



open**naru**

'웹 세션 클러스터링'

KHAN [session manager]

오픈나루

전준식 (010-5120-3249)

service@opennaru.com



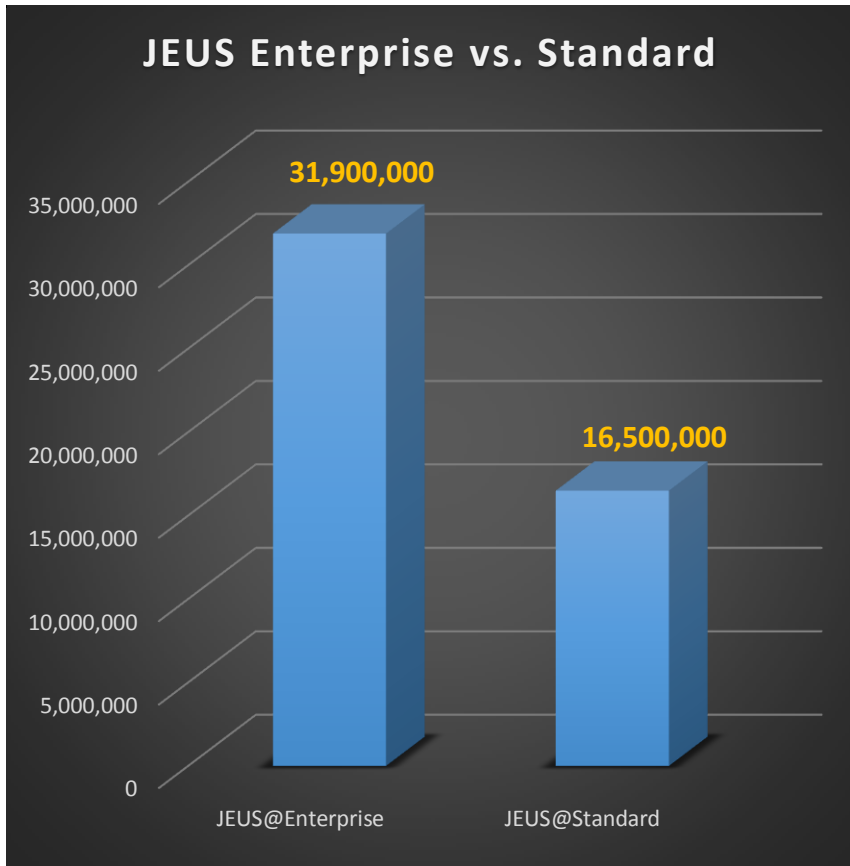
미들웨어(WAS) 제품

- 클러스터링 기능 유무에 따라 Enterprise 버전과 Standard 버전으로 나뉨
- Clustering 포함되는 기능은 EJB, Session, JNDI 등의 상태를 저장/유지하고 장애 시 자동 복구
- 클러스터링 기능 중 가장 많이 사용되는 기능은 '웹 세션 클러스터링'

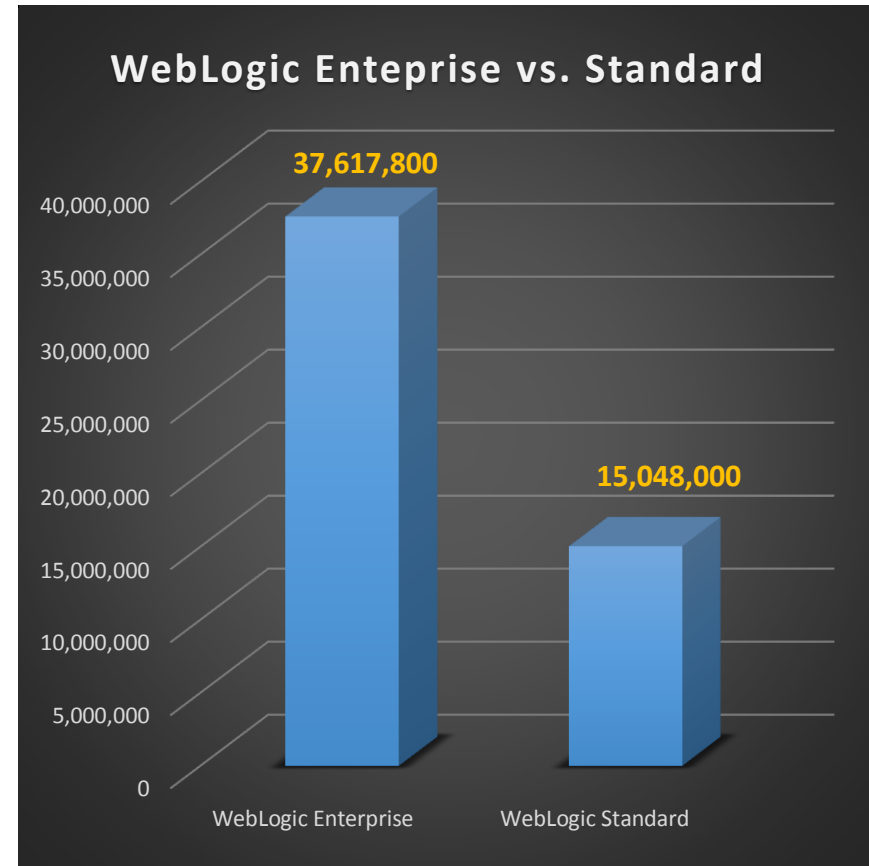
Benefits	Description	Features	WebLogic Server Standard Edition	WebLogic Server Enterprise Edition	WebLogic Suite
Enterprise-strength availability and scalability	Adds premium clustering capabilities to enable automatic failover, replications, and additional load balancing schemes	MAN/WAN replication and failover		•	•
		Whole server migration		•	•
		Side-by-side/production redeployment	•	•	•
		Asynchronous HTTP session replication reduces latency in HTTP client response		•	•
		Highly available singleton service management		•	•
		In-memory replication of EJB state		•	•
		In-memory replication of servlet session state		•	•
		Stateful session EJB failover		•	•
		Clusterwide JNDI naming service		•	•
		Automatic migration of Transaction Recovery Service (TRS)		•	•

Source : http://patriotek.com/wp-content/downloads/oracle_weblogic_server_feature_matrix.pdf

Oracle 과 TMAXSoft WAS 제품의 경우 클러스터링 유무에 따라 Enterprise 와 Standard 제품으로 구분되며, 2 배 이상의 가격차이가 있음



[TMAXSoft JEUS 가격 구분 (Core/원)]



[Oracle WebLogic 가격 구분 (Core/원)]

Source : <http://softwarecatalog.co.kr/>



KHAN [session manager]
WAS 세션 클러스터링 방식

WAS 고가용성 실현을 위한 방안 수립

현재 WAS에서 제공하는 세션 클러스터링으로는 해결이 어렵다.

