openmaru

공공기관 WAS 도입 가이드 :

왜 APM을 WAS와 별도로 구매하면 안 되는가

"WAS 라이선스에 APM 따로, 세션 클러스터링 따로, 매년 유지보수 비용은 왜 이렇게 늘어나는 겁니까"
상용 WAS 환경에서 APM·세션 클러스터링·운영 콘솔을 개별 구매하는 조립식 구조는 TCO를 기하급수적으로 증가시키고, 장애 발생 시 벤더 간 책임 공방으로 MTTR이 길어지는 구조적 문제를 안고 있습니다.
OPENMARU iAP는 WAS·Web 서버·APM·Cluster·AI 운영을 단일 플랫폼에 통합하여 이 문제를 근본적으로 해결합니다.

Contact

02-469-5426

hello@openmaru.io

www.openmaru.io

2026.03

Contents

1장: 엔터프라이즈 WAS의 구조적 문제와 OPENMARU iAP의 탄생

1.1 '비싼 WAS' 시대가 남긴 과제

엔터프라이즈 IT 환경에서 상용 WAS(Web Application Server)는 오랜 기간 동안 핵심 미들웨어로 활용되어 왔습니다. 그러나 최근 클라우드 네이티브, 오픈소스 확산, 그리고 IT 비용 절감 요구가 높아지면서 기존 상용 WAS의 구조적 한계와 비용 문제에 대한 재평가가 이루어지고 있습니다. 특히 라이선스 과금 방식, 벤더 종속성, 별도 구매가 필요한 운영 구성 요소 등은 엔터프라이즈 IT가 직면한 주요 과제로 부각되고 있습니다. 이 장에서는 상용 WAS의 비용 구조와 기술적 한계를 구체적으로 분석하고, 엔터프라이즈 IT가 당면한 문제를 심층적으로 조명합니다.

1.1.1 상용 WAS의 과도한 라이선스와 벤더 종속성

상용 WAS는 엔터프라이즈 환경에서 안정성과 신뢰성을 제공하는 미들웨어로 자리잡아 왔으나, 그 비용 구조와 벤더 종속성은 점차 심각한 문제로 인식되고 있습니다. 코어당 과금 모델은 서버의 성능 향상과 클러스터 확장에 따라 비용이 기하급수적으로 증가하는 구조를 가지고 있습니다. 예를 들어, WebLogic, JEUS, WebSphere 등 주요 상용 WAS는 CPU 코어 수에 따라 라이선스 비용이 산정되며, 클라우드 환경에서 Auto-Scaling이나 리소스 집적도를 높이려 할 때 추가 비용이 발생합니다. 이러한 과금 구조는 IT 예산에 지속적인 부담을 주고, 신규 서비스 도입이나 서버 확장 시 비용 예측이 어렵게 만듭니다. 실제로 엔터프라이즈 환경에서는 라이선스 추가 구매가 필수적이므로 장기적으로 TCO가 크게 상승하게 됩니다.

또한 상용 WAS는 기능별 에디션 구분을 통해 업셀 전략을 구사합니다. 기본 라이선스에는 핵심 기능만 포함되고, 클러스터링, 고가용성, 보안, 관리 콘솔 등 고급 기능은 상위 에디션에서만 제공됩니다. 이로 인해 실제 운영 환경에서는 추가 기능 확보를 위해 에디션 업그레이드가 불가피하며, 단일 기능 추가에도 전체 라이선스 업그레이드가 필요해 비용이 폭증합니다. 기능 제한은 운영 효율성 저하와 기술적 종속성을 심화시키는 주요 요인입니다.

벤더 종속성 또한 엔터프라이즈 IT의 유연성을 저해하는 중요한 문제입니다. 상용 WAS는 벤더 독점 API, 관리 콘솔, 배포 도구 등 자체 생태계를 구축하여 고객의 기술 선택을 제한합니다. 표준

Jakarta EE/JEE API 외에 벤더 고유 확장 기능이 많아, 애플리케이션 코드와 운영 환경이 특정 벤더에 종속되는 문제가 발생합니다. 이로 인해 타 WAS나 오픈소스 솔루션으로의 마이그레이션이 어렵고, 장기적으로 벤더의 가격 정책이나 기술 지원에 의존할 수밖에 없습니다. 벤더 종속성은 IT 전략의 유연성을 저해하며, 미래 지향적 아키텍처 설계에 장애가 됩니다.

이러한 구조적 한계는 엔터프라이즈 IT가 클라우드 네이티브, 오픈소스, 비용 절감 등 새로운 패러다임에 대응하는 데 있어 큰 장애물로 작용하고 있습니다. 실제 현장에서는 라이선스 비용 부담과 벤더 종속성으로 인해 신규 프로젝트의 추진이 지연되거나, 기술적 혁신이 제한되는 사례가 빈번하게 발생하고 있습니다.

1.1.2 숨은 비용의 주범: WAS + APM + 세션 클러스터링 별도 구매

상용 WAS 환경에서는 단순히 WAS 라이선스만으로 모든 운영 기능을 사용할 수 없습니다. 실제로 엔터프라이즈 운영에 필수적인 성능 모니터링(APM), 세션 클러스터링, 운영 콘솔 등은 별도의 솔루션 구매가 필요하며, 이로 인해 숨은 비용이 발생합니다. 특히 APM(Application Performance Monitoring)은 트랜잭션 추적, 장애 분석, 성능 진단 등 필수적인 운영 기능을 제공하지만, WAS와 별도 구매 및 연동이 필요하므로 운영 복잡성과 비용이 크게 증가합니다. 예를 들어 Dynatrace, AppDynamics, New Relic 등 주요 APM 솔루션은 WAS와 연동하여 성능 모니터링을 제공하지만, 별도의 라이선스 비용과 연간 유지보수 비용이 추가됩니다. 이로 인해 엔터프라이즈 IT는 예산 부담뿐만 아니라, 운영 복잡성 증가라는 이중의 문제를 겪게 됩니다.

세션 클러스터링 역시 엔터프라이즈 환경에서 사용자 세션의 고가용성 보장을 위해 필수적인 기능입니다. 그러나 상용 WAS는 내장 세션 복제 기능이 제한적이거나, 외부 세션 클러스터링 솔루션(예: Oracle Coherence, IBM WebSphere eXtreme Scale 등)을 별도로 구매해야 합니다. 이로 인해 라이선스 비용뿐 아니라, 네트워크 구성, 관리 포인트 증가, 장애 대응 복잡성이 추가됩니다. 세션 클러스터링은 WAS와 별도 라이선스, 별도 유지보수, 별도 기술 지원이 필요하므로 TCO가 폭증하는 구조적 원인이 됩니다.

이러한 조립식 아키텍처는 WAS, APM, 세션 클러스터링, 운영 콘솔 등 다양한 외부 구성요소가 결합된 '프랑켄슈타인 아키텍처'를 형성합니다. 각 구성요소는 별도 라이선스, 별도 유지보수, 별도 기술 지원이 필요하며, 상호 연동 과정에서 복잡한 설정과 장애 대응이 요구됩니다. 비용 구조는 단일 라이선스가 아닌, 조립식 라이선스와 유지보수의 합산으로 이루어져 예측이 어렵고, 운영

효율성이 저하됩니다. 이러한 구조는 엔터프라이즈 IT의 비용 지뢰밭을 형성하며, TCO 폭증의 주범이 됩니다.

실제 사례를 살펴보면, 국내 대형 금융기관의 WAS 환경에서는 WAS 라이선스 외에도 APM, 세션 클러스터링, 운영 콘솔 등 별도 솔루션 구매로 인해 연간 유지보수 비용이 전체 IT 예산의 30% 이상을 차지하는 경우가 있습니다. 또한 각 솔루션의 연동 과정에서 장애 발생 시 책임 소재가 불명확해져, 복구 시간이 길어지고 운영 신뢰성이 저하되는 문제가 빈번하게 발생합니다. 이러한 숨은 비용과 복잡성은 엔터프라이즈 IT가 새로운 기술 도입이나 클라우드 전환을 추진할 때 큰 장애물로 작용합니다.

1.1.3 수직계열 구조의 기술적 병목

상용 WAS 환경은 다양한 벤더의 솔루션이 Bolted-on 방식으로 결합되어 운영됩니다. WAS, APM, 세션 클러스터링, 운영 콘솔 등 각기 다른 벤더의 제품이 조합되며, 데이터 흐름이 일관되지 않습니다. 트랜잭션 추적, 장애 분석, 성능 진단 등에서 데이터 통합이 어렵고, 운영자는 각 구성 요소의 로그와 모니터링 데이터를 개별적으로 분석해야 합니다. 일관된 데이터 흐름 부재는 장애 원인 분석과 운영 효율성 저하를 초래합니다.

Bolted-on 구조는 관리 포인트가 기하급수적으로 증가하는 문제를 야기합니다. 각 구성요소의 설치, 설정, 업데이트, 장애 대응, 기술 지원이 별도로 이루어지며, 운영자는 복잡한 관리 체계를 유지해야 합니다. 관리 포인트가 많아질수록 장애 발생 시 대응이 어려워지고, 운영 인력의 부담이 증가합니다. 특히 대규모 클러스터 환경에서는 관리 포인트의 폭증이 운영 효율성 저하와 장애 대응의 병목으로 작용합니다.

장애 발생 시 책임 소재가 불명확해지는 것도 중요한 문제입니다. 이종 벤더 솔루션이 결합된 환경에서는 WAS, APM, 세션 클러스터링, 운영 콘솔 등 각 벤더는 자신의 제품에만 책임을 지며, 장애 원인 분석과 해결 과정에서 벤더 간 공방이 발생합니다. 운영자는 장애 대응 과정에서 각 벤더의 기술 지원을 개별적으로 받아야 하며, 장애 복구 시간이 길어지고 MTTR(Mean Time to Repair)이 증가합니다. 책임 소재 불명확은 엔터프라이즈 IT의 운영 신뢰성을 저하시킵니다.

실제 현장에서는 장애 발생 시 WAS 벤더와 APM 벤더, 세션 클러스터링 벤더 간에 원인 분석을 놓고 서로 책임을 미루는 사례가 많습니다. 예를 들어 트랜잭션 지연이나 세션 손실이 발생했을 때, WAS 벤더는 APM 연동 문제를 지적하고, APM 벤더는 WAS 설정 오류를 주장하며, 세션

클러스터링 벤더는 네트워크 장애를 원인으로 제시하는 등 복잡한 공방이 이어집니다. 이로 인해 장애 복구 시간이 길어지고, 운영팀의 부담이 크게 증가합니다. 이러한 수직계열 구조의 기술적 병목은 엔터프라이즈 IT가 안정적이고 효율적으로 운영되기 어렵게 만드는 핵심 원인입니다.

1.2 OPENMARU iAP의 탄생과 설계 철학

OPENMARU iAP는 상용 WAS의 구조적 한계와 비용 폭증 문제를 근본적으로 해결하기 위해 탄생했습니다. 오픈소스 WAS 전문 기업의 기술 축적을 바탕으로, WAS+Web 서버+APM+Cluster+Installer+AI 운영을 단일 플랫폼으로 통합하는 설계 철학을 구현하였습니다. 이 섹션에서는 OPENMARU Inc.의 기술 축적 과정과, 애플리케이션 중심의 통합 플랫폼이라는 해답을 구체적으로 설명합니다.

1.2.1 오픈소스 WAS 전문 기업의 기술 축적

OPENMARU Inc.는 2013년 설립 이후 오픈소스 기반 WAS 및 미들웨어 기술 연구에 집중해 왔습니다. 엔터프라이즈 환경에서 오픈소스의 가능성과 한계를 직접 검증하며, 국내외 표준 기술과 커뮤니티 참여를 통해 지속적으로 역량을 축적했습니다. 초기에는 Red Hat JBoss EAP, Tomcat, WildFly 등 오픈소스 WAS의 커스터마이징과 운영 자동화에 주력하였으며, 엔터프라이즈 요구에 맞는 기능 확장과 안정성 확보에 집중했습니다.

특히 OPENMARU Inc.는 국내외 오픈소스 커뮤니티와의 협업을 통해 최신 기술 트렌드와 표준을 적극적으로 반영하였습니다. 예를 들어, Tomcat의 커스터마이징을 통해 엔터프라이즈 환경에 적합한 고가용성 기능을 개발하고, WildFly의 모듈형 아키텍처를 활용하여 확장성과 유연성을 확보하였습니다. 또한 운영 자동화 분야에서는 Ansible, Kubernetes 등 오픈소스 도구를 활용하여 배포와 관리의 효율성을 높였습니다. 이러한 기술 축적 과정은 단순히 오픈소스 WAS를 도입하는 수준을 넘어, 엔터프라이즈 요구에 맞는 맞춤형 솔루션을 개발하는 기반이 되었습니다.

2014년 공개SW 개발자대회 금상을 수상한 KHAN Session Manager는 세션 클러스터링의 혁신적 솔루션으로, WAS 내장 세션 복제의 한계를 극복하는 외부 세션 저장소 아키텍처를 구현했습니다. KHAN Session Manager는 Redis, Memcached 등 오픈소스 데이터 저장소와 연동하여, 세션의 고가용성과 확장성을 보장합니다. 실제로 국내 대형 금융기관과 공공기관에서 KHAN Session Manager를 도입하여 세션 장애 발생률을 90% 이상 감소시키고, 운영 효율성을

크게 향상시킨 사례가 있습니다.

2015년에는 KHAN APM을 출시하여, WAS와 네이티브 연동되는 성능 모니터링과 트랜잭션 추적 기능을 제공하였습니다. KHAN APM은 JVM 내부의 트랜잭션 흐름을 실시간으로 분석하고, 장애 발생 시 자동으로 원인 분석과 조치 방안을 제시합니다. 이 두 제품은 엔터프라이즈 환경에서 고가용성, 성능 진단, 장애 대응의 핵심 솔루션으로 자리잡았으며, OPENMARU의 기술 축적 기반이 되었습니다.

2021년 GS 1등급 인증을 획득하며, OPENMARU의 제품은 공공기관과 민간기업에서 신뢰받는 미들웨어 솔루션으로 인정받았습니다. GS 인증은 국내 소프트웨어 품질 인증 제도로, 기능성, 신뢰성, 사용성 등 다양한 품질 기준을 충족해야만 획득할 수 있습니다. OPENMARU Inc.는 GS 인증을 통해 제품의 안정성과 신뢰성을 공식적으로 인정받았으며, 공공기관과 금융권 등 까다로운 환경에서도 안정적으로 운영되고 있습니다.

2023년에는 iAP(Intelligent Application Platform)로 리브랜딩하여, WAS+Web 서버+APM+Cluster+Installer+AI 운영을 단일 플랫폼으로 통합하는 설계 철학을 완성하였습니다. 리브랜딩 이후, 클라우드 네이티브 환경과 AI 기반 운영 자동화에 대응하는 혁신적 아키텍처를 제공하며, 엔터프라이즈 IT의 미래를 선도하고 있습니다. OPENMARU iAP는 기존 상용 WAS의 구조적 한계를 극복하고, 오픈소스와 AI 기술을 결합하여 새로운 미들웨어 패러다임을 제시하고 있습니다.

1.2.2 애플리케이션 중심의 통합 플랫폼이라는 해답

OPENMARU iAP는 엔터프라이즈 미들웨어의 핵심 기능을 단일 플랫폼으로 통합함으로써, 기존 상용 WAS의 조립식 라이선스 구조와 기술적 병목을 근본적으로 해결합니다. WAS, Web 서버, APM, Cluster, Installer, AI 운영 등 모든 기능을 하나의 패키지로 제공하여 운영 효율성과 비용 절감 효과를 극대화합니다. 단일 플랫폼은 설치, 설정, 운영, 장애 대응, 기술 지원까지 일관된 체계를 제공하며, 관리 포인트를 최소화하고 데이터 흐름의 일관성을 보장합니다.

