



2023 Java 에코시스템 현황

가장 널리 사용되는 프로그래밍 언어에 대한 심층 탐구

목차

- 03 개요
- 04 Java 17 사용자 도입률 1년 만에 430% 증가
- 05 가장 많이 사용되는 비 LTS 버전은 Java 14
- 06 현재 가장 보편적인 JDK 공급업체는 Amazon
- 07 우리 주변의 모든 것을 지배하는 컨테이너
 - › 컨테이너에서의 컴퓨팅 설정
 - › 컨테이너에서의 메모리 설정
- 09 가비지 인, 가비지 아웃
- 10 조사방식
- 11 뉴렐릭 소개



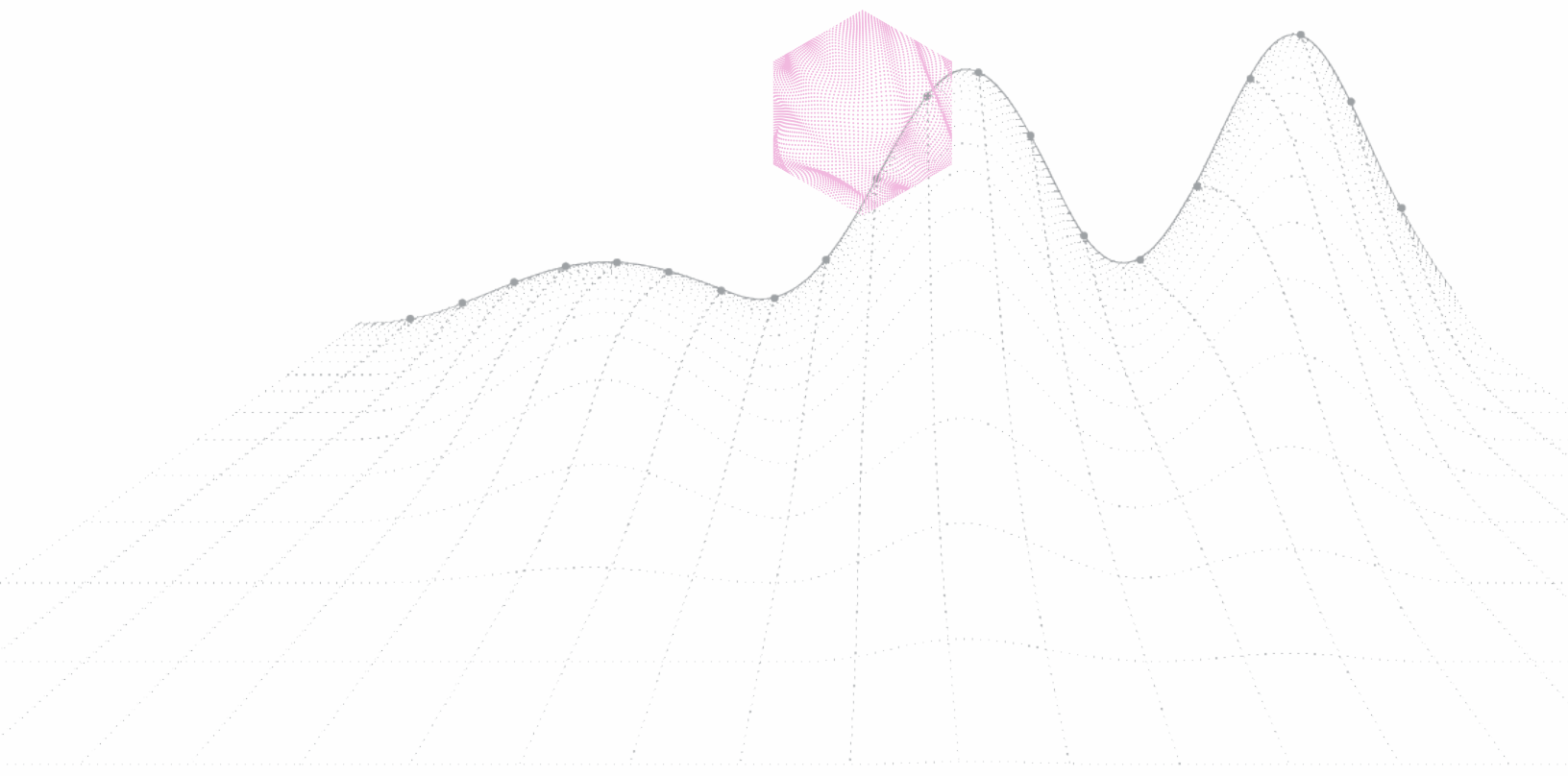
개요

1996년 썬 마이크로시스템즈(Sun Microsystems)는 현대적인 멀티미디어 애플리케이션을 개발하는데 이동 및 상호 작용 가능한 방법을 제공하기 위해 Java 프로그래밍 언어를 출시했습니다. 이후 Java는 가장 인기 있는 프로그래밍 언어의 자리를 지켜왔습니다. Java는 플랫폼에 의존하지 않고 수천 개의 라이브러리를 제공하며 잘 지원되기 때문에 거의 모든 주요 산업과 경제 부문에서 사용되고 있습니다.

