aws marketplace

OpenJDK Java

Reviews, tips, and advice from real users






Powered by PeerSpot

Contents

Product Recap

OpenJDK Java

OpenJDK Java Recap

OpenJDK Java is an open-source implementation of the Java Platform, Standard Edition. It provides developers with a robust platform for creating applications and services, ensuring compatibility and support for Java applications across diverse environments.

As a leading choice for Java development, OpenJDK Java empowers developers with a reliable and high-performance environment. It supports the development of a wide range of applications, from enterprise-level software to mobile apps, thanks to its compatibility and extensive library of tools. OpenJDK Java facilitates seamless integration and offers updates that enhance security and performance, making it a respected choice among developers.

What are the most important features?

- Open Source: Free to use, modify, and distribute.
- Cross-Platform: Ensures compatibility across different OS.
- Regular Updates: Keeps up with security and performance improvements.
- Comprehensive Libraries: Extensive collection of class libraries for efficient development.
- Community Support: Backed by a vibrant community for assistance and collaboration.

What benefits should users look for in reviews?

- Performance: High efficiency even in demanding environments.
- Scalability: Can handle applications ranging from small to large scale.
- Cost-Effective: No licensing fees for usage and deployment.
- Innovation: Continuous improvements and enhancements.

In industries like finance, OpenJDK Java is often implemented for developing secure and scalable banking applications. In healthcare, it is used to create reliable and compliant data management systems, while the tech industry employs it for innovative enterprise solutions. OpenJDK Java consistently supports industry-specific requirements, enabling development teams to deliver customized and robust applications.

Valuable Features

Excerpts from real customer reviews on PeerSpot:

- "OpenJDK Java has positively impacted my organization by significantly reducing parsing failures."

Harshwardhan Gullapalli

AI Engineer at a educational organization with 51-200 employees

- "Overall, OpenJDK Java is a very reliable and mature technology."

Yash Patel

Software developer at BISAG-N

- "OpenJDK Java is open for everyone, every developer, and every company, and it is free of cost."

Aadarsh Mishra

Sde Ii at Greytip Software Pvt. Ltd.

"Using OpenJDK Java has impacted my organization positively, as it has significantly improved productivity by providing a stable and mature platform, while the rich set of libraries and frameworks like Spring Boot and Hibernate accelerates development and reduces the time needed, allowing us to reduce development time by approximately twenty-five percent and cut production incidents and downtime by fifteen percent."

Shreya Reddy

Software Engineer at a financial services firm with 10,001+ employees

"I can observe savings in my ROI and categorize it as money-saving, as I see about 60% in money savings and 40% in time savings."

Shailesh Shandilya

Principle Consultant at Verinite

"We have found OpenJDK Java to be highly performant and reliable compared to other Java distributions such as Oracle JDK, and in some cases, even more stable due to its open-source nature and community-driven updates."

Hussain Gagan

FullStack Developer at EnactOn Technologies

"Almost a million dollars in cost savings have been realized since switching to OpenJDK Java."

Amol Tyagi

AWS solutions architect at Deloitte

What users had to say about valuable features:

"The best features that OpenJDK Java offers are the ease of using any other tools with Java; it is very easy and efficient, and troubleshooting in OpenJDK Java is the best thing you can say. It is very easy, and it will tell you where or what exact error you have. Additionally, OpenJDK Java is a pure object-oriented programming language.

"Troubleshooting in OpenJDK Java has helped me in my work while using Eclipse as a tool, which supports multiple languages including Java, Python, and Selenium. In Eclipse, at the bottom of the page under the terminal, if there are any errors, you can see the errors, and it will indicate the line number or what mistake you have made, making it very easy. Even a non-technical person will be able to understand what that error is about.

"Since using OpenJDK Java, the specific outcomes include that the Selenium automation I have done in Java on the client's home page has given us very good reliability, and after that, we never received any escalation or an email from the client that this page is down and no one is looking into it. OpenJDK Java is extremely awesome.."

