Web Application Server Clustering

OPENMARU Cluster

오픈소스

웹 어플리케이션 서버를 위한
클러스터링 솔루션

서로 다른 WAS와
어플리케이션 간 세션공유
SESSION SHARE

인 메모리 데이터그리드
기반 클러스터링
CLUSTERING

openmaru
Cluster

모바일 환경에서 세션관리
MOBILE MONITORING

고가 WAS대비
비용 효율적인 WAS로 전환
LOW COST

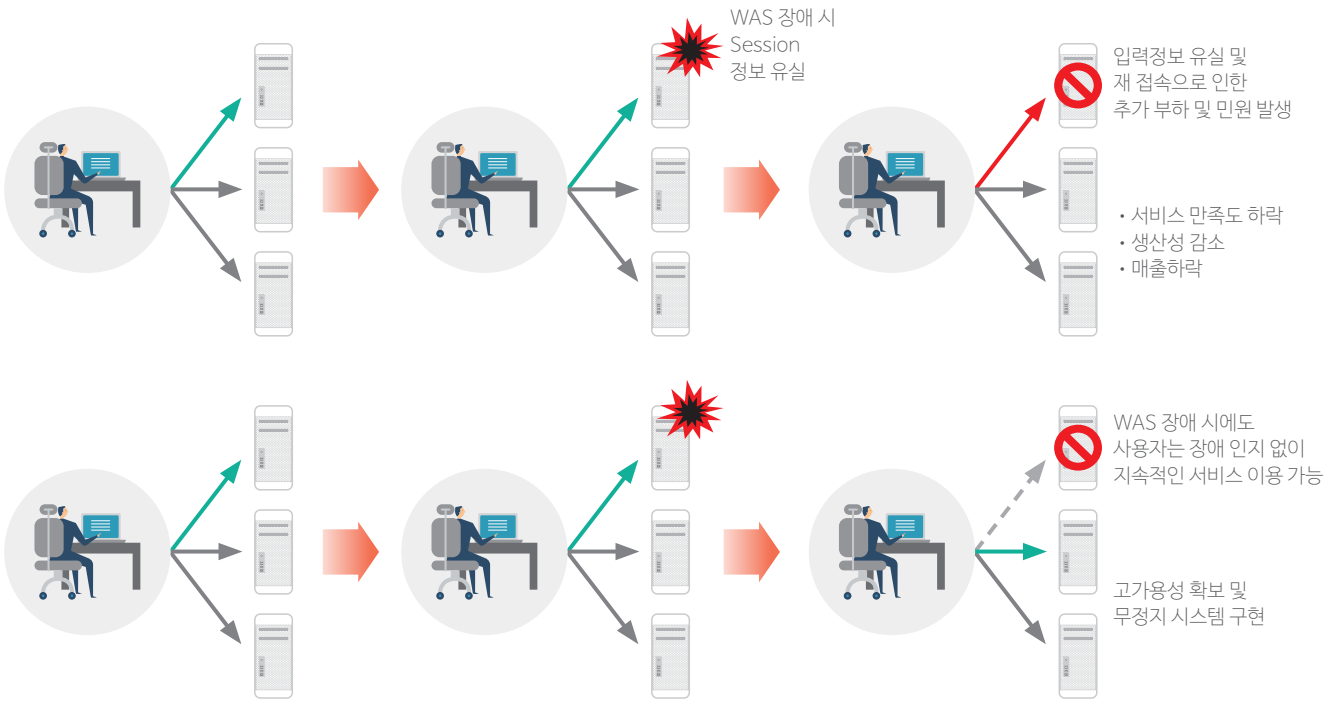
검증된 안정성
SAFETY

opennaru

세션 클러스터링은 왜 필요할까요?

세션 클러스터링은 대규모 시스템에서 여러 대의 서버로 서비스하더라도, 사용자는 투명하게 하나의 시스템으로 보일 수 있도록 합니다.

