openmaru Cluster

OPENMARU Cluster 소개

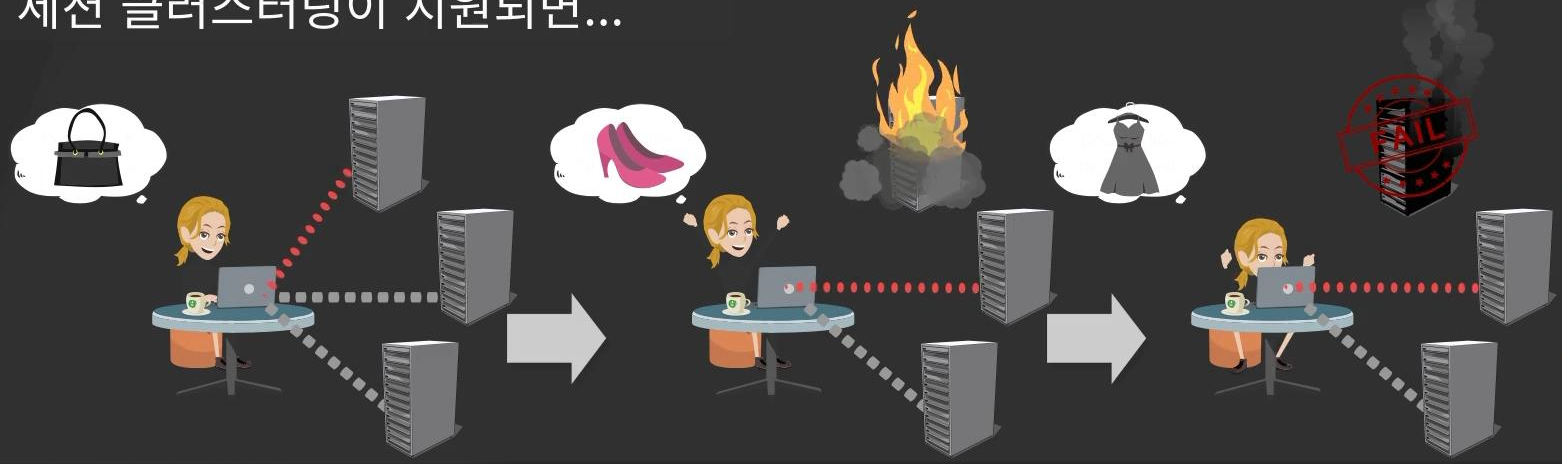
(WAS Session Clustering Solution)

세션 클러스터링이 지원되지 않으면...

openmaru Cluster

서비스 만족도 하락, 생산성 감소 (매출감소 등)

세션 클러스터링이 지원되면...



고가용성 확보 및 무정지 시스템 구현

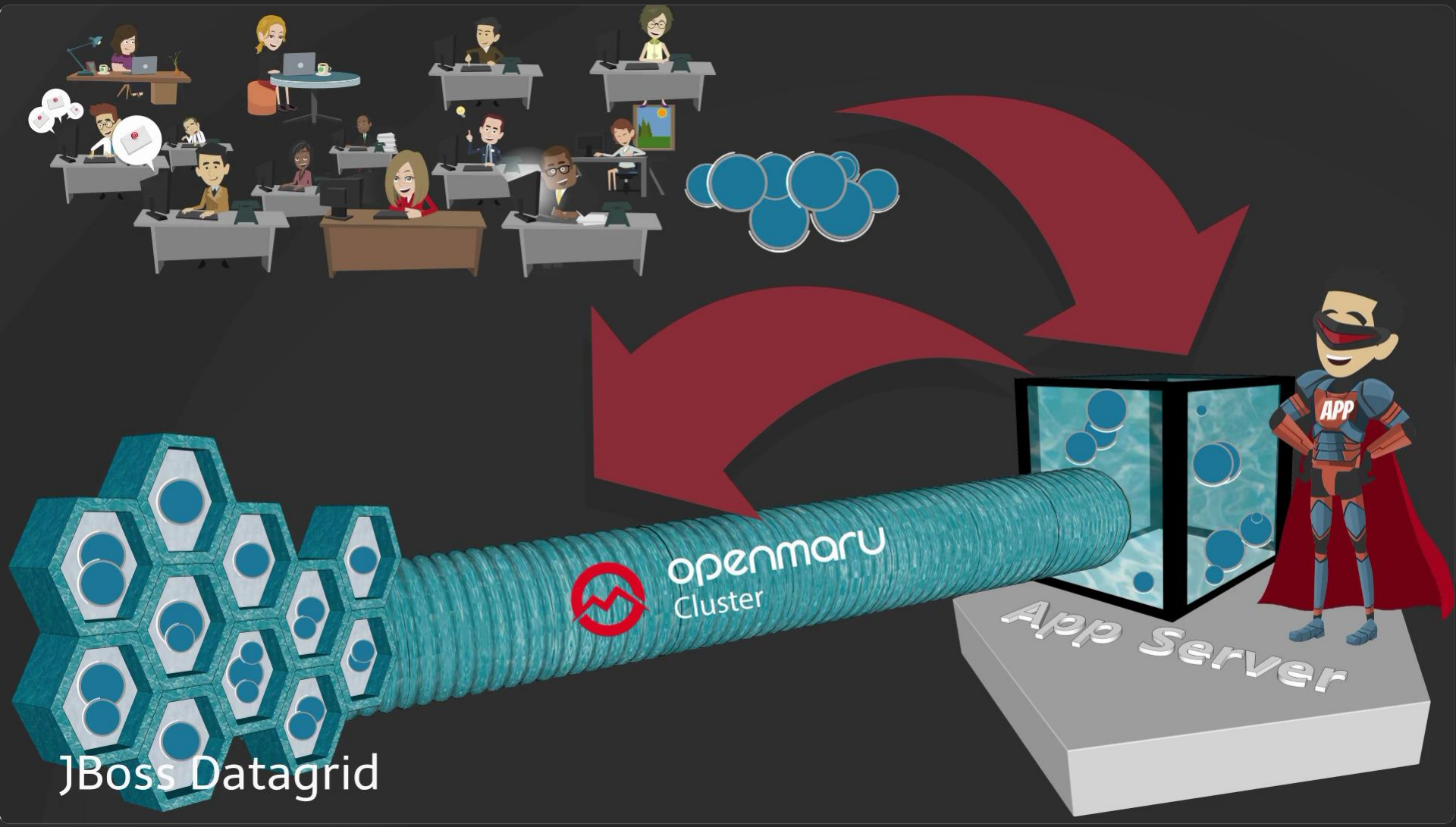
세션 클러스터링 이슈

대규모 사용자 접속에 따른 세션 이슈

대규모 사용자 접속

WAS 메모리 이슈 발생

OPENMARU Cluster overview



WAS 클러스터링 이슈

openmaru Cluster

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 복제에 따른 부하 발생	애플리케이션간 세션 공유를 통한 싱글 로그온 구현	중복로그인 방지나 강제 로그아웃 등 세션을 통한 보안 강화
			

Real Voice

WAS Advanced Clustering

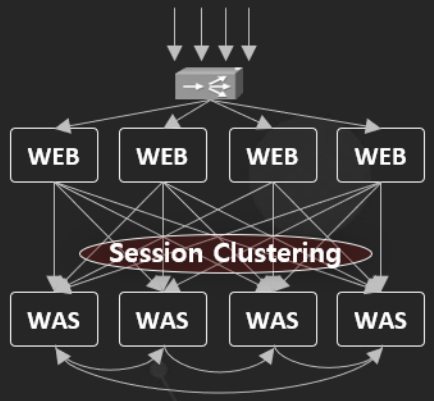
OPENMARU Cluster 배경

KHAN apm gbw

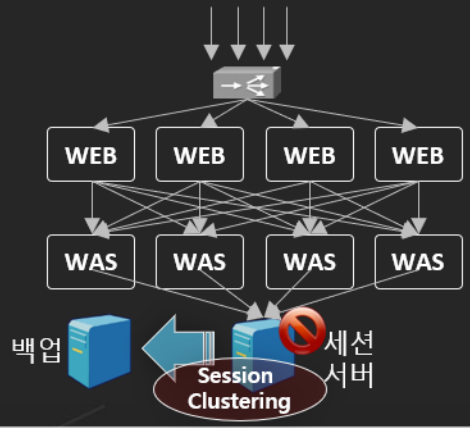
세션 클러스터링 구성 방식 별 차이점

openmaru Cluster

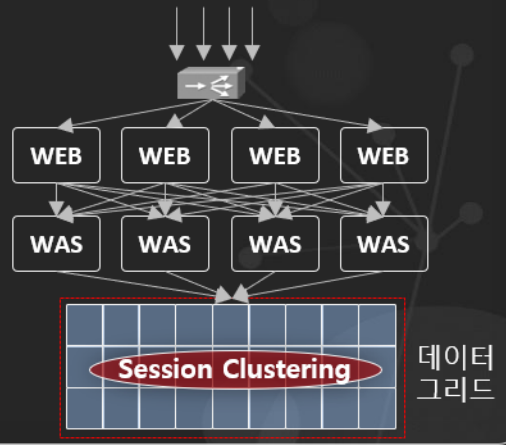
WAS 내장 세션 관리



별도 세션 관리 서버



데이터그리드 기반 세션 관리



항목	WAS 내장 세션 관리	별도 세션 관리 서버	데이터그리드 기반 세션 관리
구현 방법	• 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 부장
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
제품	• JBoss EAP • WebLogic, WebSphere	• JEUS	• JBoss EAP + JBoss Data Grid • Coherence*Web

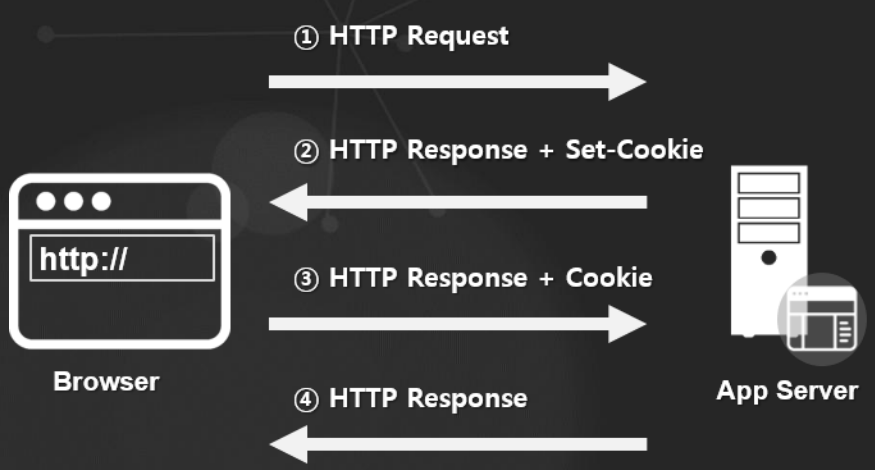
웹 페이지간 정보 유지 방안 - Cookie vs. Session

openmaru Cluster

- http 서버는 사용자의 연결을 계속 유지하지 않는 방식으로 처리됨
- 사용자의 요청에 대한 응답을 한 후 연결을 해제
- 클라이언트와의 정보 유지를 위해 Cookie와 Session을 사용

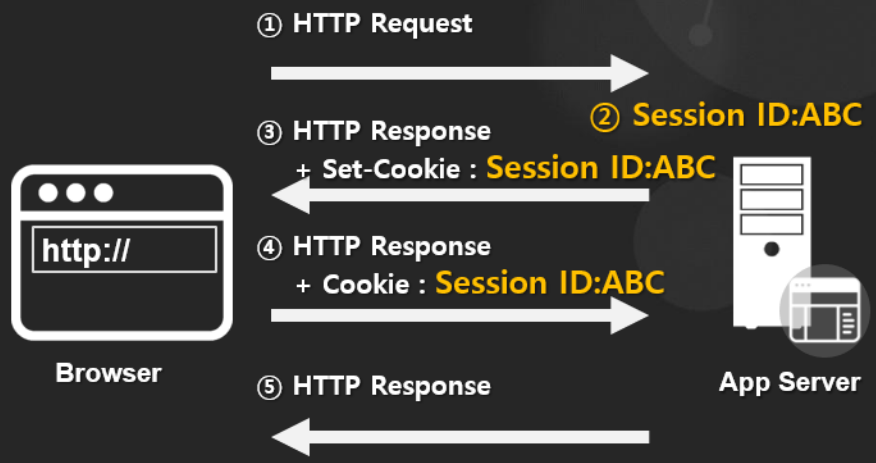
Cookie

- 사용자 브라우저에 저장되어 요청 시 헤더에 포함
- 클라이언트당 최대 20개 저장 가능
- 쿠키는 이름, 값, 유효기간, 도메인, 경로 등으로 구성
- 사용자가 쿠키 허용 안 할 경우 사용할 수 없음
- 하나의 쿠키에 최대 4K 이상 저장 불가능