Verified user

Sr Dev Ops Engineer at a tech vendor with 201-500 employees

Read full review

"OpenJDK Java offers open-source capabilities, which are significant because my laptop runs on Windows while my friend's system uses macOS. It is open-source and cross-platform, which is beneficial as it supports every system when changing the Windows operating system. Therefore, these two features are among the best for me.

"I also appreciate the security updates provided by OpenJDK Java. When I download the JDK, it comes in a specific version, but subsequent updates resolve issues with earlier applications. It is continuously updating features, which is the best aspect of OpenJDK Java.

"OpenJDK Java also has strong community support, making it the best option for me. OpenJDK Java has saved me time and effort, being one of my main resources that is free to use and helpful as a student. It provides almost all the core features required for Java development, including OOPs concepts, exception handling, file management, and database connectivity.."

Verified user

Student at a university with 501-1,000 employees

Read full review

“The best features of OpenJDK Java in my experience include write-once, run-anywhere capability thanks to the JVM, along with powerful performance optimization and garbage collection tuning options for backend systems.

“I have seen significant improvements in system stability and resource utilization thanks to OpenJDK Java's garbage collection and JVM tuning capabilities. For instance, in one project, we were able to reduce memory-related issues by over 50% by fine-tuning the JVM's garbage collection settings.

“The open-source nature of OpenJDK Java is a significant advantage, giving flexibility to choose different builds such as Amazon Corretto or Red Hat builds. Additionally, the frequent release cycles help us adopt new features quickly.

“OpenJDK Java has positively impacted our organization by helping us standardize our backend stack across teams, making onboarding easier. New developers could ramp up faster since Java and OpenJDK Java are widely known and well documented..”

Hussain Gagan
FullStack Developer at EnactOn Technologies

Read full review

The best features OpenJDK Java offers include its own garbage collection system, as it automatically removes unused objects, which is an important feature.

This feature has helped my development process because creating too many objects can lead to an out of memory situation, but the garbage collection efficiently deletes unused memory.

“OpenJDK Java stands out for its portability, as it runs on a write once, run everywhere principle due to its virtual memory and JVM, which converts every code into machine code, making it helpful to run code on any device.

“OpenJDK Java has positively impacted my organization by providing a large number of resources to find issues and creating a large environment that helps in many areas.

“Having a large environment and many resources has helped my team specifically when we tried to use Redis components, as we could consult previous blogs to maintain it, which was helpful for integrating other components into our code. .”

Verified user

Software Developer at a tech vendor with 51-200 employees

Read full review

"The best features OpenJDK Java offers, in my experience, include stability, ecosystem maturity, and solid performance. The biggest one for us is stability; it runs consistently. Once we set it up on our Linux infrastructure, it is rock solid, with minimal downtime and no unexpected crashes over the year and a half we used it. The second is the maturity of the ecosystem. Because it has been around for so long, there are incredibly robust libraries built on top of it, with Apache Tika being the perfect example for our use case. Finally, performance matters. Once the JVM warms up, the throughput for document parsing tasks is consistent and predictable, which is essential when processing batches. Those three things make OpenJDK Java valuable for us in the production financial document space.

OpenJDK Java has positively impacted my organization by significantly reducing parsing failures. Before standardizing on this stack, we dealt with manual workarounds for document format inconsistencies, where approximately 10 to 15 percent of incoming documents required manual intervention. Once we locked in Apache Tika and OpenJDK Java as our parsing foundation, that dropped to approximately 2 to 3 percent. That is a huge operational improvement for our team because it meant less manual rework and faster throughput for our clients. On the speed side, document processing time became predictable; we could reliably process a financial document from ingestion to structured output in under two minutes on average, meeting our client service level agreements. From a cost perspective, being open source meant avoiding licensing fees, freeing up budget to reinvest in our machine learning models and infrastructure. For the team workflow itself, the sustainability meant our engineers spent less time debugging parsing issues and more time on higher-value work, improving our LLM extraction logic and building new features. There was a genuine productivity gain.."

Harshwardhan Gullapalli

AI Engineer at a educational organization with 51-200 employees

Read full review

"I have been working with OpenJDK Java for about two and a half years now, mainly in backend development and end-to-end related services. Most of my work involves building REST APIs and microservices using Java on OpenJDK Java. What I appreciate is that it is stable and behaves consistently across environments, especially when deploying on Linux servers. I have also used different versions such as Java 8 and Java 17, depending on project requirements.