지난 몇 년 동안 Java 에코시스템을 모니터링한 결과, 뉴렐릭은 새로운 버전 릴리스와 컨테이너의 부상으로 인해 Java가 사용되는 방식이 변화했음을 발견할 수 있었습니다. 이 보고서는 2023년 Java 에코시스템의 현황에 대한 배경과 인사이트를 제공합니다.

다음과 같은 카테고리를 살펴보았습니다.

- [운영 환경에서 가장 많이 사용되는 Java 버전](#)
- [가장 보편적인 JDK 공급업체](#)
- [컨테이너의 부상](#)
- [가장 일반적인 힙 크기 구성](#)
- [가장 많이 사용되는 가비지 수집 알고리즘](#)



Java 17 사용자 도입률 1년 만에 430% 성장

2-3년마다 Java 장기 지원(LTS) 버전이 출시되고 분기별로 안정성, 보안 및 성능에 대한 업데이트가 제공되지만, 새로운 기능은 포함되지 않습니다.

현재 애플리케이션의 56% 이상이 운영 환경에서 Java 11을 사용하고 있습니다. 이는 2022년의 48%와 2020년의 11%에서 증가한 수치입니다. 그 다음은 운영 환경에서 약 33%의 애플리케이션이 사용하는 Java 8입니다. (2022년 46%에서 감소)

Java 11이 2년 연속 1위를 유지했지만, Java 17의 도입률은 Java 11이 처음 출시되었을 때 보다 훨씬 더 높습니다. 애플리케이션의 9% 이상이 현재 운영 환경에서 Java 17을 사용하고 있습니다. 2022년 1% 미만에서 증가한 수치로, 1년 만에 430%나 성장한 것입니다. Java 11이 그 수준에 근접하는 데는 몇 년이 걸렸습니다.

애플리케이션의 0.28%만이 운영 환경에서 여전히 Java 7을 사용하고 있으며, 이는 2022년에 Java 7에 대한 지원이 종료되었기 때문입니다. Java 7을 사용하는 대부분의 애플리케이션은 아직 업그레이드되지 않은 레거시 애플리케이션입니다.

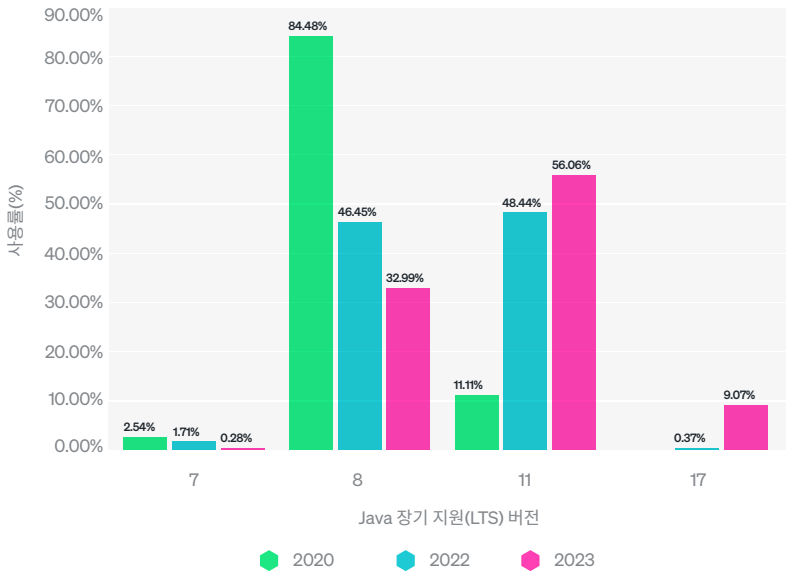


그림 01. Java LTS 버전의 사용 비율



가장 많이 사용되는 비 LTS 버전은 Java 14

Java 9부터 플랫폼의 릴리스 패턴이 변경되었습니다. 약 6개월마다 Java의 새 버전이 출시되지만, 그 버전은 다음 릴리스까지만 지원됩니다. 새 기능을 더 자주 사용하도록 만들기 위함입니다.

운영 환경에서 중간에 출시된 비 LTS Java 버전이 사용되는 비율은 LTS 버전에 비해 매우 낮으며, 비 LTS Java 버전을 사용하는 애플리케이션은 1.6%에 불과합니다. 2022년의 2.7%에서 감소한 수치입니다.