과도한 세션 사용으로
긴 GC 시간과
OOM 장애 발생



WAS 작업 시 세션
데이터 동기화와 복
제에 따른 부하 발생



애플리케이션간 세
션 공유를 통한 싱글
로그온 구현

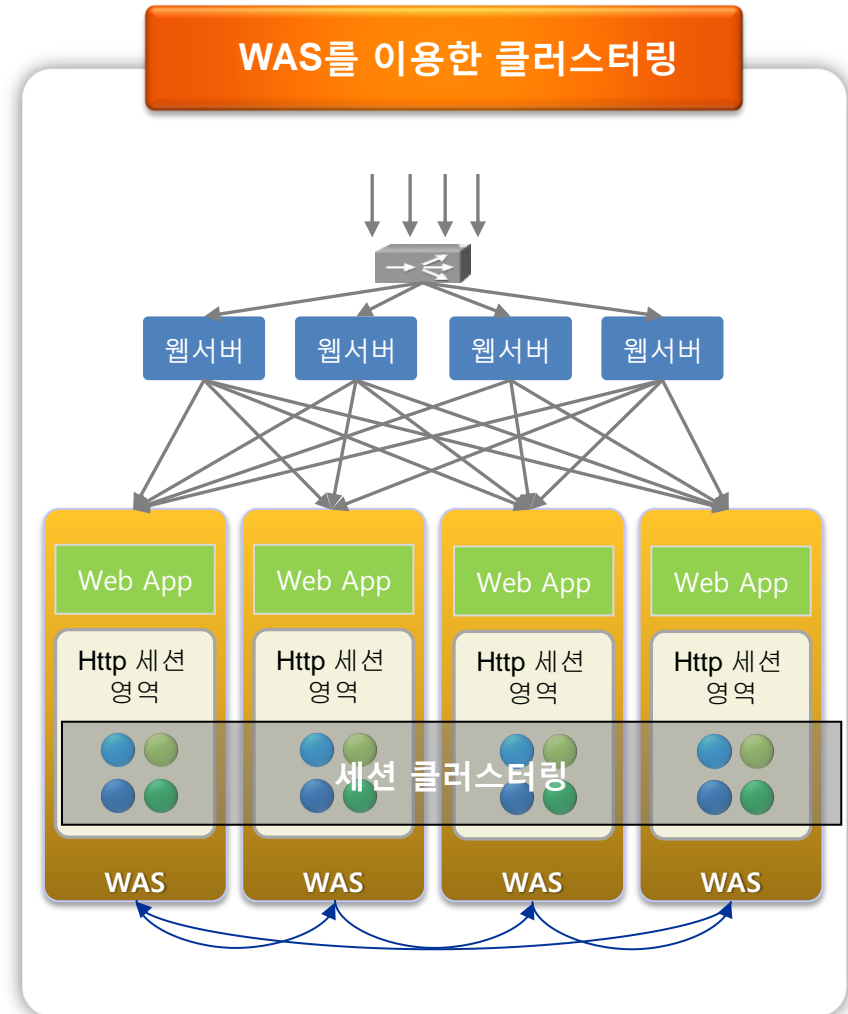


중복로그인 방지나
강제 로그아웃 등 세
션을 통한 보안 강화



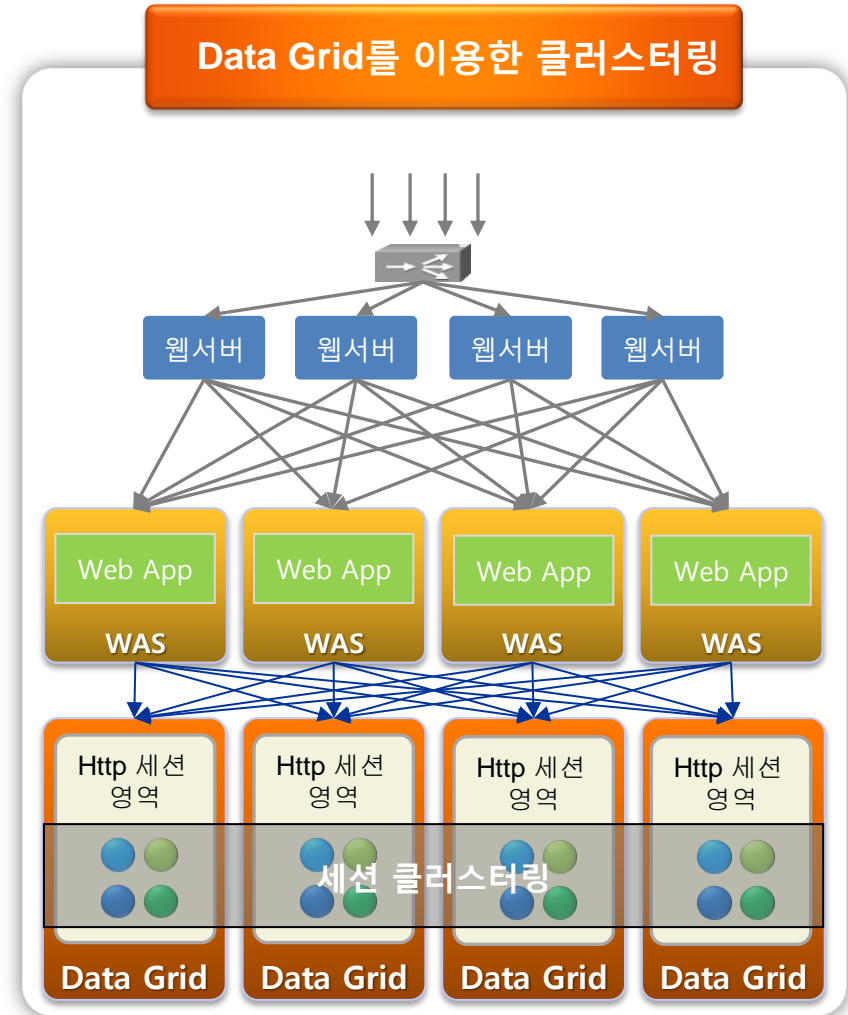
Real Voice

- Http 세션 복제 목적
 - WAS 장애 시 세션정보 보호
 - Distributed (All-to-All 이 아닌 경우) 일 때 로드밸런서의 Sticky 설정
- 문제점
 - WAS 간에 세션 정보를 복제하여 WAS의 대수를 늘려도 세션 정보 저장 영역은 스케일 아웃 불가(Replicated)
 - 세션 정보 저장 영역을 늘리기 위해서는 JVM의 Heap 사이즈를 늘릴 수 있지만 너무 늘리면 "Full GC" 처리 시간 때문에 장애요인 발생
 - 이 종류의 WAS 또는 웹 애플리케이션 간 세션 정보 공유 불가
 - 비효율적인 메모리 사용

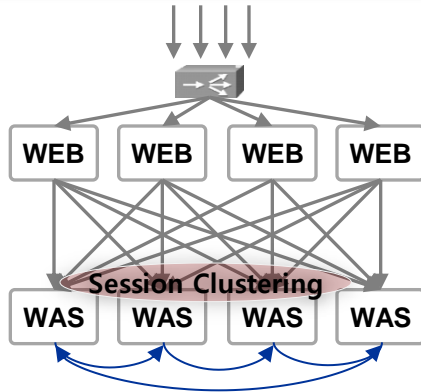


- Data Grid 방식의 이점

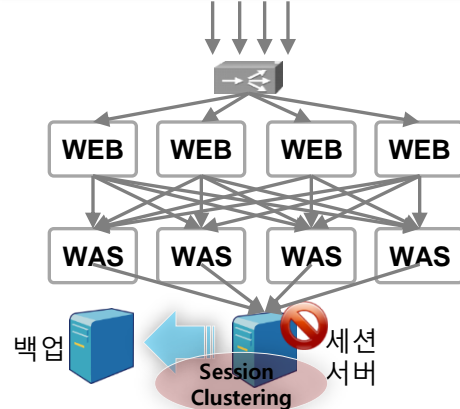
- WAS 장애 발생시 세션 데이터 보호
- 세션 정보를 분산하여 세션 저장영역에 대한 스케일 아웃
- 대용량으로 Http Session 데이터를 사용하는 웹 애플리케이션
- High Volume 웹 애플리케이션 혹은 부적절하게 설계된 Http Session을 사용하는 웹 애플리케이션에 적용
- Heap 사이즈를 크게 하는 것이 아니라 JVM의 프로세스 수를 늘리는 것으로 Full GC의 악영향을 회피하면서 시스템을 확장
 - 유휴 H/W 메모리 활용
 - 이 기종의 AP 서버 간의 세션 정보 공유
 - 웹 애플리케이션간 세션 정보 공유



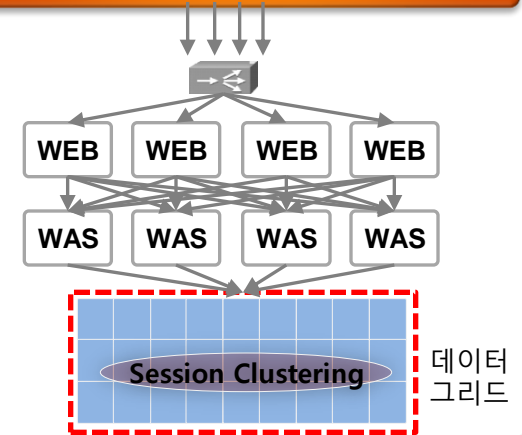
WAS 간 구성



세션 서버 구성



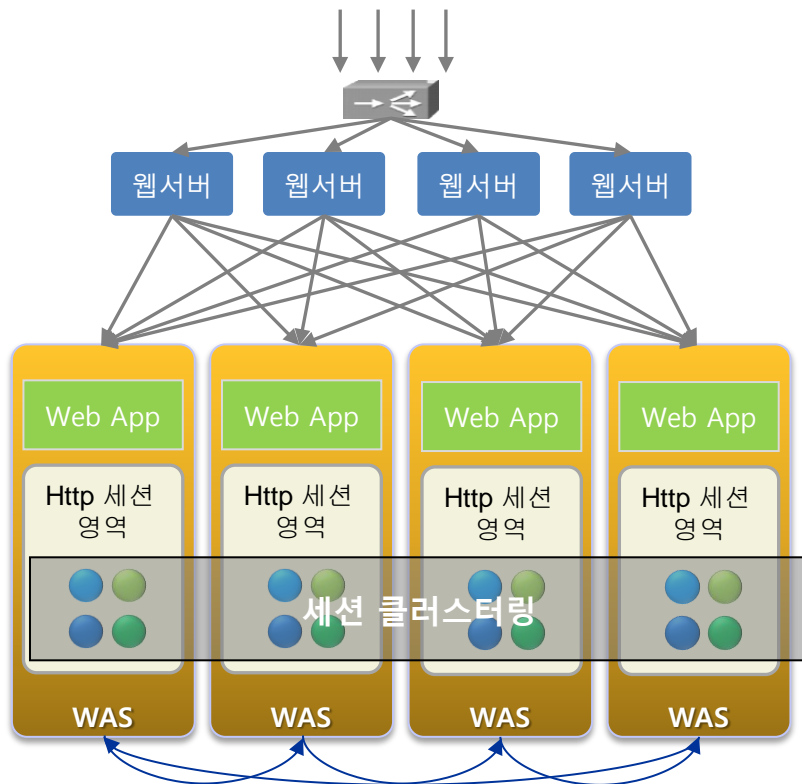
세션 데이터그리드 구성



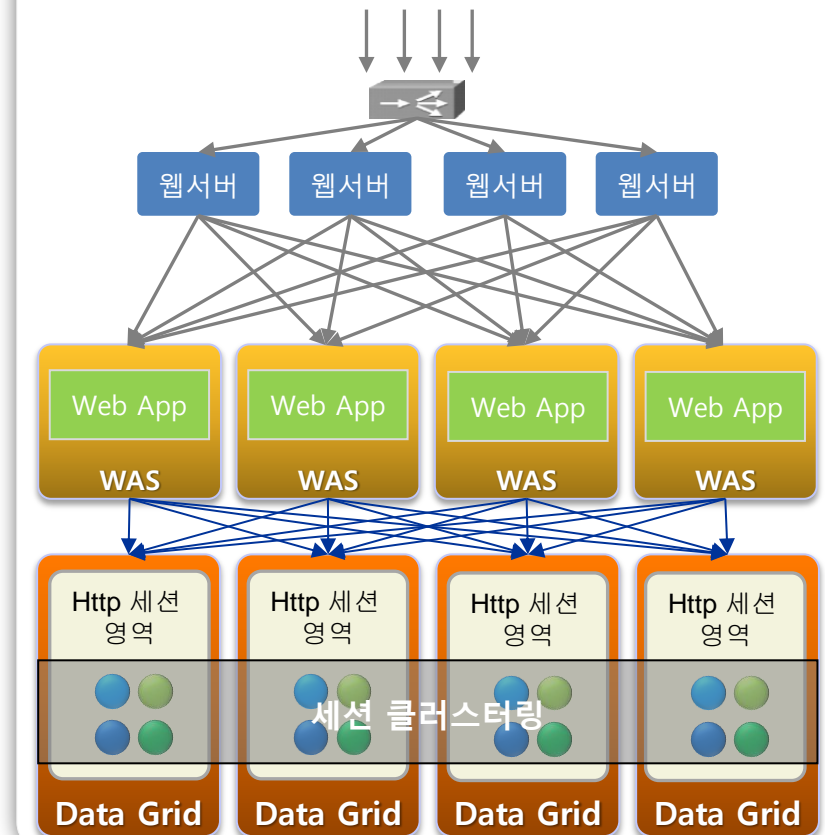
항목	WAS 내장 세션 관리	별도 세션 관리 서버	데이터그리드 기반 세션 관리
구현 방법	<ul style="list-style-type: none"> 세션 데이터별 Primary/Backup 인스턴스를 지정하여 공유 	<ul style="list-style-type: none"> 별도의 세션 서버 운영 	<ul style="list-style-type: none"> 데이터그리드에 세션 정보를 저장하여 운영
장점	<ul style="list-style-type: none"> 별도의 서버와 인프라 없이 가능 	<ul style="list-style-type: none"> 인스턴스간 애플리케이션간 세션 공유 설정이 용이 	<ul style="list-style-type: none"> 인스턴스와 애플리케이션 간 세션 공유 용이 Elastic 확장성과 안정성 부장
단점	<ul style="list-style-type: none"> 세션데이터의 백업 및 동기화 이슈 WAS 인스턴스 장애와 함께 세션 복제의 이슈가 발생 	<ul style="list-style-type: none"> 단일 장애 지점과 별도의 서버 구성에 따른 비용 제한적인 안정성 낮은 성능 	<ul style="list-style-type: none"> 별도의 서버 구성으로 인한 비용 발생 관리 포인트 증가 메모리 기반 고성능
제품	<ul style="list-style-type: none"> JBoss EAP WebLogic, WebSphere 	<ul style="list-style-type: none"> JEUS 	<ul style="list-style-type: none"> JBoss EAP + JBoss Data Grid Coherence*Web