Session

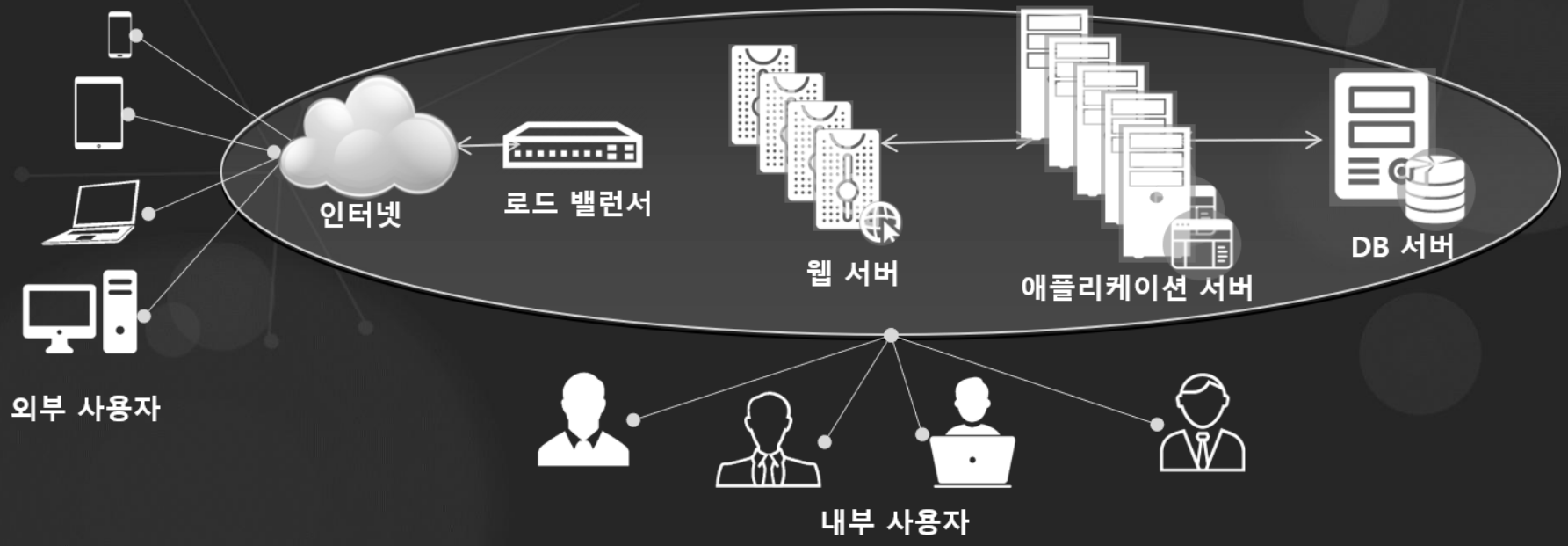
- 사용자의 데이터를 서버의 메모리에 저장
- 서버는 각각의 웹 브라우저로부터 발생한 요청에 대하여 특정한 식별자 (Session ID)를 부여
- 해시테이블에 저장: Name/Value 형태
- 사용자 정보 등에 대한 안전한 저장
- 저장 데이터에 제한이 없음



웹 시스템 일반 구성

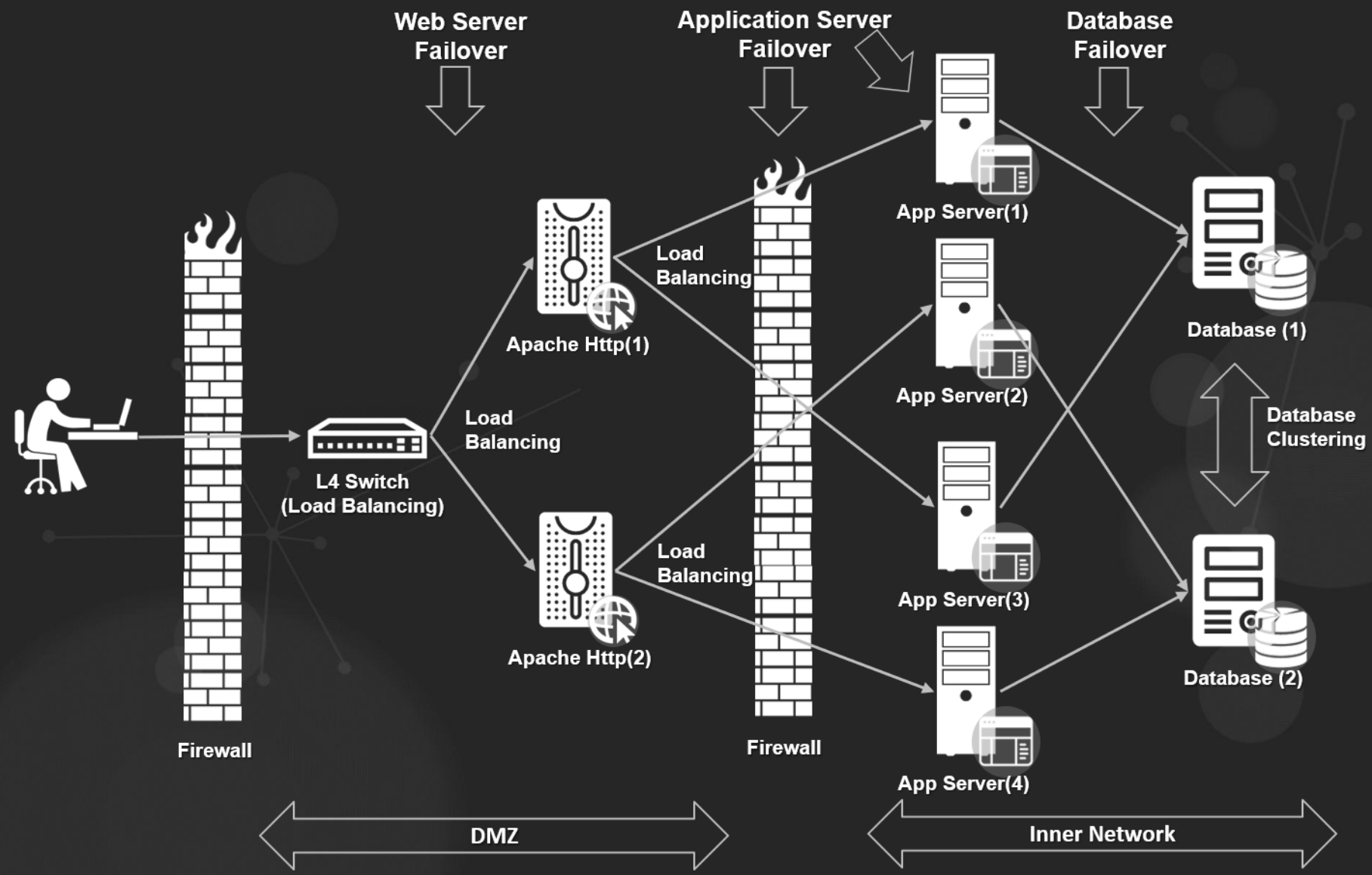
openmaru Cluster

- 웹 브라우저 에서 HTTP 액세스 요청을 분산 처리하는 웹 서버 티어
- HTTP 트랜잭션 의 일관성을 유지하고 시스템 관련 작업을 수행 백엔드에서 실행되는 데이터베이스 등의 검색 / 가공 등을 담당하는 웹 애플리케이션 층
- 시스템 데이터 및 관리 정보를 맡는 데이터베이스 티어
- 웹 시스템을 3 개의 티어로 나누어 수직 분산 된 것으로, 각 티어 별로 확장이 가능하며, 확장성 및 가격 대비 성능도 크게 향상



웹시스템 클러스터링 아키텍처

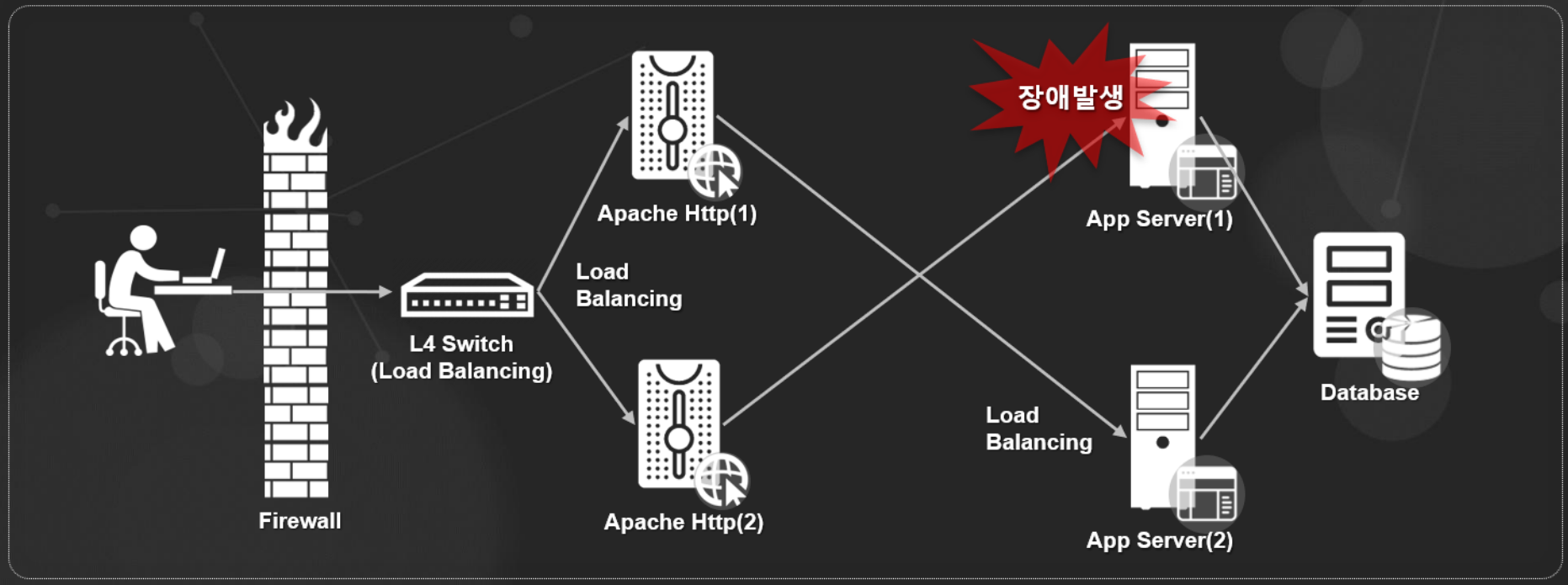
openmaru Cluster



Http-Session Clustering이란?

openmaru Cluster

- Stateless한 Http Protocol을 보완하기 위해 Client(사용자) 정보를 Cookie 및 서버 메모리에 저장하여 처리하는 방법
- 분산되어 있는 다수의 WAS Instance들을 하나의 논리적 그룹으로 묶어 서비스 요청에 대한 Client 입장의 위치 투명성을 제공
- 부하분산 및 장비나 서버의 장애 시 자동 Session Failover를 제공
- Session Clustering을 통해 Client는 중단 없는 서비스를 사용 및 안정성 보장