비 LTS 버전의 사용 감소에 영향을 미친 몇 가지 요인은 다음과 같습니다.

- 지원 부족
- 기능의 인지된 장점
- 다음 LTS 버전까지의 시간

버전 8과 버전 11 사이에는 다음 LTS 버전이 언제 출시될지 정확히 알 수가 없었습니다. 이제 2-3년으로 단축된 출시 일정이 확실시 되었습니다. 다음 LTS 버전은 23이 아닌 21이 될 것으로 예상됩니다. 일부 개발자들이 기꺼이 기다리고 있는 이유입니다.

사용 중인 비 LTS Java 버전 중에서는 Java 14가 여전히 가장 많이(0.57%) 사용되고 있습니다.(2022년 0.95%) Java 15가 근소한 차이(0.44%)로 그 뒤를 따르고 있습니다. (2022년 0.70%)

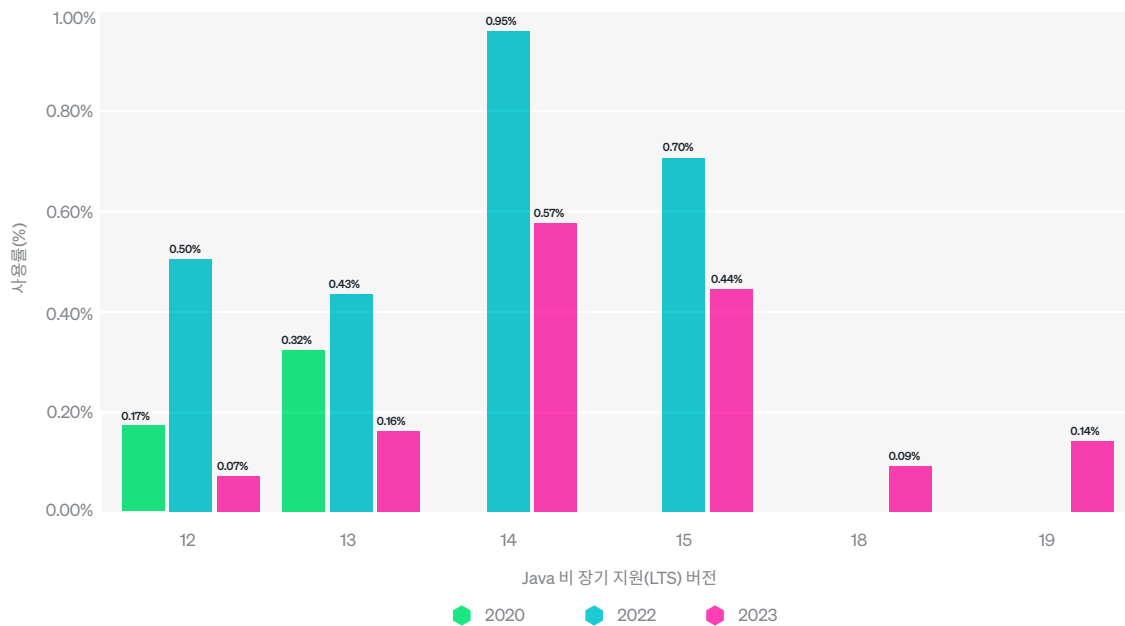


그림 02. Java 비 LTS 버전의 사용 비율

현재 가장 보편적인 JDK 공급업체는 Amazon

최근 몇 년 동안 사용 중인 자바 개발자 키트(JDK) 배포판의 소스가 바뀌었습니다. 많은 개발자들이 Oracle에서 JDK를 구입했지만 OpenJDK 프로젝트에서 Java 오픈 소싱을 통해 다양한 옵션이 생겨났습니다.

2020년, Oracle은 Java 시장의 약 75%를 차지하는 가장 인기 있는 JDK 공급업체였습니다. JDK 11 배포에서 라이선스에 보다 많은 제한이 걸리자 Oracle 바이너리에서 이탈이 두드러지게 나타났습니다. 이후 Java 17에서 보다 개방적인 입장으로 복귀했습니다. 2022년 Oracle은 34%로 최고 위치를 고수했지만, 2023년에는 28%로 점유율이 떨어졌습니다.

Amazon의 사용률이 31%로 대폭 증가하면서 (2020년 2.18%, 2022년 22%), Amazon은 이제 가장 인기 있는 JDK 공급업체가 되었습니다.

