에 등록된 데이터소스의 사용과 클러스터에 데이터소스를 등록하고 제거하는 방법 등에 대한 자세한 내용은 JEUS Server 안내서의 [데이터소스 관련 설정 동적 변경](#)을 참고한다.

5.6. 클러스터 변경

부하의 변화에 따라 클러스터에 존재하는 서버를 추가, 삭제해서 개수를 조정할 수 있다.

5.6.1. 클러스터에 서버 추가

해당 서비스의 부하가 증가하여 클러스터 사이즈를 늘리기 위해 클러스터에 서버를 추가할 수 있다. 추가할 서버는 기존의 서버와 동일한 서비스를 수행할 것이므로 기존 서버들의 설정과 동일하게 설정하는 것을 권장한다.

본 절에서는 클러스터에 서버를 추가하는 방법에 대해 설명한다.

콘솔 툴 사용

콘솔 툴(jeusadmin)을 사용하여 클러스터에 서버를 추가할 수 있다.

다음은 서버를 직접 생성, 선택하여 클러스터를 생성한 경우의 서버 추가 방법이다.

1. **add-server** 명령어를 사용하여 클러스터에 추가할 서버(server2)를 생성한다. add-server 명령어에 대한 자세한 사용법은 JEUS Reference 안내서의 [add-server](#)를 참고한다.

```
[MASTER]domain1.adminServer>add-server server2 -node node2 -addr 0.0.0.0 -port 9936
Successfully performed the ADD operation for server (server2).
NOTICE : base-addr [0.0.0.0] base-port [9936] http-port [8088]
Check the results using "list-servers or add-server".
```

2. **add-servers-to-cluster** 명령어를 사용하여 클러스터(cluster1)에 생성한 서버(server2)를 추가한다.

add-servers-to-cluster 명령어에 대한 자세한 사용법은 JEUS Reference 안내서의 [add-servers-to-cluster](#)를 참고한다.

```
[MASTER]domain1.adminServer>add-servers-to-cluster cluster1 -servers server2
Successfully performed the ADD operation for The server list for cluster(cluster1)..
Check the results using "list-clusters cluster1 or add-servers-to-cluster cluster1".
```

3. 새로 추가한 서버를 기동한다.

4. **serverinfo** 명령어를 사용하여 서버(server1, server2)가 클러스터(cluster1)에 묶여 잘 운영되고 있음을 확인할 수 있다.

```
[MASTER]domain1.adminServer>serverinfo
Information about Domain (domain1)
=====
```

Server	Status	Node Name	PID	Cluster	Latest Start Time / Shutdown Time	Need to Restart	Listen Ports	Running Engines
admin Server (*)	RUNNING(02:56:51)	node1	90288	N/A	2022-07-19 (화) 오후 12:55:20 KST	false	base-0.0.0.0:9736 http-server-0.0.0.0:8088	jms, web, ejb
server1	RUNNING(00:00:17)	node1	100083	cluster1	2022-07-19 (화) 오후 03:51:54 KST	false	base-0.0.0.0:9836 http-server-0.0.0.0:8188	jms, web, ejb
server2	RUNNING(00:00:09)	node2	100240	cluster1	2022-07-19 (화) 오후 03:52:02 KST	false	base-0.0.0.0:9936 http-server-0.0.0.0:8288	jms, web, ejb

```
=====
```

5.6.2. 클러스터에서 서버 삭제

클러스터의 크기를 줄이기 위해 클러스터에 포함된 서버를 삭제할 수 있다. 이때에는 삭제할 서버를 종료하여, 진행 중이던 서비스가 모두 완료된 것을 확인한 후 클러스터에서 삭제한다. 본 절에서는 클러스터에서 서버를 삭제하는 방법에 대해 설명한다.

콘솔 툴 사용

콘솔 툴(jeusadmin)을 사용하여 클러스터에 서버를 삭제할 수 있다.

다음은 서버를 선택하여 클러스터를 생성한 경우의 서버 제거 방법이다.

