

azul



Freedom from Oracle Java Fees.

7 things to know before renewing
your Oracle Java contract.



Azul powers the world's most mission-critical and demanding Java applications.

Azul, provider of the Java platform for the modern cloud enterprise, is the only company 100% focused on Java. Millions of Java developers, hundreds of millions of devices, and the world's most highly regarded businesses trust Azul to power their applications with exceptional capabilities, performance, security, value, and success.

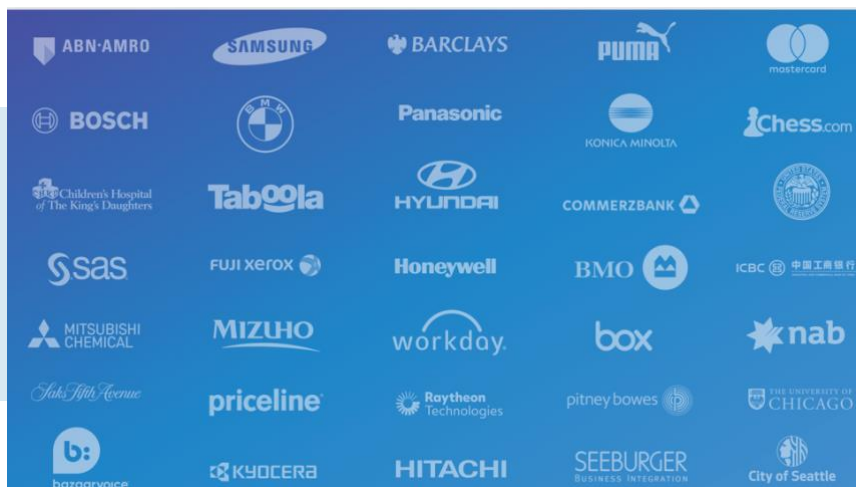
100% of the world's Top 10 Trading Companies

50% of Forbes Top 10 World's Most Valuable Brands

27% of the Fortune 100 Companies

<p>Azul Platform Core The world's most secure & stable builds of OpenJDK Low Cost Secure</p>	<p>Azul Platform Prime The world's most secure & stable builds of OpenJDK Start Fast, Run Faster, Stay Fast</p>	<p>Azul Intelligence Cloud Analyze & optimize your Java fleet with actionable intelligence More Focus, Awareness, Performance</p>
<ul style="list-style-type: none"> • TCK-tested and certified builds of OpenJDK • Stabilized builds for immediate deployment • Supports more versions of Java, including 6, 7, 8, 11, 13, 15, 17 & 18 • Millions of deployments • Curated to be production-ready • Support for Azul Zulu Builds of OpenJDK and Eclipse Temurin <p>Secure24x7x365 Support</p>	<ul style="list-style-type: none"> • Optimized OpenJDK build with higher throughput and carrying capacity • Eliminates pauses, faster startup and efficient compilation • Save infrastructure costs, in cloud or on-prem • Improve responsiveness of Java-based technologies, e.g., Kafka, Cassandra, Spark, Solr, Hadoop • Low latency Java (in trading systems) <p>Secure24x7x365 Support</p>	<ul style="list-style-type: none"> • Runtime analytics to deliver laser-focused visibility • Operational intelligence to keep runtimes dependable, secure and efficient • Cloud powered optimization for more performance with Cloud Native Compiler • Analytics providing teams with actionable intelligence • Powerful optimizations at reduced cost <p>Secure24x7x365 Support</p>

World Class Customers Who Have Switched to Azul.



azul

Freedom from Oracle Java fees.

Use your budget to build apps. Not Oracle's piggy bank.



Executive Summary: Freedom has many definitions: the power or right to act, speak or think as one wants; and the absence of necessity, coercion and constraint, in choice or action.

In open source, freedoms include and distinguish between “free as in beer” (no obligations are expected of you when you use something), and “free as in speech” (no restrictions on source code access and modification).

In this White Paper, we outline:

- Why Azul delivers freedom from Oracle Java fees
- 7 things to know before renewing your Oracle Java contract
- More about OpenJDK, Oracle and Azul
- Transitioning to Azul (it's easy)
- About Azul

Why Azul delivers freedom from Oracle Java fees

For many, Oracle is seen as the steward of Java, with Oracle having acquired Sun Microsystem, where Java was born, grew up and made (freely). This has meant Java has consistently been among the top 3 programming languages of the renowned [Tiobe Index](#) for decades!