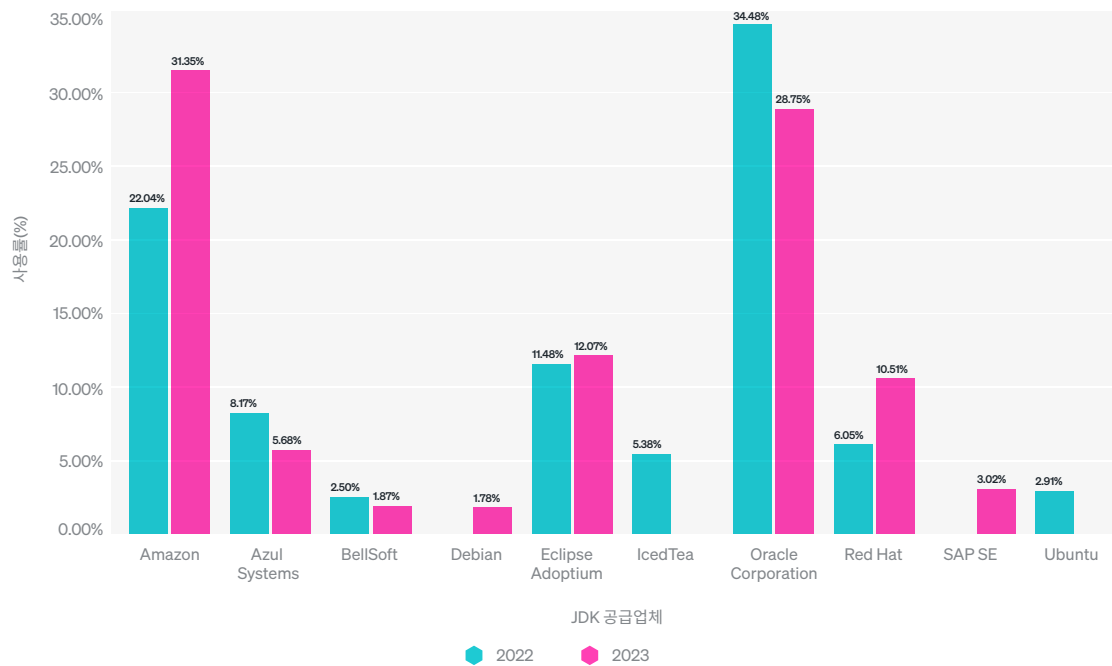
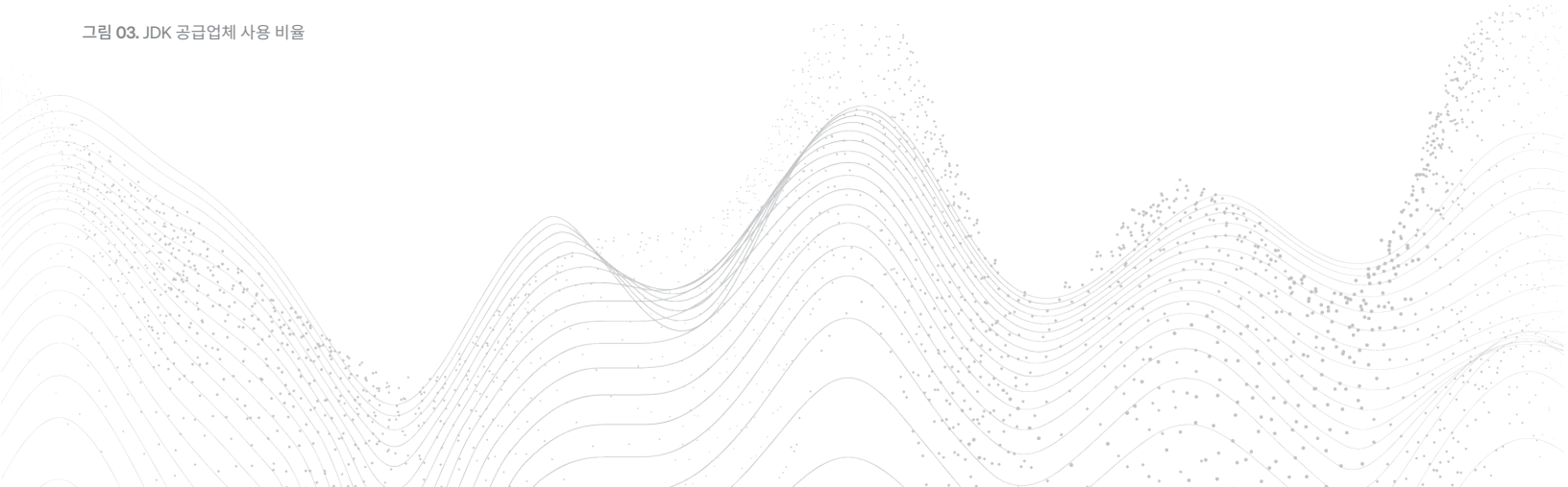


그림 03. JDK 공급업체 사용 비율



우리 주변의 모든 것을 지배하는 컨테이너

애플리케이션 컨테이너화가 주류로 자리를 잡았습니다. 뉴렐릭이 정보를 수집하는 Java 애플리케이션 중 70% 이상이 컨테이너를 사용합니다.