1. **stop-server** 명령어를 사용하여 삭제할 서버(server2)를 종료한다. stop-server 명령어에 대한 자세한 사용법은 JEUS Reference 안내서의 [stop-server](#)를 참고한다.

```
[MASTER]domain1.adminServer>stop-server server2
Stop server message to server [server2] was successfully sent.
```

2. **remove-servers-from-cluster** 명령어를 사용하여 삭제할 서버(server2)가 포함된 클러스터(cluster1)에서

서버를 삭제한다. `remove-servers-from-cluster` 명령어에 대한 자세한 사용법은 JEUS Reference 안내서의 [remove-servers-from-cluster](#)를 참고한다.

```
[MASTER]domain1.adminServer>remove-servers-from-cluster cluster1 -servers server2
Successfully performed the REMOVE operation for The server list for cluster(cluster1)..
Check the results using "list-clusters cluster1 or remove-servers-from-cluster cluster1".
```

- 클러스터에서 삭제한 서버(server2)를 단독 서버로 사용할 계획이 없다면 **remove-server** 명령어를 사용하여 완전히 삭제한다. `remove-server` 명령어에 대한 자세한 사용법은 JEUS Reference 안내서의 [remove-server](#)를 참고한다.

```
[MASTER]domain1.adminServer>remove-server server2
Successfully performed the REMOVE operation for server (server2).
Check the results using "list-servers or remove-server"
```

5.7. 클러스터 삭제

해당 서비스를 더 이상 사용할 필요가 없을 경우에는 클러스터를 삭제할 수 있다. 클러스터를 삭제할 때에는 삭제할 클러스터를 종료하여 진행 중이던 서비스의 완료를 확인한 후 삭제해야 한다. 본 절에서는 클러스터를 삭제하는 방법에 대해 설명한다.

콘솔 툴 사용

콘솔 툴(jeusadmin)을 사용하여 클러스터를 삭제하는 방법은 다음과 같다.

- stop-cluster** 명령어를 사용하여 삭제할 클러스터를 종료한다. 본 예제에서는 server1과 server2로 구성된 cluster1이라는 클러스터를 삭제한다. `stop-cluster` 명령어에 대한 자세한 사용법은 JEUS Reference 안내서의 [stop-cluster](#)를 참고한다.

```
[MASTER]domain1.adminServer>stop-cluster cluster1
Stopping servers [server1, server2].
Stop server message to the cluster [cluster1] was successfully sent.
```

serverinfo 명령어를 사용하여 종료한 클러스터(cluster1)에 포함된 서버(server1, server2)의 상태가 SHUTDOWN 상태임을 확인할 수 있다.

```
[MASTER]domain1.adminServer>server info
=====
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Server | Status | Node | PID | Clu | Latest Start | Need | Listen | Runni |
|         |         | Name |     | ster | Time /       | to   | Ports  | ng    |
|         |         |     |     |     | Shutdown Time | Restart |      | Engines |
+-----+-----+-----+-----+-----+-----+-----+-----+
| admin | RUNNING | nod | 102 | N/A | 2022-07-19 | false | base-0.0. | jms, |
| Server | (00:02:4 | e1 | 314 |   | (화) 오후   |      | 0.0:9736 | web, |
| (*)   | 3)      |   |     |   | 04:16:59 KST |      | http-serv | ejb  |
+-----+-----+-----+-----+-----+-----+-----+-----+
|         |         |     |     |     |              |      |          |      |
+-----+-----+-----+-----+-----+-----+-----+-----+
|         |         |     |     |     |              |      |          |      |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

							er-0.0.0.0		
							:8088		
+	+	+	+	+	+	+	+	+	+
	serve	SHUTDOWN	N/A	N/A	clu	2022-07-19	N/A	N/A	N/A
	r1	N(00:01:			ster1	(화) 오후			
		03)				04:18:39 KST			
+	+	+	+	+	+	+	+	+	+
	serve	SHUTDOWN	N/A	N/A	clu	2022-07-19	N/A	N/A	N/A
	r2	N(00:01:			ster1	(화) 오후			
		03)				04:18:39 KST			
+	+	+	+	+	+	+	+	+	+
=====									

2. 종료한 클러스터를 다음과 같이 **remove-cluster** 명령어를 사용해서 삭제한다. remove-cluster 명령어에 대한 자세한 사용법은 JEUS Reference 안내서의 [remove-cluster](#)를 참고한다.