However, when Sun Microsystems was acquired by Oracle, Java became a sales commodity to be sold and monetized as an overlay across its database, cloud and product portfolio. The proactive community open spirit of Sun Microsystems became superseded by a commercial sales model, and when Java 8 and 11 support was no longer provided by Oracle without an Oracle Java SE subscription, Java users were exposed.

Thankfully, Sun Microsystems had granted open source Java, **OpenJDK**, to the community prior to its acquisition by Oracle. While Oracle drives new features and is always at the helm of the latest OpenJDK versions, it does not maintain the versions people actually use in production except for customers with Java SE subscriptions. Azul, by contrast, carries the community spirit of Sun Microsystems, dedicated to ensuring the versions of Java that you use remain secure and stable, and ensuring that Java is appropriately supported to be the language of choice in the enterprise.

Azul builds are based on the same OpenJDK source code as the Oracle JDK, the Java Development Kit (“JDK”) which underpins the Oracle Java SE support product. As the code base is the same, Azul JDKs are drop-in replacements for Oracle JDKs, and provided for more Java versions.

Azul is committed to ensuring that Java distributions are and remain free, “as in beer” and “as in speech”, with a support product - Azul Platform Core - that costs 70% less than Oracle Java SE and delivered by engineers that use and care about Java, not about selling products.

As your Oracle Java renewal approaches, get armed and equipped with your support, cost and transition options, and at the very least aid your negotiation with Oracle sales representatives.

Reason 1:

Azul Platform Core costs 70% less than Oracle Java SE

A scan of pricelists and stores reveals the differentials

The Azul Solution: Just look at the numbers! The higher the volumes, the greater the savings.

Azul vCores*	Oracle Processor Licences**	Oracle Java SE Subscription (total annual price)	Azul Platform Core Support (total annual price)	Annual Savings
900	225	\$64,125	\$18,430	\$45,700
2400	600	\$144,000	\$44,240	\$99,760
9000	2250	\$472,500	\$163,070	\$309,430
24000	6000	\$1,080,000	\$410,200	\$669,800
48000	12000	\$1,800,000	\$410,200	\$1,390,000
72000	18000	\$2,700,000	\$410,200	\$2,290,000

* Assumes hyper-threading is enabled, i.e. ,1 core = 2 vCores

**Assumes Oracle Core Factor is 0.5, which is the case for Intel Xeon or similar

Remember, Oracle charges for all cores on a box, whether the Oracle JVM is in use on all or not. With Azul, you license only for where you use Java.

Azul also costs 15% to 30% less on desktops. The higher the desktop volume, the greater your savings
Azul Platform Core and Oracle Java SE are equivalent Java products, both built on OpenJDK.
Azul is open source, with great Java support and security, at 70% less cost.

Reason 2:

Azul Platform Core is a drop-In replacement for Oracle Java SE

Often you hear *FUD* from sales reps negotiating Oracle Java renewals that "it takes months to migrate your JDK." No, it does not. Azul Platform Core-supported distributions are drop-in replacements.

The Azul Solution: Simply switch your Java Home...

1. EXAMPLE CURRENT PRODUCTION STATE

Tomcat application server running on Oracle Java and hosting one web application

A. In the process list below, there is one Tomcat Application Server java process.

```
[centos@centos8 bin]$ sh startup.sh
Using CATALINA_BASE:   /opt/training/apache-tomcat-8.5.61
Using CATALINA_HOME:   /opt/training/apache-tomcat-8.5.61
Using CATALINA_TMPDIR: /opt/training/apache-tomcat-8.5.61/temp
Using JRE_HOME:        /opt/training/jdk1.8.0.202
Using CLASSPATH:        /opt/training/apache-tomcat-8.5.61/bin/bootstrap.jar:/opt/training/apache-tomcat-8.5.61/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
[centos@centos8 bin]$
[centos@centos8 bin]$
[centos@centos8 bin]$
[centos@centos8 bin]$
[centos@centos8 bin]$
[centos@centos8 bin]$
[centos@centos8 bin]$ ps -ef | grep java
centos    1473      1 99 19:18 pts/0    00:00:14 /opt/training/jdk1.8.0.202/bin/java -Djava.util.logging.config.file=/opt/training/apache-tomcat-8.5.61/conf/logging
zoo-2048  -Djava.protocol.handler.pkgs=org.apache.catalina.webresources -Dorg.apache.catalina.security.SecurityListener.IMASK=4027 -Dignore.endorsed.dirs=-classpath /
ar -Dcatalina.base=/opt/training/apache-tomcat-8.5.61 -Dcatalina.home=/opt/training/apache-tomcat-8.5.61 -Djava.io.tmpdir=/opt/training/apache-tomcat-8.5.61/temp org.
centos    1586      0 19:18 pts/0    00:00:00 grep --color=auto java
[centos@centos8 bin]$
```

