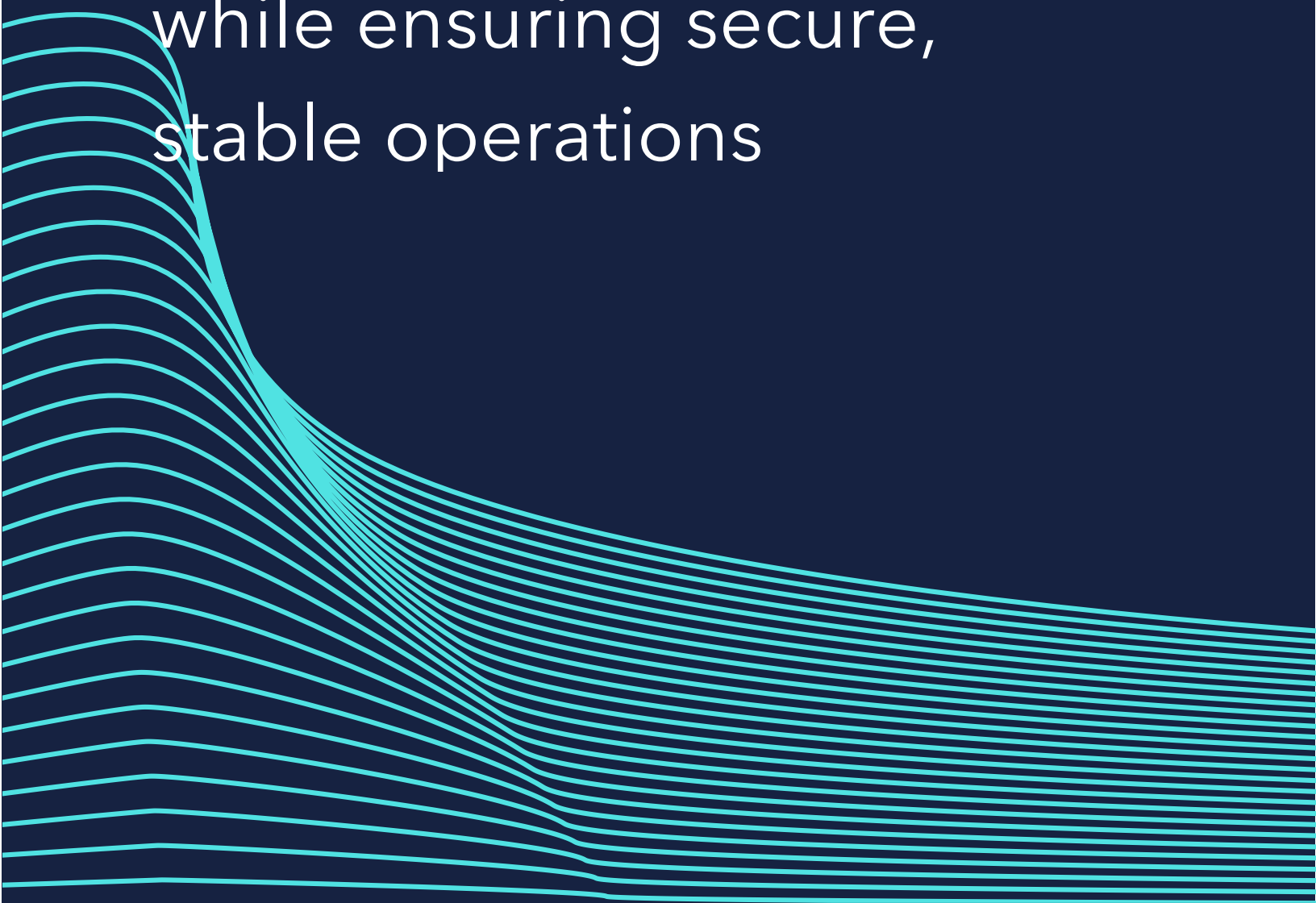
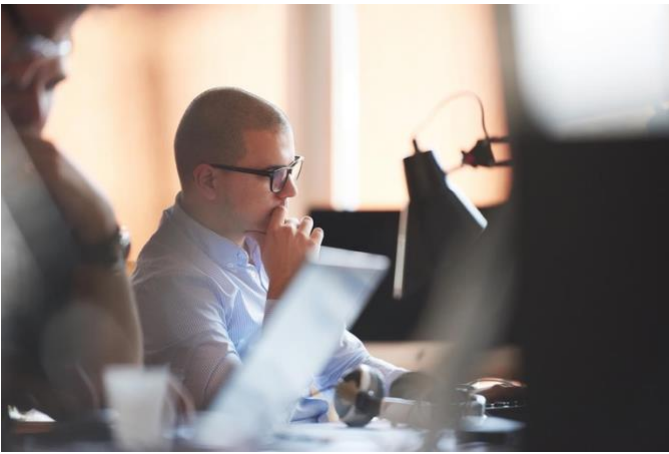