OPENMARU iAP의 영구 라이선스 정책은 '1Copy=최대 32Core/1Node'로, 코어당 과금이나 기능별 에디션 업셀 없이 단일 라이선스만으로 모든 기능을 사용할 수 있습니다. 이 정책은 클라우드 환경의 Auto-Scaling, 리소스 집적도 향상, 신규 서비스 도입 시 추가 비용 부담 없이 운영할 수 있도록 설계되었습니다. 경쟁 제품의 라이선스 구조와 비교할 때, OPENMARU iAP는

TCO를 30~40%까지 절감하는 경제적 효과를 제공합니다. 실제로 국내 대형 제조기업에서는 OPENMARU iAP 도입 후 연간 운영 비용을 기존 대비 35% 절감하였으며, 신규 서비스 도입 시 추가 라이선스 구매 없이 확장할 수 있어 IT 예산의 효율적 운용이 가능해졌습니다.

AI 기반 운영 자동화(VibeOps, PromptOps 등)는 OPENMARU iAP의 핵심 혁신 요소입니다. 플랫폼에 네이티브로 통합된 AI 운영 기능은 장애 진단, RCA(Root Cause Analysis), 조치 방안 자동 제공 등 미래 지향적 운영 체계를 구현합니다. 예를 들어 장애 발생 시 AI가 실시간으로 로그와 트랜잭션 데이터를 분석하여 원인과 해결 방안을 제시하며, 운영자는 복잡한 장애 대응 과정을 단축할 수 있습니다. 또한 AI 기반 예측 분석을 통해 성능 저하나 장애 발생 가능성을 사전에 감지하고, 자동으로 리소스 확장이나 조치 작업을 수행할 수 있습니다.

OPENMARU iAP의 아키텍처는 클라우드 네이티브, 오픈소스, AI 운영이 결합되어 엔터프라이즈 IT의 벤더 종속성 탈피, 운영 효율성 극대화, 전략적 의사결정 지원 등 다양한 비즈니스 가치를 제공합니다. 실제로 금융권, 공공기관, 제조기업 등 다양한 산업에서 OPENMARU iAP를 도입하여 운영 효율성과 비용 절감, 장애 대응 능력 향상 등 실질적인 효과를 얻고 있습니다. OPENMARU iAP는 애플리케이션 중심의 통합 플랫폼으로, 엔터프라이즈 IT의 미래를 선도하는 해답을 제시합니다.

2장: OPENMARU iAP 핵심 아키텍처 — Web/WAS·Cluster·APM의 유기적 결합

OPENMARU iAP는 엔터프라이즈 미들웨어의 구조적 한계를 극복하기 위해 Web/WAS, Cluster, APM을 하나의 플랫폼에서 유기적으로 결합한 아키텍처를 제공합니다. 이 장에서는 Jakarta EE 최신 표준 지원, 클라우드 네이티브 경량화, 세션 외부화 기반의 고가용 클러스터, 네이티브 통합 APM의 기술적 깊이 등 OPENMARU iAP의 핵심 아키텍처를 상세히 분석합니다. 각 구성 요소가 어떻게 상호 연계되어 기존 상용 WAS 대비 기술적 우위와 운영 효율성을 제공하는지, 그리고 클라우드 환경에서의 확장성과 자동화, 장애 대응 능력까지 구체적으로 설명합니다.

2.1 경량 Web/WAS: 클라우드 네이티브 시대의 런타임

클라우드 네이티브 환경에서 미들웨어의 경량화와 표준 준수는 서비스 민첩성과 확장성의 핵심 요건입니다. OPENMARU iAP는 Jakarta EE 10 Full Platform 인증과 최신 JDK 지원, 컨테이너/Kubernetes 친화적 설계, 선언적 설정과 GitOps 자동화 등 현대적 런타임 아키텍처를 구현하여 상용 WAS 대비 명확한 차별성을 확보합니다. 이러한 경량 Web/WAS 구조는 클라우드 환경에서의 빠른 배포, 자동 확장, 장애 복구 등 실무적 요구에 최적화되어 있으며, 엔터프라이즈 Java 생태계의 최신 기술 트렌드를 적극 반영하고 있습니다. 본 절에서는 OPENMARU iAP가 제공하는 경량 Web/WAS의 주요 기술적 특징과 상용 WAS 대비 우위, 그리고 클라우드 네이티브 환경에서의 실질적 운영 효과를 상세히 설명합니다.

2.1.1 Jakarta EE 10 Full Platform 인증과 JDK 17/21 지원

OPENMARU iAP는 Jakarta EE 10 Full Platform 인증을 획득하며, EE 11 Preview도 지원합니다. 이는 엔터프라이즈 Java 생태계에서 요구하는 최신 API와 기능을 제공함으로써, 개발자와 운영자가 최신 프레임워크와의 호환성을 보장받을 수 있도록 합니다. Jakarta EE 10은 RESTful API, CDI, JPA, Servlet 6.0 등 핵심 기술의 진화와 함께, 모듈화와 확장성 측면에서 기존 EE 8 대비 큰 발전을 이뤘습니다.

OPENMARU iAP는 JDK 17 및 21의 공식 호환성을 제공하며, ZGC와 Shenandoah와 같은 진보된 가비지 컬렉터(GC)를 활용해 컨테이너 환경에서의 메모리 관리 효율을 극대화합니다. ZGC는 낮은 지연 시간과 빠른 GC Pause를 제공하며, Shenandoah는 대용량 Heap에서도 Full GC 정지를 최소화함으로써 서비스의 안정성과 응답성을 높입니다. 이러한 GC 최적화는 클라우드 환경에서 Auto-Scaling, 장애 복구(RTO) 등 운영 효율성에 직접적인 영향을 미칩니다.

OPENMARU iAP는 Jakarta EE 표준 기반이면서도 Spring Framework 6, Spring Boot 3 등 최신 Java 프레임워크와의 완벽한 호환성을 자랑합니다. 이는 기존 상용 WAS(WebLogic, JEUS, WebSphere 등)에서 발생하는 프레임워크 버전 충돌이나 API 제한 문제를 근본적으로 해결하며, 개발자 생산성과 마이크로서비스 아키텍처 도입을 촉진합니다.

이처럼 OPENMARU iAP는 최신 Java 표준과 JDK 지원을 바탕으로 엔터프라이즈 환경에서의 기술적 유연성과 확장성을 극대화합니다. 예를 들어, Jakarta EE 10의 RESTful API와 CDI

기능은 클라우드 네이티브 서비스 개발에 필수적인 요소이며, JDK 17/21의 GC 최적화는 대규모 트래픽 환경에서도 안정적인 서비스 제공을 가능하게 합니다. 또한 Spring Boot 3와의 호환성은 마이크로서비스 아키텍처 도입 시 개발자의 생산성을 높이고, 기존 상용 WAS의 제한적 API 지원 문제를 해소함으로써 실질적인 운영 효율성을 제공합니다. 실제 사례로, 기존 상용 WAS에서 Spring Boot 3를 도입하려 할 때 API 충돌로 인한 장애가 빈번하게 발생하지만, OPENMARU iAP에서는 이러한 문제가 발생하지 않아 개발 및 운영 과정이 원활하게 진행됩니다. 이러한 기술적 우위는 클라우드 환경에서의 자동화, 확장성, 장애 대응 능력까지 모두 아우르며, 엔터프라이즈 미들웨어의 새로운 표준을 제시합니다.

2.1.2 컨테이너·Kubernetes 적합성과 경량화 설계

OPENMARU iAP는 경량화된 WAS(Web Application Server) 아키텍처를 통해 빠른 부팅 시간과 낮은 메모리/CPU 점유율을 실현합니다. 이는 Kubernetes 환경에서 Pod의 Auto-Scaling, 장애 복구(RTO), 자원 집적도에 직접적으로 영향을 미치며, 대규모 클러스터에서 수십~수백 개의 WAS 인스턴스를 효율적으로 운영할 수 있습니다. 상용 WAS 대비 최소 30~50%의 리소스 절감 효과를 제공합니다.

OPENMARU iAP는 공식 Docker 이미지와 Helm Chart를 제공하여, Kubernetes 및 OpenShift 환경에서 손쉽게 배포 및 운영이 가능합니다. Pod 자동 인식, 서비스 디스커버리, ConfigMap/Secret 연동 등 클라우드 네이티브 기능을 완벽히 지원하며, 롤링 업데이트, Canary 배포, HPA(수평적 오토스케일링)와 같은 현대적 운영 패턴을 적용할 수 있습니다. 이는 상용 WAS의 VM 기반 배포 방식과 비교해 명확한 운영 민첩성 우위를 제공합니다.

상용 WAS(WebLogic, JEUS, WebSphere 등)는 복잡한 라이선스 구조와 높은 리소스 요구량, 느린 기동 속도 등으로 클라우드 환경에서의 확장성에 제약이 많습니다. OPENMARU iAP는 경량화 설계와 컨테이너 친화적 아키텍처를 통해, 클라우드 환경에서의 비용 효율성과 운영 유연성을 극대화합니다.

이러한 경량화와 컨테이너 적합성은 실제 운영 환경에서 큰 차이를 만들어냅니다. 예를 들어, Kubernetes 클러스터 내에서 OPENMARU iAP WAS 인스턴스는 빠른 기동 속도로 롤링 업데이트 시 서비스 중단 없이 배포가 가능하며, Pod Auto-Scaling 기능을 통해 트래픽 변화에 따라 신속하게 인스턴스 수를 조절할 수 있습니다. 또한 Helm Chart를 활용한 자동 배포는 운영자의

수동 개입을 최소화하고, ConfigMap/Secret 연동을 통해 환경별 설정을 안전하게 관리할 수 있습니다. 상용 WAS의 경우 VM 기반 배포와 복잡한 라이선스 관리로 인해 확장성과 민첩성이 크게 저하되지만, OPENMARU iAP는 이러한 제약을 극복하여 클라우드 환경에서의 실질적인 운영 효율성을 제공합니다. 실제로 대규모 금융권 프로젝트에서 OPENMARU iAP를 도입한 사례에서는 기존 상용 WAS 대비 40% 이상의 리소스 절감과 배포 시간 단축, 장애 복구 속도 향상 등의 효과가 확인되었습니다. 이러한 기술적 우위는 클라우드 네이티브 환경에서의 미들웨어 선택 기준을 새롭게 정의하고 있습니다.

2.1.3 선언적 설정과 GitOps 기반 운영 자동화

OPENMARU iAP는 YAML 및 XML 기반의 선언적 설정(Declarative Configuration)을 지원하여, 복잡한 WAS 설정을 코드로 관리할 수 있습니다. 이는 환경별 설정 템플릿화, 버전 관리, 자동 배포 등 운영 자동화의 핵심 기반이 되며, 상용 WAS의 GUI 콘솔 기반 수동 관리 방식과 비교해 명확한 효율성 차이를 보여줍니다.

Git 리포지토리와 연동하여 설정 파일의 버전 관리와 변경 이력을 추적할 수 있으며, GitOps 워크플로우를 통해 변경 사항이 자동으로 배포됩니다. 이는 IaC(Infrastructure as Code)와 DevOps의 핵심 원칙을 실현하며, 운영자의 실수 방지, 일관된 환경 구성, 자동 롤백 등 현대적 운영 패턴을 지원합니다.

상용 WAS는 GUI 콘솔이나 수동 설정 파일 편집에 의존하는 경우가 많아, 대규모 환경에서의 설정 일관성 확보와 자동화에 한계가 있습니다. OPENMARU iAP의 선언적 설정과 GitOps 기반 자동화는 운영 표준화와 관리 효율성에서 명확한 우위를 제공합니다.

이처럼 선언적 설정과 GitOps 기반 운영 자동화는 엔터프라이즈 환경에서의 실질적인 관리 효율성을 크게 향상시킵니다. 예를 들어, YAML 템플릿을 활용하면 환경별 WAS 설정을 코드로 관리할 수 있어, 신규 서비스 배포 시 설정 오류를 최소화하고, 버전 관리 시스템을 통해 변경 이력을 체계적으로 추적할 수 있습니다. GitOps 워크플로우를 적용하면 운영자가 설정 변경을 Git에 커밋하는 것만으로 자동 배포가 이루어지며, 장애 발생 시 자동 롤백 기능을 통해 신속하게 원상 복구할 수 있습니다. 상용 WAS의 경우, GUI 콘솔에서 수동으로 설정을 변경해야 하므로 대규모 환경에서는 일관성 유지가 어렵고, 운영자의 실수로 인한 장애 발생 가능성이 높아집니다. 실제로 OPENMARU iAP를 도입한 대형 제조사에서는 GitOps 기반 운영 자동화로 인해 설정 오

류로 인한 장애가 80% 이상 감소하였으며, 신규 서비스 배포 시간이 절반 이하로 단축되었습니다. 이러한 선언적 설정과 GitOps 자동화는 DevOps, IaC 등 현대적 운영 패턴과 완벽하게 연계되어, 클라우드 네이티브 환경에서의 운영 민첩성과 안정성을 실현합니다.

2.2 In-Memory Data Grid 기반 Cluster: 세션 외부화의 기술적 우위

세션 관리와 클러스터링은 엔터프라이즈 WAS 환경에서 장애 복구, 확장성, 데이터 일관성의 핵심 요소입니다. OPENMARU iAP는 WAS 내장 세션 복제의 한계를 극복하고, 외부 In-Memory Data Grid(IMDG) 기반 세션 클러스터링을 통해 무중단 운영, 선형적 확장성, 이기종 WAS 간 세션 공유 등 기술적 우위를 제공합니다. 본 절에서는 기존 WAS 내장 세션 복제 방식의 구조적 한계와 OPENMARU iAP가 제공하는 외부 IMDG 기반 세션 클러스터링의 아키텍처 및 실무적 이점, 그리고 Stateless 아키텍처를 통한 무중단 운영 보장까지 상세히 설명합니다.

2.2.1 WAS 내장 세션 복제의 구조적 한계

상용 WAS의 내장 세션 복제는 All-to-All 방식으로, 각 WAS 인스턴스가 JVM Heap 내 세션 데이터를 서로 복제합니다. 이 방식은 Full GC 정지, OutOfMemory 오류, 네트워크 병목 등 다양한 기술적 문제를 유발합니다. 특히 대규모 트래픽 환경에서는 세션 데이터가 JVM Heap에 집중되어 GC Pause가 길어지고, 세션 복제 트래픽이 네트워크 대역폭을 소모하여 장애 발생 가능성이 높아집니다.

클라우드 네이티브 환경에서는 동적 스케일 인/아웃이 빈번하게 발생합니다. WAS 내장 세션 복제 방식은 Pod가 Scale-In될 때 세션 데이터 유실, Scale-Out 시 복제 지연, 장애 복구 시 세션 동기화 실패 등 다양한 운영 리스크를 초래합니다. 이는 서비스 가용성과 데이터 일관성에 치명적인 영향을 미칩니다.

내장 복제 방식은 WAS 인스턴스 수가 증가할수록 관리 포인트가 기하급수적으로 늘어나고, 장애 발생 시 책임 소재가 불명확해집니다. 벤더 간 공방, 데이터 흐름의 일관성 부재, 운영자의 복잡한 장애 대응 등 엔터프라이즈 환경에서의 실질적 병목을 초래합니다.

이러한 구조적 한계는 실제 운영 환경에서 다양한 문제를 야기합니다. 예를 들어, 대규모 금융 서비스에서 WAS 내장 세션 복제를 사용하면, 트래픽 급증 시 Full GC로 인한 서비스 지연이 빈번하게 발생하며, 세션 복제 트래픽이 네트워크 병목을 일으켜 장애 대응이 어려워집니다. 또한

Pod Scale-In 시 세션 데이터 유실로 인해 사용자 로그인이 강제로 종료되는 사례가 발생하며, 장애 복구 과정에서 세션 동기화 실패로 데이터 일관성이 깨지는 문제가 자주 보고됩니다. 운영자가 장애 발생 시 원인 파악과 복구에 많은 시간을 소모하게 되며, 벤더 간 책임 소재가 불명확해져 실질적인 장애 대응이 지연됩니다. 이러한 문제는 상용 WAS의 내장 복제 방식이 클라우드 네이티브 환경에 적합하지 않음을 명확히 보여주며, 엔터프라이즈 서비스의 안정성과 확장성을 심각하게 저해합니다.

2.2.2 외부 IMDG 세션 클러스터링의 아키텍처와 이점

OPENMARU iAP는 세션 데이터를 WAS 외부의 In-Memory Data Grid(IMDG)에 저장하는 세션 외부화 아키텍처를 제공합니다. IMDG(Hazelcast, Redis, Apache Ignite 등)는 분산 메모리 기반으로 세션 데이터를 저장하며, WAS 인스턴스 간 세션 공유와 무손실 Failover를 보장합니다.