```
[MASTER]domain1.adminServer>remove-cluster cluster1
Successfully performed the REMOVE operation for cluster (cluster1).
Check the results using "list-clusters or remove-cluster".
```

3. 클러스터를 해제한 이후 삭제한 클러스터에 포함되어 있던 서버들을 단독 서버로 사용할 계획이 없다면 삭제한다. **remove-server** 명령어를 사용하여 포함되어 있던 서버(server1, server2)를 삭제한다. remove-server 명령어에 대한 자세한 사용법은 JEUS Reference 안내서의 [remove-server](#)를 참고한다.

```
[MASTER]domain1.adminServer>remove-server server1
Successfully performed the REMOVE operation for server (server1).
Check the results using "list-servers or remove-server"

[MASTER]domain1.adminServer>remove-server server2
Successfully performed the REMOVE operation for server (server2).
Check the results using "list-servers or remove-server"
```

6. 서버 장애

다양한 이유로 서버가 비정상 종료될 수 있지만 클러스터를 구성하여 서버가 비정상 종료되는 경우에 서비스의 문제를 최소화 할 수 있다. 본 장에서는 서버가 비정상 종료될 경우의 문제점과 극복 방안에 대해서 설명한다.

6.1. Master Server(MASTER) 비정상 종료

MASTER가 비정상 종료되어도, Managed Server(MS)의 서비스는 영향을 받지 않고 동작한다. MASTER가 위치한 머신에 장애가 발생하여 MASTER가 비정상 종료된 상태라면, 그 머신에 함께 존재하는 MS들도 영향을 받아 비정상 종료될 가능성이 높다. 그러나 MASTER 자체의 비정상 종료가 도메인 내에 존재하는 MS의 서비스에 영향을 주지는 않는다.

6.1.1. 비정상 종료될 때 기능 제한

MASTER가 비정상 종료된 경우에는 MASTER의 역할이었던 설정의 변경이나, 애플리케이션의 추가 및 변경 등은 MASTER가 다시 시작될 때까지 기능을 할 수 없다. MASTER가 다시 시작되어 MS로 이벤트를 주면 MS는 다시 MASTER의 관리를 받을 수 있다.

MASTER와 연결이 안 되는 상태에서 MS를 시작하려고 한다면 MASTER의 설정으로 시작될 수 없고, MS가 가지고 있는 로컬 Cache를 사용하게 된다. 즉, 기존에 한 번 이상 MASTER와 연결을 맺은 적이 있어, MASTER로부터 받은 파일을 로컬에 가지고 있는 경우에만 해당된다. 로컬 Cache를 이용하여 MS를 시작한 뒤에 MASTER가 다시 시작되었을 때 MASTER의 파일과 동기화하게 된다. 물론 MS가 MASTER와 같은 머신에 존재하여 도메인 디렉터리를 공유한다면 MASTER가 종료되어 있을 때에도 MASTER와 같은 설정으로 MS를 시작할 수 있다.

6.1.2. 도메인 백업을 이용한 장애 극복

MASTER가 비정상 종료되어도 서비스에는 지장이 없지만 몇 가지 기능이 제한된다. 따라서 MASTER를 빨리 복구시켜야 하는데, MASTER가 현 상태 그대로 다시 시작할 수 없는 경우가 있다. 예를 들면 머신 자체가 비정상 종료된 경우에는 머신이 복구되기 전까지는 MASTER를 재시작할 수가 없다.

이런 경우를 대비하여 MASTER가 가지고 있는 파일들 즉, MASTER에서 유지해야 하는 설정과 애플리케이션을 평소에 백업해둘 수 있다. 미리 백업해 둔 MASTER의 파일들을 다른 머신에 복사하여 새로운 머신에서 MASTER를 시작할 수 있다. 단, 백업한 애플리케이션을 다른 머신에서 사용하기 위해서는 JEUS_HOME 경로가 기존 머신의 JEUS_HOME과 같아야 한다.

MASTER의 설정 및 애플리케이션을 백업하는 방법은 크게 세 가지를 제공한다.

- MASTER 부팅 시 도메인 백업 설정을 통한 자동 백업
- pack-domain 명령어를 통한 명시적 백업

백업 설정을 통한 자동 백업은 MASTER가 기동할 때에 자동으로 백업 파일을 생성한다. 그리고 pack-domain 명령어는 원하는 때에 백업 파일을 만들 수 있는 기능을 한다. 이 경우에 만들어진 파일은 사용자가 관리해야 한다.

도메인 백업 설정을 통한 자동 백업

MASTER가 시작할 때마다 자동으로 MASTER의 설정 및 애플리케이션을 백업한다. 도메인 백업 설정은 콘솔 툴(jeusadmin)을 통하여 설정할 수 있다.

- 콘솔 툴 사용

다음과 같이 **set-domain-backup** 명령어를 사용하여 MASTER 시작 시점에 MASTER 백업 파일을 만들 수 있도록 설정할 수 있다.