컨테이너에서의 컴퓨팅 설정

컨테이너는 엔지니어링 팀이 컴퓨팅과 메모리 리소스를 할당하는 방식에 영향을 줍니다. 일례로 뉴렐릭 데이터를 살펴보면, 컨테이너에서 4개 미만의 코어로 실행되는 애플리케이션의 비율이 훨씬 더 높아졌다는 것을 알 수 있습니다.

엔지니어링 팀들은 컨테이너의 싱글 코어 설정에서 벗어나고 있으며, 싱글 코어 사용률은 36%에 불과합니다. (2022년 42%) 이제 29% 이상이 8개 코어를 실행하며, 멀티 코어 설정으로 이동하는 추세가 나타나고 있습니다. (2022년 20%)

엔지니어링 팀들은 일반적으로 컨테이너를 자주 배포하는 클라우드 환경에서 더 작은 컴퓨팅 설정을 사용합니다. 그러나 이러한 추세는 일부 애플리케이션에서 예상치 못한 문제를 제기하여 구성이 감소되는 결과를 야기할 수 있습니다. 예를 들어, 팀이 하나의 CPU만 실행하는 경우 명시적으로 설정을 해도 예상한 가비지 컬렉터를 사용하지 못할 수가 있습니다.

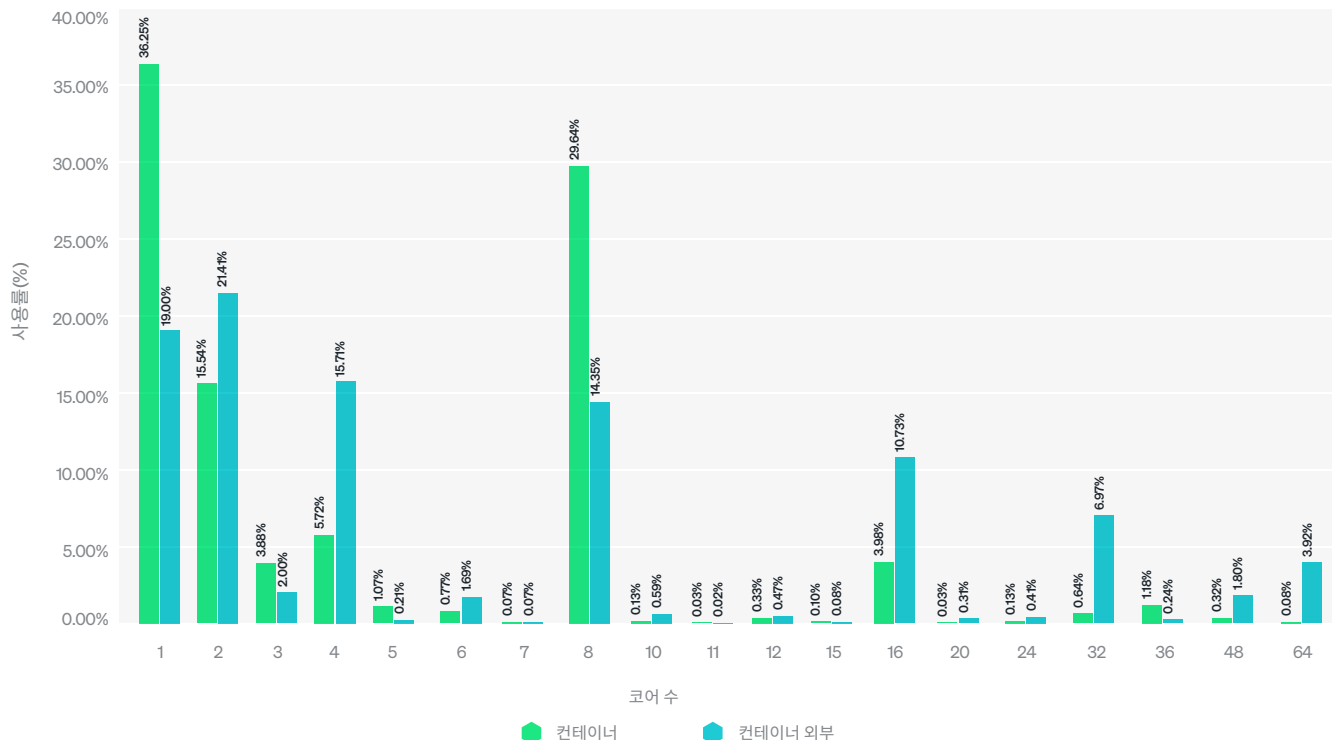


그림 04. 코어 수별 컨테이너 내/외부에서 실행되는 Java 애플리케이션 비율

컨테이너에서의 메모리 설정

메모리 설정에서도 유사한 추세가 보여지며, 컨테이너에서 더 작은 인스턴스를 실행하는 경향이 있습니다. 컨테이너 배포의 특성 상 제한이 더 엄격하게 적용되기 때문에 개발자들은 공간에 더 신경을 쓰게 됩니다.