IMDG 기반 세션 클러스터링은 WAS 인스턴스 장애 발생 시에도 세션 데이터가 외부에 안전하게 저장되어, 서비스 중단 없이 즉시 복구가 가능합니다. 이는 Full GC, OutOfMemory 등 JVM 내부 장애와 무관하게 세션 무손실을 보장하며, 엔터프라이즈급 서비스 가용성(99.9%+)을 실현합니다.

세션 데이터를 JVM Heap에서 분리함으로써 Full GC 문제를 근본적으로 해결하며, IMDG의 샤딩/복제 기능을 통해 예측 가능한 선형적 확장성을 제공합니다. WAS 인스턴스 추가/삭제 시 세션 데이터 동기화가 자동으로 이루어지며, 이기종 WAS 간 세션 공유도 가능합니다.

항목	내장 복제 방식	외부 IMDG 방식
세션 저장 위치	JVM Heap	외부 IMDG(분산 메모리)
GC 영향	Full GC 정지 발생	GC 영향 없음
확장성	관리 포인트 증가	선형적 확장성
장애 복구	세션 유실 가능성	무손실 Failover
이기종 WAS 지원	제한적	완전 지원

외부 IMDG 기반 세션 클러스터링은 실제 운영 환경에서 다양한 이점을 제공합니다. 예를 들어, Hazelcast를 활용한 세션 외부화 아키텍처는 WAS 인스턴스 장애 발생 시에도 세션 데이터가 IMDG에 안전하게 저장되어, 서비스 중단 없이 즉시 복구가 가능합니다. Redis 기반 IMDG는

빠른 데이터 처리와 높은 확장성을 제공하여, 대규모 트래픽 환경에서도 안정적인 세션 관리가 가능합니다. Apache Ignite는 분산 메모리와 샤딩 기능을 통해 수백~수천 개의 WAS 인스턴스 간 세션 데이터를 효율적으로 공유할 수 있습니다. 이러한 외부 IMDG 방식은 Full GC로 인한 서비스 지연이나 세션 유실 문제를 근본적으로 해결하며, WAS 인스턴스 추가/삭제 시 자동 동기화 기능을 통해 선형적 확장성을 실현합니다. 실제로 OPENMARU iAP를 도입한 대형 유통사에서는 외부 IMDG 기반 세션 클러스터링으로 인해 장애 발생 시 서비스 중단 없이 즉시 복구가 가능해졌으며, 이기종 WAS 간 세션 공유로 신규 서비스 도입이 용이해졌습니다. 이러한 기술적 우위는 기존 상용 WAS의 내장 복제 방식과 비교해 엔터프라이즈 환경에서의 안정성, 확장성, 운영 효율성을 크게 향상시킵니다.

2.2.3 Stateless 아키텍처와 무중단 운영 보장

OPENMARU iAP는 WAS를 Stateless로 유지함으로써 Kubernetes의 ReplicaSet 관리가 단순해집니다. Stateful 세션 관리 부담이 없어, HPA(수평적 파드 오토스케일링) 효과가 극대화되고, 롤링 업데이트 시 세션 유실이 발생하지 않습니다. 이는 클라우드 환경에서의 무중단 운영과 자동 확장에 최적화된 패턴입니다.

엔터프라이즈급 세션 관리 기능으로 중복 로그인 방지, 세션 생성 필터링, 세션 만료 정책 등 다양한 고급 기능을 제공합니다. IMDG 기반 세션 클러스터링과 연계하여, 사용자 인증 및 세션 일관성 관리가 강화됩니다.

Stateless 아키텍처는 DevOps, GitOps, IaC 등 현대적 운영 패턴과 완벽히 연계되며, 대규모 클러스터 환경에서의 운영 효율성과 장애 대응 능력을 극대화합니다. 이는 기존 상용 WAS의 복잡한 세션 복제 구조와 비교해 명확한 기술적 우위를 제공합니다.

Stateless 아키텍처는 실제 운영 환경에서 무중단 서비스와 자동 확장에 큰 효과를 발휘합니다. 예를 들어, Kubernetes 환경에서 ReplicaSet을 활용하면 WAS 인스턴스가 Stateless로 동작하여, Pod 추가/삭제 시 세션 데이터 유실 없이 자동으로 확장 및 축소가 가능합니다. 롤링 업데이트 과정에서도 세션 외부화 덕분에 사용자 경험이 저하되지 않으며, 장애 발생 시 신속한 복구가 이루어집니다. 중복 로그인 방지 기능은 IMDG 기반 세션 클러스터링과 연계되어, 동일 사용자의 다중 접속을 효과적으로 제어할 수 있으며, 세션 생성 필터링과 만료 정책은 보안과 운영 효율성을 동시에 확보합니다. 실제로 OPENMARU iAP를 도입한 글로벌 제조사에서는 Stateless

아키텍처와 IMDG 기반 세션 클러스터링을 통해 대규모 클러스터 환경에서의 운영 효율성과 장애 대응 능력이 크게 향상되었으며, DevOps 및 GitOps 패턴과 연계하여 자동화된 운영이 실현되었습니다. 이러한 기술적 우위는 기존 상용 WAS의 복잡한 세션 복제 구조와 비교해 엔터프라이즈 환경에서의 안정성, 확장성, 운영 민첩성을 크게 높여줍니다.

2.3 네이티브 통합 APM: 단일 트랜잭션 가시성의 완성

APM(Application Performance Monitoring)은 엔터프라이즈 미들웨어의 성능 관리와 장애 대응에 필수적인 요소입니다. OPENMARU iAP는 WAS, Cluster, DB까지 단일 트랜잭션 가시성을 제공하며, 커널 레벨 계측과 OpenTelemetry 기반 관측성으로 기존 외부 APM 대비 심층 모니터링 역량을 갖추고 있습니다. 본 절에서는 End-to-End 트랜잭션 추적, 네이티브 통합 APM의 기술적 깊이, OpenTelemetry 기반 관측성의 상관관계 분석 등 OPENMARU iAP의 APM 아키텍처가 제공하는 실질적 운영 효과와 기술적 우위를 상세히 설명합니다.

2.3.1 End-to-End 트랜잭션 추적(Web→WAS→Cluster→DB)

OPENMARU iAP는 사용자 요청이 Web 서버→WAS→세션 Cluster→DB를 거치는 전 과정을 하나의 트랜잭션으로 추적합니다. 트랜잭션 ID를 기반으로 각 단계의 실행 시간, 호출 체인(Call Tree), SQL 쿼리 실행 내역, DB 처리 결과까지 상세히 수집하여, 병목 구간을 직관적으로 식별할 수 있습니다.

APM 콘솔에서 Call Tree를 시각화하여, 서비스별 트랜잭션 흐름과 각 단계의 성능 지표를 한눈에 파악할 수 있습니다. 실행된 SQL 쿼리와 DB 처리 결과를 연동하여, DB 레벨의 병목도 함께 분석할 수 있습니다. 이는 기존 APM의 WAS/DB 분리 모니터링 한계를 극복하는 핵심 기능입니다.

End-to-End 트랜잭션 추적을 통해 병목 구간을 실시간으로 식별하고, 장애 발생 시 원인 파악과 대응이 신속하게 이루어집니다. MTTR(Mean Time To Repair) 단축과 서비스 안정성 향상에 직접적인 효과를 제공합니다.

이러한 End-to-End 트랜잭션 추적 기능은 실제 운영 환경에서 장애 대응과 성능 관리에 큰 효과를 발휘합니다. 예를 들어, 금융권 서비스에서 OPENMARU iAP의 APM을 도입하면, Web→WAS→Cluster→DB 전체 경로의 트랜잭션 흐름을 실시간으로 모니터링할 수 있어, 특정

단계에서 발생하는 지연이나 장애를 즉시 식별할 수 있습니다. Call Tree 시각화 기능은 복잡한 트랜잭션 구조를 한눈에 파악할 수 있게 하며, SQL 쿼리 실행 내역과 DB 처리 결과를 연동하여 DB 병목 구간까지 정확히 분석할 수 있습니다. 기존 외부 APM의 경우 WAS와 DB 모니터링이 분리되어 있어 장애 원인 분석이 어렵지만, OPENMARU iAP의 통합 APM은 단일 트랜잭션 가시성을 제공하여 MTTR을 크게 단축시킵니다. 실제로 대형 유통사에서 OPENMARU iAP APM을 도입한 이후 장애 대응 시간이 50% 이상 단축되었으며, 서비스 안정성이 크게 향상되었습니다. 이러한 End-to-End 트랜잭션 추적은 엔터프라이즈 환경에서의 성능 관리와 장애 대응에 필수적인 기능으로 자리잡고 있습니다.

2.3.2 네이티브 통합의 기술적 깊이: 커널 레벨 계측

OPENMARU iAP의 네이티브 APM은 외부 APM 에이전트가 접근할 수 없는 WAS 커널의 내부 메모리 구조, 세션 그리드 데이터 파이프라인까지 직접 계측합니다. JVM 스레드 상태, Heap Memory, Full GC 이벤트 등 심층 모니터링 정보를 왜곡 없이 포착하여, 운영자의 장애 대응 능력을 극대화합니다.

커널 레벨 계측은 JVM 내부 이벤트와 세션 클러스터 데이터 흐름까지 실시간으로 분석하며, 외부 APM의 표면적 모니터링 한계를 극복합니다. 데이터 정확성과 장애 원인 분석의 신뢰도가 높아, 엔터프라이즈 환경에서의 실질적 운영 효과를 제공합니다.

상용 WAS의 외부 APM은 WAS 내부 구조에 대한 접근이 제한되어, 심층 모니터링과 장애 분석에 한계가 있습니다. OPENMARU iAP의 네이티브 통합은 기술적 깊이와 데이터 신뢰성에서 명확한 우위를 갖습니다.

이처럼 커널 레벨 계측 기능은 실제 장애 대응과 성능 관리에서 큰 차이를 만들어냅니다. 예를 들어, JVM 내부의 Heap Memory와 스레드 상태를 실시간으로 모니터링하면, Full GC 이벤트 발생 시 서비스 지연 원인을 정확히 파악할 수 있으며, 세션 클러스터 데이터 파이프라인을 직접 계측하여 세션 동기화 실패나 데이터 흐름 이상을 신속하게 감지할 수 있습니다. 외부 APM의 경우 WAS 내부 구조에 대한 접근이 제한되어 장애 원인 분석이 표면적 수준에 머무르지만, OPENMARU iAP의 네이티브 APM은 커널 내부까지 심층적으로 분석하여 데이터 신뢰성과 장애 대응 능력을 크게 향상시킵니다. 실제로 OPENMARU iAP를 도입한 대형 제조사에서는 커널 레벨 계측을 통해 장애 발생 시 원인 분석과 복구 시간이 크게 단축되었으며, 운영자의 실무 효율성이 30%

이상 향상되었습니다. 이러한 기술적 깊이는 엔터프라이즈 환경에서의 안정성과 운영 민첩성을 실질적으로 높여주는 핵심 요소입니다.

2.3.3 OpenTelemetry 기반 관측성: Metrics·Logs·Traces 상관관계 분석

OPENMARU iAP는 OpenTelemetry 표준을 기반으로 Metrics, Logs, Traces 3요소를 유기적으로 결합하여 벤더 중립적 데이터 수집을 실현합니다. OTLP(OpenTelemetry Protocol) 기반 Collector 아키텍처를 통해 다양한 모니터링 스택(Prometheus, Grafana, ELK 등)과 연동이 가능합니다.

Metrics(CPU, 메모리, GC), Logs(이벤트 기록), Traces(트랜잭션 추적) 데이터를 통합 분석하여, 장애의 완전한 컨텍스트를 제공합니다. 예를 들어, GC 이벤트 발생 시 트랜잭션 지연과 로그 이벤트를 연계 분석함으로써, 장애 원인과 영향을 정확히 파악할 수 있습니다.

상관관계 분석은 장애 발생 시 원인 파악과 대응 방안을 신속하게 제시하며, 운영자의 업무 효율성과 서비스 안정성을 크게 향상시킵니다. OpenTelemetry 기반 관측성은 클라우드 환경에서의 확장성과 자동화에도 최적화되어 있습니다.

OpenTelemetry 기반 관측성은 실제 운영 환경에서 장애 대응과 성능 관리에 큰 효과를 발휘합니다. 예를 들어, Prometheus와 Grafana를 연동하면 Metrics 데이터를 실시간으로 시각화할 수 있으며, ELK 스택을 활용하면 로그 데이터를 체계적으로 분석할 수 있습니다. Traces 데이터는 트랜잭션 흐름을 상세히 추적하여, 장애 발생 시 Metrics와 Logs 데이터를 연계 분석함으로써 장애 원인과 영향을 정확히 파악할 수 있습니다. OTLP 기반 Collector 아키텍처는 다양한 모니터링 도구와 연동이 가능하여, 벤더 종속성을 최소화하고 운영자의 선택 폭을 넓혀줍니다. 실제로 OPENMARU iAP를 도입한 대형 금융사에서는 OpenTelemetry 기반 관측성으로 장애 대응 시간이 40% 이상 단축되었으며, 서비스 안정성이 크게 향상되었습니다. 이러한 상관관계 분석 기능은 클라우드 환경에서의 확장성과 자동화에도 최적화되어, 엔터프라이즈 미들웨어의 운영 민첩성과 안정성을 실질적으로 높여줍니다.

3장: VibeOps — LLM 기반 지능형 운영과 PromptOps

3.1 VibeOps의 개념과 운영 패러다임 전환

VibeOps는 기존의 IT 운영 방식에서 벗어나 AI와 대규모 언어 모델을 활용한 지능형 운영 체계로의 전환을 목표로 하는 플랫폼입니다. 오늘날 엔터프라이즈 환경에서는 클라우드 네이티브, 마이크로서비스, 대규모 분산 시스템의 확산으로 인해 운영 복잡성이 크게 증가하고 있습니다. 이에 따라 장애 대응, 성능 최적화, 로그 분석 등 반복적이고 인력 의존적인 업무가 늘어나고 있으며, 장애 발생 시 신속한 원인 파악과 조치가 어려워지는 문제가 빈번하게 발생합니다. VibeOps는 LLM, RAG, MCP 등 최신 AI 기술을 접목하여 운영자의 역할을 단순 복구 담당에서 시스템 최적화 지휘관으로 변화시키고, 운영 효율성과 시스템 안정성을 동시에 극대화하는 새로운 패러다임을 제시합니다. 이러한 변화는 조직의 IT 운영 방식에 근본적인 혁신을 가져오며, 사전 예방 중심의 운영 체계로의 전환을 가능하게 합니다.

3.1.1 사후 대응에서 사전 예방으로: IT 운영 패러다임의 전환

수동 로그 분석의 한계

전통적인 IT 운영 환경에서는 장애 발생 후 로그를 수동으로 분석하고, 반복적인 업무를 통해 문제를 해결하는 방식이 일반적이었습니다. 이러한 방식은 운영자의 경험과 직관에 크게 의존하며, 장애 탐지와 복구까지의 시간(MTTR)이 길어지는 문제가 있습니다. 특히 마이크로서비스 아키텍처에서는 서비스 간 복잡한 상호작용으로 인해 장애의 원인 추적이 더욱 어려워집니다. 로그의 분산, 데이터의 단편화, 인력의 전문성 부족 등은 운영 효율성을 저해하는 주요 요인입니다.

이러한 수동 로그 분석 방식은 장애 발생 후에야 문제를 인식하고 대응하는 사후 처리 중심의 운영 문화를 낳게 됩니다. 운영자는 각종 로그 파일을 일일이 확인하고, 시스템의 상태를 파악하기 위해 여러 도구를 병행하여 사용해야 하므로 업무의 복잡성과 피로도가 높아집니다. 또한, 장애의 근본 원인을 파악하는 데 시간이 오래 걸리기 때문에 서비스 중단 시간이 길어지고, 고객 불만이 증가할 수 있습니다. 최근에는 로그 분석 자동화 도구가 일부 도입되고 있지만, 여전히 운영자의 경험과 직관에 의존하는 경향이 강합니다.