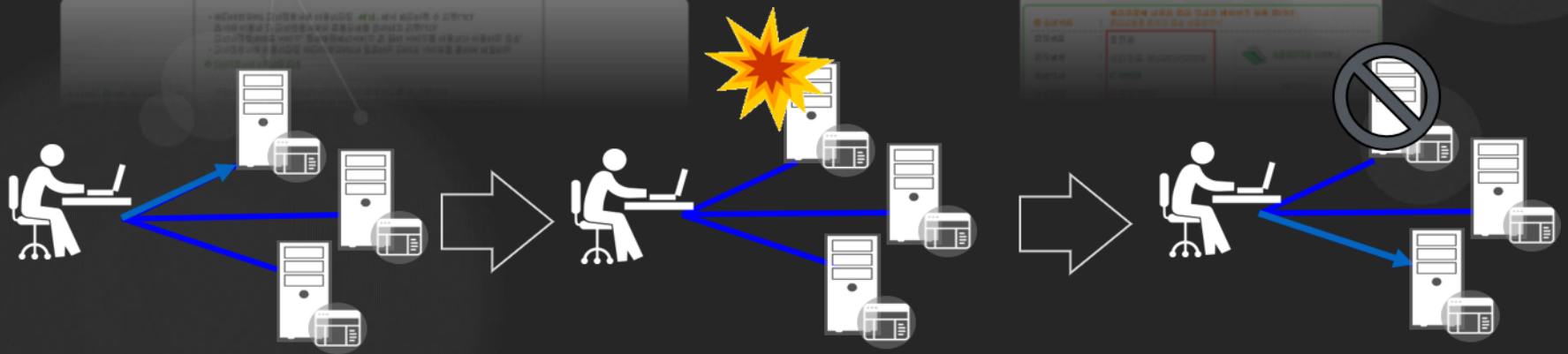


세션 클러스터링

openmaru Cluster

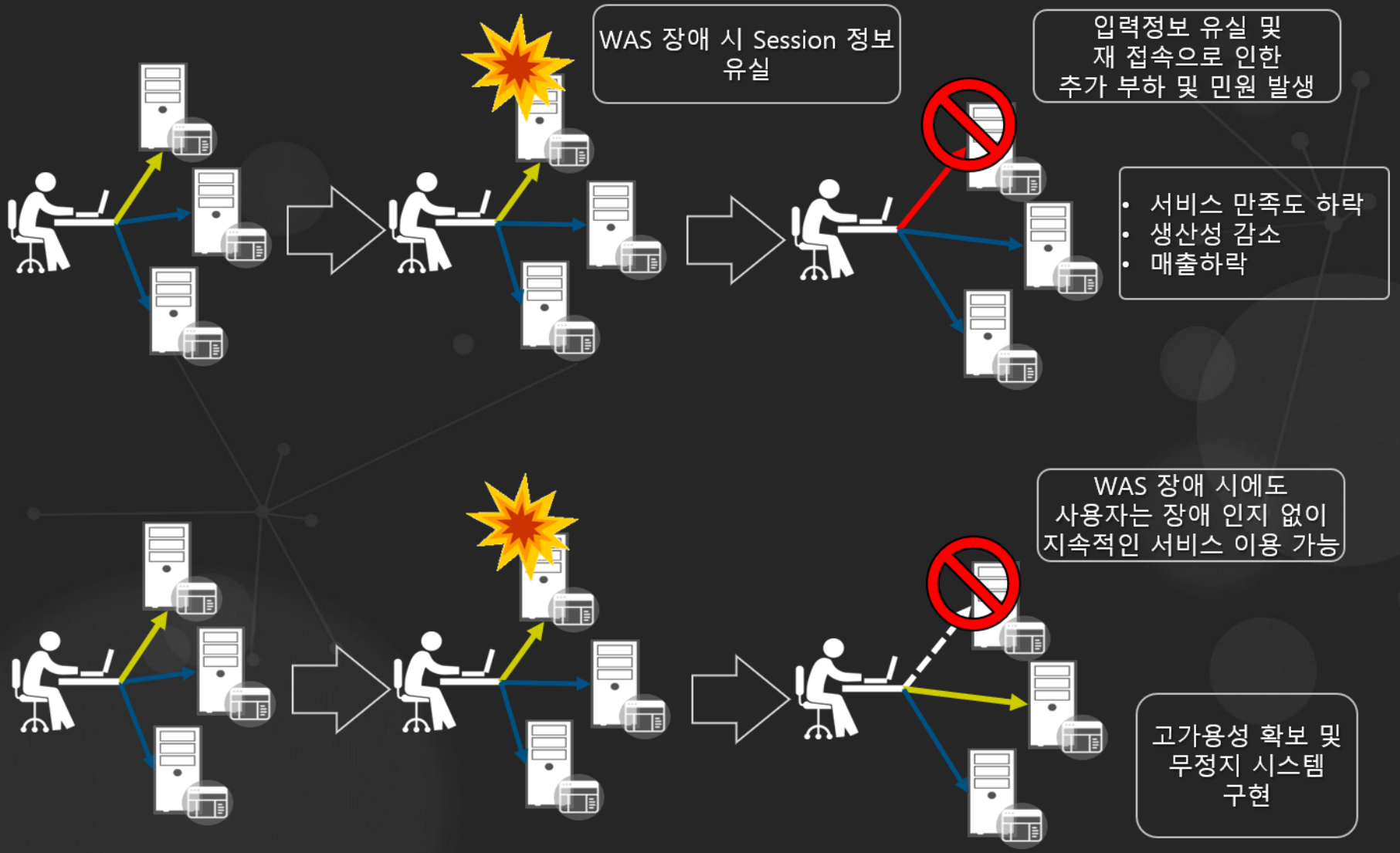
- 사용자에게 투명하게 하나의 시스템으로 보일 수 있도록 함
- 여러 개의 페이지간의 데이터를 입력하는 업무
- 주문업무/ 배송업무/ 장바구니 / 사용자 등록

계속 진행



Session Clustering 필요성

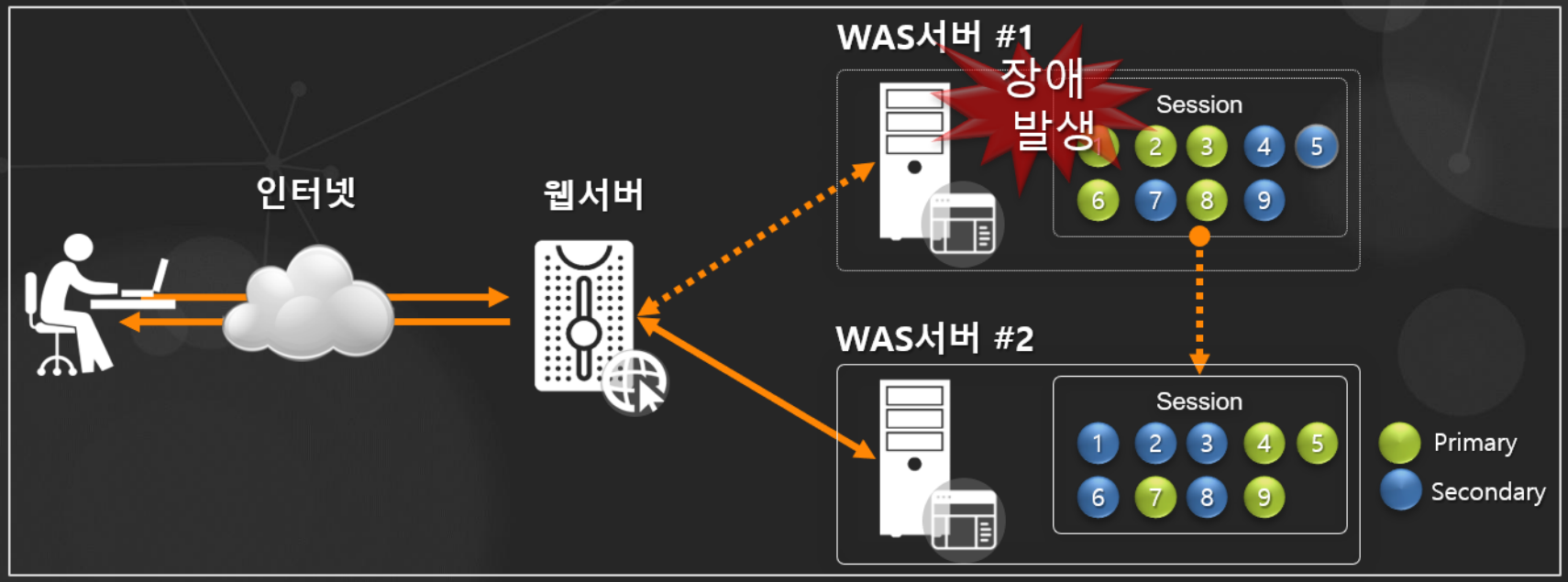
openmaru Cluster



WAS 클러스터링의 필요성

openmaru Cluster

- **웹 애플리케이션의 서비스 가용성, 운영 편이성과 자원 효율성 확보**
  - **무중단 서비스 제공**
    - WAS 장애 시에도 사용자는 장애 인지 없이 지속적인 서비스 이용 가능
    - 서비스 중단 없이 WAS 에 대한 동적 확장
  - **운영 편이성 제공**
    - 서비스 다운타임 없이 애플리케이션 배포 (Rolling Upgrade)
    - 서비스 다운타임 없이 웹시스템 유지보수 (Rolling Patch)
  - **WAS에 대한 Active-Active 구성에 따른 자원 효율성 확보**



주요 WAS 의 세션 복제

openmaru Cluster

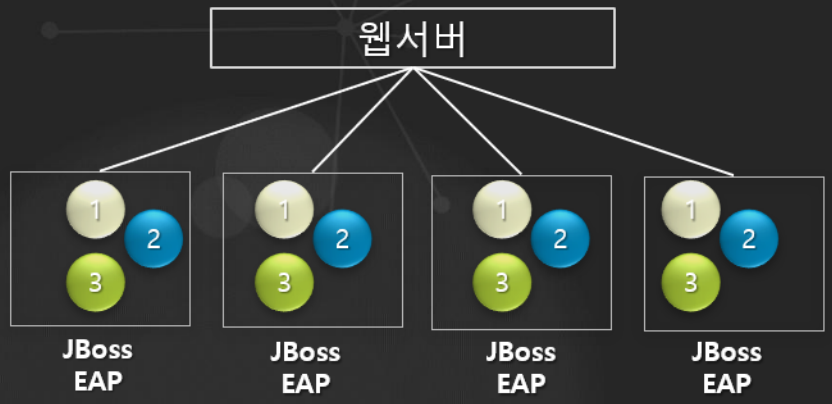
- 웹로직 기본 복제(replication) 방식
  - Primary – Secondary 복제
- JBoss의 기본 복제 방식
  - All to all – 세션 생성시 클러스터 내의 전 노드에 복제
  - 웹로직의 Primary – Secondary 방식과 동일한 Buddy Replication 설정 가능

JBoss by Red Hat

ORACLE WEBLOGIC SERVER

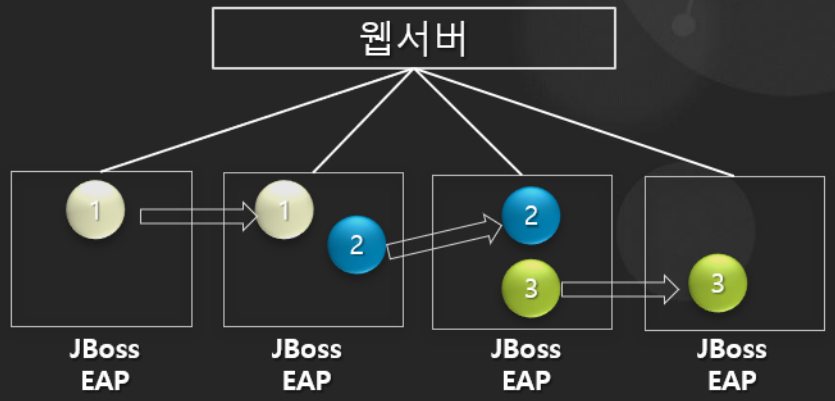
All to all

- 모든 서버 인스턴스가 동일한 정보 저장
- 고가용성 확보
- 인스턴스수가 증가하면 성능에 이슈 발생



Primary - Secondary

- owner 수(데이터를 보관 유지하는 노드수)를 설정
- replication 코스트를 최소한으로 해 퍼포먼스를 향상시키거나 데이터량 증가에 대응할 수 있는 확장성을 확보

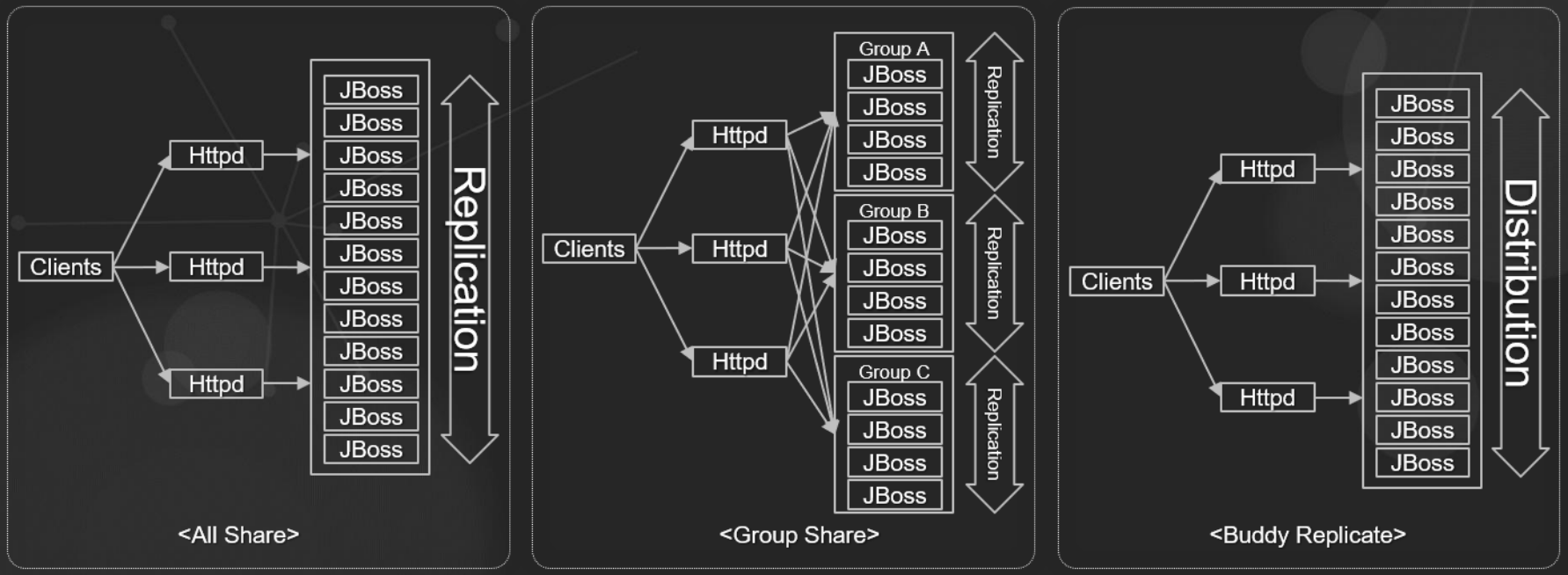


※위의 예는 owner수를 2로 했을 경우

JBoss Deployment Architecture

openmaru Cluster

- WAS Clustering 은 네트워크 상태에 민감하며, Multicast 를 주로 사용
- Grouping 방식은 웹서버의 장애 시 SPOF (Single Point Of Failure)
- AllToAll방식은 모든 인스턴스가 모든 세션을 공유해야 하기 때문에 메모리 이슈와 네트워크 이슈가 많이 발생