azul

Winning with Azul Platform Core:

Driving down Java costs
while ensuring secure,
stable operations

A decorative graphic consisting of numerous thin, parallel cyan lines that curve and flow from the left side of the page towards the bottom right, creating a sense of motion and depth.



For enterprises that need to ensure they have the latest patches and stability of their runtime this is going to require some decisions and possible changes to how they deploy the JDK/JRE and their applications.

Java is one of the most popular and widely used software development platforms on the planet. There are many reasons that make Java so popular, but one of the most important is the ability of the Java Virtual Machine (JVM) to scale to the largest of applications. Global enterprises such as Netflix, Amazon and eBay all use Java for large parts of their server-side architecture delivering services to millions of customers worldwide under demanding service level agreements (SLAs).

What has also helped drive Java's rapid and continued adoption by these types of companies is the free availability of the Java Development Kit (JDK) and Java Runtime Environment (JRE), including access to public updates that provide security and bug fixes on a regular basis.

The OpenJDK

When Sun Microsystems first developed Java, it released the JDK and JRE as free binary distributions for a variety of platforms. The source code remained closed and only available through a commercial licensing agreement. In 2006, Sun made the decision to release the source code for Sun's implementation of the JDK under an open-source license, specifically the GNU Public License (GPL) version 2 with Class path exception (CPE). This ensured that users could deliver

Java applications to run on the JVM without being affected by the copy left nature of the GPL. At this time the OpenJDK project was formed and, eventually, all code necessary to build the core JDK was made available.

In 2010, Oracle acquired Sun Microsystems. JDK 7 was the first release of Java under Oracle's stewardship. At this point, the OpenJDK source code became the reference implementation for the Java SE specification, which was defined through the Java Community Process (JCP) and umbrella Java Specification Requests (JSRs).

Sun, and subsequently Oracle continued to make binary distributions of the JDK freely available. These distributions of the JDK also included some components that were not covered by the JCP specification and were not released as open source. Examples of these components are the Java plugin for browsers, Java Webstart technologies and tools like Flight Recorder and Mission Control.

As Java developed, new major versions were released. Each release had periodic updates that provided patches to fix potential security issues and correct known bugs. For the current release, these updates were available free of charge. These are referred to as public updates. In addition, the previous release continued to have public updates; the result being that for a period there were concurrent public updates to two releases. The length of this overlap in public updates has varied but was typically for eighteen months.

Big Changes To Java Distributions

A few weeks before the 2017 JavaOne conference, Oracle announced that the way it developed and distributed Java would change. For enterprise users that need to ensure they have the latest patches and stability of their runtime this requires some decisions and possible changes to how they deploy the JDK/JRE and their applications. Azul is ideally positioned to help you ensure a secure and reliable Java platform whilst having the flexibility to migrate to new versions at a pace that suits your deployment strategy.

These are the announcements Oracle made about the future development and delivery of Java:

- There are two JDK version releases per year; one in March and one in September. Unlike in the past, the contents of each new release are based purely on which features are complete, tested and stable. New features are no longer targeted to a specific release, while preview features allow for testing and feedback prior to signing off as final production-ready features. This has enabled the Java platform to adopt a more agile development process and deliver new features more quickly.
- To simplify support, there is a Long Term Support (LTS) release of the Oracle JDK binary every three years. JDK 8 was classified as an LTS release. From April 2019, although updates to JDK 8 are still publicly available, the license for their use has changed. These updates are provided under the Oracle Technology Network License Agreement (OTNLA). This only permits personal use, in development, for Oracle approved use or Oracle cloud use. For other uses, an Oracle Java SE subscription must be purchased.
- From JDK 11, all Oracle JDK binaries will also use the OTNLA. Only the Oracle OpenJDK binaries will be free for general use. The effect of this is that, from JDK 11, there will be no free Oracle JDK public updates). To continue to receive free updates it will be necessary to upgrade the JDK every six months.
- Oracle has eliminated all differences between the binaries that are built purely from the OpenJDK source and the Oracle JDK. Specific features like Flight Recorder and Mission Control have had the source included in the OpenJDK project under the same GPLv2 license. Other features like the Java plugin for browsers and Java WebStart have been removed from the Oracle JDK binary. These changes were complete with the release of JDK 11.
- With the elimination of differences between OpenJDK binaries and Oracle JDK binaries, Oracle now delivers all JDK releases as OpenJDK binaries

under the GPLv2with Classpath Exception license. For these releases, there will be no commercial support available from Oracle. Oracle JDK binaries with commercial support will only be available for LTS releases.

- Oracle reduced the number of platforms for which it provides binary distributions of the JDK. Starting with JDK 9 there are no longer any Arm or PowerPC based binaries. Only 64-bit binaries are available for Windows, Linux, Mac OS X and Solaris on SPARC. 32-bit binary distributions for Windows and Linux have been discontinued.



What Impact Does This Have?

The most significant change affecting enterprise Java users is the termination of overlapping public updates to JDK releases. There is no longer the option to keep production systems up to date with security and bug fixes on the previous version of Java while testing and tuning applications to move to the new version. Combined with the increased possibility of changes that break backwards compatibility, enterprise Java users may face significantly greater challenges in planning which release of the JDK to deploy in production.

Having new features delivered to the Java platform more quickly is a great thing for developers. However, the reality is that most businesses will not be able to exploit these features until a new LTS release is made available. JDK 9, firmed up in JDK 11, included a significant change in the form of the Java Platform Module System (JPMS), often referred to by its project

name, Jigsaw. This breaks the core Java libraries into a set of modules enabling runtimes to be built that only contain the modules necessary for an application with an associated reduction in resource footprint. This is a feature well suited to the modern development approach of using containers.

Moving to a version of the JDK that includes JPMS (i.e. anything from JDK 9 onwards) also poses challenges. Part of the JPMS design comprises encapsulation of internal APIs that are used to provide the functionality of the public APIs. The most famous example of this is the sun.misc.Unsafe class. With encapsulation, these APIs are no longer available by default; additional command line options are required to expose the necessary classes. Many popular frameworks and third-party libraries use these internal APIs and are only gradually migrating away from them.

Azul has seen a clear need in the market for alternative support plans that are designed to help businesses gain the benefits of Java’s new release cadence without incurring incremental risk to their operation - whether they build on the latest cutting-edge feature release or only on LTS releases.



Azul’s Enhanced Java Support Initiatives

Azul Systems believes that enterprise Java users need support that allows them to continue to use Java versions for extended periods of time and with overlap between releases to allow newer versions to stabilize before being deployed in production.

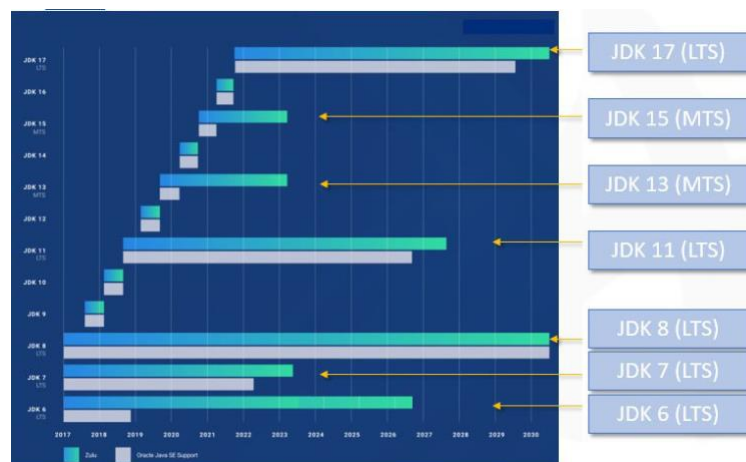
To provide this, Azul is offering two levels of commercial support for releases of the JDK. This

includes both active support (quarterly updates including fixes and security patches) and passive support (addressing specific issues reported by customers). The definitions of these two levels of support are as follows:

1. Medium Term Support (MTS): Eighteen months of active support for JDKs 13 and 15 followed by one year of passive support. These releases are intended to provide a bridge between LTS releases enabling the use of new JDK features earlier.