반복적 업무와 인력 의존성

운영 업무의 상당 부분이 반복적인 모니터링, 장애 대응, 성능 점검 등으로 구성되어 있습니다. 이 과정에서 인력의 피로도가 높아지고, 업무의 자동화 수준이 낮아 장애 발생 시 신속한 대응이 어렵습니다. 특히 신규 인력의 경우 시스템 전체를 이해하지 못해 장애 대응 능력이 떨어지고, 조직 내 지식의 축적과 공유가 제한됩니다.

반복적인 업무는 운영자의 업무 만족도를 저하시킬 뿐만 아니라, 인력의 이탈을 유발할 수 있습니다. 예를 들어, 운영자는 매일 같은 방식으로 시스템 상태를 점검하고, 장애 발생 시 동일한 절차를 반복하게 됩니다. 이러한 업무 방식은 조직 내 전문 인력의 역량을 충분히 활용하지 못하게 하며, 신규 인력의 경우 복잡한 시스템 구조와 운영 절차를 익히는 데 오랜 시간이 소요됩니다. 또한, 장애 대응 과정에서 발생하는 노하우가 문서화되지 않거나, 조직 내에 제대로 공유되지 않아 지식의 단절이 발생할 수 있습니다.

마이크로서비스 복잡성 증가

마이크로서비스 환경에서는 서비스의 수가 많아지고, 각 서비스가 독립적으로 배포되고 운영됩니다. 이로 인해 장애 발생 시 서비스 간의 상관관계를 파악하고, 전체 시스템의 상태를 종합적으로 분석하는 것이 어려워집니다. 장애의 원인이 특정 서비스에 국한되지 않고, 여러 서비스에 걸쳐 발생할 수 있어 운영자의 부담이 커집니다.

마이크로서비스 아키텍처는 각 서비스가 독립적으로 개발, 배포, 운영되기 때문에 장애 발생 시 원인 추적이 더욱 복잡해집니다. 예를 들어, 하나의 서비스에서 발생한 장애가 다른 서비스에 연쇄적으로 영향을 미칠 수 있으며, 서비스 간 호출 체인이나 데이터 흐름을 정확히 파악하지 못하면 장애의 근본 원인을 찾기 어렵습니다. 또한, 서비스의 수가 많아질수록 로그와 모니터링 데이터가 분산되어 관리되기 때문에, 운영자는 전체 시스템의 상태를 한눈에 파악하기 힘들어집니다. 이러한 복잡성은 운영 효율성 저하와 장애 대응 시간 증가로 이어집니다.

사전 예방 패러다임으로의 전환

VibeOps는 이러한 한계를 극복하기 위해 사전 예방(Proactive) 중심의 운영 패러다임을 제시합니다. AI 기반의 이상 탐지, 자동화된 장애 분석, 예측적 알림 기능을 통해 장애 발생 이전에 잠재적 위험을 탐지하고, 운영자가 시스템 최적화에 집중할 수 있도록 지원합니다. 운영자의 역할은 단순 복구 담당에서 시스템 최적화 지휘관으로 변화하며, 전체 시스템의 안정성과 효율성을 극대화할 수 있습니다.

사전 예방 중심의 운영 체계는 장애 발생 이전에 잠재적 위험을 탐지하고, 자동화된 분석과 예측적 알림을 통해 운영자가 신속하게 대응할 수 있도록 지원합니다. 예를 들어, AI 기반 이상

탐지 기능은 시스템의 성능 지표와 로그 데이터를 실시간으로 분석하여 비정상적인 패턴을 조기에 발견합니다. 자동화된 장애 분석은 복잡한 트랜잭션 흐름과 서비스 간 상관관계를 분석하여 장애의 근본 원인을 빠르게 파악합니다. 예측적 알림 기능은 트래픽 급증, 자원 사용량 증가, 네트워크 지연 등 다양한 위험 요소를 미리 알려주어 운영자가 사전에 조치할 수 있도록 합니다. 이러한 패러다임 전환은 운영 효율성 향상과 시스템 안정성 확보에 크게 기여합니다.

3.1.2 LLM+RAG+MCP 통합 아키텍처

LLM의 역할과 활용

VibeOps의 핵심 기술 스택 중 하나인 LLM(대규모 언어 모델)은 자연어 질의에 대한 이해와 응답, 운영 데이터의 요약 및 분석, 장애 원인 파악 등 다양한 기능을 제공합니다. LLM은 운영자가 “지금 문제의 원인을 알려줘”와 같은 자연어 질의에 대해 APM 데이터, 로그, 트레이스 정보를 종합적으로 분석하여 직관적인 답변을 제공합니다. 이를 통해 운영자의 업무 효율성이 크게 향상되며, 복잡한 시스템 상태를 빠르게 파악할 수 있습니다.

LLM은 기존의 복잡한 쿼리 작성이나 로그 분석 과정을 자연어 질의로 대체함으로써 운영자의 접근성을 높입니다. 예를 들어, 운영자가 장애 원인이나 성능 저하에 대해 질문하면 LLM이 다양한 데이터 소스를 통합 분석하여 명확한 답변을 제공합니다. 또한, LLM은 운영 데이터의 요약과 보고서 자동 생성 기능을 통해 경영진에게 직관적인 정보를 제공할 수 있습니다. LLM의 자연어 처리 능력은 신규 인력의 학습 부담을 줄이고, 조직 내 지식 공유와 협업을 촉진하는 효과도 있습니다.

RAG 기반 검색 증강 생성

RAG(Retrieval-Augmented Generation)은 LLM이 외부 데이터 소스에서 관련 정보를 검색하여 응답의 정확성과 신뢰성을 높이는 기술입니다. VibeOps는 실시간 세션 서버 데이터와 이력 APM 데이터를 동시에 활용하여 RAG 기반의 검색 증강 생성을 구현합니다. 이를 통해 LLM의 할루시네이션(허위 생성) 문제를 최소화하고, 운영 데이터에 기반한 신뢰성 높은 답변을 제공합니다.

RAG는 LLM이 단순히 학습된 지식에만 의존하는 것이 아니라, 실제 운영 데이터와 외부 정보 소스를 실시간으로 검색하여 응답의 근거를 강화합니다. 예를 들어, 장애 원인 분석 시 LLM이 APM 데이터, 로그, 트레이스 정보를 검색하여 정확한 분석 결과를 도출할 수 있습니다. RAG 기반 응답은 운영자의 신뢰도를 높이고, 허위 생성 문제를 줄여 실제 운영 환경에 적합한 솔루션을 제공합니다. 또한, RAG는 실시간 데이터와 과거 이력 데이터를 결합하여 다양한 시나리오에 대응할

수 있는 유연성을 제공합니다.

MCP의 역할과 통합 구조

MCP(Model Context Protocol)는 LLM, RAG, APM, 세션 클러스터링 등 다양한 운영 데이터와 모델 간의 컨텍스트를 표준화하고, 데이터 흐름을 일관되게 관리하는 프로토콜입니다. MCP를 통해 실시간 데이터와 과거 이력 데이터를 유기적으로 결합하고, 개인정보 자동 마스킹, 데이터 소스 선택, 응답 품질 제어 등 다양한 기능을 통합적으로 제공할 수 있습니다. 특히 한국어 개인정보 마스킹 기능은 국내 엔터프라이즈 환경에서 민감 정보 보호를 위한 필수 요소입니다.

MCP는 운영 데이터와 AI 모델 간의 컨텍스트를 표준화하여 데이터 흐름을 일관되게 관리합니다. 예를 들어, 실시간 세션 데이터와 과거 트랜잭션 데이터를 결합하여 장애 분석이나 성능 보고서 작성에 활용할 수 있습니다. MCP는 개인정보 자동 마스킹 기능을 통해 민감 정보가 외부로 유출되는 것을 방지하며, 데이터 소스 선택과 응답 품질 제어 기능을 통해 운영자의 요구에 맞는 최적의 답변을 제공합니다. 특히 한국어 개인정보 마스킹 기능은 국내 엔터프라이즈 환경에서 개인정보 보호법 준수를 위한 필수 요소로, VibeOps의 경쟁력을 높이는 핵심 기술입니다.

하이브리드 LLM 동적 선택과 데이터 소스 구조

VibeOps는 GPT, Claude, Gemini 등 다양한 LLM을 하이브리드로 구성하고, 운영 시나리오에 따라 동적으로 모델을 선택합니다. 실시간 세션 서버와 이력 APM 데이터의 이중 데이터 소스 구조를 통해 장애 탐지, 성능 분석, 용량 계획 등 다양한 운영 업무를 자동화할 수 있습니다. 이중 데이터 소스 구조는 운영자의 자연어 질의에 대해 현재 상태와 과거 이력을 동시에 분석하여 최적의 답변을 제공하는 데 핵심적인 역할을 합니다.

하이브리드 LLM 구성은 운영 환경의 요구에 따라 최적의 모델을 선택할 수 있는 유연성을 제공합니다. 예를 들어, 실시간 장애 탐지에는 GPT 기반 LLM을, 성능 보고서 작성에는 Claude나 Gemini를 활용할 수 있습니다. 이중 데이터 소스 구조는 실시간 세션 서버 데이터와 과거 APM 데이터를 결합하여 장애 탐지, 성능 분석, 용량 계획 등 다양한 운영 업무를 자동화합니다. 운영자는 자연어 질의만으로 현재 상태와 과거 이력을 동시에 분석할 수 있으며, 최적의 답변을 신속하게 받을 수 있습니다. 이러한 구조는 운영 효율성 향상과 시스템 안정성 확보에 크게 기여합니다.

3.2 대화형 질의와 시나리오 기반 자동 분석

VibeOps는 자연어 기반의 대화형 질의와 시나리오 기반 자동 분석 기능을 통해 운영자의 업무 효율성을 극대화합니다. 운영자는 복잡한 시스템 상태나 장애 원인을 직접 탐색하지 않고, “현재 장애의 원인을 알려줘”, “성능 보고서를 작성해줘” 등 자연어로 질의할 수 있습니다. VibeOps는 APM, 트레이스, 로그, 세션 데이터를 종합적으로 분석하여 장애 탐지, 병목 고립, 실행 권장사항, 성능 보고서 자동 생성 등 다양한 시나리오를 자동으로 수행합니다. 이를 통해 MTTR(Mean Time To Repair)을 단축하고, 운영자의 업무 부담을 크게 줄일 수 있습니다.

3.2.1 자연어 기반 장애 원인 분석과 RCA 자동화

장애 탐지 자동화

VibeOps는 실시간 APM 데이터와 트레이스 정보를 활용하여 장애 발생 시 자동으로 이상 상태를 탐지합니다. CPU, 메모리, GC, 네트워크 지표의 급격한 변화, 트랜잭션 지연, 오류 로그 등 다양한 이벤트를 실시간으로 분석하여 잠재적 장애를 신속하게 포착합니다. 장애 탐지 알고리즘은 머신러닝 기반 임계치 분석, 시계열 예측, 상관관계 분석 등 최신 기술을 적용하여 정확도를 높입니다.

장애 탐지 자동화는 운영자의 업무 부담을 줄이고, 장애 발생 시 신속한 대응을 가능하게 합니다. 예를 들어, VibeOps는 실시간으로 수집되는 APM 데이터와 트레이스 정보를 분석하여 CPU 사용량 급증, 메모리 누수, GC 지연, 네트워크 지연 등 다양한 이상 징후를 자동으로 감지합니다. 머신러닝 기반 임계치 분석은 정상 상태와 비정상 상태를 구분하는 기준을 동적으로 학습하며, 시계열 예측은 과거 데이터 패턴을 기반으로 미래의 위험을 예측합니다. 상관관계 분석은 여러 지표 간의 연관성을 파악하여 장애의 근본 원인을 찾는 데 활용됩니다. 이러한 자동화 기능은 장애 발생 시 운영자가 수동으로 로그를 분석하는 시간을 크게 단축시켜 MTTR을 줄이는 효과가 있습니다.

자연어 질의와 RCA 자동화

운영자는 “지금 장애의 원인을 알려줘”와 같은 자연어로 질의하면, VibeOps가 APM 데이터를 분석하여 장애의 원인을 자동으로 요약합니다. RCA(Root Cause Analysis) 엔진은 트랜잭션 콜 트리, 실행 SQL, 세션 상태, 외부 시스템 연동 등 다양한 데이터를 종합적으로 분석하여 장애의 근본 원인을 도출합니다. MTTR(평균 복구 시간)은 기존 수동 분석 대비 60% 이상 단축되는

효과가 있으며, 장애 탐지→요약→원인 파악까지의 전 과정이 자동화되어 운영자의 업무 부담이 크게 줄어듭니다.

RCA 자동화는 운영자가 복잡한 트랜잭션 흐름이나 서비스 간 호출 관계를 일일이 분석하지 않아도, VibeOps가 자동으로 장애의 근본 원인을 도출해줍니다. 예를 들어, 트랜잭션 콜 트리 분석을 통해 장애가 특정 서비스의 SQL 실행 지연에서 비롯된 것임을 확인할 수 있습니다. 세션 상태와 외부 시스템 연동 데이터를 결합하여 장애의 영향 범위와 파급 효과를 분석할 수 있습니다. RCA 엔진은 다양한 데이터 소스를 통합 분석하여 장애의 근본 원인을 명확하게 제시하며, 운영자는 자연어 질의만으로 신속하게 장애 원인을 파악할 수 있습니다. 이로 인해 장애 대응 시간이 크게 단축되고, 운영 효율성이 향상됩니다.

장애 요약 및 보고

VibeOps는 장애 발생 시 주요 원인, 영향 범위, 조치 내역을 자동으로 요약하여 운영자에게 제공합니다. 장애 요약 보고서는 CEO, CTO 등 경영진에게도 직관적으로 전달할 수 있도록 구성되며, 주요 성능 지표 변화, 장애 발생 이슈, 조치 내역 등을 포함합니다. 이를 통해 조직 내 의사결정의 신속성과 정확성을 높일 수 있습니다.

장애 요약 보고서는 장애 발생 시 운영자가 신속하게 경영진에게 상황을 전달할 수 있도록 자동으로 생성됩니다. 보고서에는 장애의 주요 원인, 영향 범위, 조치 내역, 성능 지표 변화 등이 포함되어 있어 경영진이 빠르게 의사결정을 내릴 수 있습니다. 예를 들어, “DB 연결 장애로 인해 전체 트랜잭션의 15%가 지연되었습니다. Connection Pool 크기 조정 및 장애 복구 조치가 완료되었습니다”와 같은 보고서가 자동으로 생성됩니다. 이러한 자동화된 보고서 기능은 조직 내 커뮤니케이션 효율성을 높이고, 장애 대응 과정의 투명성을 확보하는 데 기여합니다.

3.2.2 병목 지점 고립과 실행 권장사항 자동 생성

Trace 분석을 통한 병목 고립

VibeOps는 트랜잭션 Trace 분석을 통해 문제 트랜잭션을 즉시 고립합니다. 서비스 간 호출 체인, SQL 실행 시간, 외부 API 응답 지연 등 병목 구간을 자동으로 식별하고, 문제의 위치와 영향을 명확히 파악합니다. OpenTelemetry 기반의 Trace 데이터는 서비스 간 호출 관계, 지연 시간, 오류율 등 다양한 메트릭을 제공하여 병목 분석의 정확도를 높입니다.

Trace 분석은 장애 발생 시 문제의 원인을 신속하게 파악하는 데 핵심적인 역할을 합니다.

VibeOps는 OpenTelemetry 표준을 기반으로 트랜잭션의 호출 체인, SQL 실행 시간, 외부 API 응답 지연 등 다양한 메트릭을 실시간으로 수집합니다. 예를 들어, 특정 서비스에서 SQL 실행 시간이 급격히 증가하거나, 외부 API 호출이 지연되는 경우 해당 병목 지점을 자동으로 고립합니다. 운영자는 Trace 분석 결과를 통해 문제의 위치와 영향을 명확히 파악할 수 있으며, 장애 대응 시간을 크게 단축할 수 있습니다. 또한, Trace 데이터는 서비스 간 호출 관계와 오류율 등 다양한 정보를 제공하여 병목 분석의 정확도를 높입니다.

실행 권장사항 자동 생성

병목 지점이 고립되면 VibeOps는 Timeout 조정, 외부 시스템 상태 점검, 모니터링 지표 추가 등 구체적인 조치 방안을 자동으로 제시합니다. 예를 들어, “DB 응답 지연이 발생하므로 Connection Pool 크기를 조정하세요”, “API 호출 Timeout을 3초에서 5초로 늘리세요”와 같은 실행 가이드가 자동 생성됩니다. 운영자는 제시된 권장사항을 즉시 적용하거나, 승인 후 자동화 정책으로 반영할 수 있습니다.