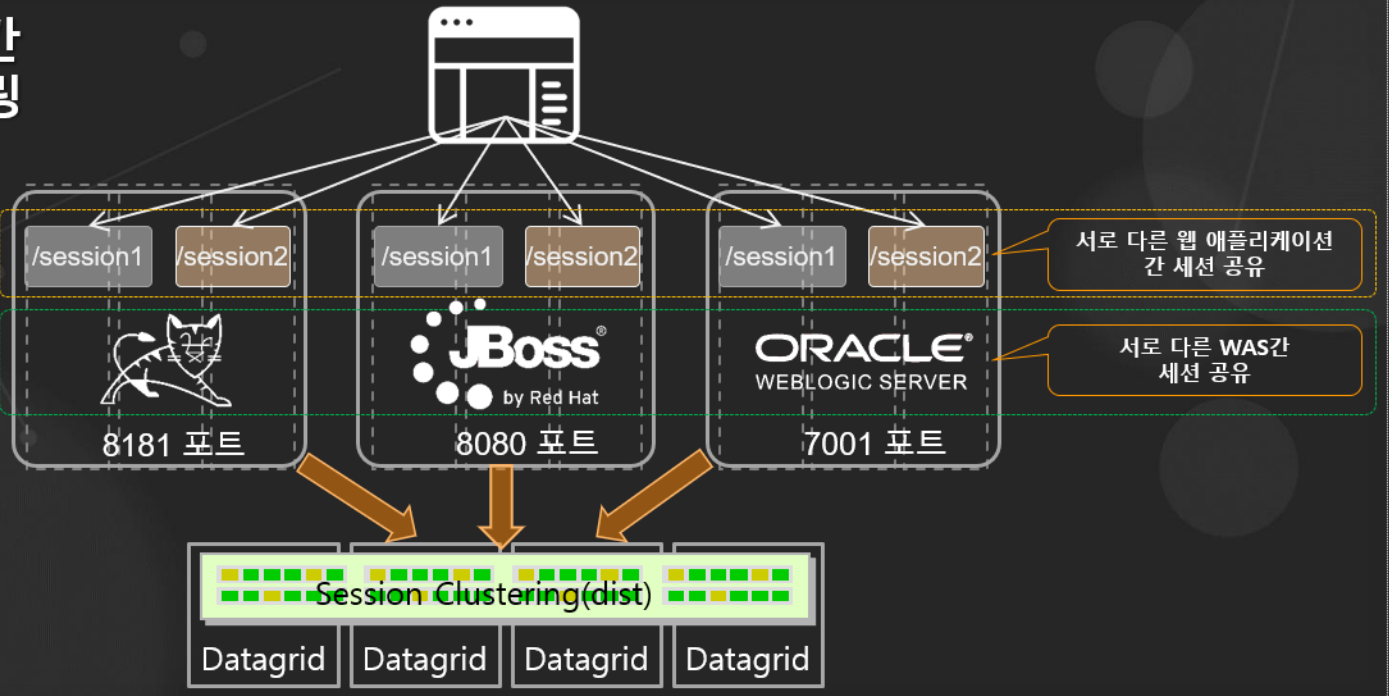


OPENMARU Cluster - 주요 기능

openmaru Cluster

Session 장애 극복(Failover)	WAS / Application 통합	Session 통합 관리 / 모니터링	Advanced Session Control
•장애 시 Client Session 자동 복구 •다양한 세션 공유 방식 지원 (Embedded/Client-Server) •간편한 Session 정보 제어	•이 기종 WAS간 Session Clustering •이 종 Web Application간 Session Clustering •시스템간 Session 통합	•Session Key 관리 기능 •WAS Session 통합 모니터링 •전체 / Node별 Session 현황 •복제 대상 / 비 대상 Session 구분 •Session 생성 / 만료 통계	• Single Log On 지원 • 중복로그인 방지 기능 지원 • Login User 관리 (별도 협의)

이 기종 WAS 간 세션 클러스터링



Session Clustering과 Grid Computing의 필요성

openmaru Cluster

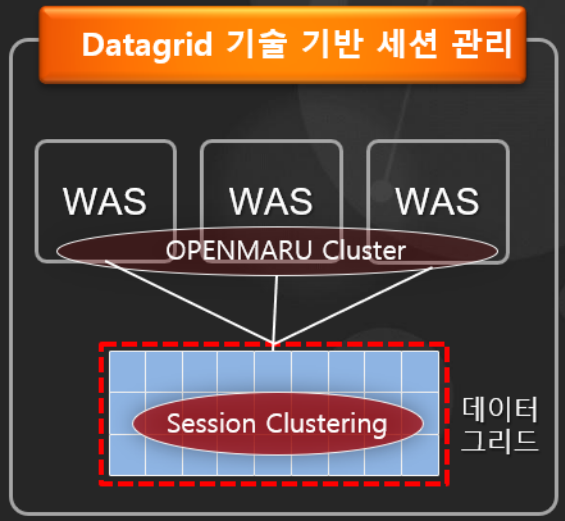
- 기업 내 시스템들간의 Session을 공유 함으로써 사용자에게 무중단 서비스 제공
- 분산된 컴퓨터 자원을 그리드 기반으로 하나의 네트워크처럼 연결
- Application과 WAS의 처리능력을 극대화 시킴.

Session Clustering 구현 기능

- 세션 유지를 통해 웹 어플리케이션의 무중단 서비스 제공
- Domain 이동, Instance 이동, 시스템 다운 시에는 Session 유지 불가

Grid Computing 기반 기술

- Grid Computing이란 분산된 컴퓨터 자원을 하나의 네트워크로 연결해 처리 능력을 향상 시키는 기술
- Scale-Out 형 인프라 구현의 필수 기술



주요 기능

openmaru Cluster

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

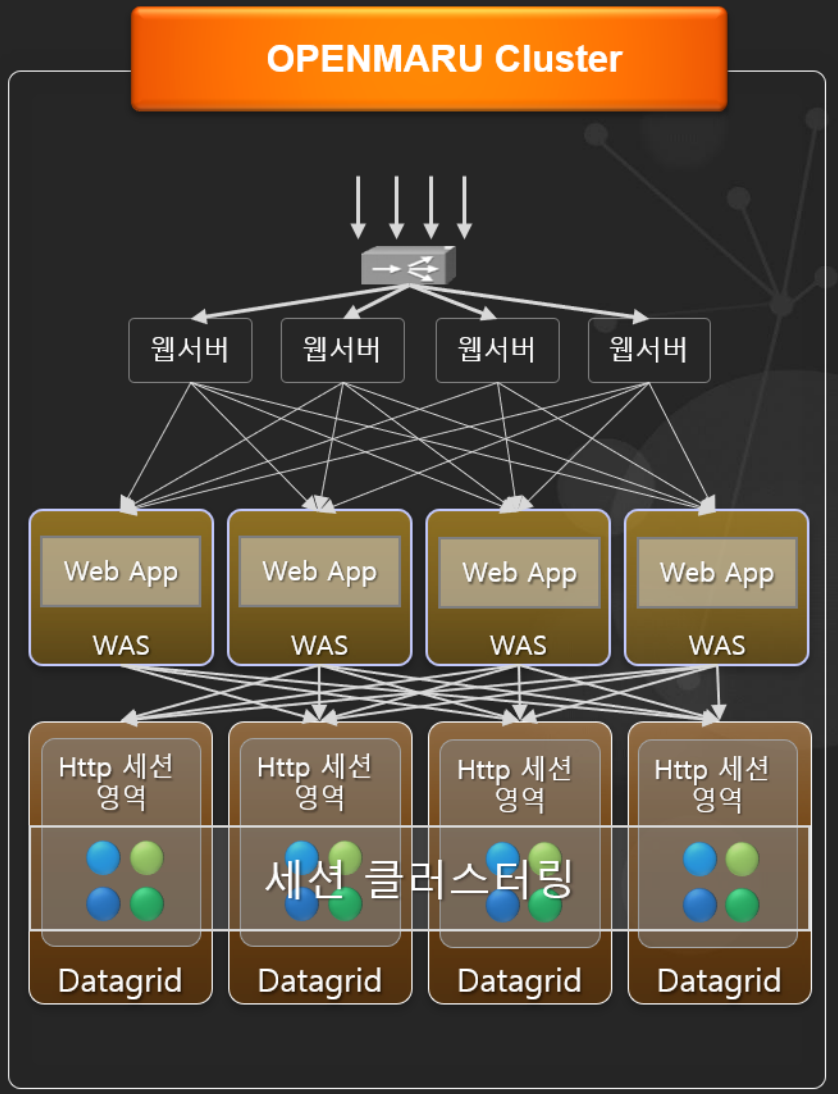
Apache Tomcat에서 세션 클러스터링이 필요할 때

서로 다른 웹 애플리케이션간의 세션 공유를 원할 때

웹 클러스터링을 안정적인 환경에서 운영하고 싶을 때

중복 로그인 방지 기능이 필요할 때

Standard Ed.에 세션 클러스터링이 필요할 때



기대효과

openmaru Cluster

openmaru Cluster

서비스 품질 향상

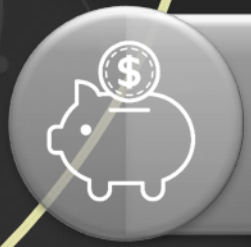
- 세션 클러스터링을 통한 장애 극복
- 서비스 안정성 향상을 통한 사용자 만족도 증대
- 서비스 품질 향상을 통한 매출 증대와 업무 생산성 향상

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



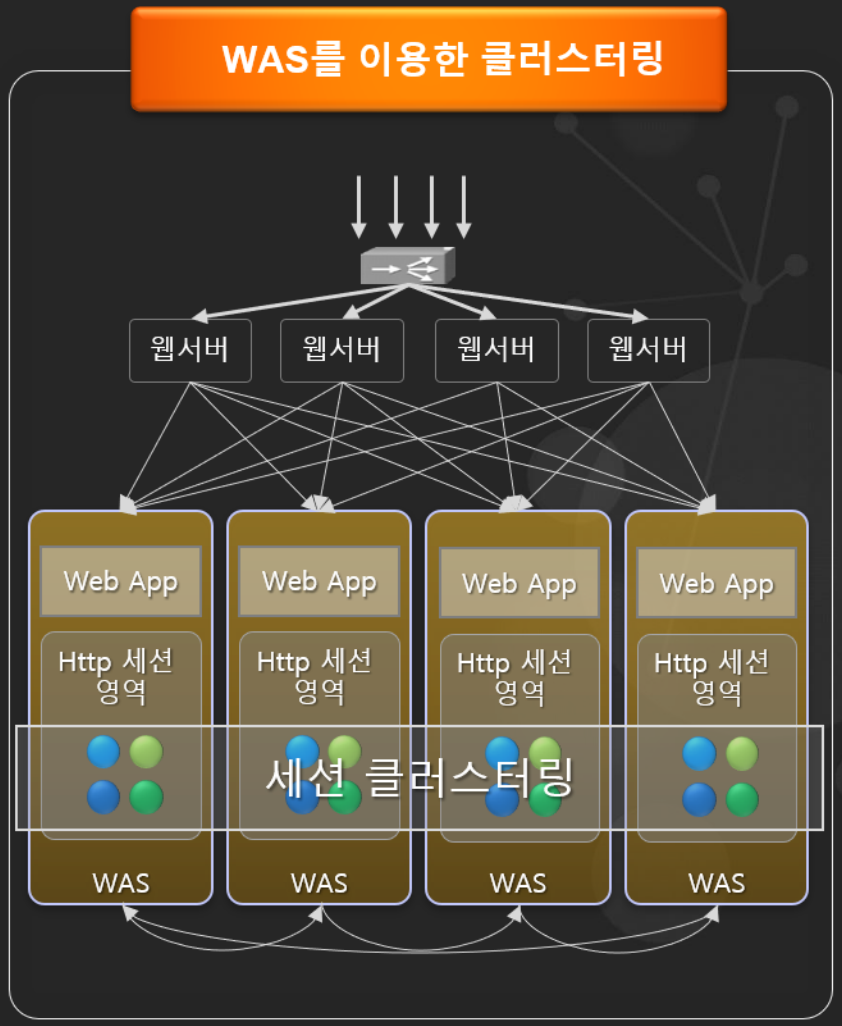
비용 절감 (WAS Standard + OPENMARU Cluster)