Java 9에는 -Xmx를 통해 정확한 힙 크기 지정을 대체하는 스타트업 플래그인 -XX:MaxRAMPercentage 같이 컨테이너에서 더 잘 작동하게 해주는 여러 기능들이 도입되었습니다. Java 가상 머신(JVM)은 컨테이너 메모리 제한을 인식하므로, -XX:MaxRAMPercentage는 JVM을 컨테이너 크기로 쉽게 확장합니다.

뉴렐릭의 데이터에 따르면 컨테이너화된 애플리케이션의 30%가 -XX:MaxRAMPercentage 플래그를 통해 명시적으로 JVM 메모리의 상한을 요청했습니다. (2022년 9%) 플래그의 사용이 늘고 있음을 알 수 있습니다.

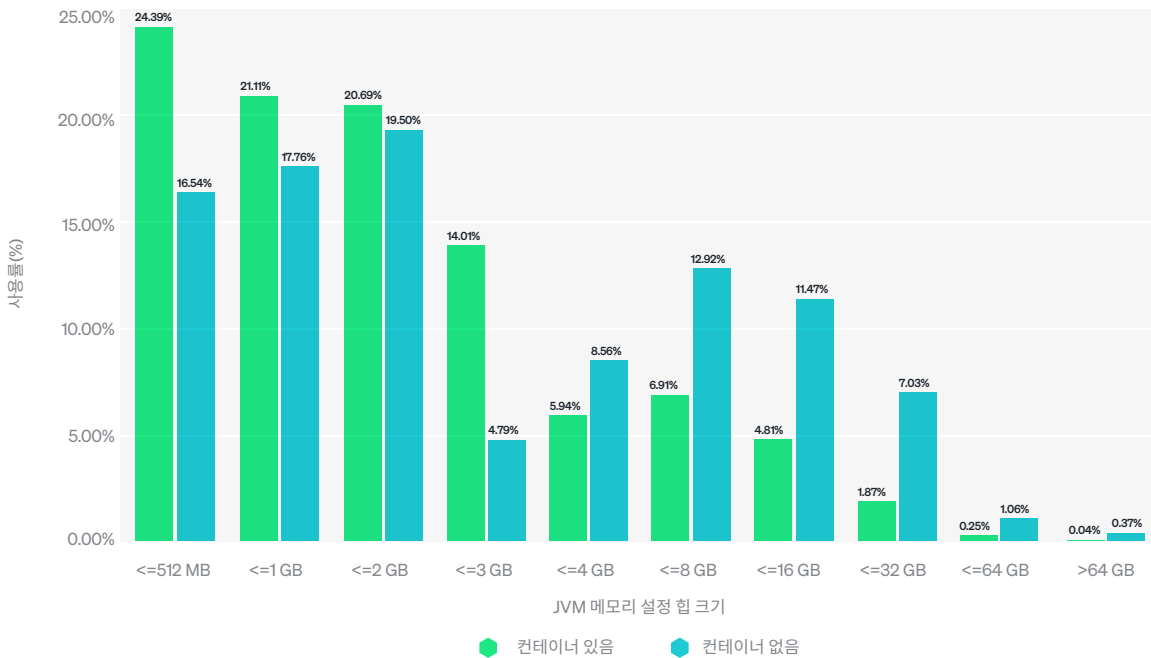
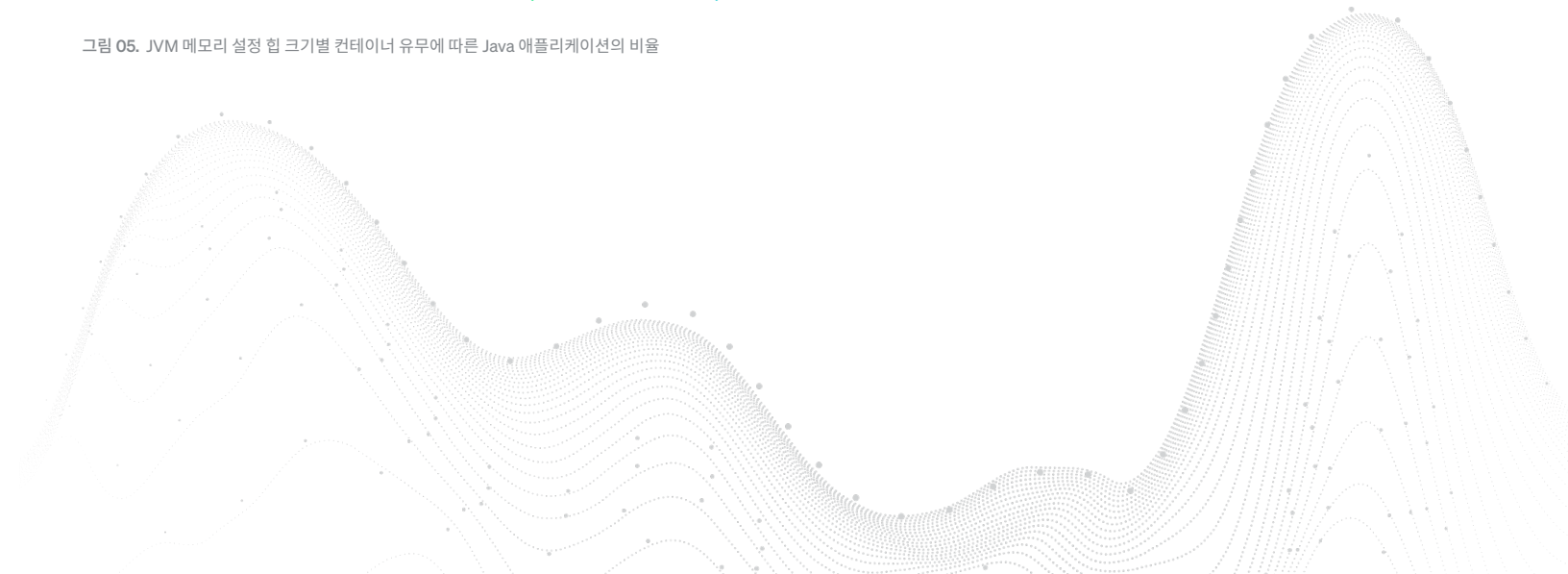