실행 권장사항 자동 생성 기능은 운영자가 병목 지점에 대해 신속하게 대응할 수 있도록 구체적인 가이드를 제공합니다. 예를 들어, DB 응답 지연이 발생한 경우 Connection Pool 크기 조정, SQL 튜닝, 인덱스 추가 등 다양한 조치 방안을 제시합니다. 외부 API 호출 지연 시 Timeout 설정 변경, API 캐싱, 네트워크 점검 등 실행 가이드가 자동으로 생성됩니다. 운영자는 제시된 권장사항을 참고하여 즉시 적용하거나, 승인 후 자동화 정책으로 반영할 수 있습니다. 이러한 기능은 장애 대응 시간을 단축하고, 시스템 안정성을 높이는 데 크게 기여합니다.

문제 발견→분석→해결 가이드 자동화

VibeOps는 문제 발견→분석→해결 가이드의 전 과정을 자동화합니다. 장애 탐지 후 RCA, 병목 고립, 실행 권장사항 제시까지 일련의 시나리오가 자동으로 수행되며, 운영자는 자연어 질의와 승인만으로 복잡한 장애 대응 업무를 효율적으로 처리할 수 있습니다. 이로 인해 운영 효율성은 크게 향상되고, 장애 발생 건수와 복구 시간이 동시에 감소합니다.

문제 발견부터 분석, 해결 가이드 제시까지의 자동화는 운영자의 업무 부담을 줄이고, 장애 대응 프로세스의 효율성을 높입니다. 예를 들어, 장애 탐지 후 RCA 자동화, 병목 고립, 실행 권장사항 제시까지 일련의 시나리오가 자동으로 수행됩니다. 운영자는 자연어 질의와 승인만으로 복잡한 장애 대응 업무를 신속하게 처리할 수 있으며, 장애 발생 건수와 복구 시간이 동시에 감소합니다. 이러한 자동화 기능은 조직의 운영 효율성 향상과 시스템 안정성 확보에 크게 기여합니다.

3.2.3 성능 보고서 자동 생성과 용량 계획 지원

성능 보고서 자동 생성

VibeOps는 LLM을 활용하여 주요 성능 지표 변화, 발생 이슈, 조치 내역을 포함한 CEO 보고 형식의 성능 보고서를 자동으로 생성합니다. 운영자는 "이번 달 성능 보고서를 작성해줘"와 같은 자연어 질의만으로 복잡한 보고서를 신속하게 받을 수 있습니다. 보고서에는 트래픽 변화, 장애 발생 현황, 주요 조치 내역, 용량 계획 등 다양한 정보가 포함되어 경영진의 의사결정에 활용됩니다.

성능 보고서 자동 생성 기능은 운영자가 복잡한 데이터를 일일이 정리하지 않아도, LLM이 주요 성능 지표 변화, 장애 발생 현황, 조치 내역, 용량 계획 등 다양한 정보를 종합하여 CEO 보고 형식의 보고서를 자동으로 작성합니다. 예를 들어, "이번 달 트래픽은 20% 증가하였으며, 장애 발생 건수는 3건으로 감소하였습니다. 주요 조치 내역은 WAS 인스턴스 추가, DB Connection Pool 조정, 캐시 TTL 변경 등이 포함됩니다"와 같은 보고서가 자동으로 생성됩니다. 이러한 기능은 경영진의 의사결정에 필요한 정보를 신속하게 제공하며, 조직 내 커뮤니케이션 효율성을 높입니다.

트래픽 증가 예측과 용량 계획 자동화

LLM 기반 예측 모델은 트래픽 증가, WAS 확장 필요성, 스레드 풀 크기 조정, 캐시 최적화 등 용량 계획을 자동으로 지원합니다. 실시간 데이터와 과거 이력 데이터를 종합적으로 분석하여 "다음 달 트래픽이 20% 증가할 것으로 예상됩니다. WAS 인스턴스를 2개 추가하세요"와 같은 구체적인 조치 가이드가 제공됩니다. 용량 계획 자동화는 운영자의 업무 부담을 줄이고, 시스템 안정성을 높이는 핵심 기능입니다.

트래픽 증가 예측과 용량 계획 자동화는 시스템의 안정성을 확보하는 데 중요한 역할을 합니다. LLM 기반 예측 모델은 실시간 데이터와 과거 이력 데이터를 분석하여 트래픽 증가, WAS 확장 필요성, 스레드 풀 크기 조정, 캐시 최적화 등 다양한 용량 계획을 자동으로 지원합니다. 예를 들어, "다음 달 트래픽이 20% 증가할 것으로 예상됩니다. WAS 인스턴스를 2개 추가하세요"와 같은 구체적인 조치 가이드가 제공됩니다. 운영자는 용량 계획 자동화 기능을 통해 업무 부담을 줄이고, 시스템 안정성을 높일 수 있습니다.

성능 최적화 사례

VibeOps는 용량 계획뿐만 아니라 성능 최적화도 자동으로 지원합니다. 예를 들어, "스레드 풀 크기를 32에서 48로 조정하세요", "캐시 TTL을 10분에서 30분으로 늘리세요"와 같은 최적화 권장사항이 자동으로 생성됩니다. 운영자는 제시된 가이드를 참고하여 시스템 성능을 지속적으로

개선할 수 있습니다.

성능 최적화 사례는 운영자가 시스템의 성능을 지속적으로 개선할 수 있도록 구체적인 가이드를 제공합니다. 예를 들어, 트래픽 급증에 대응하여 WAS 인스턴스 추가, 스레드 풀 크기 조정, 캐시 TTL 변경 등 다양한 최적화 권장사항이 자동으로 생성됩니다. 운영자는 제시된 가이드를 참고하여 시스템 성능을 지속적으로 개선할 수 있으며, 장애 발생 건수와 복구 시간을 동시에 감소시킬 수 있습니다. 이러한 기능은 조직의 운영 효율성 향상과 시스템 안정성 확보에 크게 기여합니다.

3.3 PromptOps: 세션-트랜잭션-AI 통합 운영

PromptOps는 세션 클러스터링 데이터와 APM 트랜잭션 데이터를 LLM과 통합하여 자연어 기반 운영을 실현하는 차세대 운영 패러다임입니다. 운영자는 실시간 세션 상태와 과거 트랜잭션 이력을 동시에 분석할 수 있으며, AI 기반의 예방 정비와 예측적 알림 기능을 통해 시스템 안정성과 효율성을 극대화할 수 있습니다. PromptOps는 이중 데이터 소스 구조, HyperLogLog 기반 동시접속자 집계, AI 코파일럿 운영 체계 등 다양한 기술을 결합하여 엔터프라이즈 환경에 최적화된 운영 솔루션을 제공합니다.

3.3.1 PromptOps의 정의와 이중 데이터 소스 구조

PromptOps는 세션 클러스터링의 세션 데이터와 APM의 트랜잭션 데이터를 LLM과 통합하여 자연어 기반 운영을 실현하는 패러다임입니다. 운영자는 “현재 세션 서버 상태를 알려줘”, “지난달 트랜잭션 장애 이력을 분석해줘”와 같은 질의를 통해 실시간 데이터와 과거 이력 데이터를 동시에 분석할 수 있습니다. 이로 인해 장애 탐지, 성능 분석, 용량 계획 등 다양한 운영 업무가 자동화되고, 운영 효율성이 크게 향상됩니다.

PromptOps는 기존의 운영 체계와 달리 실시간 데이터와 과거 이력 데이터를 동시에 활용할 수 있는 이중 데이터 소스 구조를 제공합니다. 실시간 세션 서버 데이터는 현재 상태 분석, 장애 탐지, 동시접속자 집계 등에 활용되며, 과거 APM 시계열 데이터는 트랜잭션 분석, 용량 계획, 성능 보고서 작성 등에 활용됩니다. 이중 데이터 소스 구조는 운영자의 자연어 질의에 대해 현재와 과거를 동시에 분석하여 최적의 답변을 제공하는 데 핵심적인 역할을 합니다. 또한, LLM을 활용하여 운영자의 자연어 질의에 대해 직관적이고 신속한 답변을 제공함으로써 복잡한 SQL, API 호출, 로그 분석 없이 운영자가 원하는 정보를 즉시 얻을 수 있습니다. 이러한 자연어 기반 운영은 신규

인력의 학습 부담을 줄이고, 조직 내 지식 공유와 협업을 촉진하는 효과도 있습니다.

3.3.2 HyperLogLog 기반 동시접속자 집계와 Seasonality 대응

PromptOps는 HyperLogLog 알고리즘을 활용하여 16KB 메모리만으로 수억 명의 고유 사용자를 집계할 수 있습니다. HyperLogLog는 해시 기반의 확률적 집계 방식으로, 대규모 분산 환경에서도 정확한 동시접속자 수를 빠르게 계산할 수 있습니다. 메모리 효율성과 집계 정확도가 뛰어나 엔터프라이즈 환경에 최적화된 집계 솔루션입니다.

동시접속자 집계는 2초→1분→5분→1시간 단위로 롤업되어 저장됩니다. 이를 통해 실시간 트래픽 변화와 장기적인 Seasonality(계절성, 패턴 변화)를 동시에 분석할 수 있습니다. PromptOps는 IP, JSESSIONID, KHANUSER 쿠키 등 3가지 사용자 식별 모드를 활용하여 다양한 패턴 분석이 가능합니다. 트래픽 급증, 이벤트 발생, 계절적 변동 등 다양한 시나리오에 대응할 수 있습니다.

HyperLogLog 기반 동시접속자 집계는 대규모 분산 환경에서도 효율적으로 고유 사용자 수를 계산할 수 있는 기술입니다. 16KB 메모리만으로 수억 명의 동시접속자를 집계할 수 있으며, 해시 기반의 확률적 집계 방식은 메모리 사용량을 최소화하면서도 집계 정확도를 높입니다. 롤업 데이터 구조는 2초, 1분, 5분, 1시간 단위로 데이터를 저장하여 실시간 트래픽 변화와 장기적인 Seasonality를 동시에 분석할 수 있습니다. IP, JSESSIONID, KHANUSER 쿠키 등 다양한 사용자 식별 모드를 활용하여 트래픽 급증, 이벤트 발생, 계절적 변동 등 다양한 시나리오에 대응할 수 있습니다. Seasonality 분석은 트래픽 패턴 변화, 이벤트 대응, 용량 계획 등 운영 시나리오에 활용되며, 운영자는 자연어 질의로 Seasonality 대응 전략을 자동으로 수립할 수 있습니다.

3.3.3 AI 코파일럿 기반 운영 체계: 예방 정비와 예측적 알림

PromptOps는 AI 코파일럿 기능을 통해 과거 데이터와 현재 상태를 분석하여 예방 정비를 자동으로 수행합니다. 예를 들어, “3시간 뒤 메모리 사용량이 임계치를 초과할 확률이 85%”와 같은 예측적 알림을 제공하며, 운영자는 자동 대응 정책과 승인 경계를 설계할 수 있습니다. 예방 정비 자동화는 장애 발생 이전에 잠재적 위험을 탐지하고, 시스템 안정성을 높이는 핵심 기능입니다.

AI 코파일럿은 실시간 데이터와 과거 이력 데이터를 종합적으로 분석하여 예측적 알림을 제공합니다. CPU, 메모리, 네트워크, 트랜잭션 지표의 변화 패턴을 분석하여 “다음 달 트래픽 급증이 예상됩니다”, “메모리 사용량이 임계치에 근접하고 있습니다”와 같은 알림을 자동으로 생성합니다.

운영자는 알림에 따라 대응 정책을 수립하고, 시스템 확장, 자원 최적화, 장애 예방 등 다양한 조치를 신속하게 수행할 수 있습니다.

AI 코파일럿 기반 운영 체계는 운영자의 역할을 단순 복구 담당에서 운영 체계 설계자로 전환시킵니다. 운영자는 자동 대응 정책, 승인 경계, 예측적 알림 설정 등 시스템 전체의 운영 체계를 설계하고, AI 코파일럿과 협업하여 시스템 최적화에 집중할 수 있습니다. 예방 정비 자동화와 예측적 알림 기능은 장애 발생 이전에 잠재적 위험을 탐지하고, 운영자가 신속하게 대응할 수 있도록 지원합니다. 예를 들어, AI 코파일럿이 “3시간 뒤 메모리 사용량이 임계치를 초과할 확률이 85%”와 같은 예측적 알림을 제공하면, 운영자는 자동 대응 정책과 승인 경계를 설계하여 장애 예방에 집중할 수 있습니다. 이러한 운영 체계 설계자 역할 전환은 조직의 운영 효율성과 시스템 안정성 확보에 크게 기여합니다.

4장: 경쟁 제품 비교와 도입 효과 — 왜 OPENMARU iAP인가

4.1 차세대 미들웨어 평가 프레임워크

최근 엔터프라이즈 미들웨어 시장에서는 단순한 기술적 스펙이나 기능 목록 중심의 평가가 점차 현실적 운영 요구와 비즈니스 가치 중심으로 변화하고 있습니다. 실제 운영 환경에서는 장애 발생 시 신속한 대응, 유지보수의 효율성, 기술 지원의 신뢰성, 그리고 총소유비용(TCO) 절감이 핵심적인 평가 요소로 부상하고 있습니다. OPENMARU iAP는 이러한 변화된 요구에 맞춰 설계된 차세대 미들웨어 플랫폼으로, 클라우드 환경에서의 컨테이너 최적화, IaC 자동화, PaaS 적합성 등 실질적인 운영 편의성과 비용 효율성을 동시에 제공합니다. 본 장에서는 기존 WAS 솔루션과 OPENMARU iAP를 현실적 관점에서 비교하고, 도입 효과를 구체적으로 분석합니다.

4.1.1 ‘운영 현실성’ 기반 비교: 유지보수, 장애 대응, TCO

운영 현실성 기반의 미들웨어 평가에서는 유지보수의 난이도가 가장 중요한 요소 중 하나로 꼽힙니다. 전통적인 상용 WAS는 복잡한 설정, 수동 관리, 벤더별 독점 API 등으로 인해 운영자의 부담이 크며, 장애 발생 시에도 복잡한 구조와 벤더 간 책임 공방으로 인해 신속한 대응이 어렵습니다.

OPENMARU iAP는 선언적 설정과 통합 관리 콘솔을 제공하여 운영 복잡성을 획기적으로 줄이고, 장애 발생 시 네이티브 통합 APM과 VibeOps AI 분석 기능이 결합되어 신속한 원인 파악과 조치가 가능합니다.

OPENMARU iAP의 유지보수 편의성은 통합 관리 콘솔과 자동화된 운영 기능에서 두드러집니다. 예를 들어, 운영자는 복잡한 설정 파일을 직접 편집할 필요 없이 선언적 YAML 파일을 통해 환경을 구성할 수 있으며, GitOps 기반의 운영 패턴을 활용하여 변경 이력 관리 및 롤백이 용이합니다. 장애 대응 측면에서는 네이티브 APM이 실시간으로 트랜잭션을 추적하고, VibeOps AI가 장애 패턴을 분석하여 원인 파악 및 해결 가이드를 자동으로 제공합니다. 이러한 기능은 기존 상용 WAS의 외부 APM 연동 한계와 세션 클러스터링의 복잡성을 극복하며, MTTR(Mean Time To Recovery)을 획기적으로 단축합니다.

기술 지원의 질 역시 OPENMARU iAP의 강점 중 하나입니다. 오픈소스 기반의 투명한 지원 체계와 GS 인증, AWS Marketplace 등록 등 신뢰성 있는 지원을 제공하며, SLA(Service Level Agreement) 이상의 실질적 지원을 보장합니다. TCO 측면에서는 영구 라이선스 정책과 통합 기능 제공으로 별도 APM, 세션 클러스터링, 운영 콘솔 구매가 불필요하여, 기존 상용 WAS 대비 30~40%의 비용 절감 효과를 실현합니다. 실제로 국내 금융기관, 공공기관 등에서 OPENMARU iAP 도입 후 운영 인력의 부담이 줄고 장애 대응 속도가 빨라졌다는 평가가 이어지고 있습니다.