```
[MASTER]domain1.adminServer>set-domain-backup -backupOnBoot true
Successfully performed the MODIFY operation for Domain Backup Policy.
Check the results using "set-domain-backup"
```

이렇게 백업된 파일은 기본적으로 **pack-domain**을 통하여 만든 파일과 동일하며, **unpack-domain**을 통하여 사용할 수 있다.

pack-domain 명령을 통한 명시적 백업

- MASTER의 설정 및 애플리케이션의 백업

다음과 같이 **pack-domain** 명령어를 사용하여 MASTER의 설정 및 애플리케이션을 백업한다.

```
offline>pack-domain domain1
Packing the domain [domain1] configuration completed successfully at the path
[JEUS_HOME/domains/domain1_packed.zip].
```

- 새로운 머신에 백업한 MASTER의 설정 및 애플리케이션 적용

다음과 같이 **unpack-domain** 명령어를 사용하여 백업한 파일의 압축을 푼다. **unpack-domain** 명령어를 사용할 때 MASTER의 IP, Port, Node 정보를 변경할 수 있다.

```
offline>unpack-domain domain1
The JEUS Master Server listener address is already set to [0.0.0.0]. Do you want to change it?
(y/n): y
Enter the JEUS Master Server base listener address: 0.0.0.0
The JEUS Master Server listener port is already set to [9736]. Do you want to change it? (y/n): n
The JEUS Master Server nodename is already set to [node1]. Do you want to change it? (y/n): n
Unpacking the domain [domain1] configuration completed successfully.
```

새로운 머신에서 기존과 같은 설정으로 MASTER가 시작되면, INDEPENDENT 모드로 있던 MS들이 MASTER로부터 설정과 애플리케이션을 동기화하고, MASTER의 관리를 받게 되어 INDEPENDENT 모드에서 해제된다.



pack-domain, unpack-domain 명령어에 대한 자세한 사용 방법은 각각 JEUS Reference 안내서의 "pack-domain"과 "unpack-domain"을 참고한다.

6.1.3. INDEPENDENT 모드의 Managed Server(MS)

MS의 서비스가 MASTER의 비정상 종료로 인한 영향을 받지 않기 위해 MS는 MASTER의 존재 여부에 의존하지 않고 서비스를 수행한다.

MS의 설정과 애플리케이션의 로컬 Cache는 서버가 시작할 때 MASTER로부터 동기화해서 가지고 있고, 변경될 때마다 MASTER와 동기화한다. 그러나 MS가 시작될 때 MASTER와 연결되지 않은 경우에는 로컬 Cache를 이용하여 시작되는데, 이것을 **INDEPENDENT 모드**라고 한다. 이 상태에서 MS는 MASTER의 제어를 받을 수 없고, MASTER 또한 해당 MS를 모니터링할 수 없다. 이는 MASTER가 실제로 비정상 종료된 경우와 MASTER의 주소를 잘못 설정했거나 네트워크 관련 오류로 연결에 문제가 발생한 경우가 해당된다.

MASTER가 재시작되어 MS가 이벤트를 받게 되면 다시 MASTER의 관리를 받게 되는 모드로 변경된다.

6.2. Managed Server(MS)의 비정상 종료

도메인 내에 MS들은 Multicast로 서로의 상태 정보를 주고받기 때문에 비정상 종료된 서버를 감지할 수 있다.

7. Security 관리