"Azul was a drop-in replacement... Had we known how seamlessly and smooth the process would be, we would have switched from Oracle Java to Azul Java much earlier on."

B. The Java/JDK version used by the above Tomcat process is Oracle Java 8

```
[centos@centos8 bin]$ /opt/training/jdk1.8.0.202/bin/java -version
java version "1.8.0_202"
Java(TM) SE Runtime Environment (build 1.8.0.202-b08)
Java HotSpot(TM) 64-Bit Server VM (build 25.202-b08, mixed mode)
[centos@centos8 bin]$
```

Michael Thalhammer
Head of Software Development,
Commend International GmbH

2. MIGRATION TO ZULU OPENJDK

Now, let's migrate the above Tomcat server from Oracle Java to Zulu OpenJDK

A. Edit file [apache-tomcat-8.5.61/bin/setenv.sh](#) and add `JAVA_HOME` variable to point to Zulu OpenJDK.

B. Save and Exit.

This is the Migration part

```
export JAVA_HOME=/opt/training/zulu8.36.0.1-ca-jdk8.0.202-linux_x64
...
...
...
"setenv.sh" 1L, 68C
```

C. Restart Tomcat Application

D. Verify Tomcat is now using Zulu OpenJDK (and not Oracle Java) by listing all the Tomcat java processes.

```
[centos@centos8 bin]$ sh startup.sh
Using CATALINA_BASE:   /opt/training/apache-tomcat-8.5.61
Using CATALINA_HOME:   /opt/training/apache-tomcat-8.5.61
Using CATALINA_TMPDIR: /opt/training/apache-tomcat-8.5.61/temp
Using JRE_HOME:        /opt/training/zulu8.36.0.1-ca-jdk8.0.202-linux_x64
Using CLASSPATH:        /opt/training/apache-tomcat-8.5.61/bin/bootstrap.jar:/opt/training/apache-tomcat-8.5.61/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
[centos@centos8 bin]$
[centos@centos8 bin]$
[centos@centos8 bin]$ ps -ef | grep java
centos    1618      1 99 19:27 pts/0    00:00:16 /opt/training/zulu8.36.0.1-ca-jdk8.0.202-linux_x64/bin/java -Djava.util.logging.config.file=
8.5.61/bin/tomcat-juli.jar -Dcatalina.base=/opt/training/apache-tomcat-8.5.61 -Dcatalina.home=/opt/training/apache-tomcat-8.5.61 -Djava.io.tmp
centos    1651      0 19:27 pts/0    00:00:00 grep --color=auto java
[centos@centos8 bin]$
```

Azul Platform Core and Oracle Java SE are equivalent Java products, both built on OpenJDK. As a result, transitioning to Azul Platform Core is a frictionless switch.

Reason 3:

Your external applications that currently run with Oracle Java will also run with Azul Platform Core. No vendor certification is required.

Sometimes application owners perceive that externally provided software applications - commercial ISV or open-source community applications - specifically require Oracle Java. They don't. Remember, Oracle Java is based on OpenJDK ! That means OpenJDK builds provided with Azul Platform Core also work with those same applications.

The Azul Solution:

Any external ISV application or community application running Oracle Java SE will run Azul Platform Core. Ideally, your software provider's documentation will state that they support OpenJDK and Oracle Java. If their documentation doesn't state OpenJDK support, raise a ticket with them, and tell them that Oracle Java and OpenJDK - and hence Azul Platform Core - are interchangeable.