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

WAS 세션 복제 방식

openmaru Cluster

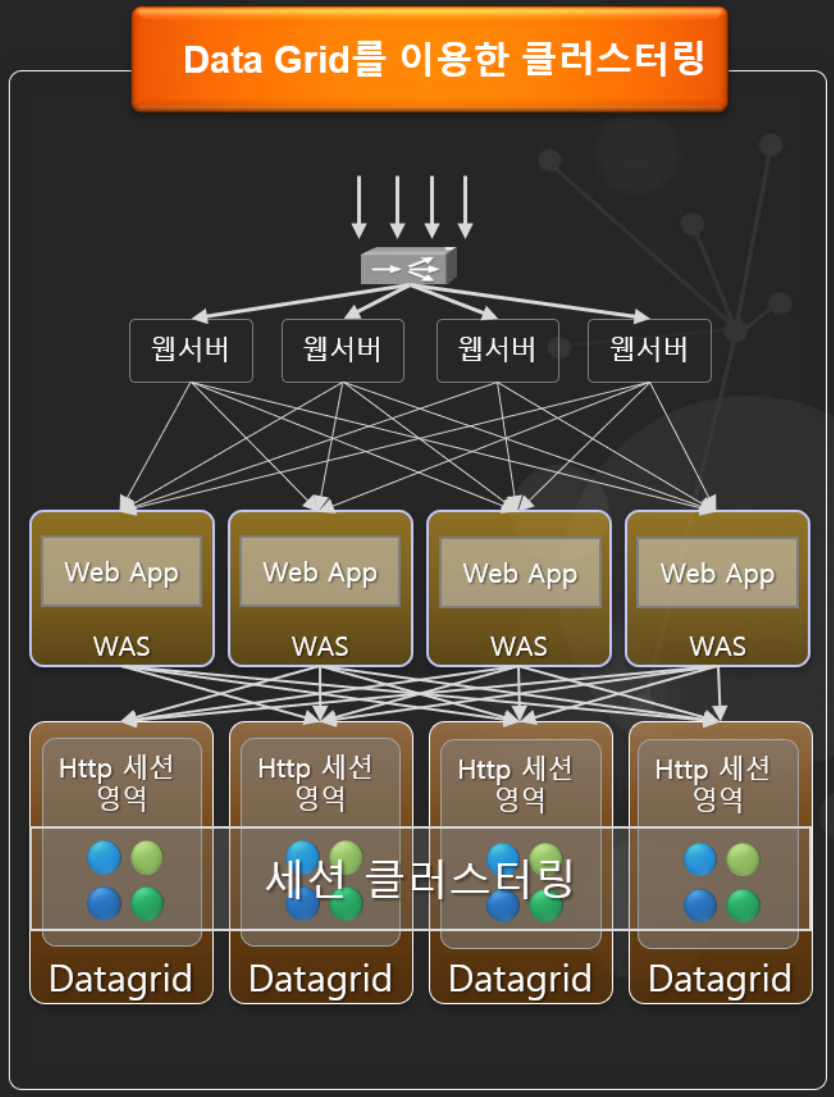
- Http 세션 복제 목적
  - WAS 장애 시 세션정보 보호
  - Distributed (All-to-All 이 아닌 경우) 일 때 로드밸런서의 Sticky 설정
- 문제점
  - WAS 간에 세션 정보를 복제하여 WAS의 대수를 늘려도 세션 정보 저장 영역은 스케일 아웃 불가(Replicated)
  - 세션 정보 저장 영역을 늘리기 위해서는 JVM의 Heap 사이즈를 늘릴 수 있지만 너무 늘리면 "Full GC" 처리 시간 때문에 장애요인 발생
  - 이 기종의 WAS 또는 웹 애플리케이션 간 세션 정보 공유 불가
  -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



WAS 세션 복제 (Data Grid 방식)

openmaru Cluster

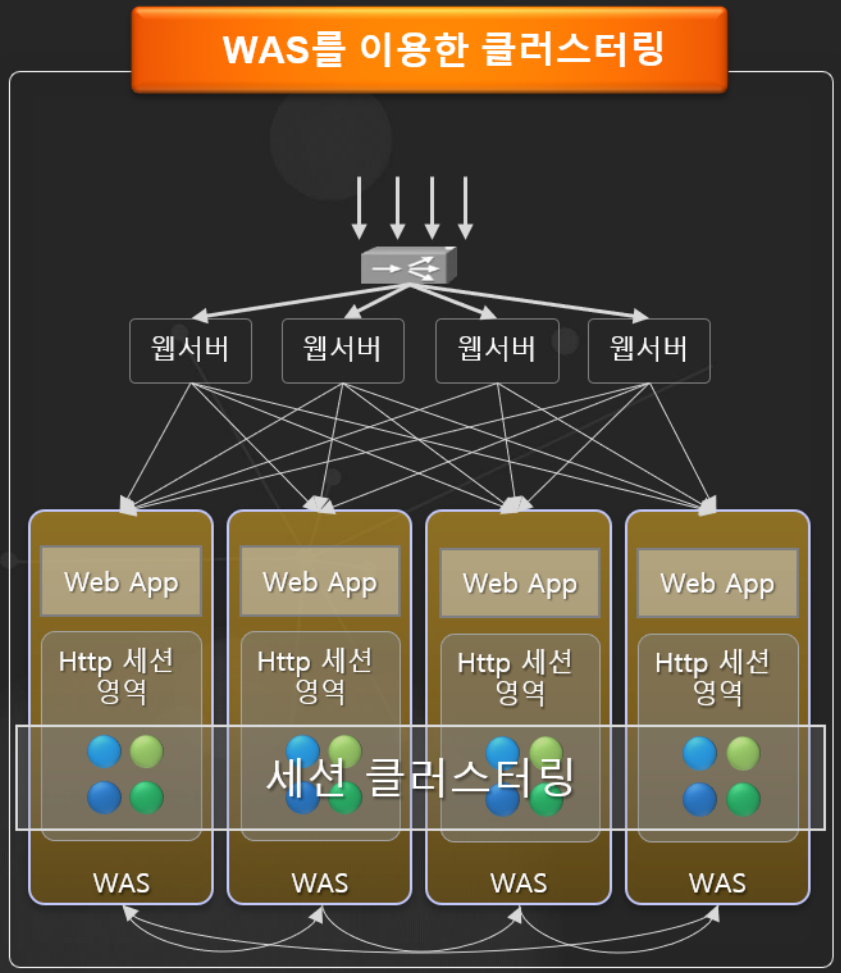
- Data Grid 방식의 이점
  - WAS 장애 발생시 세션 데이터 보호
  - 세션 정보를 분산하여 세션 저장영역에 대한 스케일 아웃
  - 대용량으로 Http Session 데이터를 사용하는 웹 애플리케이션
  - High Volume 웹 애플리케이션 혹은 부적절하게 설계된 Http Session을 사용하는 웹 애플리케이션에 적용
  - Heap 사이즈를 크게 하는 것이 아니라 JVM의 프로세스 수를 늘리는 것으로 Full GC 의 악영향을 회피하면서 시스템을 확장
    - 유휴 H/W 메모리 활용
    - 이 기종의 AP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



WAS 와 Data Grid를 이용한 세션 클러스터링 비교

openmaru Cluster

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

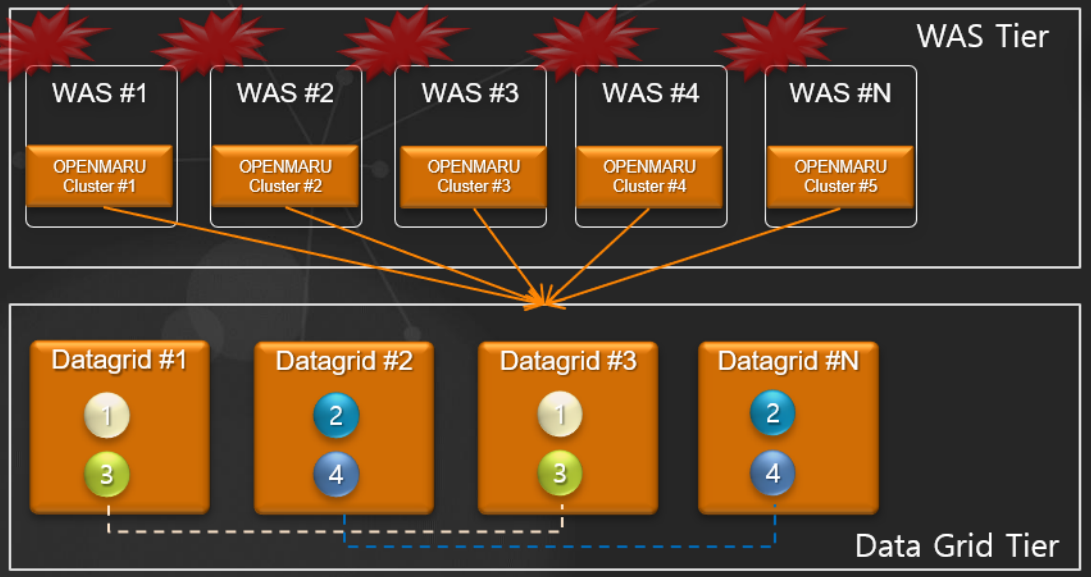


Data Grid 를 통한 WAS 고가용성 구현

openmaru Cluster

WAS 세션 관리

- All Share 에 의한 Network 부하 및 메모리 과중
- Buddy Replicate 제약 존재
- Data Grid 별도 구매

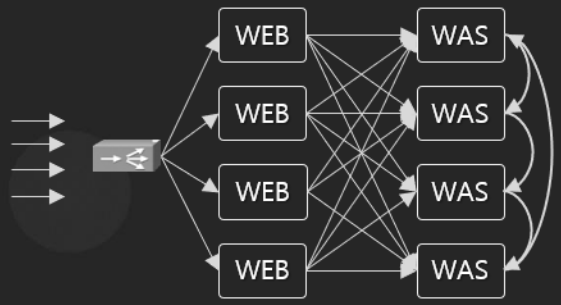
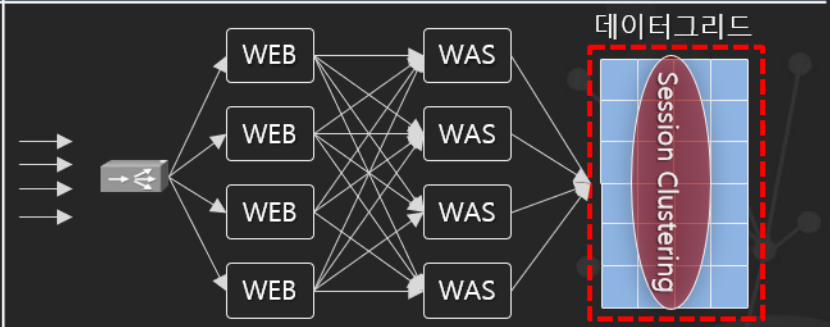


현대화된 WAS 관리 방안

- 세션 데이터 그리드 손쉬운 확장
  - WAS 전체 장애도 세션 유지
  - 세션에 의한 OOM 장애 근본 해결
  - Single Log On
- EDG (Elastic Data Grid) 로 손쉬운 확장
  - 100 ~ 1000 Node Cluster
  - Object 공유 / Cache
  - Distribute / Replicate 토폴로지 지원

WAS 클러스터링 vs. OPENMARU Cluster

openmaru Cluster

항목	WAS 세션 클러스터링		OPENMARU Cluster	
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
세션 관리	◯	• WAS 인스턴스 재시작시 세션 동기화와 복제 • 애플리케이션 배포 시 세션 동기화와 복제 • 애플리케이션 별 세션 정보 관리	●	• WAS 인스턴스 재 시작 시 세션 복제 작업 제거 • 애플리케이션 재배 포 시 세션 복제 작업 제거 • 복수의 애플리케이션 간 세션 정보 공유

WAS Clustering vs. OPENMARU Cluster

openmaru Cluster

- WAS에서 제공하는 클러스터링은 단일 애플리케이션에 대한 세션 공유 지원
- OPENMARU Cluster는 이 기종의 WAS 나 애플리케이션 간 세션 공유 지원

항목	WAS Enterprise Edition	OPENMARU Cluster
Session Clustering	지원	지원
EJB/JMS Clustering	지원	미지원
다른 Web Application간 Session Clustering	미지원	지원
2차 Domain간 Session Clustering	미지원	지원
서로 다른 WAS간 Session Clustering	미지원	지원
중복 로그인 방지	미지원	지원
전체 WAS 장애 세션 유지 여부	미지원	지원
Session 관리 기능	미지원	지원