"I would like to highlight the strong ecosystem around OpenJDK Java. Libraries and frameworks such as Spring, Hibernate, and Kafka integrate seamlessly. The security features are also quite robust, especially for enterprise applications. Multithreading support is excellent for handling concurrent workloads. Plus, tools such as JV, JV, and VisualVM help in profiling and debugging performance issues. One of the best features OpenJDK Java offers is platform independence. Write once, run anywhere actually works. The JVM is highly optimized and handles memory management efficiently. Garbage collection tuning is another strong point.

"Since it is open source, there is a huge community backing it. Regular updates and long-term support versions make it production-ready. The ROI has been quite strong. I saved on licensing costs completely. Development speed improved due to mature tooling. Operational efficiency increased with stable performance. Overall, I got high value with minimal investment. It also helped me onboard new developers quickly since Java is widely known.."

Yash Patel
Software developer at BISAG-N

Read full review

Other Solutions Considered

"We were initially using Oracle JDK but switched to OpenJDK Java due to licensing costs and flexibility concerns, as OpenJDK Java offered the same core functionality without the restrictions.."

Hussain Gagan
FullStack Developer at EnactOn Technologies

Read full review

"I evaluated Oracle JDK and some lightweight runtimes. Oracle JDK was powerful but came with licensing concerns. Other alternatives did not have the same ecosystem support. OpenJDK Java offered the best balance of cost and performance. That is why I chose it.."

Yash Patel
Software developer at BISAG-N

Read full review

"Before choosing OpenJDK Java, we evaluated other options including Oracle JDK and some JVM alternatives, but most were either expensive or less flexible compared to OpenJDK Java.."

Hussain Gagan
FullStack Developer at EnactOn Technologies

Read full review

"Before OpenJDK Java, some teams were using older proprietary JDK versions. I switched mainly to reduce licensing costs and move to open source. OpenJDK Java provided the same performance and features. It also aligned better with my cloud-native approach, so it was a logical transition.."

Yash Patel
Software developer at BISAG-N

Read full review

"We considered several alternatives before choosing OpenJDK Java. We explored a pure Python-based solution using libraries such as PyPDF2 or PDFPlumber for document parsing. The appeal was that our core application logic was already in Python, making for a simpler tech stack. The trade-off was in limitations with complex and scanned financial documents as those libraries are not as robust for edge cases. We landed on Apache Tika running on OpenJDK Java because it struck the right balance. It is mature, battle-tested for handling complex document formats, open source, meaning no licensing overhead, and the ecosystem is solid. Introducing Java into our stack meant stability and reliability gains justified the architectural decision.."

Harshwardhan Gullapalli
AI Engineer at a educational organization with 51-200 employees

Read full review

“Before standardizing on OpenJDK Java, we used a more fragmented approach. We had a mix of Python-based text extraction tools and some proprietary PDF parsing libraries. The problem was inconsistency; different document formats failed unpredictably with different tools. We ended up with manual workarounds, with team members extracting data from PDFs that the automated pipeline could not handle reliably. We switched to OpenJDK Java and Apache Tika because we required something battle-tested and stable for production use with financial documents. The Java ecosystem had proven libraries that could handle edge cases and unusual document formats that were giving us trouble. Tika's maturity and the JVM's stability made it the right choice, simplifying our tech stack significantly. Instead of maintaining multiple parsing tools across different languages, we have one solid foundation that reduces operational complexity. In hindsight, it was absolutely the right decision; the consistency and reliability improvements justified the learning curve my team had to undergo coming from a Python-heavy background..”

Harshwardhan Gullapalli
AI Engineer at a educational organization with 51-200 employees

Read full review

ROI

Real user quotes about their ROI:

"I can observe savings in my ROI and categorize it as money-saving. If I quantify the savings, I see about 60% in money savings and 40% in time savings.."