- 세션 정보를 데이터그리드에 저장하여 WAS 의 부하를 감소
- 세션 수가 증가하더라도 데이터그리드를 추가하여 확장

WAS를 이용한 클러스터링



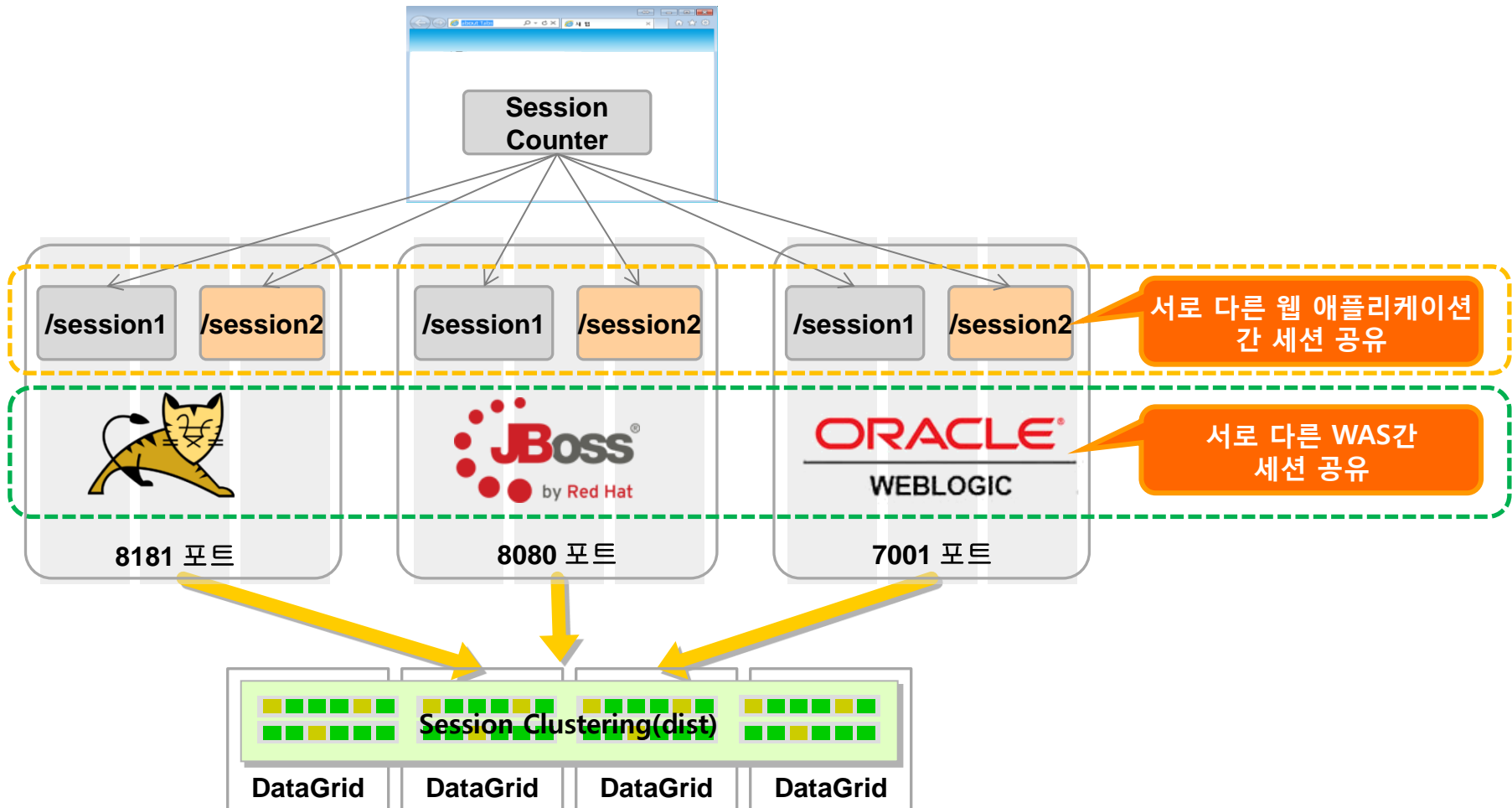
Data Grid를 이용한 클러스터링



A close-up photograph of several pink chrysanthemum flowers. The central flower is in sharp focus, showing its dense, spherical arrangement of small petals. Other flowers are visible in the foreground and background, slightly out of focus. The background is a soft, blurred green, suggesting foliage.

KHAN [session manager] 주요 특징 및 기능

- 이 기종의 WAS 와 웹 애플리케이션 간의 세션 정보를 공유
- 지원하는 WAS - WebLogic Server / Tomcat / JBoss





Apache Tomcat에서 세션 클러스터링이 필요할 때



서로 다른 웹 애플리케이션간의 세션 공유를 원할 때



웹 클러스터링을 안정적인 환경에서 운영하고 싶을 때

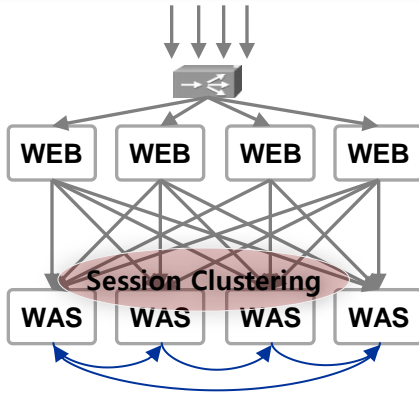


중복 로그인 방지 기능이 필요할 때

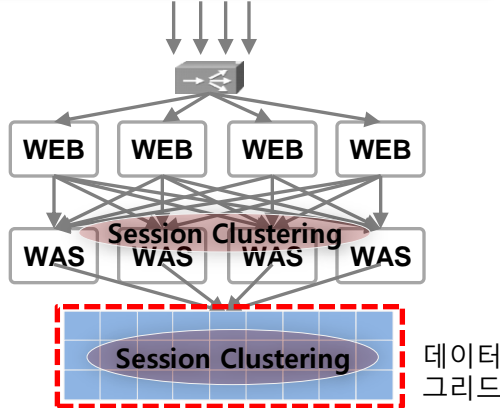


WebLogic Standard Ed.에 세션 클러스터링이 필요할 때

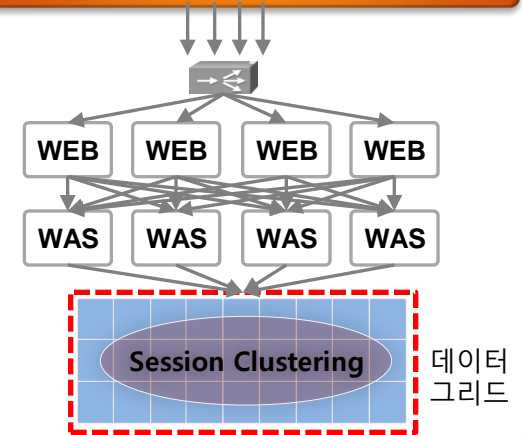
WAS 간 구성



WAS간 + 데이터 그리드 구성

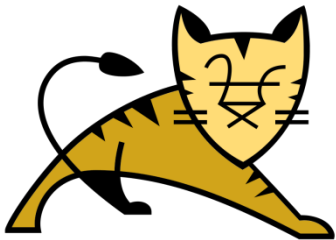


세션 데이터그리드 만 구성



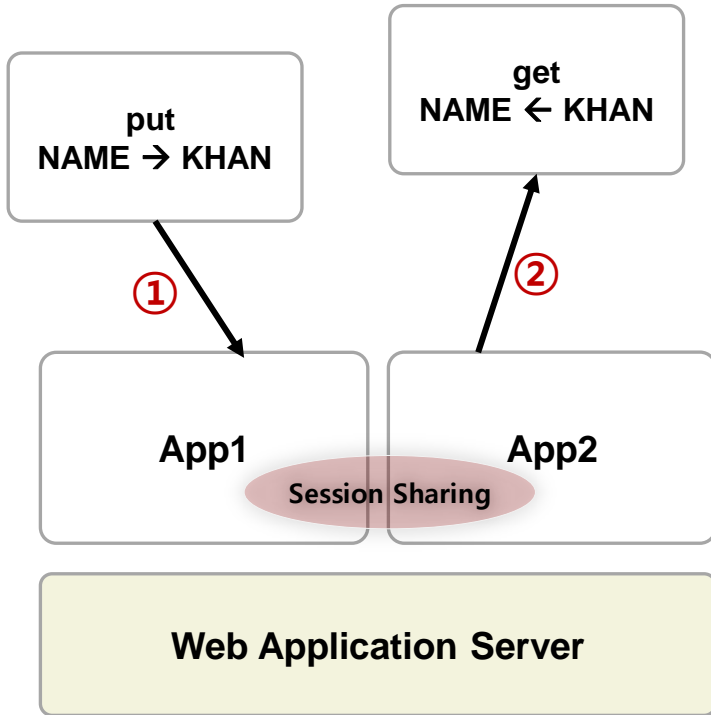
항목	WAS간 세션 관리	별도 세션 관리 서버	데이터그리드 기반 세션 관리
구현 방법	<ul style="list-style-type: none"> 세션데이터별 Primary/Backup 인스턴스를 지정하여 공유 	<ul style="list-style-type: none"> WAS간 세션을 데이터를 일부 공유하고 나머지 데이터는 데이터 그리드에 저장하여 운영 	<ul style="list-style-type: none"> 데이터그리드에 세션 정보를 저장하여 운영
장점	<ul style="list-style-type: none"> 별도의 서버와 인프라 없이 가능 	<ul style="list-style-type: none"> 인스턴스와 애플리케이션간 세션 공유 용이 확장성과 안정성 보장 성능 보장 	<ul style="list-style-type: none"> 인스턴스와 애플리케이션 간 세션 공유 용이 Elastic 확장성과 안정성 보장
단점	<ul style="list-style-type: none"> 세션데이터의 백업 및 동기화 이슈 WAS 인스턴스 장애와 함께 세션 복제의 이슈가 발생 	<ul style="list-style-type: none"> 관리포인트 증가 별도의 서버 구성이 필요함 	<ul style="list-style-type: none"> 별도의 서버 구성으로 인한 비용 발생 관리 포인트 증가 네트워크로 인한 성능 저하

- Servlet 표준을 사용하기 때문에 코드 변경은 없음
- Servlet 표준 web.xml에 설정 추가

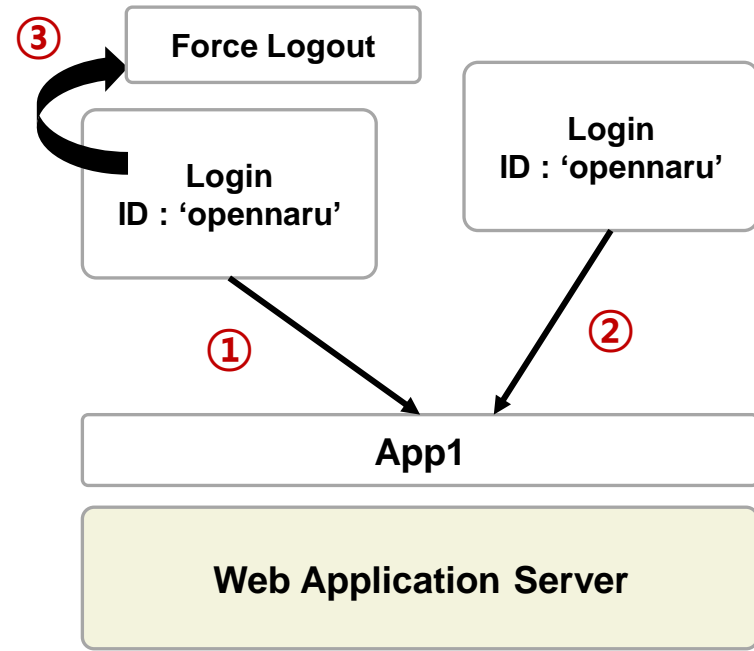


계속 테스트, 추가 중...