2. Long Term Support (LTS): Eight years of passive support followed by two years of passive support.

The diagram below shows how different versions of the JDK will fit into this new scheme.



To deliver this extended support, Azul will use the source code for fixes and security patches from whatever releases currently have public updates and back-port these to the appropriate release before building and testing binaries.

Azul’s engineering team have decades of combined experience working with the JVM and have been integrating code in this way for several years. In addition, Azul’s engineering teams have the skills required to triage bugs reported by a customer, identify the root cause of the problem and create independent resolutions that can be supplied to customers, as needed.

Azul's support plans are extremely cost-effective; Java support for thousands of servers is typically available for less than the cost of one full-time engineer.

Azul Platform Core Includes Certified Azul Zulu Builds of Open JDK with Great Support and the Tightest SLAs on the Market

Azul Platform Core is founded about the tested and certified Azul Zulu Builds of Open JDK. Azul Platform Core includes the commercial support offering for enterprise internal usage on servers and desktops, with distribution options if required for ISVs/OEMs who integrate a JVM with their product and distribute the bundled product to third parties.

To ensure that Azul Platform Core fully conforms to the Java SE specification all binaries are tested using the Technology Compatibility Kit (TCK).

Every commercial binary deployed with Azul Platform Core is built only from open source, which is also verified to ensure non-contamination. Through the use of specifically developed tools, Azul scans and analyzes the full set of build artifacts and object materials and the more than 7 million lines of Open JDK sources that they are produced from (including intermediate and dynamically-generated source files). This analysis covers all topological paths and relationships between any code that might run on the resulting Azul Platform Core product and any internal components. Azul verifies that the relationships between the multitude of OSS licenses involved within the Azul Platform Core product and any code that may run on the Azul Platform Core platform will not result in contamination, and do not impose any requirements or restrictions on licensing of the code that runs on JDKs or JREs provided by Azul Platform Core.

Azul Zulu Builds of OpenJDK are available, free of charge, for a variety of platforms:

- Windows x86 64 and 32-bit
- Linux x86 64 and 32-bit
- macOS X 64-bit and Arm 64-bit
- Linux Arm v6 and v7 32-bit
- Linux Arm v8 64-bit
- Linux PPC64

Azul's Commitment to Java

Azul's goal is to provide Java users with the most comprehensive range of JVMs possible. For those looking for higher performance, there is Azul Platform Prime. Azul Platform Prime delivers a JVM, founded on OpenJDK but with an alternative implementation of both garbage collection (GC) and JIT compilation to eliminate GC pause concerns and provide higher application throughput, typically delivering 30%+ infrastructure savings.

With Azul Platform Core, users can use a standard build of the reference OpenJDK source code with extended support. Azul's extended support for all Java releases ensures continued security and stability for enterprise operations. In addition, this allows users to migrate to new releases of the JDK at a time of their choosing, mitigating risks associated with being forced to change platform.

In addition, Azul is actively engaged in helping to shape the development of Java and the JVM. Azul has been along-standing member of the Executive Committee of the Java Community Process (JCP) since 2011 and is a member of the Java SE Expert Group (EG) since Java SE9. In addition, Azul employees lead the OpenJDK 7, 13 and 15 projects, submitted the first JDK Enhancement Proposal (JEP) from outside Oracle to be accepted and included in the JDK (JEP 285), and successfully delivered Open JDK for Mac Silicon in 2020 (JEP 391).

Contact Azul for more details of commercial support terms and conditions. Learn more about Azul's Java support options and roadmap.

Contact Azul

385 Moffett Park Drive, Suite 115

Sunnyvale, CA 94089 USA

☎ +1.650.230.6500

www.azul.com