사용자 측면



시스템 운영자 측면

WAS 클러스터링 이슈가 자주 발생되어, 변경 작업을 새벽에만 합니다.

WAS 담당자

쇼핑몰과 교육 사이트는 사용자 로그인 상태 유지가 매우 중요합니다.

현업

Tomcat에 적용 할 수 있는 괜찮은 클러스터링 제품은 없을까요?

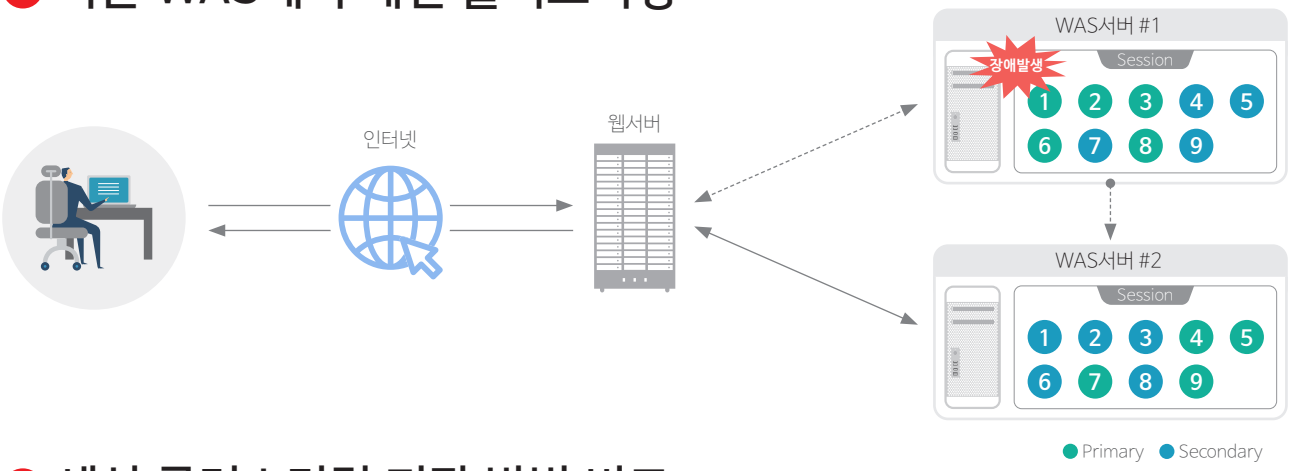
시스템 담당자

긴급 반영해야 하는데 클러스터링이 지원되지 않아 저녁까지 기다려야 해요

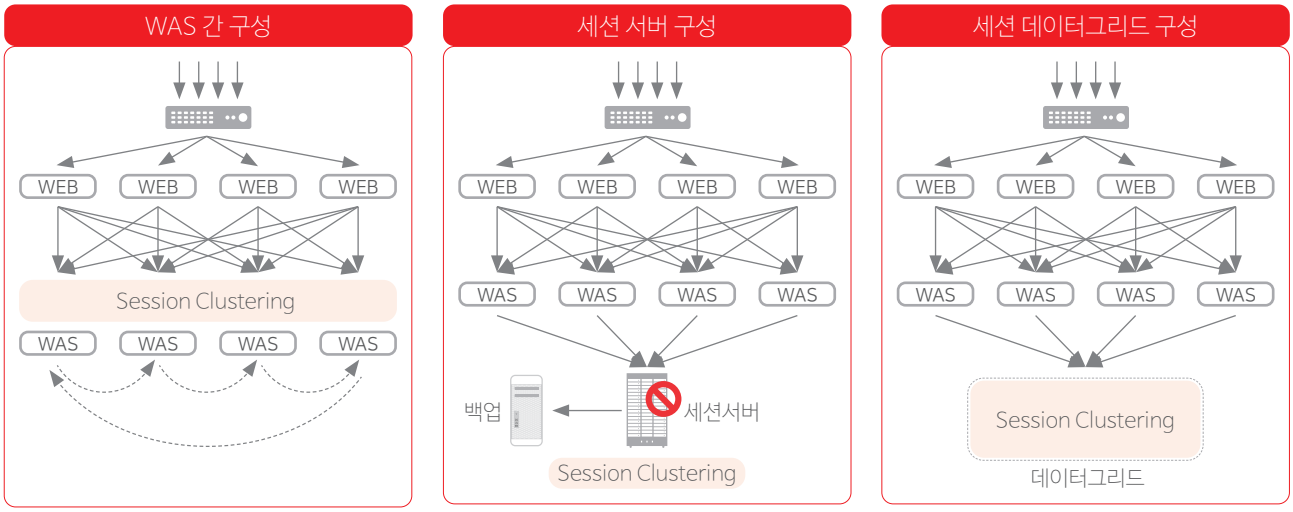
개발자

세션 클러스터링 방식은?

기존 WAS에서 세션 클러스터링



세션 클러스터링 저장 방법 비교



항목	WAS 내장 세션 관리	별도 세션 관리 서버	데이터그리드 기반 세션 관리
구현방법	• 세션 데이터별 Primary/Backup 인스턴스를 지정하여 공유	• 별도의 세션 서버 운영	• 데이터그리드에 세션 정보를 저장하여 운영
장점	• 별도의 서버와 인프라 없이 가능	• 인스턴스간 애플리케이션간 세션 공유 설정이 용이	• 인스턴스와 애플리케이션 간 세션 공유 용이 • Elastic 확장성과 안정성 보장
단점	• 세션데이터의 백업 및 동기화 이슈 • WAS 인스턴스 장애와 함께 세션 복제의 이슈가 발생	• 단일 장애 지점과 별도의 서버 구성에 따른 비용 • 제한적인 안정성 • 낮은 성능	• 별도의 서버 구성으로 인한 비용 발생 • 관리 포인트 증가 • 메모리 기반 고성능