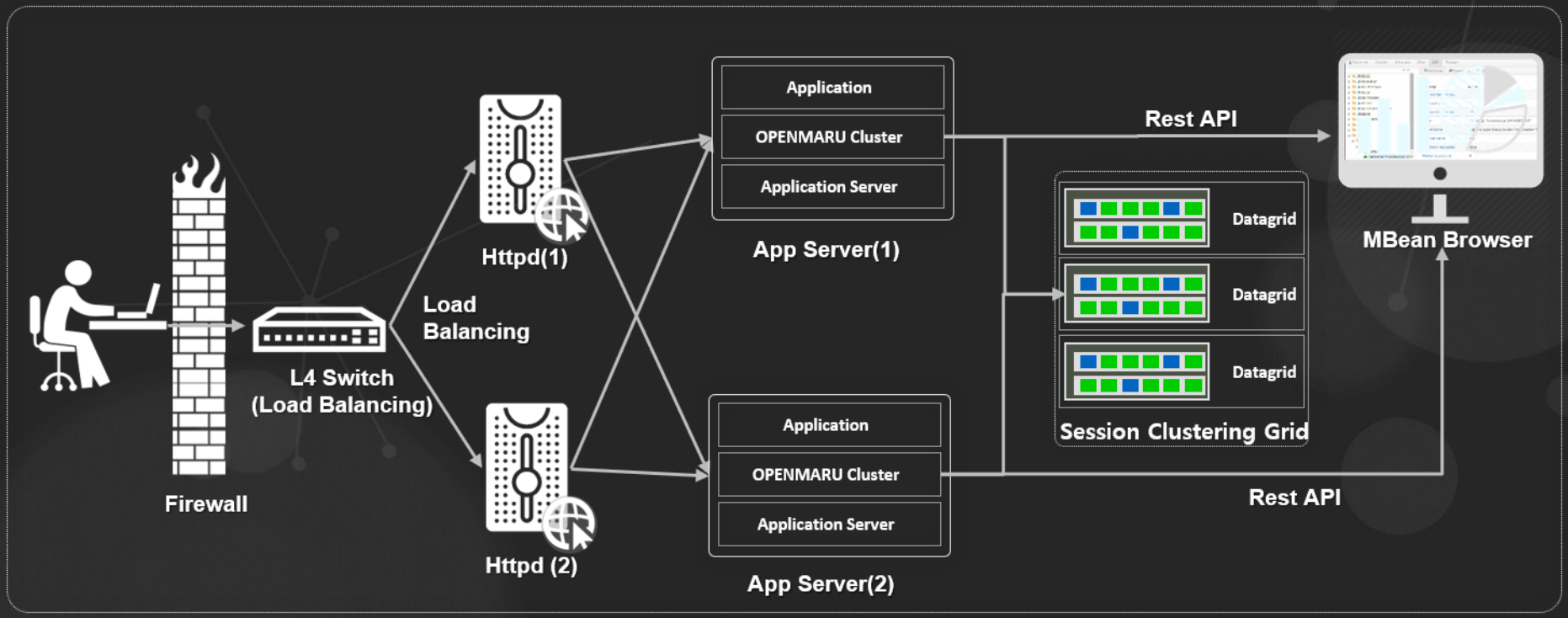
WAS Advanced Clustering

OPENMARU Cluster
주요 기능

openmaru
APM

OPENMARU Cluster – Architecture

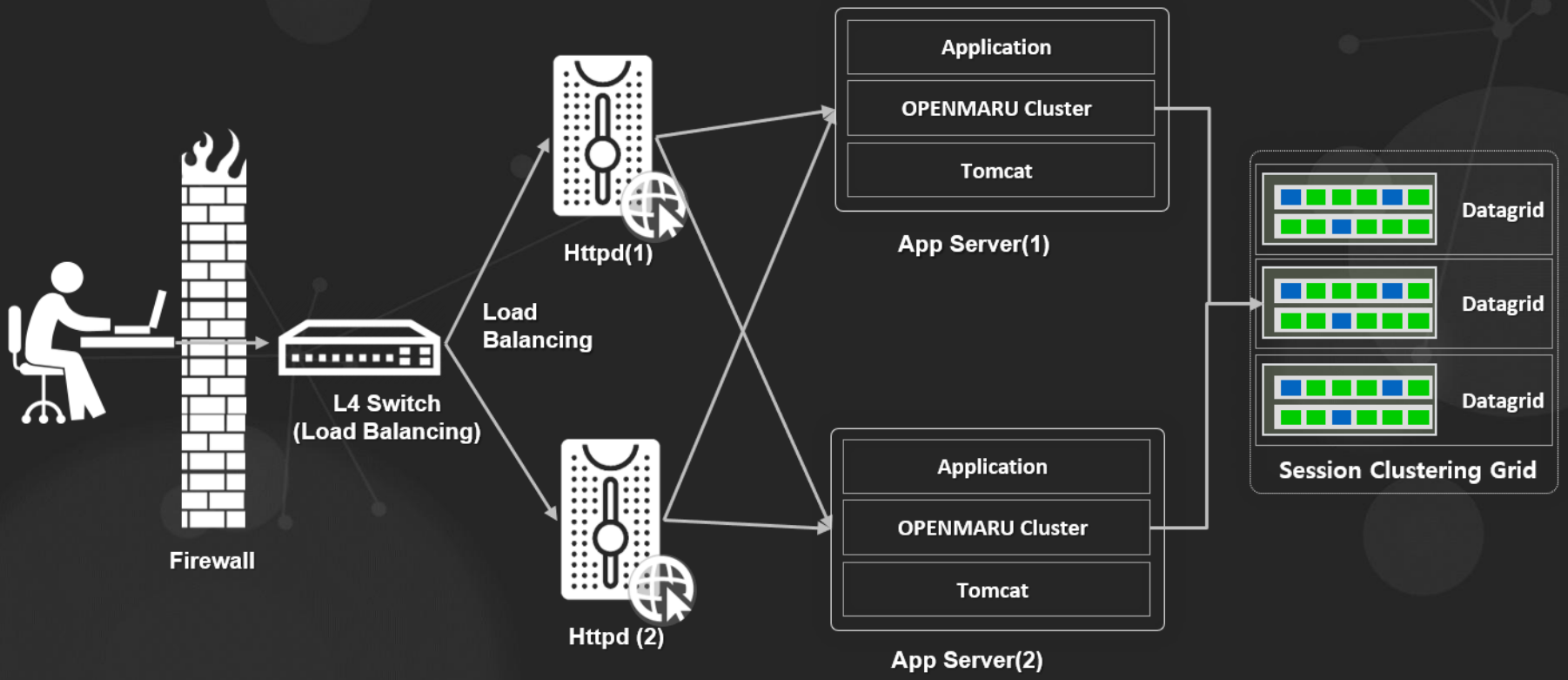
- 다양한 WAS와 JAVA가 지원되는 어떠한 플랫폼이나 프레임워크에도 적용이 가능하여 이 기종의 WAS 간에도 Session 통합이 가능.
- OPENMARU Cluster 는 서브 도메인으로 구성된 시스템 간에도 Session 공유를 통한 SLO(Single Log On)기능을 제공



OPENMARU Cluster – Architecture

openmaru Cluster

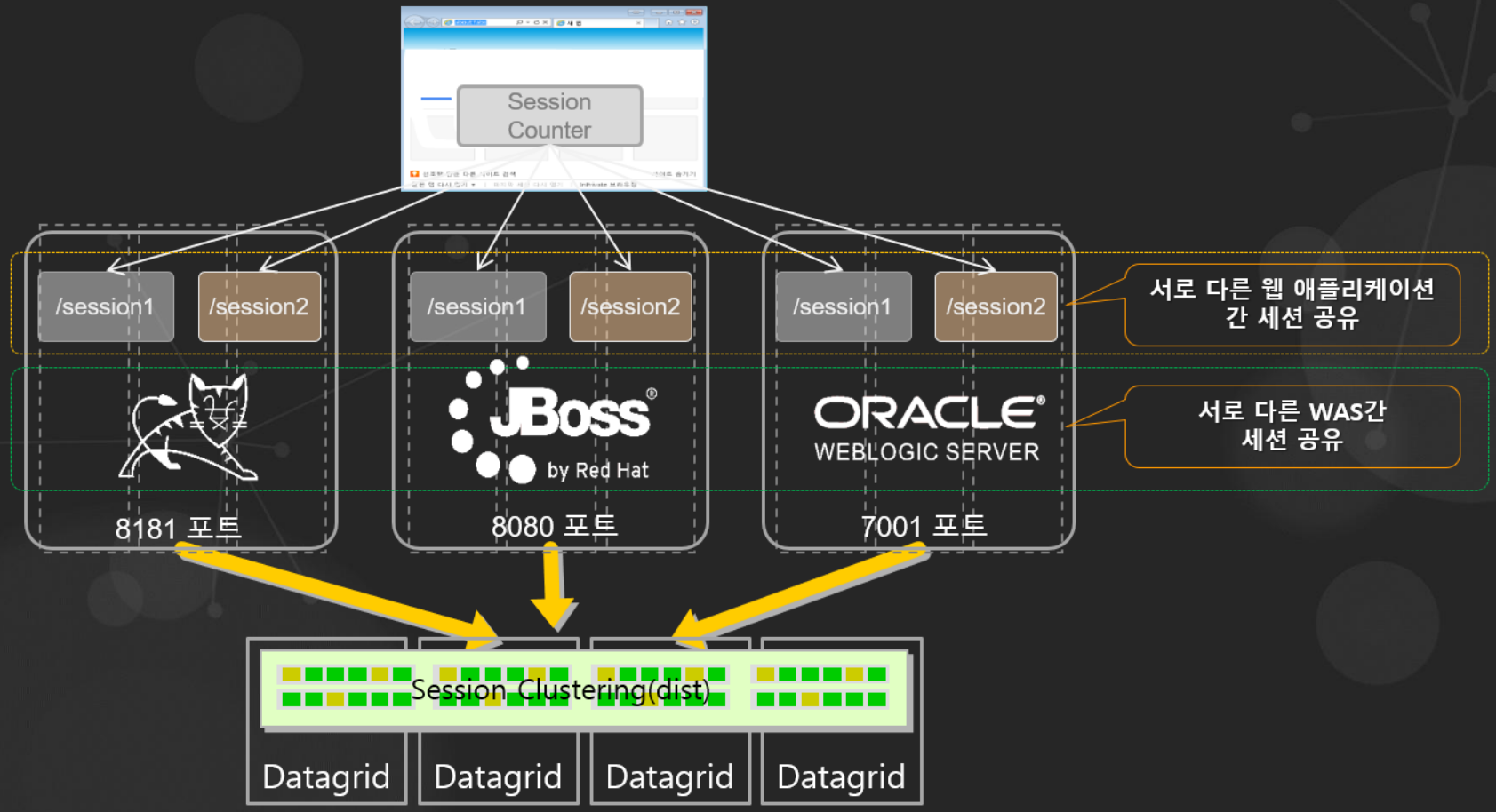
- 이 기종의 WAS 와 웹 애플리케이션 간의 세션 정보를 공유
- 지원하는 WAS - WebLogic Server / Tomcat / JBoss



다른 WAS 제품 간의 세션 클러스터링

openmaru Cluster

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



OPENMARU Cluster가 필요할 때 - Use Cases

openmaru Cluster

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

ORACLE WEBLOGIC SERVER

WebLogic Standard Ed.에 세션 클러스터링이 필요할 때

다양한 웹 애플리케이션 서버 지원

openmaru Cluster

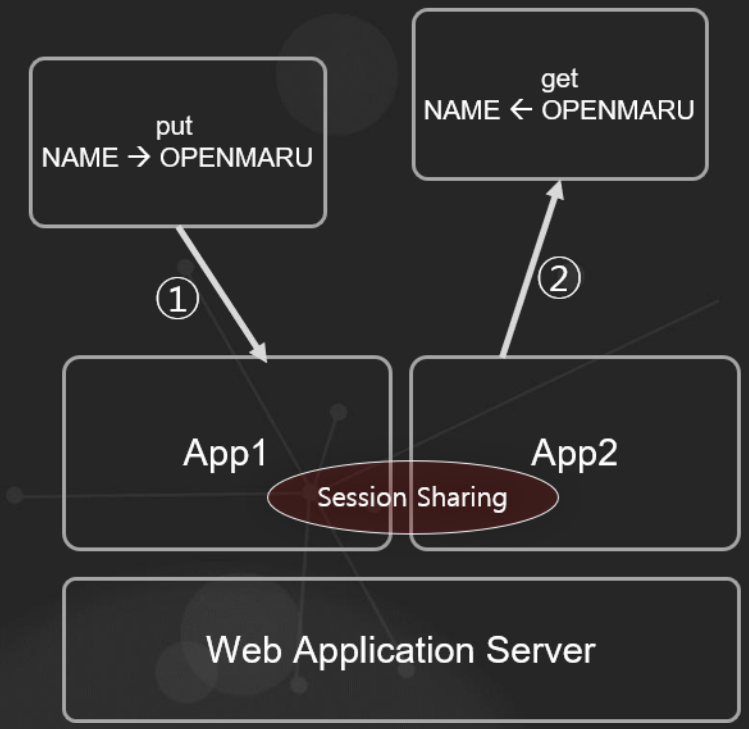
- Servlet 표준을 사용하기 때문에 코드 변경은 없음
- Servlet 표준 web.xml에 설정 추가