서로 다른 웹 애플리케이션 간 세션 공유



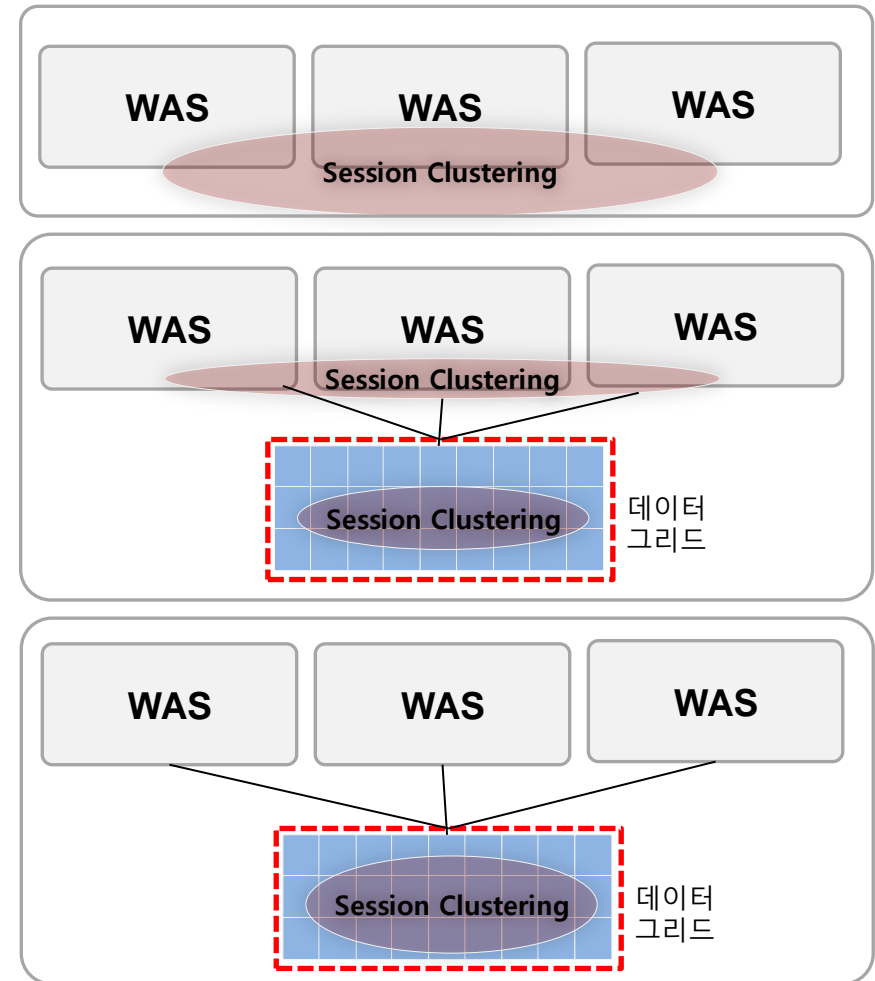
중복 로그인 방지



세션 타입아웃 제어

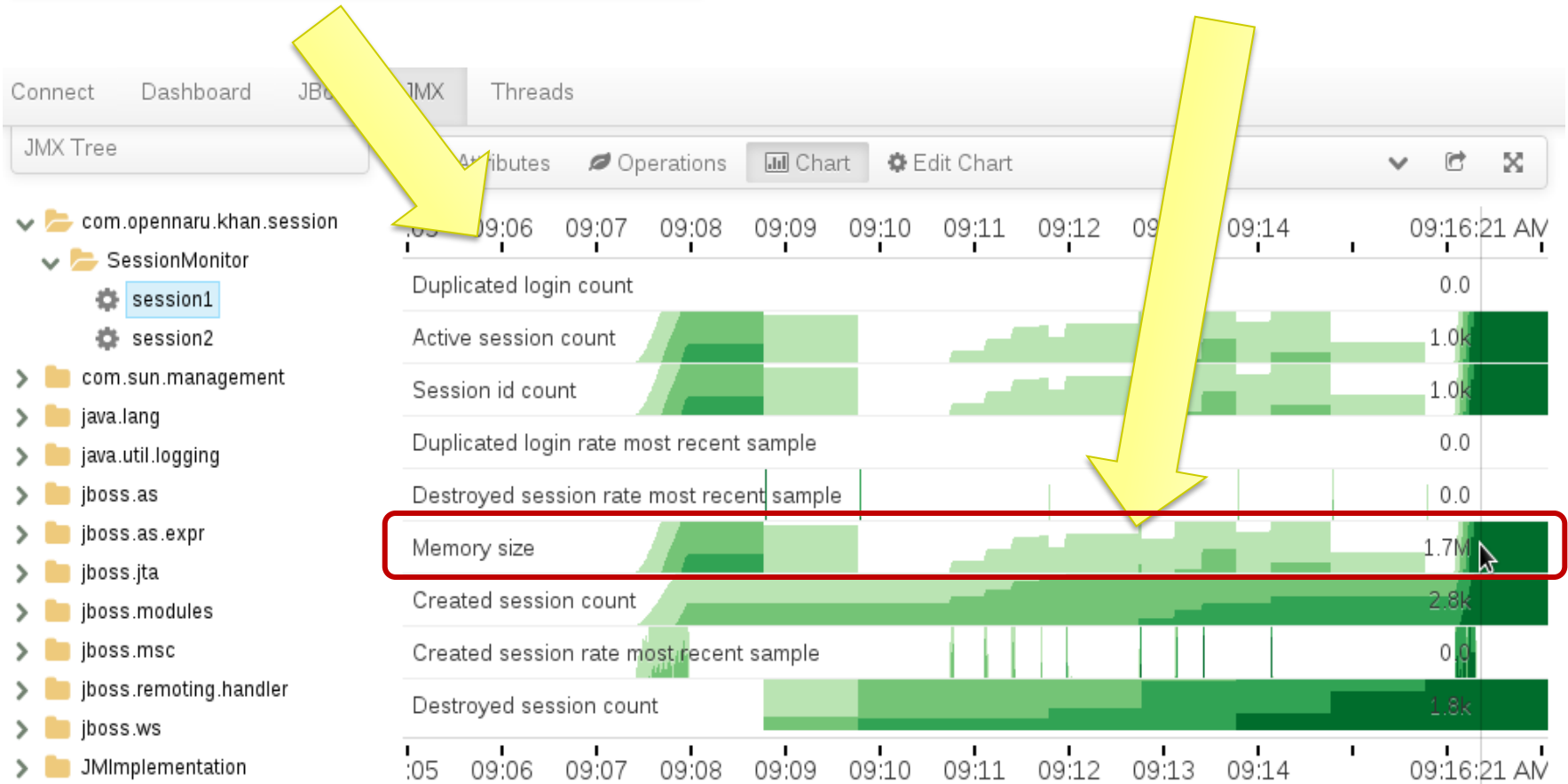


다양한 토폴로지 지원



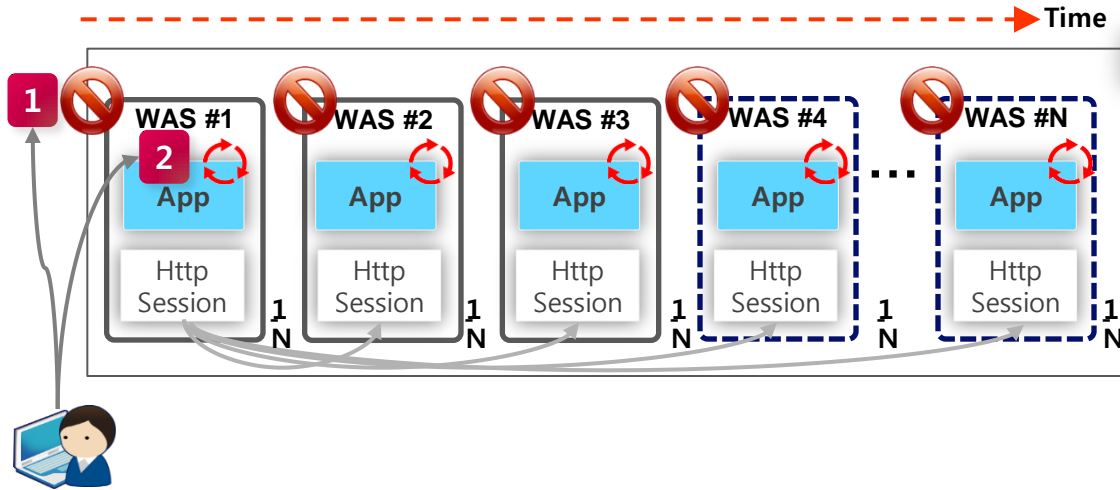
세션 생성, 소멸, 중복 로그인 수 모니터링

세션의 메모리 사이즈 모니터링



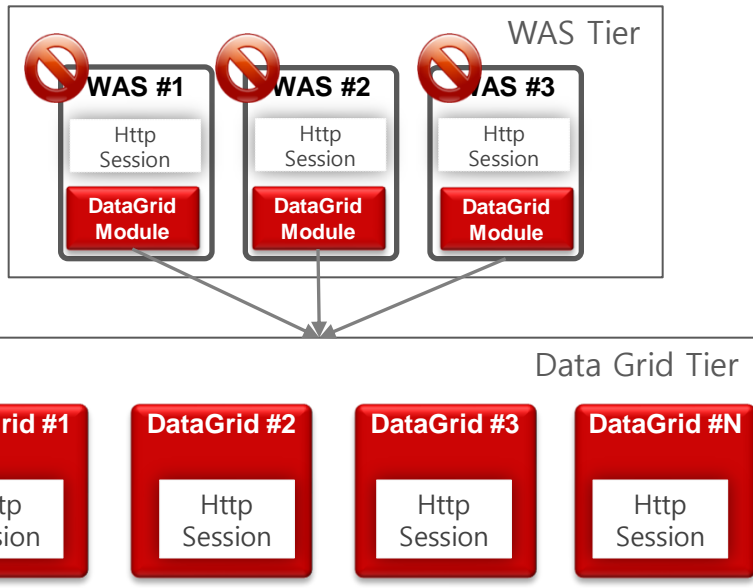


KHAN [session manager]
효율적인 세션 관리 방안



WAS 관리 작업 시 세션 복제 부하 발생

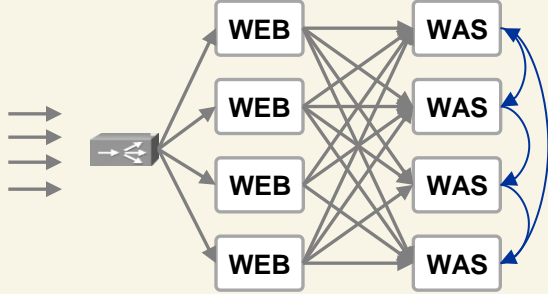
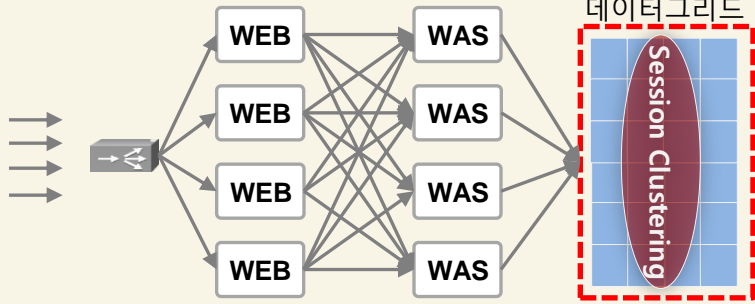
- 주요 WAS 관리 작업시 세션 복제 부하 발생
1. WAS 인스턴스 재 시작 시 세션 복제 발생
 - WAS 인스턴스 재 시작시 소유하고 있던 세션을 다른 인스턴스에 백업
 2. 애플리케이션 재배포 시 세션 복제 발생
 - 웹 애플리케이션 재배포시 세션을 백업하고 새로운 세션을 생성함