WAS 클러스터링의 한계

과도한 세션 사용으로
긴 **GC 시간과**
OOM 장애 발생

서로 다른
애플리케이션이나 WAS간
세션 공유

애플리케이션간
세션 공유를 통한
싱글 로그온 구현

중복로그인 방지나
강제 로그아웃 등
세션을 통한 보안 강화

openmaru Cluster 소개

Apache Tomcat에서
세션 클러스터링이 필요할 때

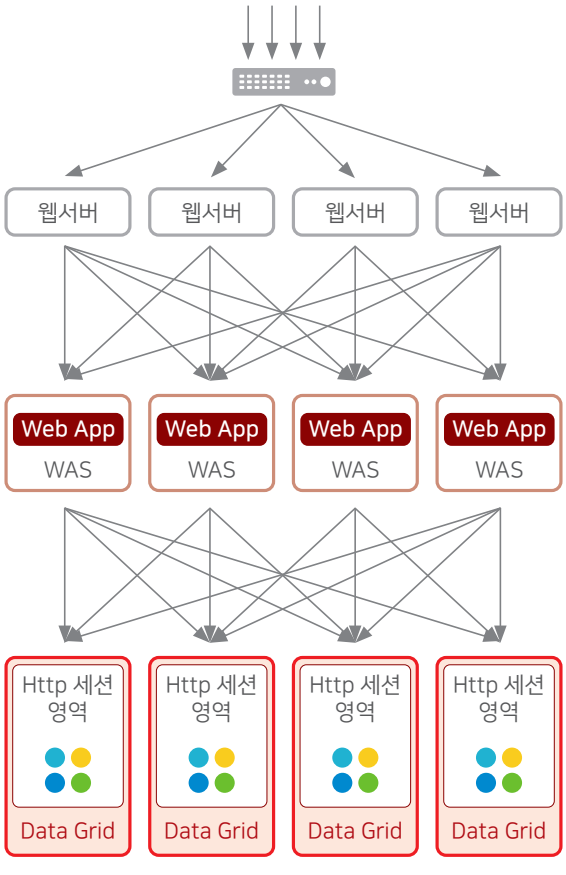
서로 다른 웹 애플리케이션 간의
세션 공유를 원할 때

웹 클러스터링을 안정적인 환경에서
운영하고 싶을 때

중복 로그인 방지 기능이 필요할 때

WAS Standard Ed.에
세션 클러스터링이 필요할 때

Data Grid를 이용한 클러스터링



openmaru Cluster 기대 효과