ISV Ecosystem and Java Support

- Most ISVs don't bundle a JDK, but will support/certify one or more (e.g.Oracle JDK, OpenJDK, Azul Zulu JDK, etc)
- While Oracle Java was once the gold standard, "OpenJDK" is the new default
 - Most ISVs already support "an" OpenJDK distro: that means Azul builds of OpenJDK are supported as it meets "TCK" (Technology Compatibility Kit" compliance.
- Azul has **never** encountered a Java application Azul couldn't support
- Oracle JDK is based on OpenJDK.

Azul has a 100% track record that if the ISV product runs with a Oracle JVM (Java Virtual Machine) Oracle JRE (Java Runtime Environment), or Oracle JDK (Java Development Kit) or OpenJDK, it will work with Azul Platform Core distributions (this does not include Oracle propriety extensions).

Reason 4:

Azul Platform Core delivers security vulnerability fixes immediately (like Oracle Java SE) and as secure, stable "CPU" builds that can be immediately deployed without fear of regression (like Oracle Java SE).

As many regulatory and governmental edicts remind us (e.g., Australia's "Essential Eight" government guidance, the Payments Industry's PCI-DCC regulation, US utilities NERC CIP-013 protocol, and many others, rapid delivery and deployment of "uncoupled" secure, stable builds (in minutes and hours) are essential.

Tightly coupled security and features patches

It is always recommended that security patches be applied as soon as possible. However, some vendors do not provide separate security and feature update patches. require a new feature, being forced to apply the new feature by a vendor process risks, as certain business processes may rely on features remain

Risks mitigated by Azul
(and Oracle)

The Azul Solution

The actual Azul delivery track record for *all* Platform products is immediate, like Oracle's.

Why Stable Secure Builds Are Important?

- Most enterprises prefer to deploy stabilized security builds into production, common practice for more than a decade, in Java and in other languages
- Azul Platform Core gives you stabilized security builds, the same ones enterprises traditionally prefer to use, so security fixes can be deployed with confidence, including when they are urgently needed.
- Other upstream OpenJDK quarterly updates include (along with its security changes) hundreds of non-critical changes, not tested in production prior to the update
 - Stability regressions are inevitable, with a regular "Russian roulette" occurrence (e.g., 8u262, July 2020).
 - It broke several common applications
 - For example, Hadoop clusters, Solr/Lucene, and more
- Regressions, when they do happen, are usually remedied in the next quarterly update or a mid-quarter re-spin, but this significantly delays access to security fixes to vulnerabilities where attackers now have access to source code

Two Azul engineers are members of the Java Security Vulnerability Group.

Azul Platform Core and Oracle Java SE are equivalent Java products that consistently deliver updates immediately when security vulnerabilities are announced and as "uncoupled" stable secure builds.

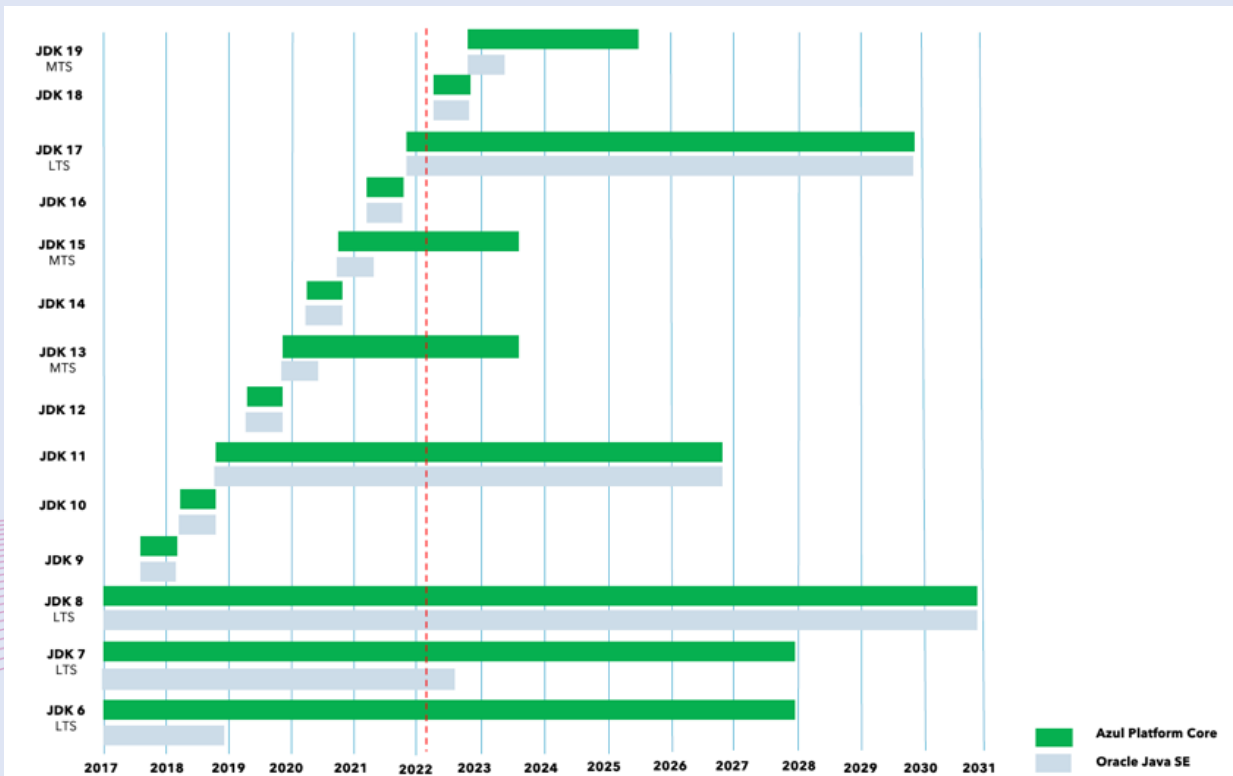
Reason 5:

Which Java versions do you use?

Oracle Java SE only supports a few Java versions.

Check your Java estate, especially your legacy estates. Do you use any Java 6 or 7? One bank told us 70% of their Java production application estate is 7-based, a version which Oracle will stop providing support for in July 2022.

The Azul Solution: Azul Platform Core simply supports more Java versions, including Java 6 and 7. Users of Java 7 and 6 can ask Azul for extended support for their Platform Core binaries, ensuring continuous access to “security and stable” releases. This extended support will help enterprises minimize migration costs and potential disruption by enabling application upgrades based on flexible timelines. First released in 2011 and 2006 respectively, Java 7 and 6 were significant LTS releases and still support many mission-critical applications in enterprises, government and public sector organizations today.



Azul Platform Core supports more Java versions, meaning that you can upgrade your Java applications in your own time, not at the licensing whims of Oracle.

Reason 6:

Oracle is Oracle; Azul is easy to work with

Oracle's sales tactics are well known, and as Oracle compliance-specialists, Palisade Compliance makes clear on their [web-site](#), Oracle Java is complex - "Java went from free, to not free, to sometimes free under specific circumstances. As typical with changes like this, not everyone gets the memo as sales teams put the pressure on to sign new Java ULAs. The result is billions to Oracle."

Furthermore, in their December 2021 blog entitled "[Oracle is now auditing Java](#)," Palisade Compliance claims "Oracle LMS is now involved in Java audits: "[Until recently] Oracle was not auditing their customers over Java. Oracle was using their sales teams to put pressure on customers and threaten audits. That has now changed. Oracle's audit team, License Management Services, or LMS, is officially involved in these activities."

While no audit letters have been received, LMS is getting on the phone with Oracle customers, pushing those customers to run scripts..... It may not be an audit supported by an official letter, but it's still an audit. A good rule of thumb, if walks like an audit, talks like an audit, and smells like an audit, then it's an audit. And here we have something that smells very bad, just like an Oracle LMS audit."

The Azul Solution:

Azul is simple and easy to work with, no lock-in, no strings and with engineers that want to talk about Java. With Azul, you can keep calm and carry on!

Results that speak for themselves.

"In talking with Azul, the trust came quickly, and was continually reinforced by the excellent support we received. Azul gives us peace of mind."

Travis Spencer, CEO, Curity



Reason 7:

Do you actually use Oracle support services?

Can you get through to Support? What is the quality of service? Do you have to explain your problem multiple times with different people?

The Azul Solution: With Azul, you actually get Java-dedicated Support from Java experts that organizations *want* to learn from.

The Azul support team - all Java experts, selected for their industry experience - are available 24 x 7 x 365 to discuss any CVE security concerns you might have, and help with any out-of-cycle updates and patches that might happen. With the recent log4shell vulnerability - which in no way impacted any Azul products; it impacted Java applications rather than foundation OpenJDK and Java - the Azul Support Team provided a statement on the Support site within a day, over the weekend.