4.1.2 클라우드 전환 요건: 컨테이너 최적화, IaC, PaaS 적합성

클라우드 네이티브 환경에서는 경량 아키텍처와 빠른 기동 속도가 필수적입니다. OPENMARU iAP는 오픈소스 기반의 경량 WAS와 외부 세션 클러스터링, 통합 APM을 결합하여 Kubernetes, Docker, OpenShift 등 다양한 컨테이너 플랫폼에서 최적의 실행 효율성을 보장합니다. Auto-Scaling, 장애 복구(RTO) 등 클라우드 특화 기능을 지원함으로써 자원 집적도와 운영 민첩성을 높입니다.

OPENMARU iAP는 컨테이너 환경에서의 최적화된 실행을 위해 공식 컨테이너 이미지와 Helm Chart, Kustomize 템플릿을 제공합니다. 이를 통해 Kubernetes 환경에서 손쉽게 배포가 가능하며, ReplicaSet 관리와 HPA(수평적 파드 오토스케일링)에 최적화된 아키텍처를 갖추고 있습니다. 또한, 롤링 업데이트와 스케일 인/아웃 기능을 통해 서비스 중단 없이 확장 및 축소가 가능합니다. 기존 상용 WAS는 컨테이너 환경에서의 배포가 제한적이며, 별도의 구성요소 연동과

복잡한 설치 절차로 인해 운영 민첩성이 떨어집니다.

IaC(Infrastructure as Code) 자동화는 현대 IT 운영의 핵심입니다. OPENMARU iAP는 Terraform, Ansible 등 주요 IaC 도구와의 연동을 지원하며, 선언적 YAML 설정과 GitOps 기반 운영 패턴을 통해 환경 표준화와 자동 프로비저닝을 실현합니다. 예를 들어, 운영자는 Terraform 스크립트를 통해 인프라 리소스를 자동으로 생성하고, Ansible 플레이북을 활용하여 WAS 환경을 일관되게 배포할 수 있습니다. 이는 기존 상용 WAS의 수동 GUI 콘솔 관리 방식과 차별화되는 지점으로, 운영 효율성과 표준화 수준을 한층 높여줍니다.

PaaS 환경에서는 멀티테넌시, Auto-Scaling, Auto-Healing 등 플랫폼 특화 기능이 요구됩니다. OPENMARU iAP는 세션 외부화와 Stateless 아키텍처를 통해 ReplicaSet 관리와 HPA에 최적화되어 있으며, 다양한 클라우드 플랫폼(AWS, Azure, GCP, NCP)과의 호환성을 갖추고 있어 공공기관 및 대규모 민간 도입에 적합합니다. 실제로 OPENMARU iAP는 공공 클라우드 규정 준수와 대규모 트래픽 처리에 최적화된 구조를 제공하며, 멀티테넌시 지원을 통해 여러 서비스가 하나의 플랫폼에서 안정적으로 운영될 수 있도록 설계되어 있습니다.

4.2 핵심 요구사항별 상세 비교

OPENMARU iAP와 주요 경쟁 솔루션(WAS 그룹별) 간의 표준 적합성, 운영 민첩성, 세션 클러스터링, APM 통합, AI 연계성, TCO 등 핵심 요구사항을 구체적으로 비교합니다. 최신 표준 대응, 세션 외부화 아키텍처, 네이티브 APM, VibeOps AI 운영 등 OPENMARU iAP만의 차별적 강점을 표와 함께 제시하며, 실제 도입 사례와 기술적 세부사항을 통해 각 요구사항별 경쟁력의 차이를 분석합니다.

4.2.1 표준 적합성과 운영 민첩성 비교

OPENMARU iAP는 Jakarta EE 10 Full Platform 인증과 EE 11 Preview를 신속히 지원합니다. 글로벌 상용 WAS(WebLogic, WebSphere), 국내 상용 WAS(JEUS), 오픈소스 상용 WAS(JBoss, Tomcat) 대비 최신 표준 대응 속도가 빠르며, Spring Framework 6/Spring Boot 3 등 최신 프레임워크와의 호환성이 뛰어납니다. 실제로 OPENMARU iAP는 Jakarta EE 11 Preview 기능을 빠르게 반영하여, 최신 표준 API와 보안 패치, 성능 개선 사항을 즉시 적용할 수 있습니다. 이는 기존 상용 WAS가 표준 대응에 있어 벤더 내부 검증과 라이선스 정책으로 인해

느린 반영 속도를 보이는 것과 대조적입니다.

컨테이너 이미지 제공 및 IaC 지원 측면에서도 OPENMARU iAP는 공식 컨테이너 이미지와 Helm Chart, Kustomize 템플릿을 제공하여 Kubernetes 환경에 즉시 배포가 가능합니다. IaC 자동화 지원(Terraform, Ansible 연동)은 글로벌 상용 WAS 대비 높은 수준으로, 운영 민첩성과 프로비저닝 속도에서 우위를 보입니다. 예를 들어, 운영자는 Terraform 스크립트로 인프라를 자동 생성하고, Ansible을 활용하여 WAS 환경을 일관되게 배포할 수 있습니다. 이는 기존 상용 WAS가 수동 설치와 복잡한 라이선스 등록, 별도 구성요소 연동으로 인해 초기 구축 시간이 길고, 운영 효율성이 떨어지는 것과 차별화됩니다.

프로비저닝 속도 비교에서는 OPENMARU iAP가 Installer 및 CLI(lenactl.sh)를 통한 자동화된 설치·복제 기능으로 'Days to Hours' 수준의 프로비저닝을 실현합니다. 기존 상용 WAS는 복잡한 설치 절차와 라이선스 등록, 별도 구성요소 연동으로 인해 초기 구축 시간이 길며, 운영자가 직접 여러 단계를 거쳐야 하는 번거로움이 있습니다. OPENMARU iAP는 자동화된 설치와 복제 기능을 통해 운영자의 부담을 줄이고, 신속한 환경 전환이 가능합니다.

아래 비교표는 각 솔루션 그룹별 표준 적합성과 운영 민첩성의 차이를 정리한 것입니다.

솔루션 그룹	Jakarta EE 11 대응	컨테이너 이미지	IaC 자동화	프로비저닝 속도
글로벌 상용 WAS	느림	제한적	제한적	Days~Weeks
국내 상용 WAS	보통	제한적	제한적	Days~Weeks
오픈소스 상용 WAS	빠름	있음	있음	Hours~Days
OPENMARU iAP	매우 빠름	있음	있음	Hours

실제 도입 사례에서는 OPENMARU iAP가 금융권, 공공기관 등에서 신속한 표준 대응과 자동화된 프로비저닝으로 구축 기간을 단축하고, 운영 민첩성을 높인 것으로 평가받고 있습니다.

4.2.2 세션 외부화·APM 통합·AI 운영 연계성 비교

세션 외부화 방식의 비교에서 글로벌/국내 상용 WAS는 대부분 JVM Heap 내장 세션 복제 방식(All-to-All)을 사용하며, Full GC 정지, OutOfMemory, 네트워크 병목 등 구조적 한계가 있습니다. 이러한 방식은 세션 데이터가 JVM 내부에 저장되어 GC(가비지 컬렉션)로 인한 서비스 중단, 장애 발생 시 세션 손실, 확장 시 네트워크 병목 등의 문제가 빈번하게 발생합니다. OPENMARU

iAP는 외부 IMDG(In-Memory Data Grid) 기반 세션 클러스터링으로 세션 무손실, GC 문제 근본 해결, 이기종 WAS 간 세션 공유, 선형적 확장성을 제공합니다. 실제로 IMDG 기반 세션 클러스터링은 대규모 트래픽 환경에서 안정적인 세션 관리와 빠른 확장성을 보장하며, 장애 발생 시에도 세션 데이터가 외부에 저장되어 서비스 무중단 전환이 가능합니다.

APM 통합 수준의 비교에서는 상용 WAS는 별도 APM 구매와 연동이 필요하며, WAS 커널 내부 계측이 불가능합니다. 이는 운영자가 장애 원인 분석이나 성능 모니터링을 위해 추가 비용과 복잡한 연동 작업을 해야 하는 단점이 있습니다. OPENMARU iAP는 네이티브 APM을 내장하여 End-to-End 트랜잭션 추적, 커널 레벨 계측, OpenTelemetry 기반 관측성 통합을 실현합니다. 예를 들어, 운영자는 별도의 APM 라이선스 없이 실시간 트랜잭션 추적과 장애 분석을 수행할 수 있으며, OpenTelemetry 표준을 활용한 데이터 연동으로 다양한 모니터링 도구와의 통합이 가능합니다.

AI 운영 연계성의 비교에서는 AI 기반 운영(VibeOps)은 OPENMARU iAP만이 제공하는 기능으로, LLM+RAG+MCP 통합 아키텍처를 통해 장애 진단, RCA 자동화, 예측적 알림, 운영 보고서 자동 생성 등 지능형 운영을 실현합니다. 경쟁 솔루션은 AI 연계 기능이 미제공 또는 제한적이며, OPENMARU iAP는 VibeOps AI를 통해 자연어 질의 기반 장애 분석, 예측적 알림, 자동화된 운영 보고서 생성 등 차별적 기능을 제공합니다. 실제 운영 환경에서는 VibeOps AI가 장애 발생 시 원인 파악과 해결 가이드를 자동으로 제시하여 운영자의 부담을 줄이고, 시스템 안정성을 높이는 효과를 발휘합니다.

아래 비교표는 세션 외부화, APM 통합, AI 운영 연계성의 차이를 정리한 것입니다.

솔루션 그룹	세션 외부화 방식	APM 통합 수준	AI 운영 연계성
글로벌 상용 WAS	내장 복제	별도 구매	미제공
국내 상용 WAS	내장 복제	별도 구매	미제공
오픈소스 상용 WAS	내장 복제/외부화	별도 구매/연동	제한적
OPENMARU iAP	외부 IMDG	네이티브 통합	VibeOps 제공

실제 사례에서는 OPENMARU iAP의 외부 세션 클러스터링과 네이티브 APM, VibeOps AI 연계 기능이 장애 발생 시 빠른 대응과 서비스 무중단 운영을 가능하게 하며, 운영 효율성과 안정성을 동시에 확보할 수 있습니다.

4.2.3 TCO 실질 비용 구조 비교

초기 도입 비용과 라이선스 정책 측면에서 상용 WAS는 코어당 과금, 에디션 업셀, 연간 유지보수 등 복잡한 라이선스 구조로 초기 도입 비용이 높습니다. OPENMARU iAP는 1Copy=최대 32Core/1Node 영구 라이선스 정책으로, 단일 라이선스에 WAS+Web+APM+Cluster+Installer+AI 운영이 모두 포함됩니다. 이는 운영자가 별도의 라이선스 업그레이드나 추가 비용 없이 모든 기능을 사용할 수 있는 구조로, 도입 비용의 예측 가능성과 비용 절감 효과를 극대화합니다.

포함 기능 범위와 유지보수 비용 측면에서 상용 WAS는 기능별 조립식 구매(별도 APM, 세션 클러스터링, 운영 콘솔)로 유지보수 비용이 누적됩니다. OPENMARU iAP는 단일 플랫폼에 모든 기능이 통합되어 추가 비용이 발생하지 않으며, 기술 지원 서비스도 GS 인증, 조달청 등록 등 신뢰성 기반으로 제공됩니다. 실제로 OPENMARU iAP 도입 기관에서는 유지보수 비용이 기존 상용 WAS 대비 30~40% 절감되었으며, 운영 인력의 부담도 줄어드는 효과를 경험하고 있습니다.

TCO 절감 효과의 수치는 OPENMARU iAP가 단일 라이선스 구조와 통합 기능 제공으로 기존 상용 WAS 대비 TCO를 30~40% 절감할 수 있다는 점에서 두드러집니다. 이는 라이선스 비용, 유지보수 비용, 운영 인력 비용, 장애 대응 비용의 합산 결과로 산출되며, 실제 도입 사례에서는 운영 비용 절감과 효율성 향상, 장애 대응 속도 개선이 동시에 이루어지고 있습니다.

아래 비교표는 각 솔루션 그룹별 TCO 실질 비용 구조를 정리한 것입니다.

솔루션 그룹	초기 도입 비용	라이선스 정책	포함 기능 범위	유지보수/기술지원 비용	TCO 절감 효과
글로벌 상용 WAS	매우 높음	서브스크립션	조립식	높음	낮음
국내 상용 WAS	높음	서브스크립션	조립식	높음	낮음
오픈소스 상용 WAS	보통	영구/서브스크립션	제한적	보통	보통
OPENMARU iAP	낮음	영구	단일	낮음	30~40%

실제 운영 환경에서는 OPENMARU iAP 도입 후 라이선스 비용, 유지보수 비용, 장애 대응 비용이 크게 줄어들고, 운영 효율성이 향상되는 효과가 확인되고 있습니다.

4.3 도입 효과와 비즈니스 가치

OPENMARU iAP 도입을 통해 실제로 얻을 수 있는 운영 효율성, 시스템 안정성, 전략적 가치에 대해 정량적·정성적으로 분석합니다. 네이티브 APM과 VibeOps AI 분석 결합으로 MTTR 단축, 장애 감소, 서비스 가용성 향상, 벤더 종속성 탈피, 비즈니스 민첩성 확보 등 다양한 효과를 구체적으로 제시하며, 실제 도입 사례와 기술적 세부사항을 통해 비즈니스 가치의 실질적 향상을 설명합니다.

4.3.1 운영 효율성: MTTR 60% 단축, 장애 발생 40% 감소

OPENMARU iAP는 End-to-End 트랜잭션 추적과 VibeOps의 LLM 기반 AI 분석이 결합되어 장애 탐지→요약→원인 파악→해결 가이드의 전 과정을 자동화합니다. 운영자는 자연어 질의로 장애 원인을 즉시 파악할 수 있으며, 병목 지점 고립과 실행 권장사항이 자동 생성됩니다. 예를 들어, 장애 발생 시 운영자는 “최근 1시간 내 트랜잭션 병목 원인과 해결 방안”을 자연어로 질의하면 VibeOps AI가 로그 분석, 트랜잭션 추적, 성능 지표를 종합하여 원인과 해결 가이드를 자동으로 제공합니다.

MTTR 단축 효과는 실제 운영 환경에서 OPENMARU iAP 도입 후 MTTR이 60% 이상 단축된 사례가 보고되고 있습니다. 이는 네이티브 APM의 실시간 모니터링과 AI 기반 RCA 자동화, 운영 표준화가 결합된 결과입니다. 예를 들어, 금융기관에서는 장애 발생 시 기존 WAS 대비 복구 시간이 절반 이하로 줄었으며, 운영자가 반복적 장애 대응에서 벗어나 시스템 최적화와 예방 정비에 집중할 수 있게 되었습니다.

장애 발생 건수 감소 역시 OPENMARU iAP의 선제적 이상 탐지와 예측적 알림 기능으로 장애 발생 건수가 40% 이상 감소합니다. VibeOps AI는 실시간으로 이상 패턴을 탐지하고, 예측적 알림을 통해 운영자가 사전 대응 정책을 설계할 수 있도록 지원합니다. 실제로 공공기관에서는 장애 발생 건수 감소와 운영 효율성 향상으로 사용자 만족도가 높아졌으며, 운영 인력의 부담이 줄어드는 효과를 경험하고 있습니다.

4.3.2 시스템 안정성: 서비스 가용성 99.9%+ 확보

OPENMARU iAP는 외부 세션 클러스터링을 통해 Failover 시 세션 무손실을 보장하며, APM의 실시간 모니터링은 장애 발생 즉시 탐지와 대응을 가능하게 합니다. 롤링 업데이트, 스케일 인/아웃 등 클라우드 환경에서도 서비스 중단 없이 운영이 가능하며, 실제로 대규모 트래픽 환경에서도 안정적인 서비스 가용성을 확보할 수 있습니다.

예측적 알림과 안정성 확보 측면에서는 VibeOps AI 코파일럿이 “3시간 뒤 메모리 사용량 임계치 초과 확률 85%”와 같은 예측적 알림을 제공하여, 운영자가 사전 대응 정책을 설계할 수 있습니다. 이를 통해 서비스 가용성 99.9% 이상을 달성하며, 공공기관에서는 규정 준수와 사용자 만족도 향상 효과가 기대됩니다. 실제 도입 사례에서는 OPENMARU iAP의 예측적 알림 기능이 장애 예방과 신속한 대응에 크게 기여하고 있으며, 서비스 중단 없이 안정적인 운영이 가능하다는 평가가 이어지고 있습니다.