Shailesh Shandilya
Principle Consultant at Verinite

Read full review

"The ROI has been quite strong. I saved on licensing costs completely. Development speed improved due to mature tooling. Operational efficiency increased with stable performance. Overall, I got high value with minimal investment.."

Yash Patel
Software developer at BISAG-N

Read full review

"The ROI has been strong with OpenJDK Java due to zero licensing costs, stable performance, and reduced operational overhead, overall improving efficiency by around 25% across backend teams.."

Hussain Gagan
FullStack Developer at EnactOn Technologies

Read full review

"A return on investment has been realized, specifically in terms of money saved and time saved.

"Approximately two hours daily have been saved in terms of maintenance since implementing OpenJDK Java, and in terms of project estimation, about four to six story points are saved.."

Amol Tyagi
AWS solutions architect at Deloitte

Read full review

"I have indeed seen a return on investment with OpenJDK Java, particularly in operational efficiency. As mentioned earlier, we reduced manual document processing interventions from around 10 to 15 percent down to 2 to 3 percent. That means our team of three document processors can handle approximately 40 to 50 percent more documents per week without adding headcount. Over a year and a half, that is a significant capacity gain without hiring additional people. On the cost side, being open source means we incurred zero licensing fees. For a financial document processing platform, that represents real money saved since commercial Java solutions would have cost us thousands annually in licensing alone. Development time is another critical metric; our engineers spend significantly less time debugging parsing failures and environment issues. I estimate we save approximately 15 to 20 percent of engineering time that would have gone to infrastructure maintenance, which we redirect toward building new features and improving our LLM extraction accuracy. The speed of processing a document from ingestion to final structured output in under two minutes means we meet aggressive client turnaround SLAs, translating directly to client retention and upsell opportunities.."

Harshwardhan Gullapalli
AI Engineer at a educational organization with 51-200 employees

Read full review

Use Case

My main use case for OpenJDK Java is to build Java backend code and backend logic. A specific example of a project where I used OpenJDK Java for backend logic is in my current organization, where we are using it to build our Deva code, which is an application that functions as a digital wallet.

In addition to my main use case, while studying, I used JDK to build backend logic for my personal project, which is a form-fill assist application. .”

Verified user

Software Developer at a tech vendor with 51-200 employees

Read full review

"My primary use case for OpenJDK Java has been building scalable backend services. I use it for handling API logic, database interaction, and asynchronous processing. In one of my projects, it powered a system that handled user authentication and real-time data updates. It worked very well with frameworks such as Spring Boot, which make development faster. OpenJDK Java is essentially the backbone of the server-side applications I build.

"One real example is a healthcare-related app I worked on where OpenJDK Java handled patient data processing and prescription management. I built REST APIs that process thousands of requests daily. By optimizing JVM memory settings, I reduced API latency by around 30%. It also helped me maintain data consistency across modules. The system has been running smoothly in production with minimal downtime.."

Yash Patel
Software developer at BISAG-N

Read full review

"My main use case for OpenJDK Java revolves around incoming financial documents such as PDFs, scanned balance sheets, and trial balances from clients. Apache Tika, which runs on OpenJDK Java, parses those documents and extracts the raw text. From there, we pass that cleaned text into our OCR pipeline using Tesseract and then feed it into our large language models for structured extraction, such as account mapping and compliance classification. OpenJDK Java gives a stable, reliable foundation for that parsing layer.

One thing worth mentioning about my use case is that consistency is crucial for us. Financial documents are sensitive, and we need reliable parsing. OpenJDK Java-backed tooling provides that stability. We run it on Linux servers in production for over a year with minimal issues. The concurrent processing capability is also important because we often batch process multiple documents at once, and it handles that load without memory problems. It is not just about parsing; it is about having a dependable backbone that we can trust in the production environment handling real clients.."

Harshwardhan Gullapalli
AI Engineer at a educational organization with 51-200 employees

Read full review

"I have been using OpenJDK Java for around two years, mainly for building backend services and APIs in microservices architecture.

"My primary use case with OpenJDK Java has been building scalable backend systems, including REST APIs, async job processors, and event-driven services.

"One example is a booking system I worked on where we handled around 50,000 daily requests using OpenJDK Java with proper JVM tuning, which reduced API response times by about 30% and improved throughput significantly without increasing infrastructure costs.