본 장에서는 기본적으로 JEUS 서버를 사용하는 데 있어서 필요한 암호화된 비밀번호 및 계정 관리 등 몇 가지 보안 방식에 대해서 간단하게 설명한다. Security 정보는 도메인에 속한 모든 서버가 공유하며, MASTER를 통해 설정해야 한다.



JEUS Security에 대한 자세한 내용은 [JEUS Security 안내서](#)를 참고한다.

7.1. 계정 관리

JEUS에서 계정은 개인 단위인 User(사용자)와 User들의 집합인 Group(그룹)으로 관리된다.

JEUS의 권한 부여 모델은 Resource(리소스)들에 대한 권한을 Role(역할)이 갖고 Role에 User를 지정하여, Role이 가진 Resource 권한을 User가 갖는 구조이다.

다음은 Role에 User를 지정하는 3가지 방법이다.

- Role에 User의 Name을 직접 지정한다.
- Role에 User가 속한 Group의 Name을 지정한다.
- Role에 User가 속한 Group의 상위 Group의 Name을 지정한다.

계정의 구조가 User, Group, Subgroup 등의 계층을 갖기 때문에 JEUS에서 비슷한 역할을 하는 User에게 한 번에 비슷한 권한을 부여할 수 있다. 보안 정책에 대한 자세한 설정 방법은 JEUS Security 안내서의 [보안 시스템 정책 설정](#)을 참고한다.

7.2. 비밀번호 암호화 저장

시스템 관리를 위한 사용자의 비밀번호를 간편하게 암호화하여 저장할 수 있다.

암호화 정보는 다음의 형식으로 저장된다.

{알고리즘}암호문

가능한 암호화 알고리즘은 다음과 같다.

- AES
- DES
- DESede
- blowfish
- SEED
- base64

- SHA
- SSHA

패스워드 암호화 저장은 콘솔 툴(jeusadmin)을 사용하거나, 암호화 툴을 통해 직접 설정하는 방법이 있다. 본 절에서는 각 방법에 대해 설명한다.

7.2.1. 콘솔 툴 사용

다음은 콘솔 툴(jeusadmin)을 통해 'administrator'라는 계정의 'password'라는 암호를 암호화 알고리즘 'AES'를 사용하여 암호화하고 저장하는 예제이다.

```
[MASTER]domain1.adminServer>set-password administrator password -algorithm AES
The password is set for [administrator].

[MASTER]domain1.adminServer>exit

JEUS_HOME/bin$ jeusadmin -u administrator -p password

Attempting to connect to 127.0.0.1:9736.
The connection has been established to JEUS Master Server [adminServer]
in the domain domain1.
JEUS 9 Administration Tool
To view help, use the 'help' command.
```

7.2.2. 암호화 툴을 통한 직접 설정

암호화 툴을 통해 직접 암호화를 수행하여 저장할 수도 있다. 예를 들어 DB 패스워드, accounts.xml 계정들의 패스워드의 경우 일반 문자열 대신 암호화된 문자열로 입력할 수 있다.

암호화된 문자열은 다음의 형식으로 패스워드가 필요한 위치에 설정한다.

{알고리즘}암호문

이러한 암호화된 문자열을 사용하기 위해서는 JEUS가 제공하는 암호화 툴(Encryption Tool)을 사용해야 한다. 암호화 툴은 암호화될 대상이 되는 문자열과 알고리즘을 입력받아 결과물인 암호문을 출력하는데, 그 암호문을 위에서 언급한 형식에 맞춰서 기록한다.



암호화 툴의 사용법에 관한 자세한 사항은 JEUS Reference 안내서의 [encryption](#)을 참고한다.

base64 또는 Hash 알고리즘(sha 등)은 평문이 정해지면 결과값도 항상 동일하게 출력되지만, 그 외의 알고리즘들은 수행하기 위한 비밀 Key(secret key)가 필요하다. 이러한 비밀 Key 관리 방법에 관한 자세한 사항은 [비밀 Key 파일 관리](#)를 참고한다.

다음은 accounts.xml 편집을 통해 암호화된 문자열을 패스워드로 설정하는 예제이다.

암호화된 문자열 비밀번호 설정 : <accounts.xml>