그림 05. JVM 메모리 설정 힙 크기별 컨테이너 유무에 따른 Java 애플리케이션의 비율



가비지 인, 가비지 아웃

자동 가비지 수집은 힙 메모리를 살펴보고 사용되고 있는 개체와 사용되고 있지 않은 개체를 식별하여 사용되지 않는 개체를 삭제하는 프로세스입니다. JVM 성능에서 가비지 수집이 핵심적인 역할을 하기 때문에, Java 커뮤니티에서는 여전히 이에 대해 많은 논의가 이뤄지고 있습니다.

뉴렐릭의 데이터에 따르면 Java 11 이상 버전 사용자들 사이에서 가장 선호되는 가비지 컬렉터는 65%의 고객들이 사용하고 있는 Garbage-First(G1)입니다. G1의 주요 이점 중 하나는 큰 영역을 한 번에 클리어하는 것이 아니라 보다 작은 영역들을 클리어하여 수집 프로세스를 최적화한다는 것입니다. 또한 실행 중에 중단되는 경우가 거의 없고 구세대와 새로운 세대를 동시에 수집할 수 있으므로 엔지니어를 위해 훌륭한 기본 설정이 됩니다.

Java 8 이후에 등장한 다른 실험적인 가비지 컬렉터(ZGC 및 Shenandoah)는 운영 시스템에서 사용되는 경우가 아직 많지 않습니다. 두 가지 모두 운영 환경에서 사용할 수 있는 릴리스가 나와 있지만 일반 프로세싱에서의 활용도는 미미합니다.