WAS 와 분리하여 그리드에서 세션관리

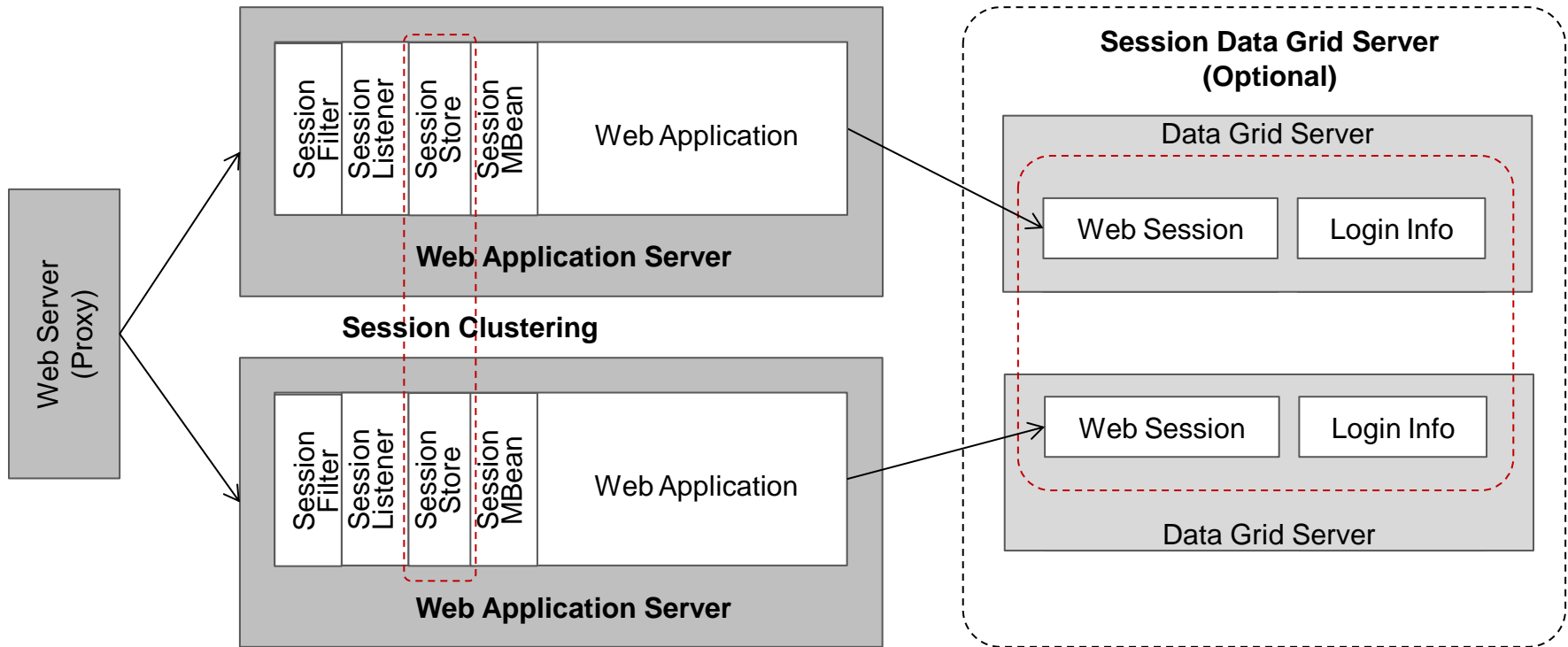
- WAS 관리와 성능 향상
 - 세션 동기화와 복제를 배제하여 성능향상
 - 세션 데이터에 의한 OOM 해결
 - 전체 WAS 장애 시에도 세션 데이터 유지
- 효율적인 세션 공유
 - 서로 다른 애플리케이션 세션 공유
 - 중복 로그인 방지
 - 세션 관리

WAS 클러스터링 vs. KHAN [session manager]

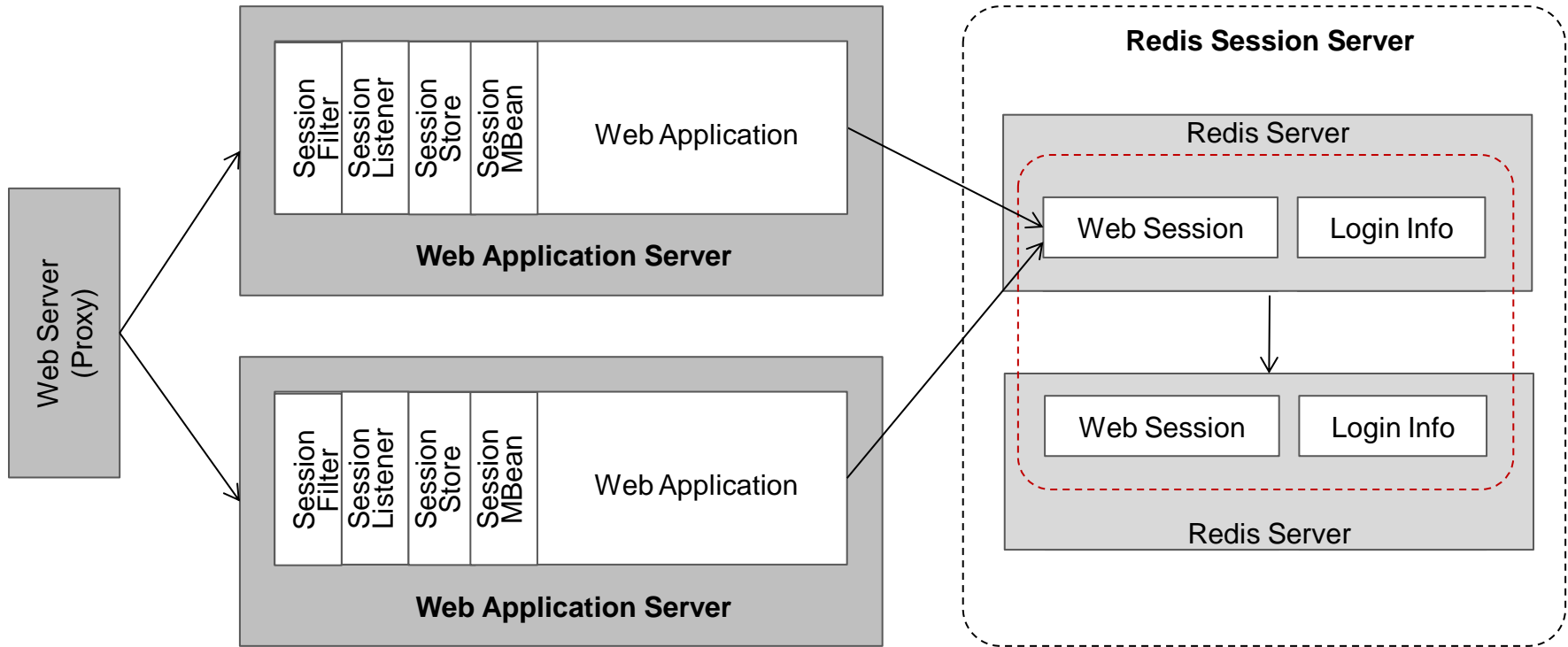
항목	WAS 세션 클러스터링	KHAN [session manager] 세션 클러스터링
아키텍처		
안정성	<p>○</p> <ul style="list-style-type: none"> 해당 인스턴스와 다른 인스턴스에 세션 데이터를 복제하고 동기화하여 관리 과도한 세션 사용시 OOM 메모리 장애 발생 세션데이터에 의한 GC가 장시간 발생 	<p>●</p> <ul style="list-style-type: none"> 세션 데이터를 데이터그리드에 저장하고 공유하기 때문에 거래가 증가되어도 가용성을 유지한 채 안정적으로 분산 관리 가능 WAS 노드 장애 시 상호 공유된 세션 정보를 통해 세션유실방지
성능	<p>◐</p> <ul style="list-style-type: none"> WAS 인스턴스 관리 세션 복제와 동기화에 따른 성능 이슈 	<p>●</p> <ul style="list-style-type: none"> 세션 복제나 동기화 과정이 생략되어 신속한 WAS 관리 작업이 가능
확장성	<p>◐</p> <ul style="list-style-type: none"> WAS 인스턴스 확장 	<p>●</p> <ul style="list-style-type: none"> 애플리케이션 메모리와 세션 메모리를 분리하여 예측 가능한 확장성 보장
세션 관리	<p>○</p> <ul style="list-style-type: none"> WAS 인스턴스 재시작시 세션 동기화와 복제 애플리케이션 배포시 세션 동기화와 복제 애플리케이션 별 세션 정보 관리 	<p>●</p> <ul style="list-style-type: none"> WAS 인스턴스 재 시작시 세션 복제 작업 제거 애플리케이션 재배포 시 세션 복제 작업 제거 복수의 애플리케이션 간 세션 정보 공유