```
<accounts xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  <users>
    <user>
      <name>administrator</name>
      <password>{base64}amV1czEyMw==</password>
    </user>
    <user>
      <name>user1</name>
      <password>{AES}i06wYRz3Gqun2sKtXHIq+Tw3vUcc=</password>
    </user>
    . . .
  </users>
  . . .
</accounts>
```



최초 도메인을 구성했을 때에는 설정 파일(XML)을 직접 수정할 수 있으나, 도메인 구성 후에는 콘솔 툴(jeusadmin)을 통해 설정하는 것을 권장한다.

7.3. 비밀 Key 파일 관리

본 절에서는 비밀 Key 파일을 생성하고 관리하는 방법과 비밀 Key 파일의 보호 옵션 및 서버 기동 방법에 대해 설명한다.

7.3.1. 비밀 Key 파일의 생성과 관리

JEUS_HOME/bin에 위치한 암호화 툴(Encryption Tool)이 제공하는 AES, DES, DESede, SEED, BlowFish 등의 알고리즘은 암호화, 복호화를 수행하기 위해 비밀 Key가 필요하다. 이를 위해서 JEUS는 비밀 Key 파일을 사용한다. 비밀 Key 파일은 암호화 툴을 처음으로 사용할 때 자동으로 생성되며, 그 이후로는 그 파일을 계속 사용한다.

비밀 Key 파일은 기본적으로 다음 경로의 **security.key** 파일이다.

```
JEUS_HOME/domains/<domain-name>/config/security
```

또는 jeus.security.keypath 시스템 프로퍼티로 지정할 수 있다. 이 경로는 절대 경로, 상대 경로 모두 가능하며, 상대 경로의 기준 경로는 JVM을 실행한 경로이다. 경로를 디렉터리로 지정할 경우에는 해당 디렉터리의 security.key 파일을 이용하며, 파일로 지정할 경우에는 지정된 파일을 사용한다.

7.3.2. 비밀 Key 파일 보호 옵션과 서버 기동 방법

JEUS에서는 위에서 언급한 비밀 Key 파일 자체를 보호하는 옵션도 제공한다. 이를 위해서는 비밀 Key 파일을 암호화하기 위한 비밀번호가 필요하다. 이를 마스터 비밀번호(Master Password)라고 한다. 암호화 툴을 실행할 때 -protectkey 옵션을 입력하면 이 마스터 비밀번호를 이용해서 비밀 Key 파일을 보호할 수 있다.

암호화 툴을 통해 마스터 패스워드가 설정되었다면 서버를 기동할 때 반드시 마스터 패스워드가 필요하게 된다.

마스터 패스워드를 입력하는 방법은 다음과 같이 크게 2가지가 있다.

- 서버를 백그라운드 프로세스로 시작할 경우 시스템 프로퍼티 `jeus.security.master`로 지정하는 방법
- JEUS에서 제공하는 스크립트를 이용하여 시작할 때 콘솔 툴로 입력받는 방법

`jeus.security.master`로 지정하는 방법은 셸 스크립트에 마스터 패스워드를 입력해야 하므로 보안상 안전하지 않다. 만약 마스터 패스워드를 사용해야 하는 상황이라면 되도록 후자의 방법을 추천한다.



1. 서버 기동에 대한 자세한 내용은 JEUS Reference 안내서의 [Script 모드 사용법 및 Script 작성법](#)을 참고한다.
2. 비밀 Key에 대한 자세한 내용은 JEUS Reference 안내서의 [encryption](#)을 참고한다.

7.4. Keystore와 Truststore 관리

본 절에서는 SSL 통신에 사용되는 Keystore와 Truststore의 관리에 대해 간단히 설명한다.

Keystore와 Truststore는 모두 Java Key Store(JKS) 형식으로 본질적으로 동일하며, JDK에서 제공하는 Key 툴을 사용하여 생성 및 관리할 수 있다.



Keystore, Truststore 및 SSL에 대한 자세한 정보를 원한다면 Java SE 문서나 [Oracle Java 기술 문서](#)에서 JSSE, JKS, Key 툴(keytool) 관련 정보를 찾아볼 수 있다.

JEUS는 기본적으로 다음 경로의 파일들을 각각 Keystore, Truststore로 지정해서 사용하며, Keystore, Truststore가 다른 경로에 혹은 다른 이름으로 존재한다면 이를 직접 설정해야 한다.

- Keystore

Keystore는 개인 Key(Private Key)들만 별도로 모아서 저장해 놓은 저장소이다.