openmaru Cluster

서비스 품질 향상

- 세션 클러스터링을 통한 장애 극복
- 서비스 안정성 향상을 통한 사용자 만족도 증대
- 서비스 품질 향상을 통한 매출 증대와 업무 생산성 향상

운영 편이성 제공

- 서비스 다운타임 없이 애플리케이션 배포 (Rolling Upgrade)
- 서비스 다운타임 없이 웹시스템 유지보수 (Rolling Patch)
- 긴급 배포와 패치를 통한 웹시스템 유지 관리

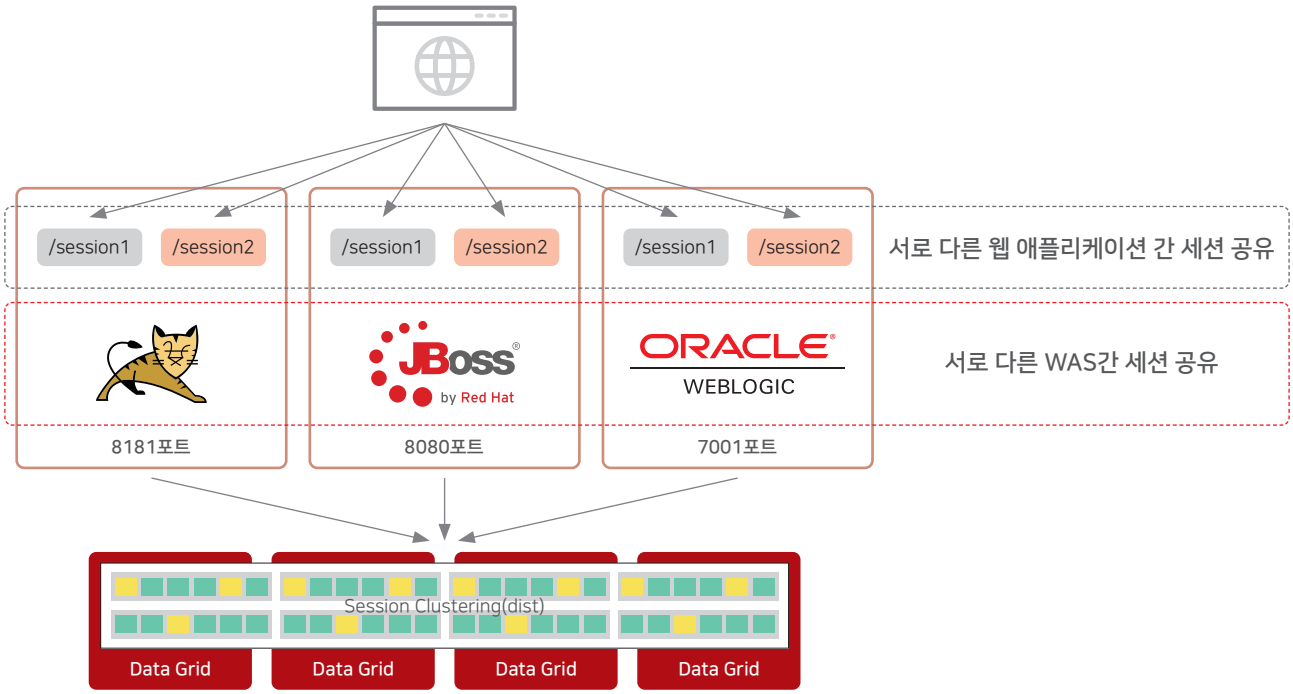
자원 활용 극대화

- WAS에 대한 Active-Active 구성에 따른 자원 효율성 확보
- 업무별 WAS 통합/이 기종 WAS 통합
- Standard Edition WAS 재활용