ORACLE®
WEBLOGIC SERVER

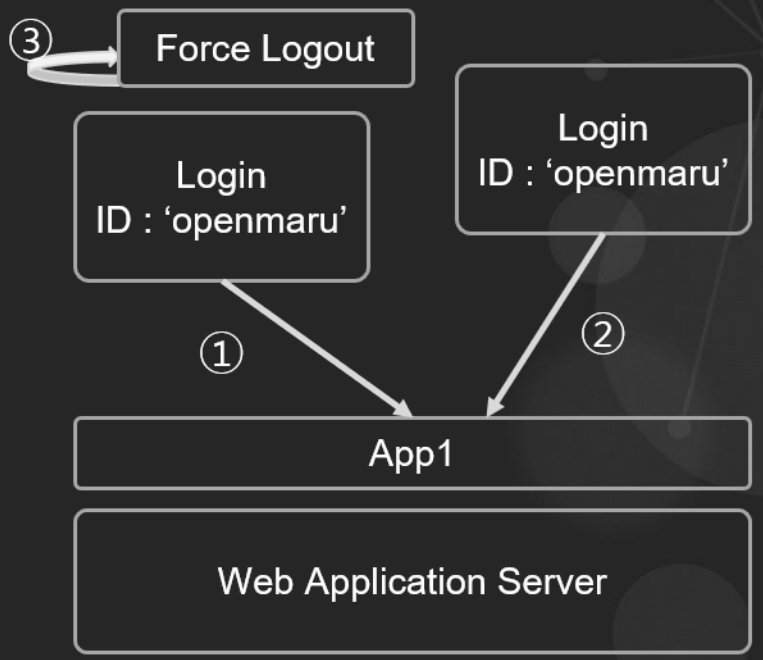
주요 기능

openmaru Cluster

서로 다른 웹 애플리케이션 간 세션 공유



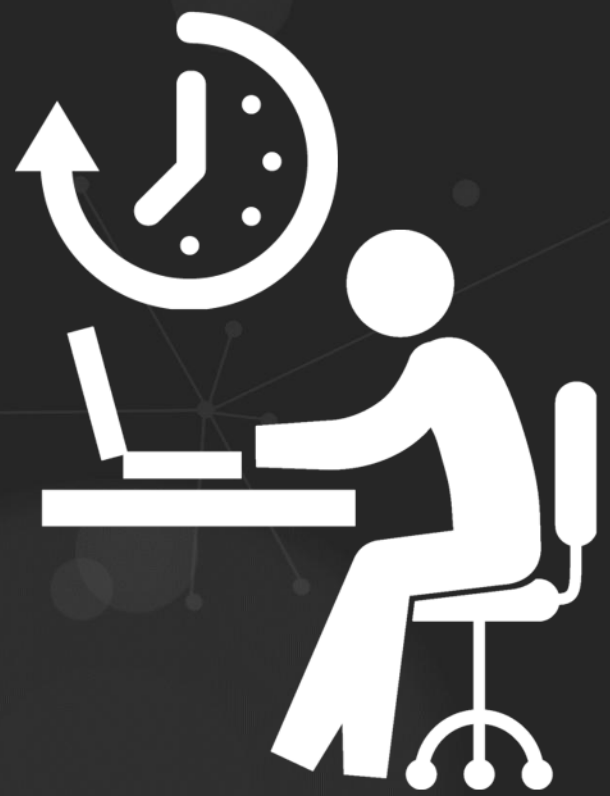
중복 로그인 방지



주요 기능

openmaru Cluster

세션 타임아웃 제어



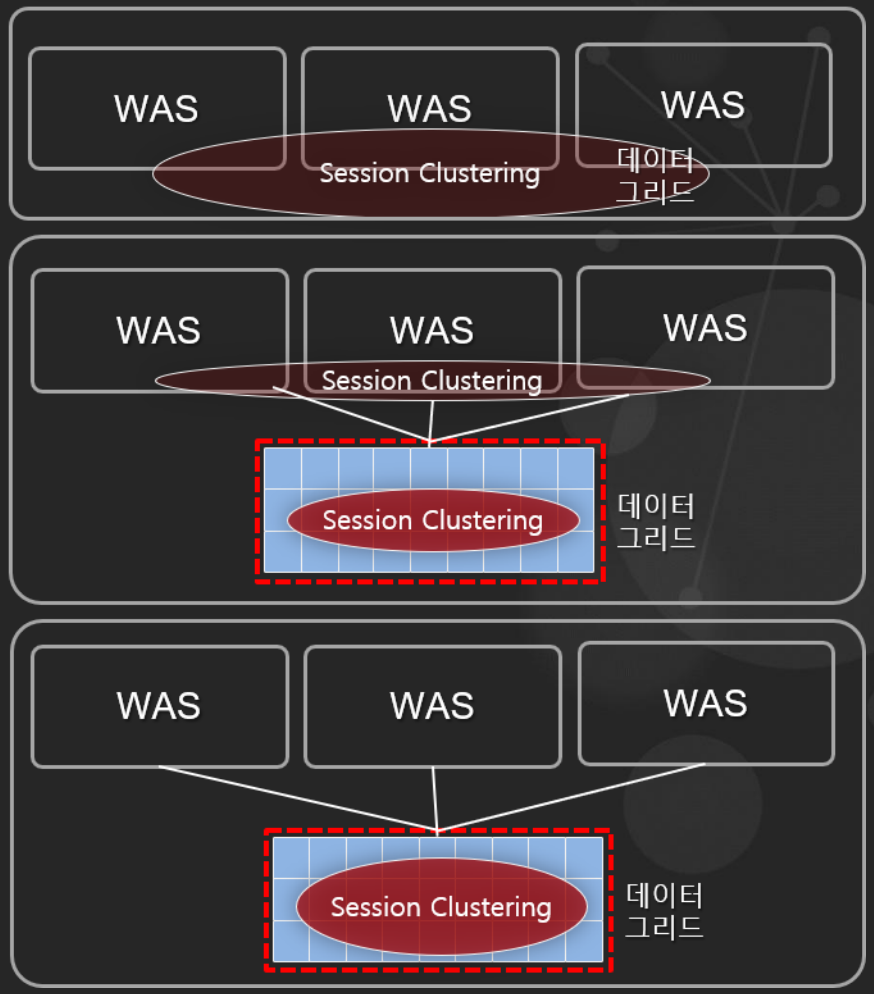
다양한 토폴로지 지원

주요 기능

openmaru Cluster

세션 타입아웃 제어

다양한 토폴로지 지원



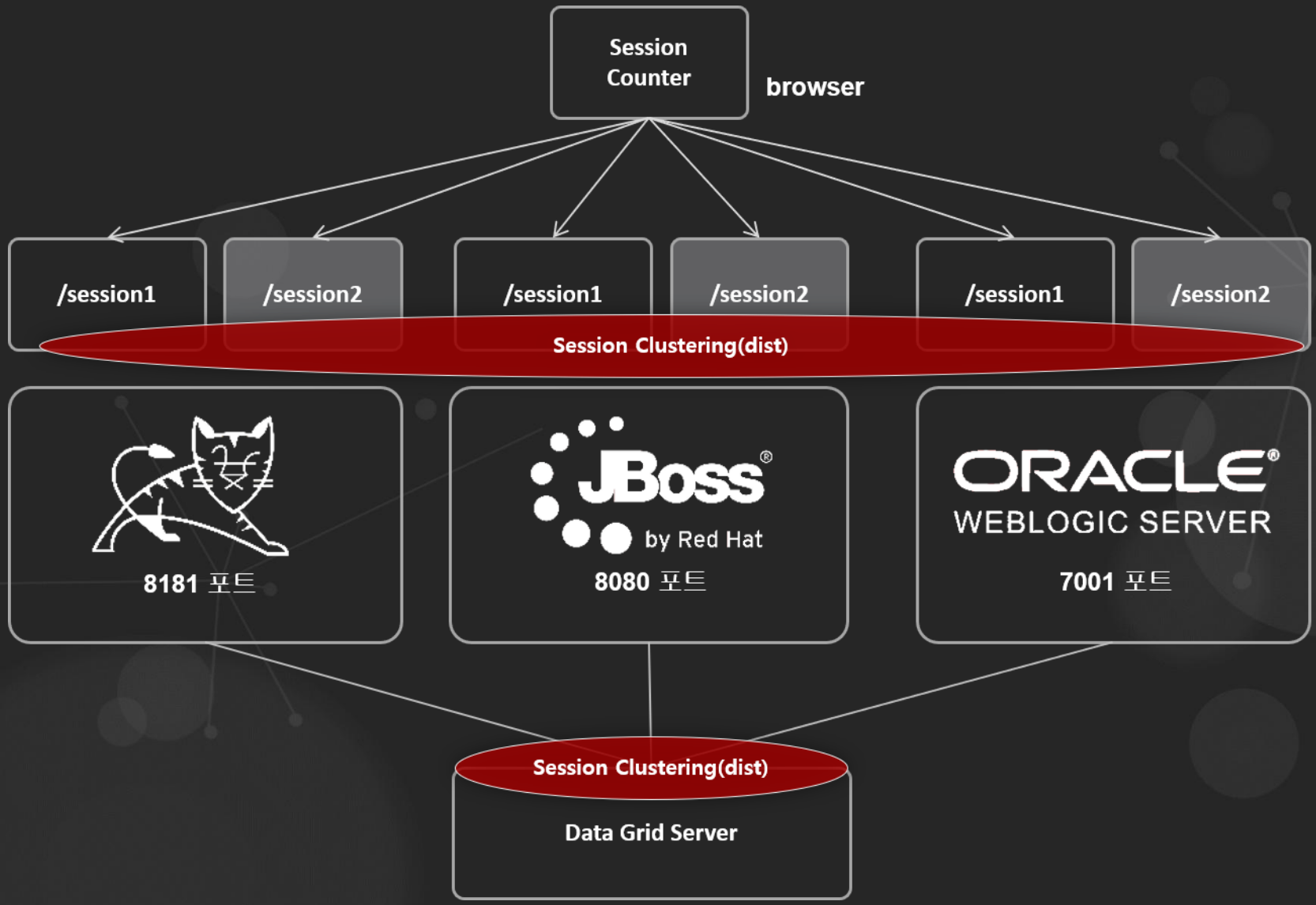
WAS Advanced Clustering

OPENMARU Cluster 데모

openmaru APM

데모 시스템 환경 구성

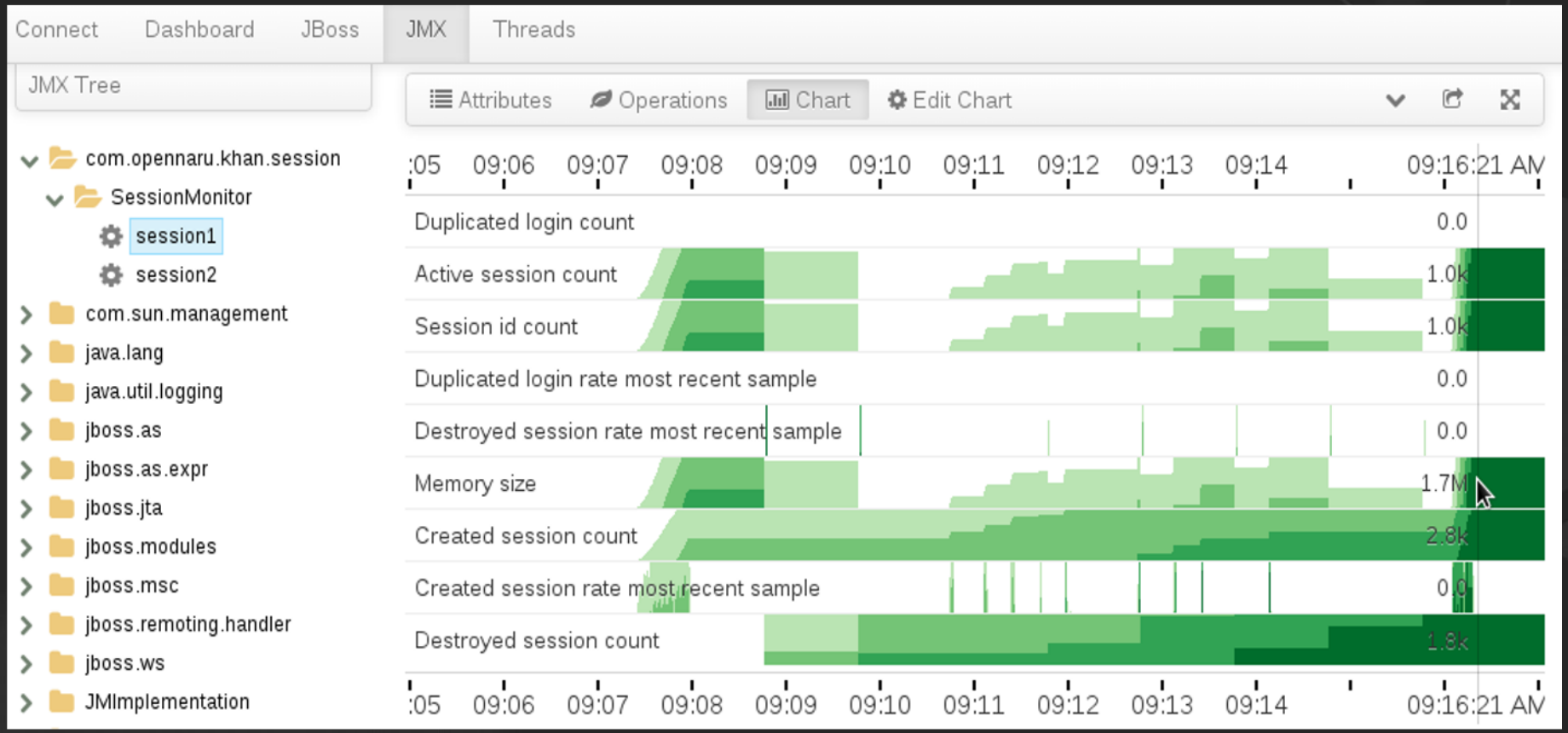
openmaru Cluster



애플리케이션 부하 테스트 및 모니터링

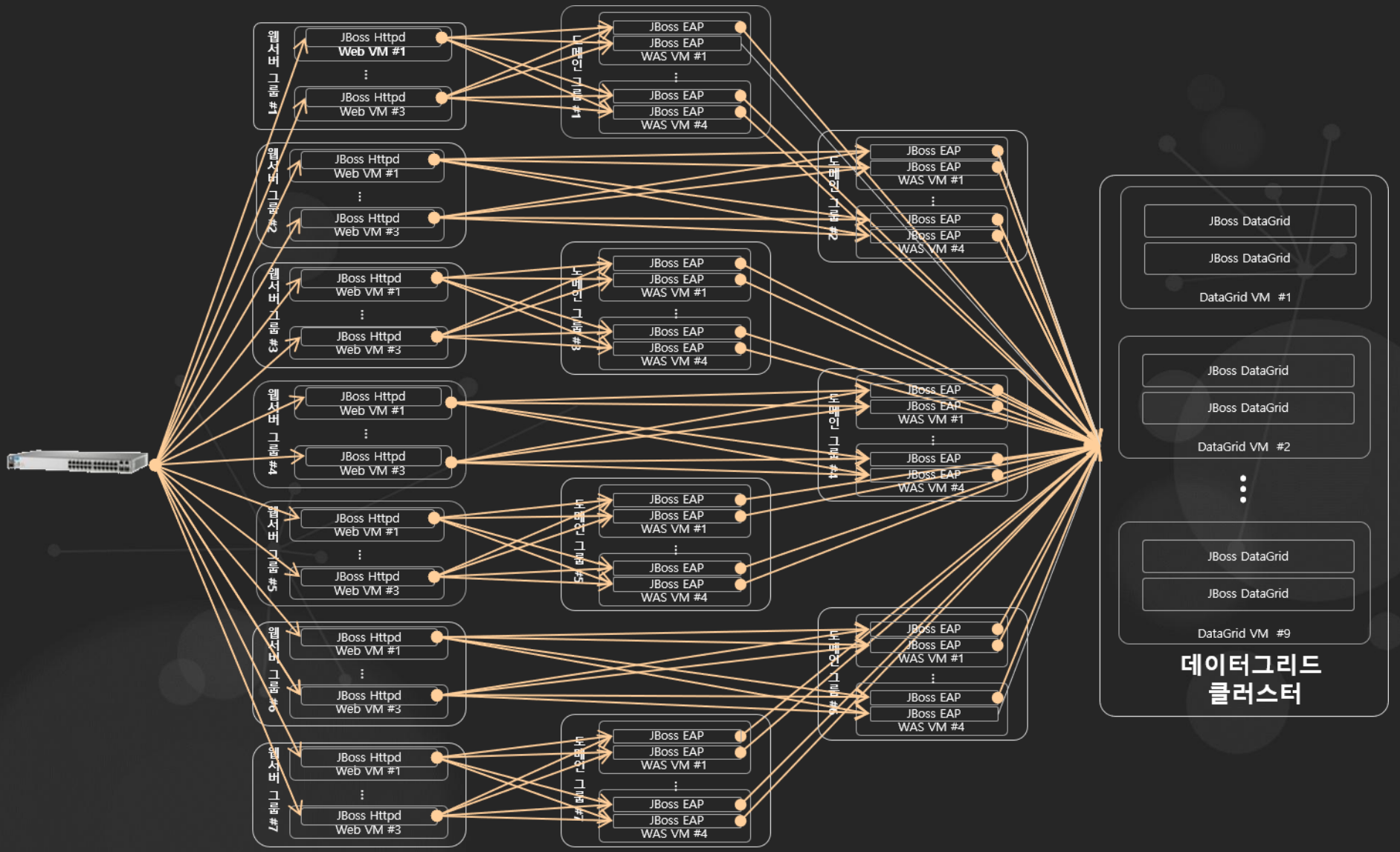
openmaru Cluster

- 샘플 애플리케이션에 대한 부하 테스트
- 부하 테스트 상황에서 OPENMARU Cluster가 제공하는 MBean의 정보 모니터링



인구주택총조사 시스템 인스턴스 구성도

openmaru Cluster



적용사례 : 모 쇼핑몰 사이트 구축 사례

openmaru Cluster

- 선택적 복리후생 쇼핑몰 운영사
- 950여 고객사 보유
- 141만명 고객 사용 중

OOOO 의 영업실적 추이 및 전망

	2013	2014F	2015F
고객사(개)	824	950	1,000
직원수(만명)	130	141	148
위탁복지예산(십억원)	870	920	945
총사용(십억원)	877.1	923.5	955.0

자료: 회사 자료, 신한금융투자 추정

- OPENMARU Cluster를 이용한 안정적인 세션 클러스터링 환경 구축
- 멀티 도메인의 세션을 모두 JBoss Data Grid에 저장하여 세션 클러스터링
- OPENMARU APM을 통한 빠르고 정확한 운영시스템 구축
- 기존 세션 클러스터링 방식의 SPOF(Single Point Of Failure) 문제를 해결
- 애플리케이션 성능 향상 : JBoss Data Grid를 활용한 Global Data Cache 적용

제 품	머신 수	인스턴스 수
웹서버	6	6
JBoss EAP – WAS	21	40
JBoss Data Grid	7	11
Total	34	47

금융회사 모바일 서비스 세션 통합 방안

openmaru Cluster

모바일 Native App

모바일 Web

Single Log On

openmaru Cluster

Datagrid | Datagrid | Datagrid | Datagrid