In addition, Azul provides regular security briefings which provides credible guidance concurrent with published Java vulnerabilities and their fixes. Azul customers use this forum to raise concerns, get clarifications and determine a patch strategy based on their infrastructure and risk appetite, keeping Azul customers steps ahead of hackers and cyberbullies, delivering peace of mind to their stable, secure Java environments.

100% customer satisfaction.
100% Java.

Our team of Java-only engineers—the largest in the world—average 20+ years of experience and are passionate about empowering the modern cloud enterprise.

"Azul gives us peace of mind"

"I've become a raving fan of Azul in no small part because of the support we get," says Spencer. "We've opened half a dozen cases and have gotten great help each time. The cases we open are really complicated ones -- things like TLS 1.3 support, ChaCha20-Poly1305 support, RSA-PSS support, HSM fixes - tricky stuff, yet we always get the help we need and help in workable timeframes. It feels really good to get the help from knowledgeable people that aim to please."

Azul provides much more than secure Java.

Azul provides exceptional support, loved by customers, which is dedicated to your maximum security and stability, and reducing your risks 24 x 7 x 365.

Do Apache Log4j CVE-2021-44228, CVE-2021-45046, or CVE-2021-45105 affect Azul's JVMs?

December 10, 2021 20:18

Follow

Question:

Do Apache Log4j CVE-2021-44228, CVE-2021-45046, or CVE-2021-45105 affect Azul's JVMs?

Answer:

These CVEs do not affect the JDK directly, and while there have been related vulnerabilities which have affected JDK versions in 2009, 2017, and 2018, versions since then have been secure against those issues.

However, the new CVEs are reported against Apache Log4j, which is a logging framework used almost everywhere in Web Application frameworks. You will need to take steps to find and secure any environment which uses Log4j.

A safe long-term strategy that mitigates all three of the recent CVEs is to upgrade to Log4j 2.17.0 everywhere. It's possible that deploying this update may cause issues with your web

Join thousands of customers that trust Azul to deliver.

"There are no licensing restrictions with AzulGreat service. Flat rate. Flexible. No reporting. Easily reachable and responsive. That's Azul."

Bernd Eckenfels,
Chief Architect SEEBURGER.com



"We chose to work with Azul to de-risk the technology build and allow our team to focus on developing business logic. We got our Java signed, sealed, and delivered from the Java experts, and could keep our singular focus on creating the best exchange."

Dominick Paniscotti,
Chief Technology Officer, MEMX



**Azul provides the same open source Java
LOWER COST. BEST-IN-CLASS SUPPORT.**

**Customers can reduce their Java TCO up to 90%
GUARANTEED MINIMUM SAVING OF 25%.**

More Java versions supported: 6, 7, 8, 11, 13, 15, 17, 18

"It's been surprisingly amazing working with Azul®. They're always very responsive when I do need some help, and one of the great things is that the support team has helped me find bugs in other people's software."

Ted Boehm
Chief Platform Architect, Mastercard



"At Software AG, we have used Azul Platform Core as well as other products for over a year, and we can say first hand that our customers have remained wholly satisfied with the quality of the Azul JDK and Azul services."

Dr Gudrun Gelfand
Senior Director, R&D, Software AG



**Azul is the largest company 100% focused on Java and the JVM
Azul has supported Java deployments for over 19 years**

More on OpenJDK, Oracle and Azul

Java drives commercial leading/bleeding edge commercial technologies, often cloud-centric, driving the latest technology innovation, for example as part of Software/Product/Application-as-a-service technologies. Java also drives stable mature enterprise mission-critical applications, at the heart of many commercial and public sector applications, including financial services, healthcare, industrial manufacturing, defense, civilian agencies, and utilities. While many such applications are actively maintained, others are “legacy”. Java also supports embedded use cases, for example IoT devices, in connected car environments, smart devices. and in telecommunications.

Whatever the deployment and Java use case, it is vitally important to install updates where necessary to ensure the Java platform is as secure as possible. That means understanding how JDKs and runtimes are packaged, and ensuring you have appropriate Java Support services, such as Oracle Java SE or Azul Platform Core.