KHAN [session manager]
시스템 아키텍처



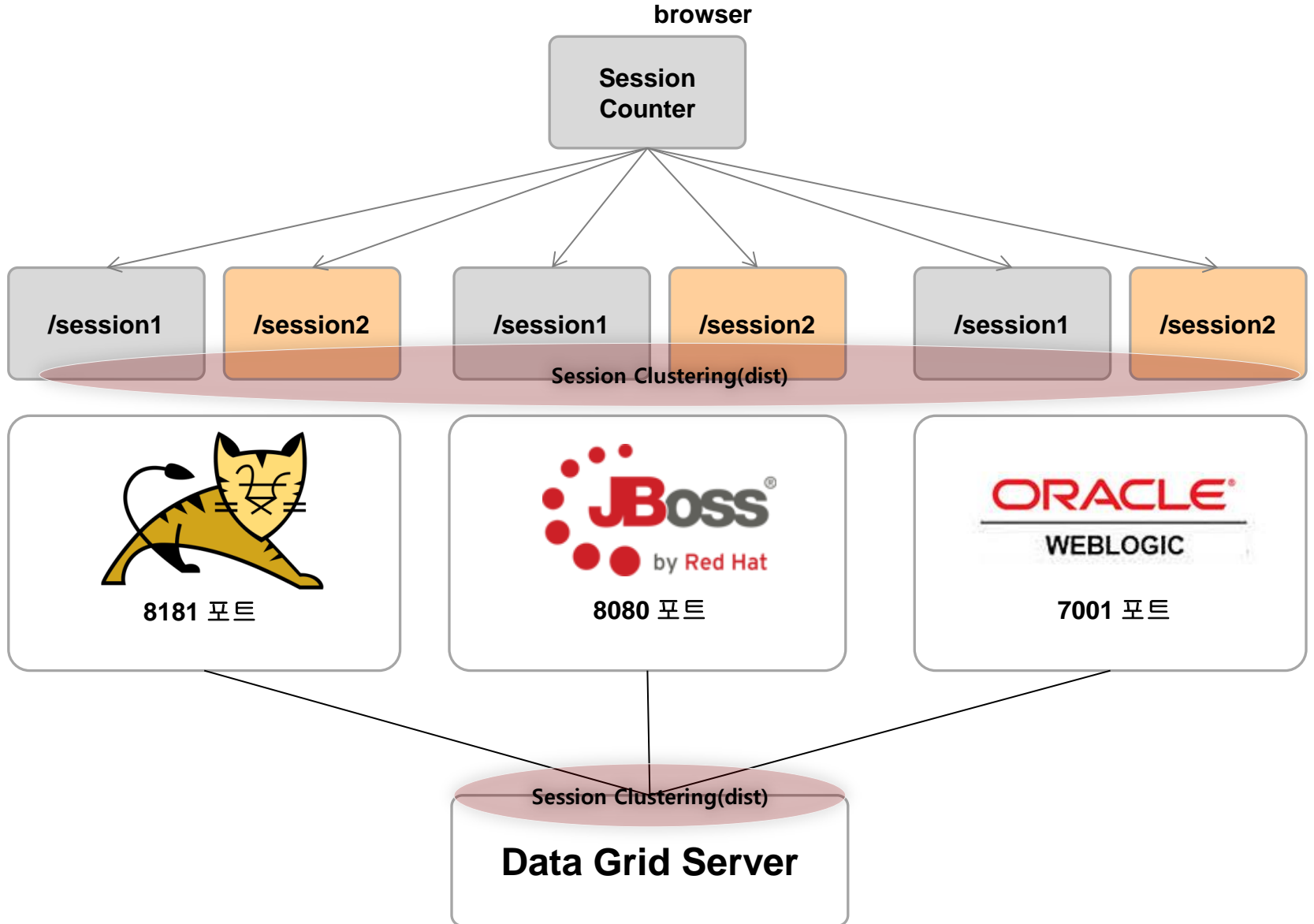
KHAN [session manager] 구조도



KHAN [session manager] 구조도



'웹 세션 클러스터링'
KHAN [session manager] Test

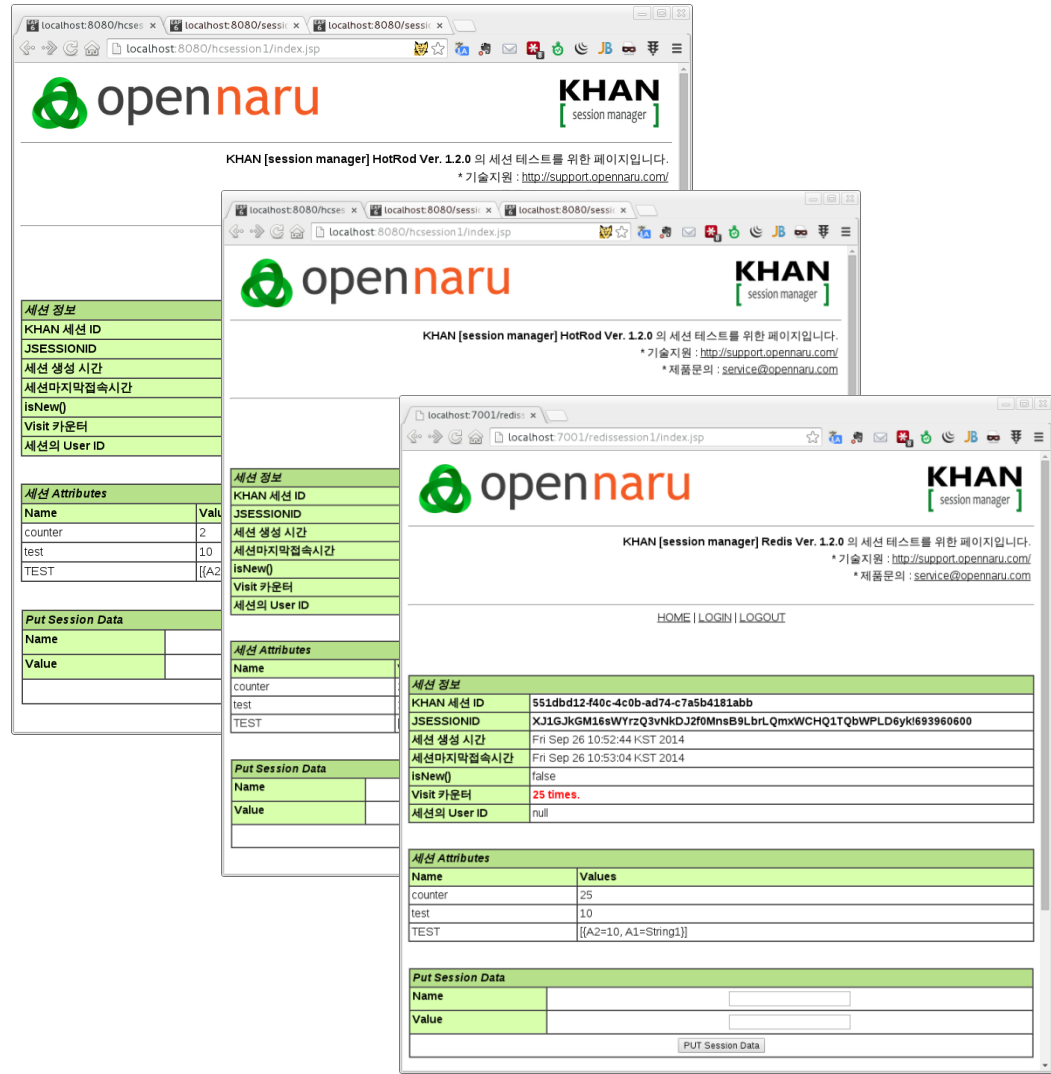


- 방식

- Library Mode 방식
- Hotrod 방식
- Redis Store 방식

- 테스트 애플리케이션

- 세션 ID 정보 표시
- 세션의 메타데이터 표시
- 세션 카운터 증가 확인
- 로그인 정보 표시
- 세션 리스너 테스트
- 세션에 임의의 값 추가



The image displays three browser windows showing the KHAN session manager interface. The top window shows the HotRod mode, the middle window shows the Redis mode, and the bottom window shows the session data for the Redis mode. Each window includes a header with the opennaru logo and KHAN session manager branding, and a main content area with session information and a 'Put Session Data' form.

HotRod Mode Session Data:

세션 정보	
KHAN 세션 ID	
JSESSIONID	
세션 생성 시간	
세션마지막접속시간	
isNew()	
Visit 카운터	
세션의 User ID	

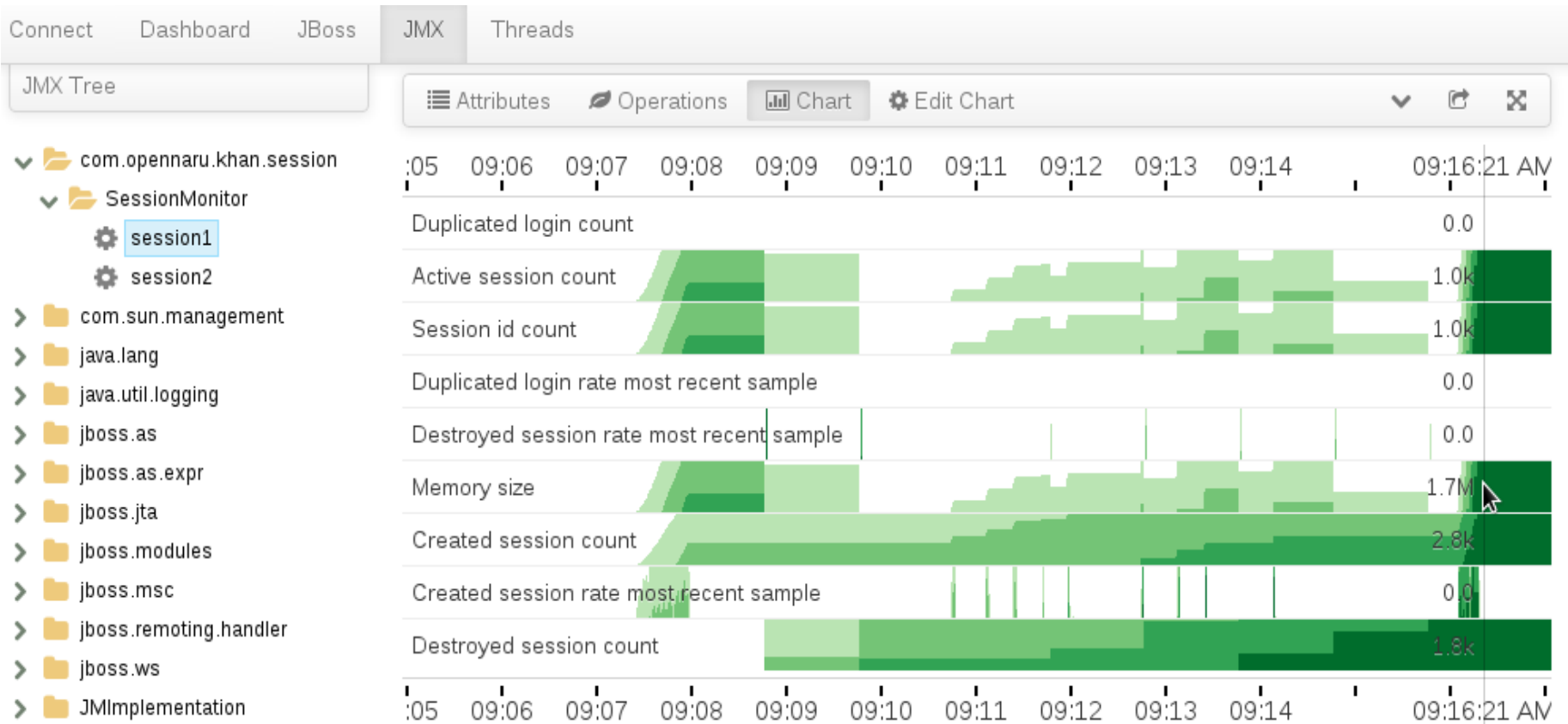
Redis Mode Session Data:

세션 정보	
KHAN 세션 ID	551dbd12-f40c-4c0b-ad74-c7a5b4181abb
JSESSIONID	XJ1GJkGM16sWYrzQ3vNkDj2f0MnsB9LrLQmxWCHQ1TqbWPLDeyk69396000
세션 생성 시간	Fri Sep 26 10:52:44 KST 2014
세션마지막접속시간	Fri Sep 26 10:53:04 KST 2014
isNew()	false
Visit 카운터	25 times.
세션의 User ID	null

Redis Mode Session Attributes:

세션 Attributes	
Name	Values
counter	25
test	10
TEST	[[A2=10, A1=String1]]

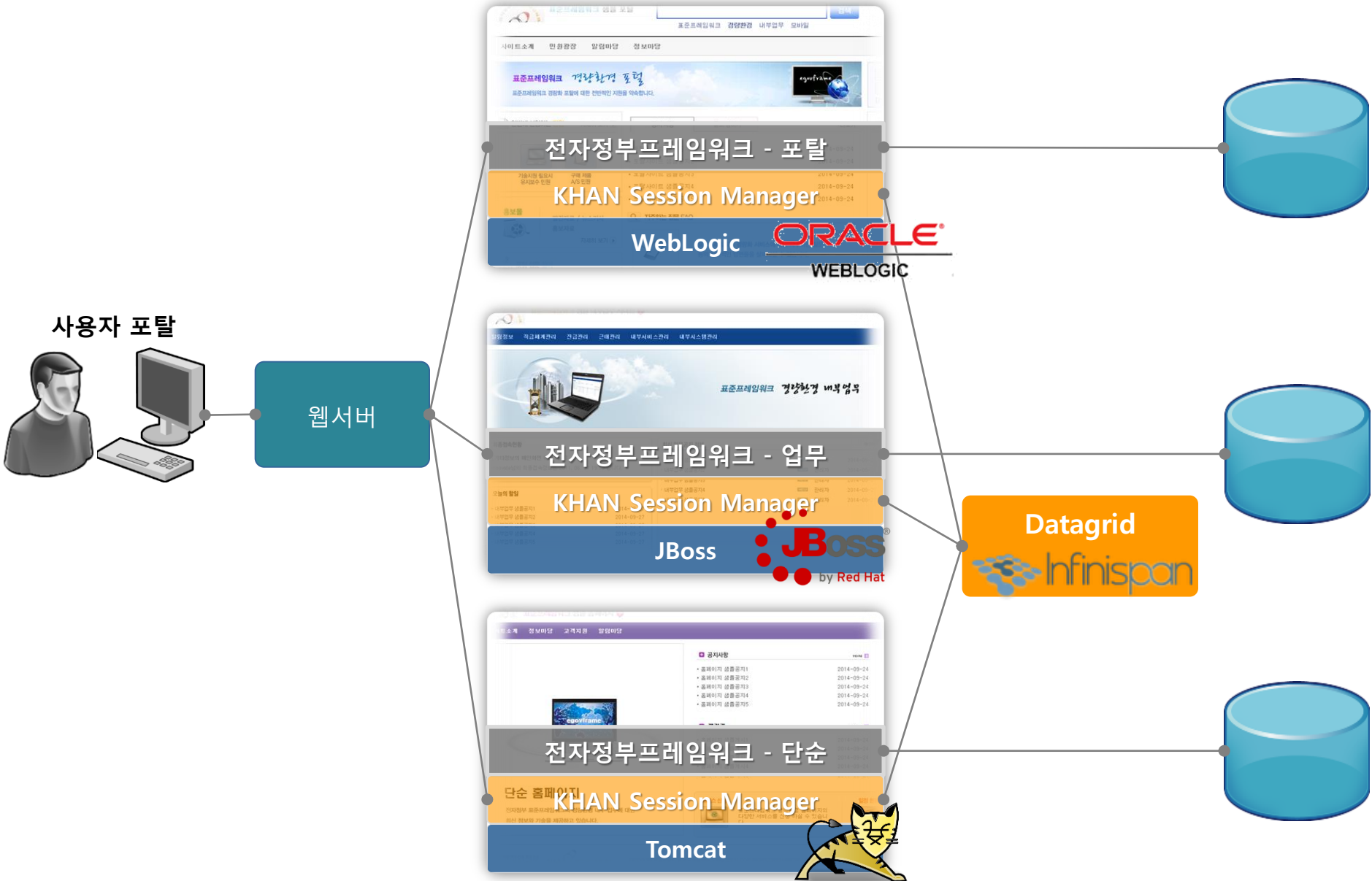
- 샘플 애플리케이션에 대한 부하 테스트
- 부하 테스트 상황에서 KHAN [session manager]가 제공하는 MBean의 정보 모니터링