"Beyond APIs, I have also used OpenJDK Java for batch processing jobs and background workers.."

Hussain Gagan
FullStack Developer at EnactOn Technologies

Read full review

“From my first or second semester, I have been using OpenJDK Java on my local PC as part of my college journey and academic learning. I have been using OpenJDK Java for personal projects and during my internship, where I also work on a project utilizing OpenJDK Java mainly for developing and testing Java-based applications such as those in banking sectors and many other systems.

“In my third semester, I had one subject, OOPs, where I needed to perform practicals and create an application in Java. I mainly use OpenJDK Java because without it I cannot support my coding needs.

“My application relies on OpenJDK Java, as my source code is in Java. Without OpenJDK Java, I cannot manage or run the application or source code because OpenJDK Java is the installer package that is needed. Otherwise, I cannot run my simple code.

“For my internship, I only use OpenJDK Java for my college project. While working at my internship, I also needed OpenJDK Java for my college project. It is compatible with almost every development studio such as VS Code and IntelliJ IDEA, which is the Java software. I used both for my college project, starting with IntelliJ IDEA to run my source codes, and in the next semester, I had a subject on advanced Java, where I learned more about OpenJDK Java and Java.

“Throughout many subjects in my college academic journey, I needed OpenJDK Java as part of my role as a support engineer. It provides almost all the core features required for Java development, including support for OOPs concepts. For instance, in one subject called OOPs, I needed JDK support to perform my Java programming. I also use OpenJDK Java for database work in DBMS with MS SQL, needing it to run their workbench..”

Verified user

Student at a university with 501-1,000 employees

Read full review

"OpenJDK Java is one of the open-source Java development kits that I use. With all the Java libraries included in the JRE, I can handle all cases regarding object-oriented programming. However, it does not include J2EE (Java Enterprise Edition). OpenJDK Java is used for Java development and can be utilized in Spring Boot, Spring framework, and other play frameworks.

OpenJDK Java provides nearly all the functionality that Java uses. When comparing OpenJDK Java to Java EE, OpenJDK Java offers better code reusability through OOPs features, class collections, and collection frameworks. These features also bring some J2EE capabilities through beans. A main feature is that OpenJDK Java has Long-Term Support (LTS). For example, Java 21 has long-term support, and Java 1.8, Java 11, and Java 17 also have long-term support. Additionally, there is no cost for development, whereas Java EE is a paid feature where beans are handled by Java. From Spring Boot and Spring framework, I can use OpenJDK Java effectively.

OpenJDK Java runs on Windows, Linux, and macOS across every type of operating system. Similar to Java, OpenJDK Java compiles with Java code into bytecode, which runs in the JVM machine. All the features included in JEE are already present, such as Java compile code, Java run program, Javadoc, and job running and debugging. Multithreading is supported by OpenJDK Java. These are powerful tools, and high performance is available. JIT compilation and adaptive optimization are also present, along with regular updates as a main feature.

Java typically updates twice a year, with updates occurring every six months. Regular updates include bug fixes that do not impact the code if I use LTS (Long-Term Support). If a small update comes, that is fine. If a big update comes, I can also handle it. LTS support will always be available as a key feature. Currently, Java 21 has LTS support, and any update that comes will support my code.."

Aadarsh Mishra
Sde Ii at Greytip Software Pvt. Ltd.

Read full review

Setup

The setup process involves configuring and preparing the product or service for use, which may include tasks such as installation, account creation, initial configuration, and troubleshooting any issues that may arise. Below you can find real user quotes about the setup process.

"Since OpenJDK Java is open source, there is no licensing cost, which is a huge advantage. Setup is straightforward, especially on Linux systems. Installation and configuration usually take just a few minutes compared to paid JDKs. It is very cost-effective. Overall, the setup experience is smooth.."

Yash Patel

Software developer at BISAG-N

Read full review

"The experience with pricing, setup cost, and licensing for OpenJDK Java is straightforward and completely free since it is open-source. We usually install it via package manager or Docker images, making onboarding new services quick.."