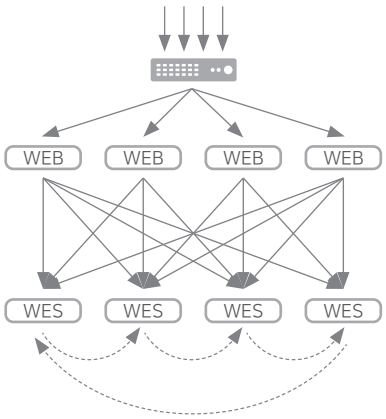
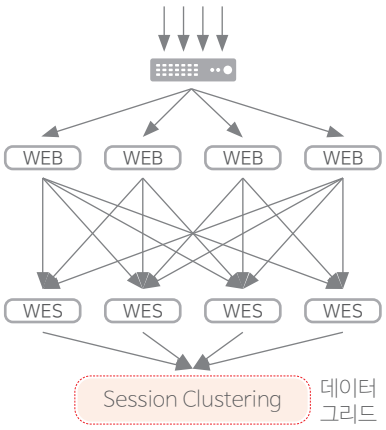
비용 절감

- 비용 절감 (WAS Standard + openmaru Cluster)
- 상용 WAS Enterprise Edition 대비, 40~60% 도입비용 절감
- SSO 도입 비용 절감
- 유지보수 비용 절감(80~40% 절감)

openmaru Cluster 구성



기존 WAS 클러스터링 vs. Grid 기반 클러스터링 비교

항목	WAS 세션 클러스터링		openmaru Cluster 세션 클러스터링	
아키텍처				
안정성	○	• 해당 인스턴스와 다른 인스턴스에 세션 데이터를 복제하고 동기화하여 관리 • 과도한 세션 사용시 OOM 메모리 장애 발생 • 세션데이터에 의한 GC가 장시간 발생	●	• 세션 데이터를 데이터그리드에 저장하고 공유하기 때문에 거래가 증가되어도 가용성을 유지한 채 안정적으로 분산 관리 가능 • WAS 노드 장애 시 상호 공유된 세션 정보를 통해 세션유실방지
성능	◑	• WAS 인스턴스 관리 세션 복제와 동기화에 따른 성능 이슈	●	• 세션 복제나 동기화 과정이 생략되어 신속한 WAS 관리 작업이 가능
확장성	◑	• WAS 인스턴스 확장	●	• 애플리케이션 메모리와 세션 메모리를 분리하여 예측 가능한 확장성 보장
세션관리	○	• WAS 인스턴스 재시작시 세션 동기화와 복제 • 애플리케이션 배포시 세션 동기화와 복제 • 애플리케이션 별 세션 정보 관리	●	• WAS 인스턴스 재 시작시 세션 복제 작업 제거 • 애플리케이션 재배포 시 세션 복제 작업 제거 • 복수의 애플리케이션 간 세션 정보 공유