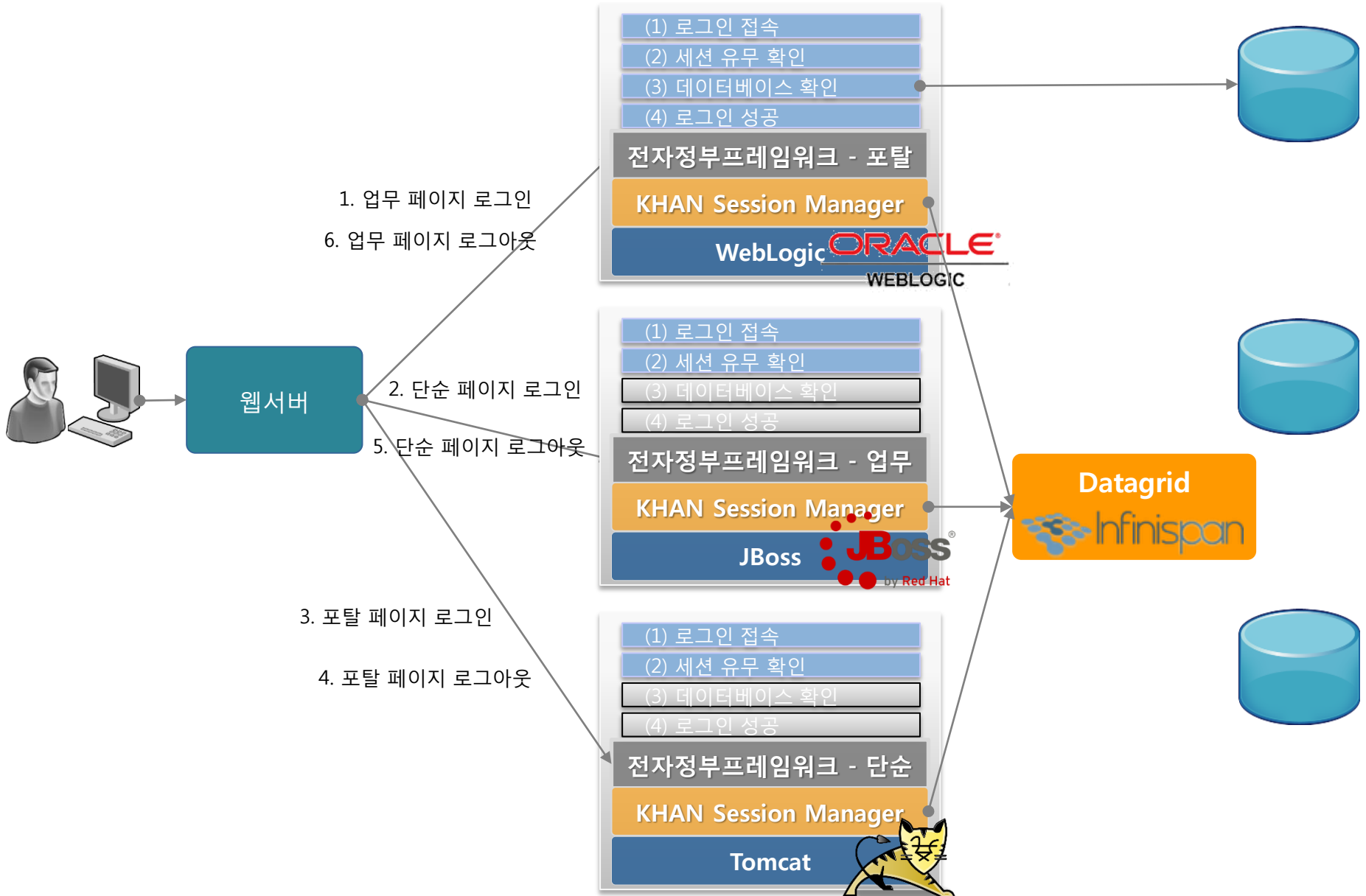


'웹 세션 클러스터링'
KHAN [session manager] Demo

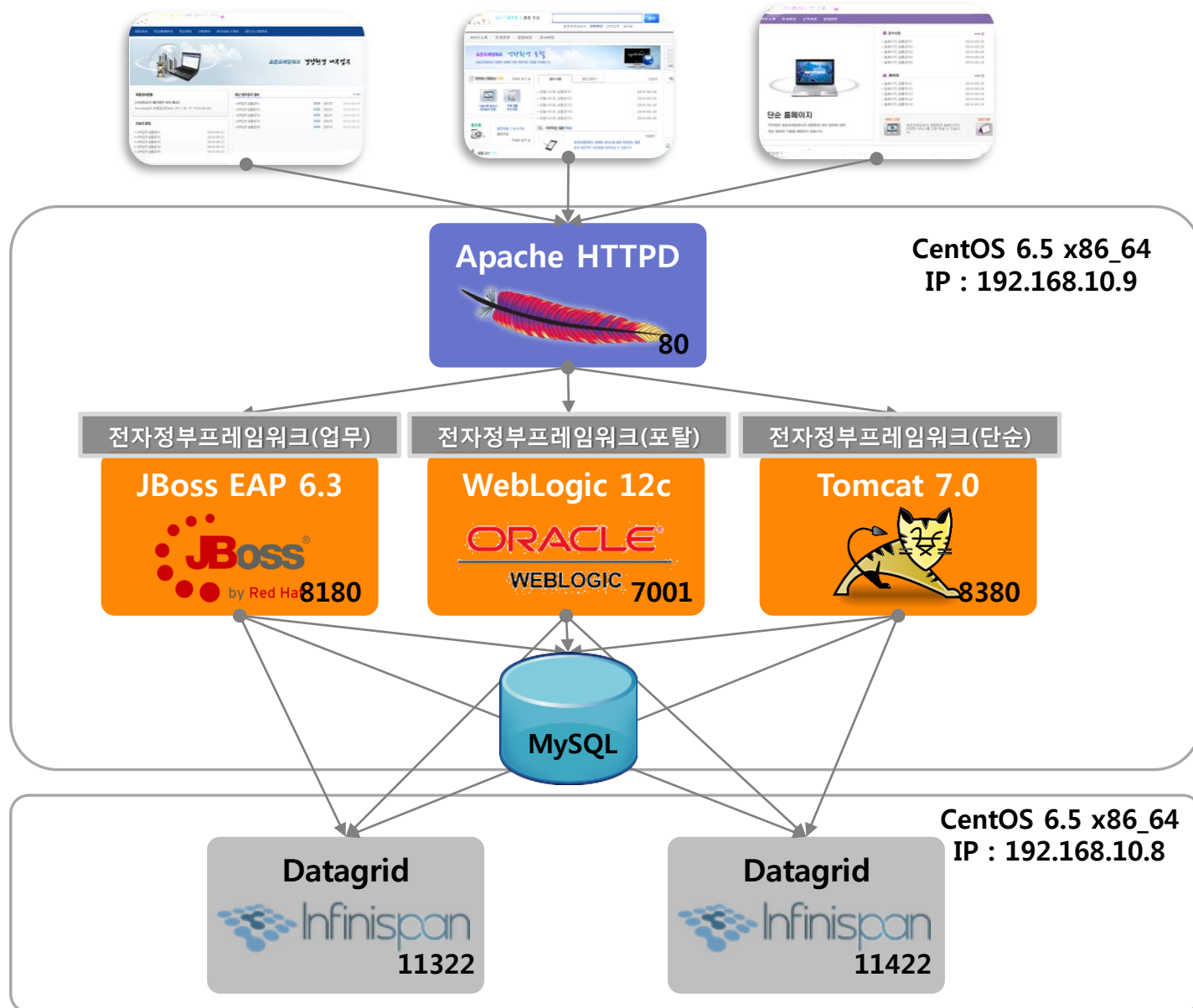
KHAN [session manager] 데모 아키텍처



KHAN [session manager] 데모 시나리오



KHAN [session manager] 시스템 구성도



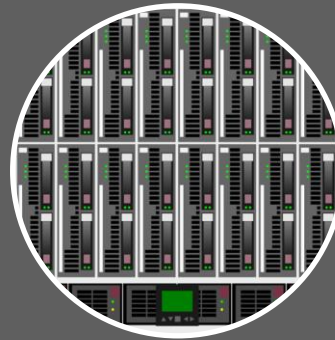
JBoss EAP, Tomcat, JBoss Data Grid 설치도 지원