Hussain Gagan

FullStack Developer at EnactOn Technologies

Read full review

"I do not think there are any challenges with initial setup. However, when installing any application, everything should be set up automatically, similar to how Java is installed. There should be no need to write environment variables. They should already be set up in the system. At the beginner level, developers can only write the code and do not have to use environment variables at all.."

Aadarsh Mishra

Sde Ii at Greytip Software Pvt. Ltd.

Read full review

"We deployed OpenJDK Java on-premises. We run our OpenJDK Java infrastructure on Linux servers hosted in our own data center setup, which is important for our use case because we handle sensitive financial documents for clients including balance sheets, trial balances, and compliance data. Having on-premises infrastructure gives us direct control over data residency and security, crucial for client trust and regulatory compliance. We consider cloud options eventually; Docker containerization makes it theoretically portable to cloud environments if needed. However, we decided to keep it on-premises for data sovereignty reasons. The on-premises setup started well for us because our document volumes are predictable, allowing us to manage infrastructure costs directly. We do not need the auto-scaling flexibility that cloud would offer.."

Harshwardhan Gullapalli

AI Engineer at a educational organization with 51-200 employees

Read full review

Customer Service and Support

"Since OpenJDK Java is open source, there is no direct vendor support. However, the community support is very strong. Forums, Stack Overflow, and documentation cover most issues. For enterprise needs, some teams opt for paid support or distribution. I did not face major support challenges.."

Yash Patel
Software developer at BISAG-N

Read full review

"Community support is available, and there is also technical support with many discussions happening in different communities since it is an open platform, offering a lot of assistance. In my case, community support has been sufficient, and I did not need to reach out to technical support for the product.."

Shailesh Shandilya
Principle Consultant at Verinite

Read full review

"The community support for OpenJDK Java is massive and active. When we encounter issues or have questions, Stack Overflow usually has answers. OpenJDK Java documentation itself is comprehensive, though it is dense and assumes some Java experience. For our team, coming from Python and JavaScript backgrounds, we had to do additional learning, but the resources are there. Specifically, the Apache Tika community has helped us greatly; the documentation for Tika is solid, with good examples, and the community forums are responsive. When we run into edge cases with parsing specific document formats, we typically find discussions or solutions within the community. The trade-off with open source is that we do not pay for dedicated support; however, we are also not dependent on the vendor support queue. We need to be more self-sufficient, but this approach works for our team. We troubleshoot, read documentation, check community discussions, and usually find answers within hours rather than waiting for support tickets.."

Harshwardhan Gullapalli
AI Engineer at a educational organization with 51-200 employees

Read full review

Other Advice

"My advice to others looking into using OpenJDK Java is to please go ahead and use it because it is the best. If you are using Python, then you understand the syntax is easy in Python, but the indentation is problematic, whereas OpenJDK Java is extremely awesome; we do not have an indentation issue, and the code is understandable and not that difficult. I rate this product a 10.."

Verified user
Sr Dev Ops Engineer at a tech vendor with 201-500 employees

Read full review

"OpenJDK Java is open for everyone, every developer, and every company, and it is free of cost. There is no deployment cost or any such pricing. OpenJDK Java can be used for developing any type of software that uses API level functionality, such as backend software. I gave this review a rating of 8.5 out of 10. If a company is choosing Java tools, they should go for OpenJDK Java.."

Aadarsh Mishra
Sde Ii at Greytip Software Pvt. Ltd.

Read full review

"My advice would be to invest time in understanding JVM tuning. That is where you unlock real performance benefits. Also, use modern Java versions such as Java 17 or above. Pair it with frameworks such as Spring Boot for faster development and always monitor your application properly.

"Overall, OpenJDK Java is a very reliable and mature technology. It is ideal for

building scalable, enterprise-level applications. The open-source nature makes it cost-effective. With proper tuning, it delivers excellent performance. It is definitely a long-term, dependable choice. One real example is a healthcare-related app I worked on where OpenJDK Java handled patient data processing and prescription management. I built REST APIs that process thousands of requests daily. By optimizing JVM memory settings, I reduced API latency by around 30%. It also helped me maintain data consistency across modules. The system has been running smoothly in production with minimal downtime. I would rate this solution a 9 out of 10.."