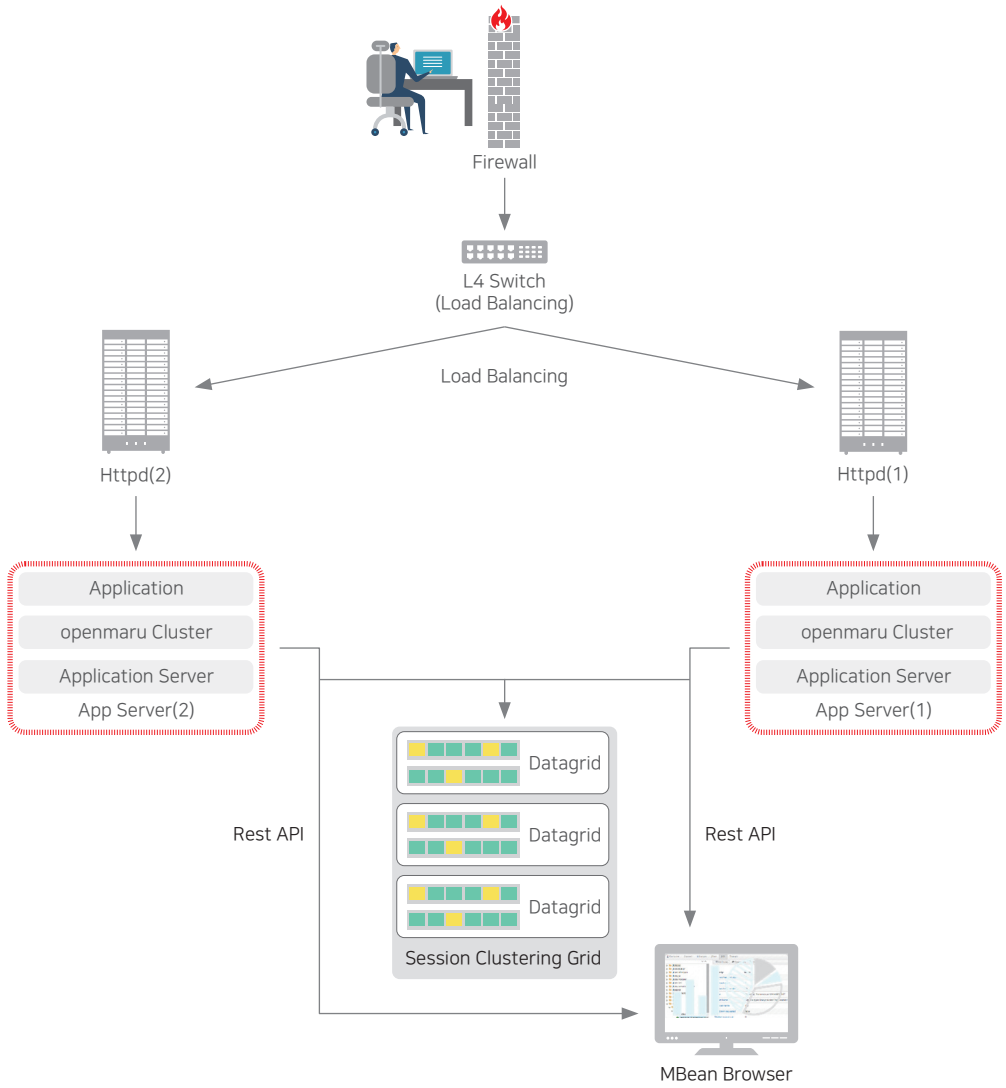
지원 환경

항목	지원 내용	
지원 운영체제	Red Hat Linux 6.x/7.x	X86_64
	CentOS 6.x/7.x	X86_64

항목	지원 내용	최소환경	권장 환경
openmaru Cluster 최소 사용 환경	Core 수	2 Core	4 core 이상
	Memory	8 GB	16GB 이상
	머신	1대	3대 이상

openmaru Cluster 구축 아키텍처

- 기업 내 시스템들간의 Session을 공유 함으로써 무중단 서비스 제공
- 분산된 컴퓨터 자원을 그리드 기반으로 하나의 네트워크처럼 연결
- Application과 WAS의 처리능력을 극대화



지원 OS	지원 JDK	지원 WAS
IBM AIX HP-UX Oracle Solaris Linux Microsoft Windows	Oracle JDK 1.7, 1.8 Open JDK 1.7, 1.8	Red Hat JBoss EAP Red Hat JBoss EWS Apache Tomcat Oracle WebLogic IBM WebSphere Application Server Tmaxsoft JEUS Caucho Technology Resin

openmaru
Cluster

www.opennaru.com
서울시 성동구 뚝섬로1길 31, 906~907호
(성수동1가, 서울숲 M타워) (우)04778
Tel. 02 469 5426 Fax. 02 469 7247
Email. sales@opennaru.com

opennaru