The OpenJDK project provides an open-source implementation of the Java SE specification, which is free to use under the GPLv2 with the Classpath exception (CPE) license. Many providers use this source code to build JDK distributions. A major analyst firm recently predicted that by 2025, more than 70% of Java applications will be deployed on third-party Java runtimes, up from 30% in 2019. That has been corroborated by recent surveys, such as those provided by Snyk, JRebel and Payara, which while focused on limited sample sizes and specific personas, indicate that that projection is coming to fruition.

Indeed, when Oracle changed the license of the Oracle JDK (in September 2018), they realized this would lead to a proliferation of OpenJDK distributions, both free and with paid support. To ensure that the security of Java continued at the highest possible level, they created the OpenJDK Vulnerability Group. This is a group of engineers representing companies and organizations producing distributions of the JDK. Even though the group is part of an open-source project, it is a closed group. All communication between members is encrypted, and none of the messages are published. This is because of the sensitive nature of the group's work which examines details of vulnerabilities (CVEs) reported in the OpenJDK. Members of the group work collaboratively to produce patches to remedy the vulnerabilities. Several Azul engineers are members of this group. All patches are developed *only* for the current release of the JDK.

Did you know?

Azul employs 21 “Committers” to the umbrella OpenJDK Project, runs 3 Version-specific projects and 1 major Java enhancement project. (CRaC)

Since Azul engineers are aware of all security patches included in an OpenJDK, they can work on backporting these patches to older versions of OpenJDK that Azul support.

The current support timelines for Azul Java versions are shown in the table below:

JDK Version	End of Azul Commercial Support
JDK 6	December 2027
JDK 7	December 2027
JDK 8	December 2030
JDK 11	September 2026
JDK 13 (MTS)	March 2023
JDK 15 (MTS)	March 2023
JDK 17	September 2029

For Azul, JDK 13 and JDK 15 are medium-term support versions; all others are long-term support (LTS).

Scheduled updates are provided four times a year in January, April, July, and October. As a result, Azul engineers can ensure that all necessary update packages are built and pass all TCK tests *before* Oracle release an update. Azul can deliver updates effectively simultaneously with Oracle's (in this case, simultaneous means within one hour).

Azul ensures that *all* security patches are applied to *all* versions where applicable.

The caveat here is that if there is a feature introduced in a newer version of Java that requires a patch, it will not be possible to backport it to an older version since the feature does not exist.

The only situation where Oracle will provide additional security patches is for components that are not part of the OpenJDK. Prior to JDK 11, Oracle included certain components they termed *commercial features*. These features are for the deployment of applications in a desktop environment, specifically the Applet Browser Plugin and Java Web Start. These components remain proprietary and closed source. However, Oracle terminated support for the Browser Plugin in March 2019, so there will be no security patches for this component.



Since Azul uses the same OpenJDK source code as Oracle and is a member of the OpenJDK Vulnerability Group, Azul Platform Core provides Zulu builds of OpenJDK, identical to the Oracle JDK since JDK 11. For earlier releases, the only potential difference will be in the closed source Java Web Start deployment technology, which is not included in Azul's distribution.

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.

When Oracle acquired Sun Microsystems in 2010, Oracle continued to make binary distributions of the JDK freely available. However, a few weeks before the 2017 JavaOne conference, Oracle announced how 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 required 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, all for support at 70% less cost than Oracle.

Here are some of the support licensing details:

- Oracle Support: In simple terms, Oracle support Long Term Support (LTS) releases of the Oracle JDK binary historically in recent years every three years, and every two years from JDK 17 onwards. JDK 8, 11, and most recently 17 were classified as an LTS release. However, from April 2019, while updates to JDK 8 and 11 were still publicly available, the license for their use changed to one provided under the Oracle Technology Network License Agreement (OTNLA) and 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**, the pricelist publicly listed at <https://www.oracle.com/java/technologies/java-se-subscription-faq.html>.
- Azul Support Cost Comparison: For Azul pricing, see <https://www.azul.com/products/pricing/> and specifically the [Price comparison calculator](#) comparing Azul Platform Core to Oracle Java SE. Azul Java support fees typically cost 70% less than Oracle on server environments, on desktops between 15% to 30%.
- Oracle has eliminated all differences between the binaries that are built purely from the OpenJDK source and the Oracle JDK. Specific features that historically shipped with Java 8 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.
- Oracle reduced the number of platforms for which it provides binary distributions of the JDK. After JDK 9 there were 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. Azul supports a greater range of (so-called legacy) platforms. See <https://docs.azul.com/core/zulu-openjdk/supported-platforms>
- Azul sees a clear need 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 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.