```
JEUS_HOME/domains/<domain-name>/config/security/keystore 파일
```

- Truststore

Truststore는 공개 Key(Public Key)들만 별도로 모아서 저장해 놓은 저장소이다.

```
JEUS_HOME/domains/<domain-name>/config/security/truststore 파일
```

설정 방법은 시스템 전체적으로 시스템 프로퍼티로 설정할 수도 있고, Keystore와 Truststore가 필요한 부분에서 각각 설정할 수도 있다. 각각 설정하는 방법은 안내서의 각 파트에 별도로 설명이 되어 있다.

사용하는 시스템 프로퍼티는 다음과 같다.

시스템 프로퍼티	설명
jeus.ssl.keystore	SSL에서 사용하는 Keystore 파일 경로를 설정한다. (기본값: JEUS_HOME/domains/<domain-name>/config/security/keystore)
jeus.ssl.truststore	SSL에서 사용하는 Truststore 파일 경로를 설정한다. (기본값: JEUS_HOME/domains/<domain-name>/config/security/truststore)
jeus.ssl.keypass	SSL에서 사용하는 Keystore 패스워드를 설정한다. (기본값: jeuskeypass)
jeus.ssl.trustpass	SSL에서 사용하는 Truststore 패스워드를 설정한다. (기본값: jeustrustpass)

Keystore와 Truststore를 사용하기 위해서는 패스워드가 필요하며 위에서 설명한 것처럼 JEUS는 Keystore와 Truststore의 패스워드를 설정할 수 있다. 경로를 설정하는 것과 마찬가지로 시스템 프로퍼티로 전체적으로 설정하거나 필요한 부분에서 각각 설정할 수도 있다.

특히 Keystore는 저장소 자체의 패스워드 외에 Keystore 내에 존재하는 Key별로 각각 패스워드가 요구된다. 하지만 대부분의 경우에 이 패스워드들이 Keystore의 패스워드와 동일하기 때문에 JEUS 내부적으로 Keystore 패스워드와 동일하게 사용된다.

Keystore의 Key 패스워드를 저장소 패스워드와 구분해야 하는 경우에는 각 설정에 <keystore-keypassword> 지정하는 태그가 있으므로 이를 지정하여 사용한다. 또한, Keystore 내의 Key들의 패스워드가 서로 다를 경우 해당 Keystore를 이용할 수 없으며 동일 Keystore 내부의 Key들은 모두 동일한 패스워드로 설정되어 있어야 한다.



JEUS를 설치할 때 기본적으로 Keystore와 Truststore가 포함되어 있지만 직접 Keystore와 Truststore를 새로 생성하여 설정하는 것을 권장한다.

7.5. SSL이 설정된 도메인에서의 실행 스크립트 관리

본 절에서는 도메인에서 각 서버의 Base Listener를 SSL로 동작하도록 설정하였을 때 각종 실행 스크립트에 추가해야 할 설정에 대해서 설명한다.

도메인에 속하는 각 서버들을 SSL로 설정할 경우 서버들 간의 통신은 서버에 설정된 SSL 설정들이 이용되지만, 콘솔 툴 혹은 실행 스크립트 등 클라이언트 입장에서 동작하는 프로세스들에는 SSL 설정이 따로 필요하게 된다.

다음의 경우 SSL 설정이 필요하다.

구분	설명
jeusadmin	콘솔 툴을 실행하기 위한 스크립트
startManagedServer	MS를 직접 구동하기 위한 스크립트
stopServer	서버의 동작을 정지시키기 위한 스크립트
기타 Standalone Client	스크립트는 아니지만 SSL을 이용하여 서버와 통신해야 하는 Standalone Client

다음 예제에서 볼드체로 강조된 부분은 스크립트에 추가해야 할 내용이다.

SSL과 관련된 클라이언트 설정을 직접 명시한 jeusadmin 스크립트