OPENMARU Cluster

웹서버 | 웹서버 | 웹서버 | 웹서버

Web App WAS | Web App WAS | Web App WAS | Web App WAS

Http 세션 영역 | Http 세션 영역 | Http 세션 영역 | Http 세션 영역

세션 클러스터링

Datagrid | Datagrid | Datagrid | Datagrid

금융회사 모바일 서비스 세션 통합 방안

openmaru Cluster

Real-Time Monitoring

80 Port
HTTP,
WebSocket

80 Port

80 Port

WEB Servers

모바일앱 Web#1~4

오픈웹 Web#1~2

SYS Agent

JBoss Servers

OPENMARU Cluster

모바일앱
WAS#1~4

App Push
WAS#1~2

상품서비스
WAS#1~2

배치
WAS#1~2

신용평가시스템
WAS#1~2

관리업무서버
WAS#1~2

WAS Agent

FDS
WAS#1~3

OPENMARU Cluster

Datagrid

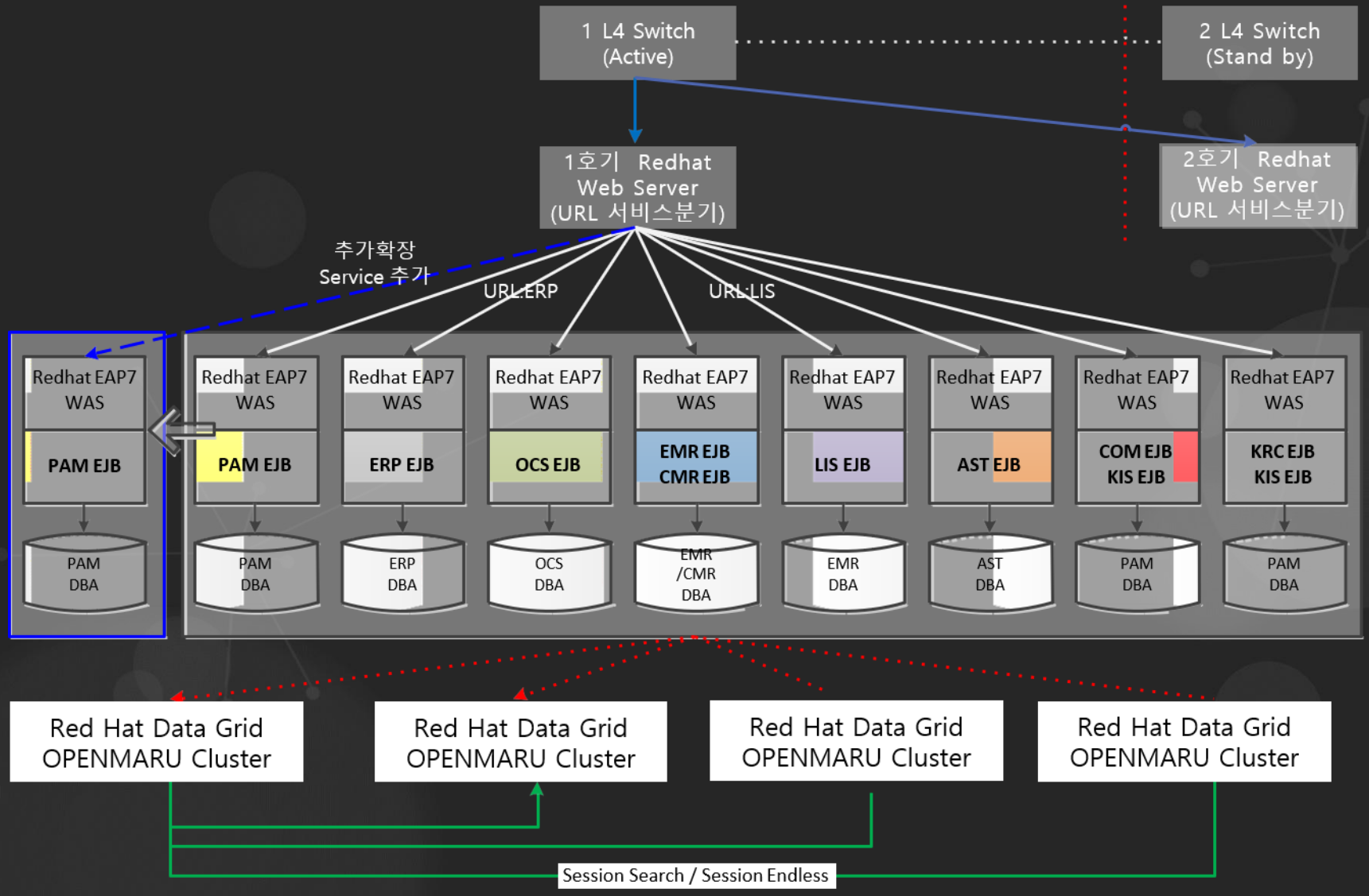
OPENMARU APM Server

OPENMARU APM

Time Series DB

건국대학교병원 MSA 와 OPENMARU Cluster

openmaru Cluster



세션 클러스터링 적용 가이드

openmaru Cluster

- 적용 가이드
  - Session Size를 최대한 적은 크기가 되도록 한다
  - Session Attribute는 단순한 Object를 사용한다
  - 큰 Attributes를 사용하지 말 것
    - Hashtable이나 Collection Object들을 사용하지 말라
  - Session에 복잡한 Object Tree를 저장하지 말라
  - 더 이상 사용되지 않는 Attribute는 반드시 Remove한다

클러스터 HTTP 세션 복제 실패의 주요 원인

openmaru Cluster

- 증상
  - 애플리케이션이 자꾸 로그아웃 된다
  - 서버 로그에 에러 or 경고 메시지가 출력된다
  - Fail-over가 동작하지 않는다
- Multicast나 네트워크 문제
  - Multicast 문제 원인을 찾아서 해결해야 함
- Cluster 구성이나 Web Application 설정의 문제
  - 설정 파일을 검토
- Session Data가 Serializable Interface를 구현하지 않았음
  - 세션 데이터가 Serializable 를 구현해야 함
- 애플리케이션의 세션 사용 프로그램의 문제

제품 / 서비스에 관한 문의

- 콜 센터 : 02-469-5426 (휴대폰 : 010-2243-3394)
- 전자 메일 : sales@opennaru.com