Yash Patel
Software developer at BISAG-N

Read full review

"If someone is considering using OpenJDK Java, I advise being clear about your use case; OpenJDK Java is excellent for backend services, document processing, and financial systems where stability and long-term reliability are crucial but less ideal for lightweight microservices or serverless workloads where startup time is critical. Secondly, invest in proper tooling from day one; use Docker for environment consistency, SDKMAN for Java version management, and set up proper JVM tuning parameters early. Do not underestimate this investment; it saves months of friction later. Third, embrace the ecosystem; the Java ecosystem is mature and battle-tested. Proven solutions such as Apache Tika, Jackson, and Spring Boot should not be reinvented. Fourth, understand memory and performance tuning; the JVM is powerful but requires thoughtfulness around garbage collection and heap settings. Spend time understanding these basics upfront. Finally, if you come from a Python and JavaScript background such as my team, budget extra time for the learning curve. Java has different paradigms and conventions, but it is worth learning because the payoff in stability and production reliability is genuine.

OpenJDK Java proved itself in a real mission-critical scenario for us. Financial

document processing is not forgiving; if your parsing fails, your entire downstream pipeline breaks. The fact that we can rely on OpenJDK Java and Apache Tika to handle that responsibility consistently over 18 months speaks volumes. I would emphasize the longevity factor; OpenJDK Java is backed by the Java community and Oracle, providing confidence that this platform will be supported and maintained for years to come. That is important when building systems that clients depend on. Furthermore, if you evaluate solutions for backend infrastructure or document processing, do not overlook OpenJDK Java just because it has been around for decades. Its age is actually a strength, not a weakness, signifying that the platform has been battle-tested in countless production scenarios, where bugs have been found and fixed, and the best practices are well established. I provided an overall rating of 8 out of 10 for OpenJDK Java.."

Harshwardhan Gullapalli
AI Engineer at a educational organization with 51-200 employees

Read full review

"I have experience with TiDB Cloud as I have been using that product before. I am working right now with multiple vendors and not only with a particular database, as I am using front end and back end languages including React and Angular. I am currently writing some Python code and back end Java.

"The languages I am using include Angular, React, Java, and Python. PowerCARD is the product I am currently working on, which is a card product and an HPS product.

"I am using OpenJDK Java and confirm that I use OpenJDK Java. I have around 10 plus years of experience with OpenJDK Java. I work with the product not only as a user but also as a consultant, working as a consultant with the vendor.

"APIs impact my development speed positively as it is fast and secure, and it is easy to integrate across platforms with different tools and technologies. Almost all the tools available in the market are Java JDK enabled, making integration with OpenJDK Java very easy.

"The cloud-native capabilities in OpenJDK Java are beneficial for my projects, as about 80 to 90% of the projects I work on in the card and payment industry are running on OpenJDK Java. I work in a hybrid environment, using a mix of cloud and on-premises solutions.

"I have solutions available on AWS Cloud, and OpenJDK Java is compatible with AWS and Azure, as I use both. As of now, I have not purchased anything from the AWS marketplace.

"I give this review a rating of 8 out of 10.."

Shailesh Shandilya
Principle Consultant at Verinite

Read full review

"I have found that the learning curve for new developers adopting OpenJDK Java in our team is relatively moderate. They typically already have a background in Java, allowing them to get up to speed quickly and start contributing to our projects with some guidance and resources.

"I have had a smooth experience integrating OpenJDK Java with other technologies, particularly Spring Boot and containerization tools such as Docker. OpenJDK Java works seamlessly with Spring Boot, allowing us to develop and deploy applications quickly and efficiently.

"I handle security and updates for OpenJDK Java by regularly checking for updates and applying them as soon as possible, while also making sure to follow best practices for secure coding and configuration. I have found that using tools such as Docker and Kubernetes helps streamline the process of keeping our deployments up to date and secure.

"We monitor and manage the performance of our applications running on OpenJDK Java using tools such as Java Mission Control and VisualVM, which provide detailed insights into memory usage, CPU usage, and other key metrics.