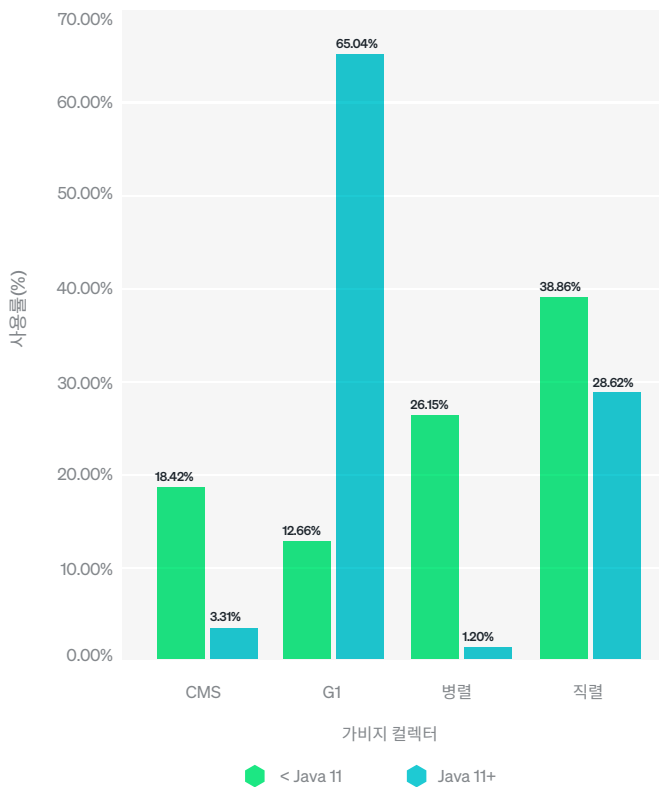
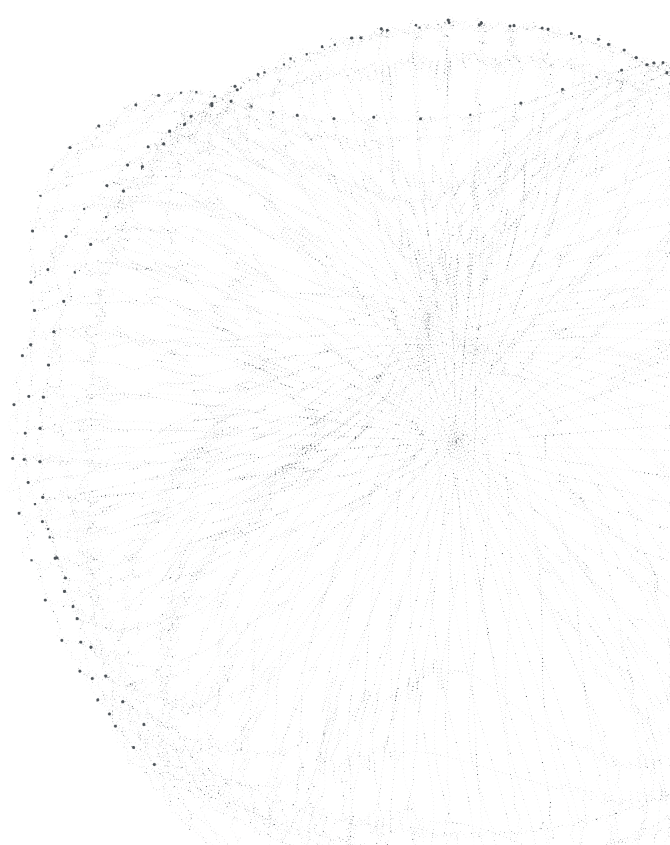


그림 06. Java 버전별 사용되는 가비지 컬렉터의 비율

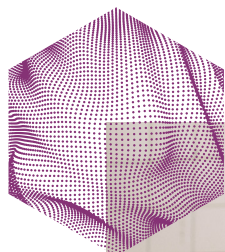


조사방식

이 보고서에 사용된 데이터는 모두 2023년 1월 뉴렐릭에 보고하는 애플리케이션에서 추출되었으며, Java의 전 세계적인 사용 정보가 아닙니다. 뉴렐릭은 Java 에코시스템에 대한 일반적인 개요를 제공하기 위해 적절한 데이터를 익명 처리한 후 개괄적으로 분류했습니다. 공격자 및 기타 악의적인 행위자에게 도움이 될 수 있는 세부적인 정보는 고의적으로 누락시켰습니다.

지금 뉴렐릭으로 Java 데이터 모니터링을 시작하십시오.

Java 퀵스타트 설치하기



뉴렐릭 소개

뉴렐릭은 오피저빌리티 분야의 리더로, 엔지니어들에게 탁월한 소프트웨어를 계획, 구축, 배포 및 실행할 수 있는 데이터 중심 접근 방식을 제공합니다. 뉴렐릭은 메트릭, 이벤트, 로그, 트레이스 등 모든 텔레메트리 데이터를 갖춘 유일한 통합 데이터 플랫폼과 강력한 풀스택 분석 툴을 결합하여 엔지니어들이 의견이 아닌 데이터에 기반해 작업을 최대한 효과적으로 수행할 수 있도록 지원합니다.

업계에서 유일하게 직관적이고 예측 가능한 사용량 기반 요금제를 제공하는 뉴렐릭은 계획 주기 시간, 변경 실패율, 릴리스 빈도, 평균 해결 시간(MTTR)을 개선하여 엔지니어가 비용 대비 더 많은 가치를 실현할 수 있도록 지원합니다. 이러한 요금제는 세계 유수의 브랜드와 고성장 스타트업들이 업타임, 안정성 및 운영 효율성을 개선하고 혁신과 성장을 촉진하는 탁월한 고객 경험을 제공할 수 있도록 지원합니다.