OPENMARU iAP의 시스템 안정성은 외부 세션 클러스터링, 네이티브 APM, VibeOps AI 연계 기능이 결합되어 장애 발생 시 빠른 대응과 서비스 무중단 운영을 가능하게 하며, 운영 효율성과 안정성을 동시에 확보할 수 있습니다.

4.3.3 전략적 가치: 벤더 종속성 탈피와 비즈니스 민첩성

OPENMARU iAP는 오픈소스 기반 아키텍처로 벤더 종속성에서 탈피할 수 있습니다. 표준 API 지원, 커뮤니티 기반 기술 발전, 투명한 라이선스 정책은 IT 의사결정자에게 전략적 자유도를 제공하며, 실제로 OPENMARU iAP 도입 기관에서는 벤더 종속성에서 벗어나 다양한 기술 선택과 확장, 표준화된 운영 정책을 활용할 수 있게 되었습니다.

자동화 프로비저닝과 민첩성 확보 측면에서는 Installer 및 CLI 자동화로 ‘Days to Hours’ 수준의 구축이 가능하며, IaC 연동과 GitOps 운영으로 비즈니스 민첩성이 극대화됩니다. 신속한 환경 전환과 확장, 표준화된 운영 정책은 변화하는 시장 요구에 빠르게 대응할 수 있게 하며, 실제로 금융권, 공공기관 등에서 OPENMARU iAP 도입 후 환경 전환과 확장 속도가 크게 빨라졌다는 평가가 이어지고 있습니다.

데이터 기반 전략적 의사결정 지원에서는 VibeOps LLM 분석이 운영 데이터, 장애 이력, 성능 지표를 종합적으로 분석하여 CEO 보고서, 용량 계획, 최적화 가이드 등 전략적 의사결정에 필요한 정보를 자동으로 제공합니다. 데이터 기반 의사결정은 경쟁력 있는 IT 조직을 만드는 핵심

요소이며, 실제 도입 사례에서는 VibeOps LLM 분석을 활용한 전략적 의사결정이 조직의 경쟁력 향상과 효율적 운영에 기여하고 있습니다.

5장: 도입 가이드 — 설치 자동화, 마이그레이션, 운영 표준화

OPENMARU iAP의 도입은 기존 상용 WAS 환경에서의 복잡한 설치와 운영을 혁신적으로 단순화하고, 표준화된 환경과 자동화된 프로비저닝을 통해 운영 효율성과 보안, 확장성을 극대화합니다. 본 장에서는 Installer를 통한 설치 자동화, 마이그레이션 프로세스, 운영 표준화 패턴, 기술 지원 체계 등 실제 도입에 필요한 모든 핵심 절차와 기술적 세부사항을 다룹니다. 이를 통해 IT 조직은 고가 상용 WAS의 복잡성과 비용 부담에서 벗어나, 클라우드 네이티브와 DevOps에 최적화된 미들웨어 환경을 신속하게 구축할 수 있습니다.

5.1 OPENMARU Installer: 자동화된 프로비저닝과 환경 표준화

OPENMARU iAP Installer는 기존 미들웨어 설치의 복잡함을 해소하고, 운영 환경의 표준화와 자동화된 프로비저닝을 실현하는 핵심 도구입니다. Installer는 웹 UI와 CLI를 모두 지원하여 다양한 운영 시나리오에 맞게 서버 설치, 설정 복제, 리비전 관리 등 주요 기능을 제공합니다. 이를 통해 대규모 환경에서도 일관된 미들웨어 배포와 운영이 가능하며, 보안과 품질을 유지하는 데 중요한 역할을 합니다. 특히 프로파일 기반 설정과 리비전 관리 기능은 운영 환경의 안정성과 신속한 장애 복구를 보장합니다.

5.1.1 서버 복제(Clone)와 프로파일 기반 설정 표준화

OPENMARU Installer는 직관적인 웹 UI(Manager Console)와 강력한 CLI(lenactl.sh)를 모두 제공합니다. 웹 UI는 시각적 인터페이스를 통해 서버 설치, 등록, 복제, 삭제 등 주요 작업을 쉽게 수행할 수 있으며, CLI는 자동화 스크립트와 연동하여 대규모 환경에서 빠른 배포를 지원합니다. 특히 lenactl.sh 명령어를 활용하면 서버 설정을 템플릿화하여 반복적 작업을 최소화할 수 있습니다.

Installer는 검증된 마스터 서버의 설정을 단 몇 분 만에 복제할 수 있는 기능을 제공합니다. 기존 상용 WAS 환경에서는 신규 서버 설치와 설정에 수일에서 수주가 소요되었으나, OPENMARU iAP에서는 프로파일 템플릿과 자동화된 복제 기능으로 'Days to Hours' 수준의 신속한 프로비저닝이 가능합니다. 이는 운영 효율성을 극대화하고, 장애 복구 시에도 빠른 복원력을 제공합니다.

Installer는 Connector, Session, Logging, Datasource 등 주요 설정을 프로파일로 관리할 수 있습니다. 각 프로파일은 템플릿 형태로 저장되어, 신규 서버나 클러스터에 동일한 설정을 적용할 수 있습니다. 리비전 관리 기능을 통해 설정 변경 이력을 추적하고, 필요 시 이전 버전으로 롤백할 수 있어 운영 환경의 안정성과 일관성을 보장합니다.

프로파일 기반 표준화는 운영 환경의 품질을 유지할 뿐만 아니라, 보안 정책과 컴플라이언스 요구사항을 일관되게 적용하는 데 필수적입니다. 다만, 템플릿 적용 시 환경별 특수 설정(예: 네트워크, 스토리지, 사용자 권한 등)은 별도의 검증 절차를 거쳐야 하며, 대규모 배포 시에는 설정 충돌이나 누락이 발생하지 않도록 리비전 관리와 검증 프로세스를 반드시 포함해야 합니다.

실제 도입 사례를 살펴보면, 대형 금융기관에서는 Installer의 프로파일 템플릿 기능을 활용하여 수십 대의 서버를 단일 기준으로 일괄 배포하였고, 장애 발생 시에도 리비전 관리 기능을 통해 신속하게 이전 설정으로 복구할 수 있었습니다. 또한, 운영 자동화 스크립트와 연동하여 신규 인스턴스 추가, 설정 변경, 보안 정책 적용 등을 반복적으로 수행하면서 운영 효율성과 품질을 동시에 확보하였습니다. 프로파일 기반 표준화는 조직 내 다양한 팀이 동일한 환경에서 개발과 운영을 진행할 수 있도록 지원하며, 컴플라이언스 감사 시에도 일관된 설정 이력을 제공하여 신뢰성을 높입니다. Installer의 템플릿화 기능은 클라우드 환경, 온프레미스, 하이브리드 환경 모두에서 적용 가능하며, 대규모 확장 시에도 설정 충돌이나 누락을 최소화할 수 있습니다. 이를 위해 리비전 관리와 검증 프로세스를 반드시 포함하여 운영 환경의 안정성과 품질을 지속적으로 유지해야 합니다.

5.1.2 보안 베이스라인과 멀티환경 일관성

OPENMARU Installer는 SSL/TLS 구성, 암호화 설정, 키 관리 등 보안 항목을 템플릿에 포함하여 모든 인스턴스에 일관된 보안 정책을 적용할 수 있습니다. 각 서버는 설치 시 자동으로 인증서를 배포받고, 암호화된 채널을 통해 데이터가 전송됩니다. 이는 외부 공격이나 내부 데이터 유출 위험을 최소화하고, 규정 준수(예: 개인정보 보호법, ISMS-P 등)를 지원합니다.

Installer는 Role-Based Access Control(RBAC) 정책을 템플릿화하여 서버마다 동일한 접

근 제어 정책을 적용합니다. 관리자, 운영자, 개발자 등 역할별 권한을 명확히 구분하고, 불필요한 권한 부여를 방지합니다. RBAC 설정은 웹 UI와 CLI 모두에서 관리 가능하며, 정책 변경 시 실시간으로 전체 인스턴스에 반영됩니다.

OPENMARU iAP는 온프레미스, VM, Docker, Kubernetes 등 다양한 환경에서 동일한 미들웨어 구성과 운영 정책을 적용할 수 있습니다. Installer는 환경별 특성을 자동으로 감지하여, 필요한 설정(네트워크, 볼륨, 리소스 제한 등)을 템플릿에 포함시킵니다. 이를 통해 멀티환경에서의 운영 일관성을 확보하고, 클라우드 네이티브 환경으로의 전환을 용이하게 합니다.

보안 베이스라인은 신규 인스턴스 추가, 환경 변경, 정책 업데이트 시에도 일관되게 유지되어야 합니다. Installer의 리비전 관리와 템플릿화 기능은 이러한 요구에 부합하며, 대규모 배포나 클러스터 확장 시에도 보안 수준이 저하되지 않도록 설계되어 있습니다. 다만, 환경별 특수 보안 요구사항(예: 공공기관, 금융권 등)은 별도의 커스텀 템플릿으로 관리하는 것이 바람직합니다.

보안을 위한 베이스라인 유지와 멀티환경 일관성 확보는 실제 운영에서 매우 중요한 요소입니다. 예를 들어, 금융권에서는 SSL/TLS 인증서 관리와 RBAC 정책의 일관성 유지가 필수적이며, 클라우드 환경에서는 네트워크 격리와 리소스 제한 정책을 템플릿화하여 적용함으로써 보안 수준을 높이고 있습니다. Installer는 이러한 요구에 맞춰 환경별 특수 설정을 자동 감지하고, 템플릿에 반영하여 운영자가 별도의 수작업 없이 일관된 정책을 적용할 수 있도록 지원합니다. 또한, 리비전 관리 기능을 통해 정책 변경 이력을 추적하고, 필요 시 이전 버전으로 롤백할 수 있어 장애 발생 시 신속한 복구가 가능합니다. 멀티환경 지원은 온프레미스, VM, 컨테이너, 클라우드 등 다양한 인프라에서 동일한 미들웨어 구성과 보안 정책을 적용할 수 있게 하며, 운영 일관성과 확장성을 동시에 확보할 수 있습니다. 공공기관이나 금융권처럼 특수 보안 요구가 있는 환경에서는 커스텀 템플릿을 활용하여 별도의 정책을 적용하고, 규정 준수와 컴플라이언스 요구를 충족할 수 있습니다. 이러한 기능들은 OPENMARU Installer가 엔터프라이즈 환경에서 신뢰받는 이유이며, 실제 도입 시 운영 품질과 보안 수준을 지속적으로 유지하는 데 핵심적인 역할을 합니다.

5.2 상용 WAS에서 OPENMARU iAP로의 마이그레이션

OPENMARU iAP로의 마이그레이션은 기존 상용 WAS 환경의 복잡성을 해소하고, 체계적인 구축 프로세스와 실전 도구 활용을 통해 효율적인 전환을 지원합니다. 엔터프라이즈 환경에서 요구되는 세션 클러스터링, APM 데이터 연속성, 레거시 코드 분석 등 핵심 항목을 체크리스트로 관리하며,

데이터 일관성과 성능 검증을 위한 PoC 환경 구축도 포함합니다. 마이그레이션 과정에서는 호환성, 데이터 이관, 성능 최적화 등 다양한 리스크를 사전에 식별하고 대응할 수 있도록 설계되어 있습니다.

5.2.1 5단계 구축 프로세스와 PoC 구성 방법

마이그레이션의 첫 단계는 기존 환경의 현황 분석입니다. 서버 수, WAS 버전, 세션 관리 방식, APM 구성, 데이터베이스 연동 현황 등을 상세히 파악해야 합니다. 이를 통해 전환 대상 시스템의 범위와 요구사항을 정의하고, 마이그레이션 시 고려해야 할 주요 리스크(호환성, 데이터 이관, 성능 저하 등)를 사전에 식별할 수 있습니다.

분석 결과를 바탕으로 전환 설계와 상세 계획을 수립합니다. 이 단계에서는 OPENMARU iAP의 아키텍처(Cluster, APM, Session Externalization 등)와 기존 시스템의 구조를 비교하고, 필요한 커스텀 설정, 데이터 마이그레이션 방식, 테스트 시나리오 등을 문서화합니다. PoC 환경 구성 계획도 이 단계에서 포함되며, 성능, 호환성, 세션 이관, APM 데이터 연속성 등 핵심 검증 항목을 체크리스트로 관리합니다.

설계에 따라 Installer를 활용해 OPENMARU iAP를 설치하고, 프로파일 기반 설정을 적용합니다. 기존 WAS의 세션 데이터를 IMDG로 이관하고, APM 연동을 통해 데이터 연속성을 확보합니다. 이 과정에서 자동화된 프로비저닝 기능을 최대한 활용하여 설치 시간을 단축하고, 설정 오류를 최소화합니다.

설치 완료 후에는 성능 최적화와 안정화 작업을 수행합니다. JVM 튜닝, GC 정책 설정, 네트워크/스토리지 최적화, 세션 클러스터링 검증, APM 모니터링 등 운영 환경의 품질을 높이기 위한 작업이 포함됩니다. PoC 환경에서 핵심 검증 항목을 반복 테스트하며, 장애 시나리오와 복구 절차도 함께 점검합니다.

마이그레이션 완료 후에는 운영 및 관리 지원 체계를 구축합니다. 정기 점검, 장애 대응, 업데이트 및 패치 관리, 기술 지원 프로그램 연동 등을 포함하며, 운영 표준화와 교육 프로그램을 통해 조직 전체의 운영 역량을 강화합니다.

PoC 체크리스트 예시로는 성능(TPS, 응답 시간, 리소스 사용량), 호환성(Jakarta EE, Spring, DB 연동), 세션 이관(IMDG 연동, 세션 무손실 검증), APM 데이터 연속성(트랜잭션 추적, 장애 로그 분석), 운영 자동화(Installer, CI/CD 연동) 등이 있습니다.

실제 마이그레이션 사례에서는, 대형 제조기업이 기존 WebLogic 환경에서 OPENMARU iAP로 전환하면서 5단계 프로세스를 적용하였습니다. 현황 분석 단계에서 서버별 세션 관리 방식과 DB 연동 현황을 상세히 파악하였고, 설계 단계에서는 기존 시스템과 iAP의 아키텍처를 비교하여 커스텀 설정과 데이터 이관 방식을 문서화하였습니다. 설치 및 구성 단계에서는 Installer를 활용하여 프로파일 기반 설정을 일괄 적용하였고, 세션 데이터는 IMDG로 이관하여 데이터 무손실을 검증하였습니다. 최적화 및 안정화 단계에서는 JVM 튜닝과 GC 정책 설정, 네트워크 최적화 작업을 반복적으로 수행하였으며, PoC 환경에서 장애 시나리오와 복구 절차를 테스트하였습니다. 마지막으로 운영 및 관리 지원 단계에서는 정기 점검과 장애 대응 체계를 구축하고, 기술 지원 프로그램과 교육을 통해 운영 역량을 강화하였습니다. 이러한 체계적인 프로세스와 PoC 체크리스트 관리는 마이그레이션 성공률을 높이고, 운영 품질과 신뢰성을 동시에 확보할 수 있게 해줍니다.

5.2.2 레거시 WAS 전환 시 주의사항과 도구 활용

WebLogic, JEUS, WebSphere 등 상용 WAS에서 OPENMARU iAP로 전환할 때, EJB 등 레거시 코드 분석과 변환이 필수적입니다. MTA(Migration Toolkit for Applications)는 소스 코드 분석, 벤더 독점 API 탐지, 표준 Jakarta EE API로의 자동 변환 기능을 제공합니다. 이를 통해 코드 호환성을 높이고, 전환 리스크를 최소화할 수 있습니다.

상용 WAS의 독점 API(예: WebLogic-specific, JEUS-specific 등)는 OPENMARU iAP에서 지원되지 않으므로, 표준 API로의 변환이 필요합니다. MTA를 활용한 자동 변환과 수동 리팩토링을 병행하며, 테스트 시나리오를 통해 기능 동등성을 검증해야 합니다. API 변환 과정에서 발생하는 예외나 성능 저하에 대한 대응책도 마련해야 합니다.