"We have found OpenJDK Java to be highly performant and reliable compared to other Java distributions such as Oracle JDK, and in some cases, even more stable due to its open-source nature and community-driven updates. It has consistently delivered high-quality performance and reliability across various applications and deployments.

"We handle version upgrades with OpenJDK Java by regularly checking for updates and applying them as soon as possible to ensure that we have the latest security patches and features.

"Since OpenJDK Java is community-driven, we rely mostly on forums and documentation. For critical systems, we sometimes use vendor-supported builds for SLA coverage.

"I would recommend starting with OpenJDK Java if you are building backend systems, as it is cost-effective and production-ready. I suggest investing some time in understanding JVM tuning early on.

"OpenJDK Java is a very mature and reliable platform, giving enterprise-grade capabilities without the cost overhead, making it an excellent choice for most backend systems. I would rate OpenJDK Java a nine out of ten.."

Hussain Gagan
FullStack Developer at EnactOn Technologies

Read full review

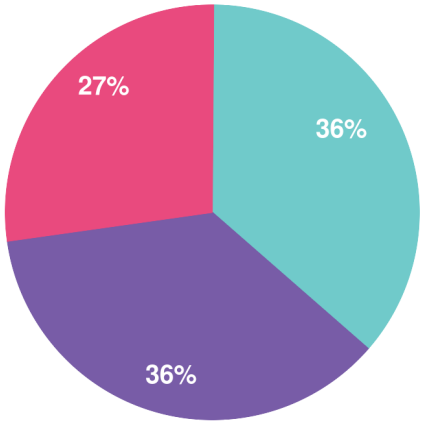
Top Industries

by visitors reading reviews

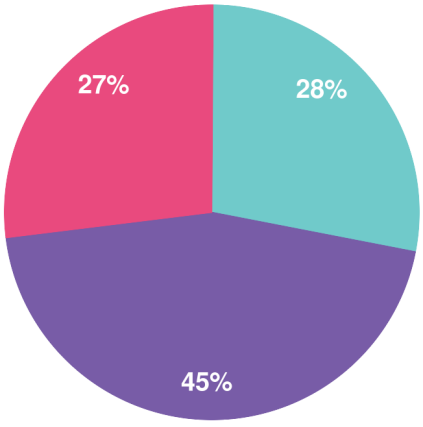
Construction Company	33%
Financial Services Firm	14%
Comms Service Provider	12%
Computer Software Company	6%

Company Size

by reviewers



by visitors reading reviews



Large Enterprise Midsize Enterprise Small Business

About this buyer's guide

Thanks for downloading this PeerSpot report.

The summaries, overviews and recaps in this report are all based on real user feedback and reviews collected by PeerSpot's team. Every reviewer on PeerSpot has been authenticated with our triple authentication process. This is done to ensure that every review provided is an unbiased review from a real user.

Get a custom version of this report... Personalized for you!

Please note that this is a generic report based on reviews and opinions from the collective PeerSpot community. We offer a customized report of solutions recommended for you based on:

- Your industry
- Company size
- Which solutions you're already considering

The customized report will include recommendations for you based on what other people like you are using and researching.

Answer a few questions in our short wizard to get your customized report.
Get your personalized report here

About PeerSpot

PeerSpot is the leading review site for software running on AWS and other platforms. We created PeerSpot to provide a trusted platform to share information about software, applications, and services. Since 2012, over 22 million people have used PeerSpot to choose the right software for their business.

PeerSpot helps tech professionals by providing:

- A list of products recommended by real users
- In-depth reviews, including pros and cons
- Specific information to help you choose the best vendor for your needs

Use PeerSpot to:

- Read and post reviews of products
- Access over 30,000 buyer's guides and comparison reports
- Request or share information about functionality, quality, and pricing

Join PeerSpot to connect with peers to help you:

- Get immediate answers to questions
- Validate vendor claims
- Exchange tips for getting the best deals with vendor

Visit PeerSpot: www.peerspot.com

PeerSpot

244 5th Avenue, Suite R-230 • New York, NY 10001

reports@peerspot.com

+1 646.328.1944