OS 만 설치되어 있으면
수분 이내에 설치 환경을
테스트하고 웹서버
와 WAS서버를 설치하
고 즉시 서비스할 수 있
는 환경 제공



미들웨어 전문가가 아니
어도 전문가 수준의 시
스템 튜닝이나 난이도
높은 구성을 할 수 있도
록 기능 제공



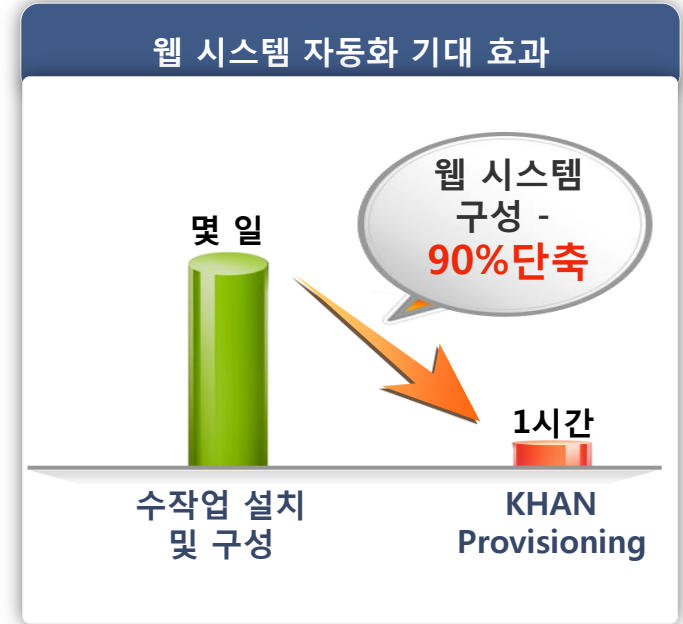
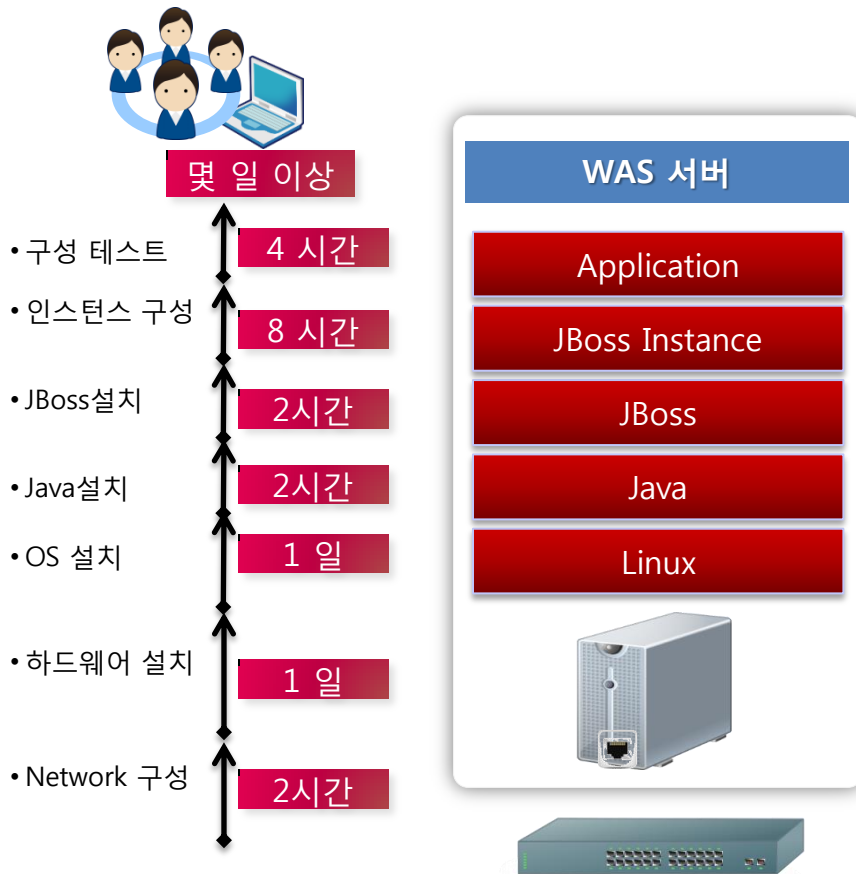
서버 구성에 대한 정보
만 입력하면 한대에서
수 십대까지 규모에 상
관없이 자동으로
웹시스템 운영환경을
구성



웹시스템 설치/구성
보고서를 시스템에 맞
게 자동으로 생성하여
개발팀과 운영 팀에게
제공

수 분 내 튜닝된 웹 서버와 WAS 서버로 웹 시스템 구축

- ✓ 웹시스템 구성하는데 “ 몇 일? ”
- ✓ “ 튜닝은? ”
- ✓ “ 설치보고서는? ”



“1 시간 이내”

- 웹 시스템 인스턴스 자동 구성
- 기본 튜닝
- 설치 보고서

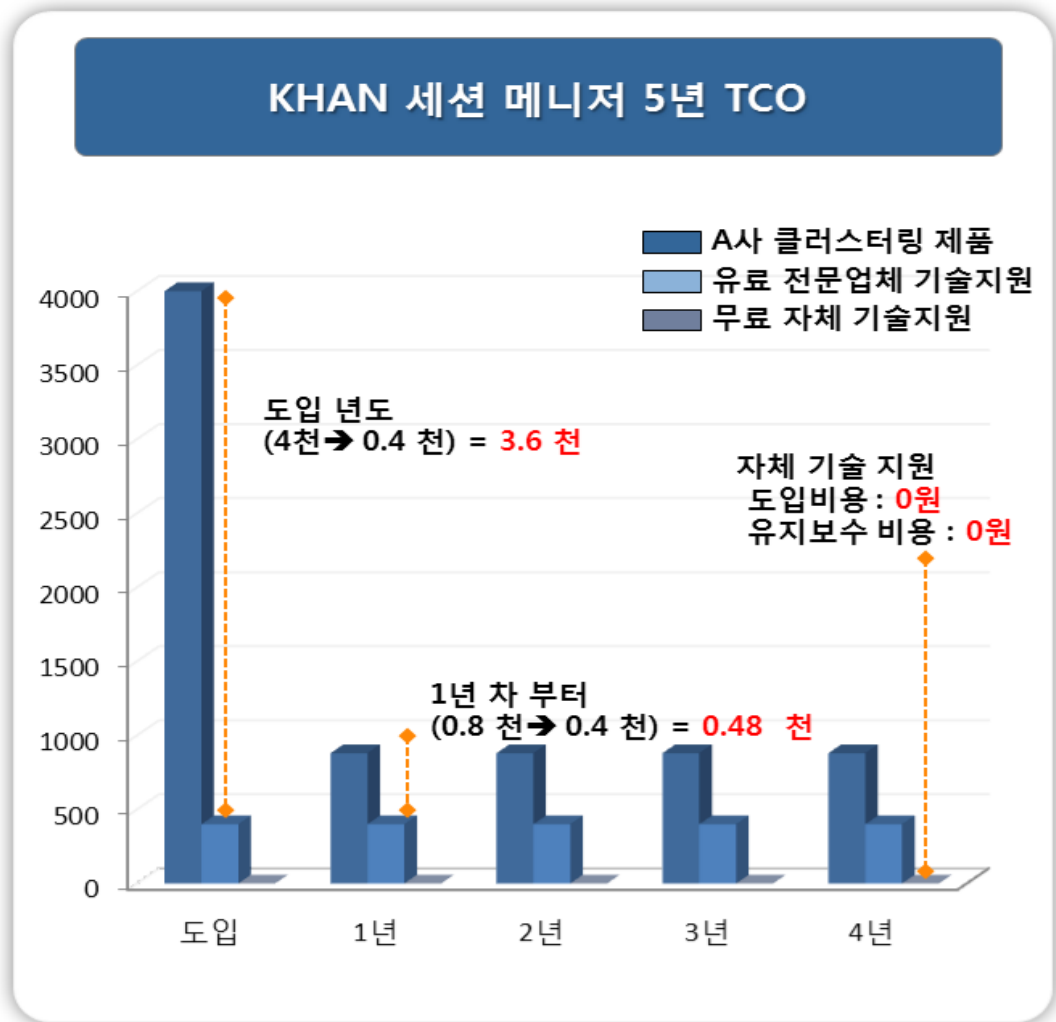
- 1) 업무에서 로그인 → 포탈/홈페이지에 접근하면 로그인 되어 있음
- 2) 포탈에서 로그아웃 → 홈페이지/업무를 Refresh하면 로그아웃 되어 있음



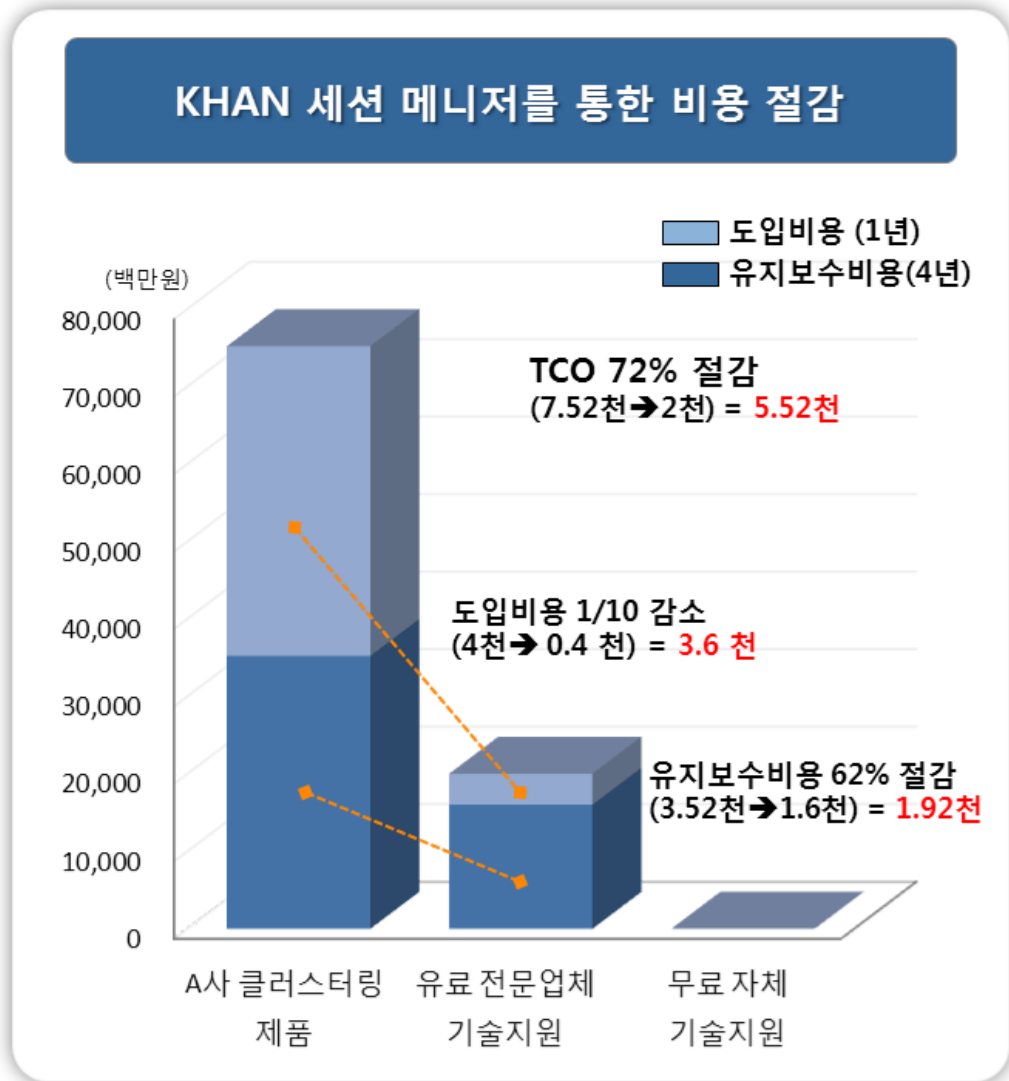


오픈 소스를 통한 비용절감 효과

- 시중에 판매되고 있는 A사의 클러스터링을 제품의 소비자 실 판매가를 기준
- 4 Core 기준
- 오픈소스를 자체 기술로 활용할 수 있음



- 무료 자체 기술로 사용할 수 있음
- 전문업체 기술지원을 받아도 72%이상 절감 효과



- KHAN [session manager]를 이용한 안정적인 세션 클러스터링 환경 구축
- KHAN [provisioning]을 통한 구축 시간 감소
 - JBoss EWS, JBoss EAP, JBoss Data Grid 116개 머신
 - 설치/구성/튜닝 → 3시간 이내 완료
- 환경 자동 구성
 - 배포환경, 테스트 환경 자동 구성
- 장애 포인트 최소화

제 품	머신 수	인스턴스 수
JBoss EWS – 웹서버	44	44
JBoss EAP – WAS	56	108
JBoss Data Grid	16	34
Total	116	186

감사합니다.
Opennaru