IMDG(In-Memory Data Grid) 기반 세션 클러스터링은 메모리/네트워크 설정이 핵심입니다. 세션 데이터의 크기, TTL(Time-To-Live), 샤딩/복제 정책, 네트워크 대역폭 등을 환경별로 최적화해야 하며, 클라우드 네이티브 환경에서는 동적 스케일 인/아웃 시 세션 유실이 발생하지 않도록 설계해야 합니다.

세션 클러스터링과 데이터 이관 과정에서는 Strong Consistency와 Eventual Consistency 정책을 선택해야 합니다. 금융, 공공 등 민감한 환경에서는 Strong Consistency를 적용하여 데이터 무결성을 보장하고, 대규모 트래픽 환경에서는 Eventual Consistency로 확장성과 성능을 우선할 수 있습니다. 정책 선택 시 비즈니스 요구와 운영 환경을 종합적으로 고려해야 합니다.

실전 주의사항으로는 코드 변환 후 기능 테스트 필수, IMDG 설정 변경 시 성능/데이터 유실 검증, 네트워크/스토리지 환경별 특수 설정, 운영 자동화와 모니터링 연동 검증 등이 있습니다.

레거시 WAS에서 OPENMARU iAP로의 전환은 단순한 설치를 넘어, 코드 호환성과 데이터 일관성, 운영 자동화까지 종합적으로 고려해야 하는 복합적인 작업입니다. 실제 금융권 도입 사례를 보면, MTA를 활용하여 WebLogic의 독점 API를 Jakarta EE 표준 API로 자동 변환한 후, 수동 리팩토링을 통해 기능 동등성을 검증하였습니다. 이 과정에서 IMDG 기반 세션 클러스터링을 적용하여 세션 데이터의 무손실 이관을 실현하였고, Strong Consistency 정책을 선택하여 데이터 무결성을 확보하였습니다. 또한, 네트워크 대역폭과 스토리지 환경별 특수 설정을 템플릿화하여 적용하였으며, 운영 자동화 스크립트와 모니터링 연동을 통해 장애 발생 시 신속한 대응이 가능하도록 설계하였습니다. API 변환 과정에서는 예외 발생 시 대응책을 마련하고, 성능 저하가 우려되는 부분은 별도의 튜닝 작업을 통해 최적화하였습니다. 이러한 도구 활용과 실전 주의사항 관리는 마이그레이션 성공률을 높이고, 운영 품질과 데이터 일관성을 동시에 확보할 수 있게 해줍니다. 특히, 클라우드 네이티브 환경에서는 동적 스케일 인/아웃 시 세션 유실을 방지하기 위한 설계가 필수적이며, IMDG 설정과 네트워크 정책을 환경별로 최적화하여 안정적인 서비스 제공이 가능합니다.

5.3 운영 표준 패턴과 기술 지원 체계

OPENMARU iAP는 IaC, CI/CD, 클라우드 네이티브 환경에 최적화된 운영 표준 패턴을 제공하며, 교육 프로그램과 기술 지원 서비스 체계를 통해 조직의 운영 역량과 신뢰성을 높입니다. 이를 통해 운영 자동화, 장애 대응, 보안, 확장성 등 모든 측면에서 엔터프라이즈급 미들웨어 환경을 실현할 수 있습니다. 운영 표준화는 반복적 작업의 자동화, 품질 관리, 신속한 장애 대응 등 조직 전체의 IT 역량을 강화하는 데 핵심적인 역할을 합니다.

5.3.1 CI/CD 파이프라인 통합과 클라우드 환경 운영

OPENMARU iAP는 Ansible, Terraform 등 Infrastructure as Code(IaC) 도구와 연동하여, 미들웨어 설치와 설정을 코드로 관리할 수 있습니다. 이를 통해 서버 배포, 설정 변경, 환경 복제 등 반복적 작업을 자동화하고, 운영 환경의 품질과 일관성을 유지합니다. IaC 스크립트는 Installer와 연동하여 신규 인스턴스 설치, 설정 적용, 보안 정책 배포 등을 자동화합니다.

iAP는 Docker 컨테이너, Kubernetes, OpenShift 환경에서 최적화된 운영 패턴을 지원합니다. Pod 자동 인식, Auto-Scaling, 롤링 업데이트, 장애 복구 등 클라우드 네이티브 기능을 Installer와 연동하여 구현할 수 있습니다. 특히, Kubernetes의 ReplicaSet, HPA(수평적 파드 오토스케일링)와 연계하여, 대규모 트래픽 환경에서도 안정적인 서비스 제공이 가능합니다.

OPENMARU iAP는 AWS, Azure, GCP, NCP 등 주요 클라우드 플랫폼과 호환되며, Installer와 IaC 도구를 활용해 멀티 클라우드 환경에서 일관된 미들웨어 배포와 운영을 실현할 수 있습니다. 각 플랫폼의 특수 기능(예: Auto-Healing, Load Balancer, VPC 등)과 연동하여, 확장성과 가용성을 극대화합니다.

iAP는 OpenTelemetry 표준을 기반으로 Metrics, Logs, Traces를 수집하고, Prometheus, Grafana, ELK Stack 등 관측성 스택과 연동할 수 있습니다. OTLP Collector를 통해 벤더 중립적 데이터 수집이 가능하며, 장애 탐지, 성능 모니터링, 트랜잭션 추적 등 운영 자동화와 품질 관리에 필수적인 기능을 제공합니다.

실제 운영 환경에서는 IaC 도구를 활용하여 신규 인스턴스 배포, 설정 변경, 보안 정책 적용 등을 자동화하고 있습니다. 예를 들어, 대형 유통기업에서는 Terraform 스크립트와 Installer를 연동하여 수십 대의 서버를 일괄 배포하였고, Ansible을 활용하여 설정 변경과 보안 정책 적용을 자동화하였습니다. Docker/Kubernetes/OpenShift 환경에서는 Pod 자동 인식과 Auto-Scaling 기능을 활용하여 트래픽 변화에 신속하게 대응하고, 롤링 업데이트와 장애 복구를 통해 서비스의 안정성을 유지하였습니다. 멀티 클라우드 환경에서는 AWS, Azure, GCP 등 다양한 플랫폼의 특수 기능을 Installer와 연동하여 확장성과 가용성을 극대화하였고, OpenTelemetry 기반 모니터링 스택을 활용하여 장애 탐지와 성능 모니터링, 트랜잭션 추적을 자동화하였습니다. 이러한 운영 표준 패턴은 조직 전체의 IT 역량을 강화하고, 품질 관리와 신속한 장애 대응을 가능하게 하며, 엔터프라이즈급 미들웨어 환경을 실현하는 데 핵심적인 역할을 합니다.

5.3.2 교육 프로그램과 기술 지원 서비스

OPENMARU iAP는 기본 사용자 교육, 관리자 전문가 교육, 맞춤형 교육 등 3단계 교육 프로그램을 제공합니다. 기본 교육은 설치, 설정, 운영의 기초를 다루며, 전문가 교육은 장애 대응, 성능 튜닝, 보안 정책 등 심화 내용을 포함합니다. 맞춤형 교육은 조직별 요구에 따라 커스텀 커리큘럼을 제공하여, 실제 운영 환경에 최적화된 역량을 확보할 수 있습니다.

정기 점검 서비스는 시스템 상태, 성능, 보안, 장애 이력 등을 주기적으로 점검하고, 상세 리포트를 제공합니다. 이를 통해 운영 환경의 품질을 지속적으로 관리하고, 장애 예방과 성능 최적화를 지원합니다.

OPENMARU iAP는 온/오프라인 장애 지원, 긴급 대응, 패치 및 업데이트 서비스 등을 제공합니다. 장애 발생 시 실시간 대응과 원인 분석, 복구 지원을 통해 MTTR을 최소화하며, 최신 기능과 보안 패치 적용을 통해 운영 환경의 안정성과 신뢰성을 높입니다.

OPENMARU iAP는 조달청 디지털서비스몰 등록, GS 1등급 인증, AWS Marketplace 등록 등 공공기관과 민간기업에서 신뢰할 수 있는 도입 근거를 제공합니다. 이를 통해 도입 조직은 안정성과 신뢰성을 확보하고, 규정 준수와 컴플라이언스 요구를 충족할 수 있습니다.

실제 도입 사례에서는, 대형 공공기관이 OPENMARU iAP의 교육 프로그램을 통해 운영팀의 역량을 강화하였고, 정기 점검 서비스와 장애 지원 체계를 활용하여 운영 품질과 신뢰성을 높였습니다. 맞춤형 교육을 통해 조직별 특수 요구에 맞는 커리큘럼을 제공하였으며, 관리자 전문가 교육에서는 장애 대응과 성능 튜닝, 보안 정책 등 심화 내용을 집중적으로 다루었습니다. 정기 점검 서비스는 시스템 상태, 성능, 보안, 장애 이력 등을 주기적으로 점검하고, 상세 리포트를 통해 운영 환경의 품질을 지속적으로 관리하였습니다. 장애 지원 및 업데이트 서비스는 온/오프라인 실시간 대응과 원인 분석, 복구 지원을 통해 MTTR을 최소화하였고, 최신 기능과 보안 패치 적용을 통해 운영 환경의 안정성과 신뢰성을 지속적으로 유지하였습니다. 공공/민간 도입 신뢰성 근거로는 조달청 디지털서비스몰 등록, GS 1등급 인증, AWS Marketplace 등록 등이 있으며, 이를 통해 도입 조직은 안정성과 신뢰성을 확보하고, 규정 준수와 컴플라이언스 요구를 충족할 수 있습니다. 이러한 교육 프로그램과 기술 지원 서비스는 조직 전체의 운영 역량을 강화하고, 엔터프라이즈급 미들웨어 환경을 실현하는 데 필수적인 요소입니다.

Appendix

References

1. (Web Scraper로 스크래한 모든 URL을 포함하세요)

2. (본문에서 인용하거나 참조한 모든 외부 출처를 나열하세요)
3. Author/Organization. (Year). “APM 및 세션 클러스터링 기술 설명”. https://openmaru.io/docs/apm-cluster/
4. Author/Organization. (Year). “AWS Marketplace OPENMARU iAP”. https://aws.amazon.com/marketplace/pp/prodview-xyz
5. Author/Organization. (Year). “Grafana Documentation”. https://grafana.com/docs/
6. Author/Organization. (Year). “Hazelcast IMDG Documentation”. https://docs.hazelcast.com/
7. Author/Organization. (Year). “Jakarta EE Specification”. https://jakarta.ee/specifications/
8. Author/Organization. (Year). “Kubernetes 공식 문서”. https://kubernetes.io/docs/
9. Author/Organization. (Year). “LLM, RAG, MCP 통합 아키텍처”. https://openmaru.io/docs/llm-rag-mcp/
10. Author/Organization. (Year). “OPENMARU iAP 공식 문서”. https://openmaru.io/docs/
11. Author/Organization. (Year). “OPENMARU iAP 공식 홈페이지”. https://www.openmaru.com/
12. Author/Organization. (Year). “OPENMARU iAP 백서 목차”. (https://openmaru.com)
13. Author/Organization. (Year). “OPENMARU iAP 백서 목차”. Provided by user.
14. Author/Organization. (Year). “OpenTelemetry Documentation”. https://opentelemetry.io/docs/
15. Author/Organization. (Year). “OpenTelemetry 공식 문서”. https://opentelemetry.io/docs/
16. Author/Organization. (Year). “OpenTelemetry 표준”. https://opentelemetry.io/docs/
17. Author/Organization. (Year). “Prometheus Documentation”. https://prometheus

us.io/docs/

18. Author/Organization. (Year). “PromptOps 기술 아키텍처”.https://openmaru.io/docs/promptops/
19. Author/Organization. (Year). “Spring Framework Documentation”.https://docs.spring.io/spring-framework/docs/current/reference/html/
20. Author/Organization. (Year). “Title”. URL
21. Author/Organization. (Year). “VibeOps 공식 소개”.https://openmaru.io/docs/vibeops/
22. Author/Organization. (Year). “WildFly Documentation”.https://docs.wildfly.org/
23. Author/Organization. (Year). “WildFly 공식 문서”.https://docs.wildfly.org/
24. GS 인증센터. (2021). “GS 1등급 인증”.https://gs.korea.kr/
25. IBM. (2024). “IBM WebSphere Application Server”.https://www.ibm.com/products/websphere-application-server
26. OPENMARU Inc. (2023). “OPENMARU iAP 공식 소개”.https://openmaru.com/iap
27. Oracle. (2024). “Oracle WebLogic Server Licensing”.https://www.oracle.com/middleware/technologies/weblogic-server.html
28. 공개SW 개발자대회. (2014). “KHAN Session Manager 금상 수상”.https://oss.kr/award/2014

Glossary

용어	정의
APM	Application Performance Monitoring, 성능 모니터링 및 장애 분석 도구
Bolted-on 구조	이종 벤더 솔루션이 조립식으로 결합된 아키텍처
Call Tree	트랜잭션 호출 체인을 시각적으로 표현하는 구조.
Failover	장애 발생 시 서비스 무중단 전환
GS 1등급 인증	국내 소프트웨어 품질 인증 제도
HPA	Horizontal Pod Autoscaler, Kubernetes의 수평적 파드 오토스케일링

HyperLogLog	확률적 집계 알고리즘으로, 대규모 고유 사용자 집계를 위한 메모리 효율적 기술.
IaC	Infrastructure as Code, 인프라를 코드로 정의하고 관리하는 방식
iAP	Intelligent Application Platform, OPENMARU의 통합 미들웨어 플랫폼
IMDG	In-Memory Data Grid, 메모리 기반 데이터 분산 저장소
Installer	미들웨어 설치 및 설정을 자동화하는 도구
Jakarta EE	엔터프라이즈 Java 표준 플랫폼으로, RESTful API, JPA, CDI 등 다양한 Java EE 기술을 포함.
JDK	Java Development Kit, Java 애플리케이션 개발 및 실행 환경.
KHAN APM	OPENMARU가 개발한 네이티브 성능 모니터링 솔루션
KHAN Session Manager	OPENMARU가 개발한 세션 클러스터링 솔루션
LLM	Large Language Model, 대규모 언어 모델
MCP	Model Context Protocol, 운영 데이터와 AI 모델 간 컨텍스트 표준화 프로토콜.
MTA	Migration Toolkit for Applications, WAS 마이그레이션 지원 도구
MTTR	Mean Time to Repair, 장애 복구 평균 시간
OpenTelemetry	벤더 중립적 관측성 표준 프레임워크
OTLP	OpenTelemetry Protocol, 관측성 데이터 전송 표준.
PaaS	Platform as a Service, 플랫폼 서비스 환경
PoC	Proof of Concept, 개념 검증 환경
PromptOps	세션-트랜잭션-AI 통합 운영 패러다임으로, 자연어 기반 운영 자동화를 실현.
RAG	Retrieval-Augmented Generation, 외부 데이터 검색을 결합한 생성 AI 기법.
RBAC	Role-Based Access Control, 역할 기반 접근 제어
RCA	Root Cause Analysis, 장애 원인 분석
Seasonality	트래픽 등 운영 데이터의 계절성 또는 패턴 변화 분석 개념.
Session Clustering	사용자 세션의 고가용성 및 복제 기능
Stateless	상태를 저장하지 않는 아키텍처로, 확장성과 장애 복구에 유리함.
TCO	Total Cost of Ownership, 총소유비용
VibeOps	OPENMARU iAP의 LLM 기반 AI 운영 체계
WAS	Web Application Server, 웹 기반 애플리케이션 실행 환경
WildFly	오픈소스 WAS(Web Application Server)로 Jakarta EE 표준 구현체.
ZGC/Shenandoah	Java의 최신 가비지 컬렉터(GC)로, 낮은 지연 시간과 빠른 GC Pause를 제공.

OPENMARU Blog

운영/모니터링 인사이트와
실무 가이드를 전하는
기술 아카이브

OPENMARU eBook

거침없이 배우는
JBoss EAP

Contact Us

02-469-5426

hello@openmaru.io

www.openmaru.io

openmaru

오픈마루 | OPENMARU

전화 : (02) 469 - 5426
팩스 : (02) 469 - 7247
메일 : hello@msap.ai