```
. . .
# execute jeusadmin
"${JAVA_HOME}/bin/java" -Xmx128m -classpath "${BOOTSTRAP_CLASSPATH}" ${TOOL_OPTION}
-Djmx.remote.x.request.timeout=600000
-Djeus.home="${JEUS_HOME}"
-Djeus.baseport=${JEUS_BASEPORT}
-Djeus.tool.console.useJLine="false"
-Djava.naming.factory.initial=jeus.jndi.JEUSContextFactory
-Djava.naming.factory.url.pkgs=jeus.jndi.jns.url
-Djava.util.logging.config.file="${JEUS_HOME}/bin/logging.properties"
-Djeus.net.client.use-ssl=true
-Djavax.net.ssl.trustStore=${JEUS_HOME}/domains/domain1/config/truststore
-Djavax.net.ssl.trustStorePassword=changeit
-Djavax.net.ssl.trustStoreType=JKS
-Dssl.TrustManagerFactory.algorithm=SunX509
-Djavax.net.ssl.keyStore=${JEUS_HOME}/domains/domain1/config/keystore
-Djavax.net.ssl.keyStorePassword=changeit
-Djavax.net.ssl.keyStoreType=JKS
-Dssl.KeyManagerFactory.algorithm=SunX509
-Djeus.security.keypath=${JEUS_HOME}/domains/domain1/config/security/security.key
${JAVA_ARGS}
jeus.tool.console.ConsoleBootstrapper ${BOOT_PARAMETER}
```

또는 다음과 같이 대부분의 설정을 별도의 파일에 명시한 후 해당 파일을 지칭할 수 있다.

SSL 설정을 별도 파일에 명시하도록 한 jeusadmin 스크립트

```
. . .
# execute jeusadmin
"${JAVA_HOME}/bin/java" -Xmx128m -classpath "${BOOTSTRAP_CLASSPATH}" ${TOOL_OPTION}
-Djmx.remote.x.request.timeout=600000
-Djeus.home="${JEUS_HOME}"
-Djeus.baseport=${JEUS_BASEPORT}
-Djeus.tool.console.useJLine="false"
-Djava.naming.factory.initial=jeus.jndi.JEUSContextFactory
-Djava.naming.factory.url.pkgs=jeus.jndi.jns.url
-Djava.util.logging.config.file="${JEUS_HOME}/bin/logging.properties"
-Djeus.net.client.ssl-properties-file=ssl.properties
${JAVA_ARGS}
jeus.tool.console.ConsoleBootstrapper ${BOOT_PARAMETER}
```

다음은 SSL과 관련된 클라이언트 설정을 별도 파일에 명시한 예제이다. 파일에 설정을 할 경우에는 \${JEUS_HOME}으로 시작하는 상대 경로는 사용할 수 없으므로 주의한다.

SSL과 관련된 클라이언트 설정 : <ssl.properties>

```
javax.net.ssl.keyStore=/Users/tmax/JEUS9/domains/domain1/config/keystore
javax.net.ssl.trustStore=/Users/tmax/JEUS9/domains/domain1/config/truststore
javax.net.ssl.keyStorePassword=jeus
javax.net.ssl.trustStorePassword={AES}i06wYRz3Gqun2sKtXHIq+Tw3vUcc=
jeus.ssl.keystorekeypass={base64}amV1czEyMw==
```

```
jeus.security.keypath=/Users/tmax/JEUS9/domains/domain1/config/security/security.key
```

password가 들어가는 property 항목에는 accounts.xml에 사용되는 것과 동일한 방식으로 인코딩 혹은 암호화된 문자열을 사용할 수 있다. 이때는 JEUS에서 제공하는 암호화 툴을 사용하도록 한다.



암호화 툴 사용법에 대한 자세한 사항은 JEUS Reference 안내서의 [encryption](#)을 참고한다.