- Azul offers 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): Usually eighteen (18) 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): Usually eight (8) years of passive support followed by two years of passive support.

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.

Transitioning from Oracle Java to Azul Platform Core

Good news: every workload transitioned from Oracle JDK to Azul Zulu Builds of OpenJDK, the premier build of Azul Platform Core, has been successful - that is millions and millions of workloads!

This is **not** a long, daunting process as Oracle would like you to believe. One of Azul's banking customers converted 2500 applications over a weekend, while one of the largest entertainment companies in the world, project managed a transition of thousands of applications within a couple of months. The switch itself, easy, the project management to ensure all JDK and runtime instances are known, switched, and tested, a little more onerous but still relatively straightforward, equivalent to project managing a dot- or point- release of a popular application.

The Azul Zulu JDK, Azul's supported distribution provided with Azul Platform Core bundle, is a drop-in replacement to Oracle JDK, providing **identical** levels of service and updates. Oracle JDK is built from the OpenJDK repository and core functionality like the JVM, libraries, etc. are completely interchangeable with the Azul Platform Core's Zulu OpenJDK binary distribution. No modifications to source code are required nor is a recompilation of application code necessary.

In addition, each Azul binary passes all the tests of the Technology Compatibility Kit (TCK) provided as part of the relevant Java Specification Request (JSR). There are over 150,000 tests that ensure a binary conforms to the defined specification and provides a high level of confidence of functional equivalence between tested JDKs.

Based on customer feedback, and to show the ease of transition upfront, we will start this guide with the Implementation Phase, addressing the multiple approaches to Planning later in the document.

Azul suggests a 3-phase project management approach to transitioning applications to Azul Platform Core:

- Planning
- Implementation
- Testing

Remember! The switch itself is simple and instantaneous!

Phase 1: Planning - Identify machines/instances needing to transition and the major JDK versions in use.

Phase 2: Implementation - transition time!

Download the Azul Zulu JDK - Azul Platform Core customers have credentials to access the Azul binary download site. A variety of formats for each supported version are provided to give maximum flexibility.

Install the Azul Zulu JDK: The installation process depends on which format of distribution is being used:

Zip file. Use a utility or tool to unzip the archive dependent on which operating system is being used. This is a manual installation method so the Azul Zulu JDK can be installed in a directory of your choosing.

Compressed tar file (.tar.gz). Use the UNIX command `tar -xvf <zulu-package>.tar.gz`. This is also a manual installation method so any directory can be chosen.

Windows MSI file. From the Windows command line, run `msiexec /i <zulu-package>.msi /qn`. This will install Zulu into the `C:\Program Files\Zulu\<zu-lu-jdk>\` directory.

Linux RPM file: Install from a command prompt using the command:

- On Red Hat Enterprise Linux: `yum install <zulu-package>.rpm`
- On SUSE Linux Enterprise Server: `zipper install <zulu-package>.rpm`

Linux DEB file: On Ubuntu or Debian use the command: `apt install <zulu-package.deb>`.

MacOS DMG file. This can be installed graphically from the desktop or the command line using the command: `hdiutil mount <zulu_package>.dmg`

Full instructions for installation can be found in the Azul documentation at <https://docs.azul.com/core/#>.

Additional Fonts - Oracle JDK prior to JDK 11 included additional Lucida fonts. To provide compatibility with the Oracle JDK, Azul provides the Azul Commercial Compatibility Kit (CCK). For desktop machines, the CCK should be installed if graphical applications are being used. Details of the available CCK files and installation instructions can be found at <https://www.azul.com/products/components/commercial-compatibility-kit/>.

Desktop Transition - if you use desktop applications with Browser Plug-ins (Applet) and Java Web Start

functionality, please see section Desktop Machine Transition at the end of this document. These are not included in the Azul Zulu OpenJDK binary distributions.

Application Configuration - Having installed the Azul Zulu JDK on a machine, it may be necessary to reconfigure applications to use the new JDK.

Each application will vary in how it determines where the Java executable is located. Here are some common scenarios.

1. The PATH environment variable. This is set differently depending on which operating system is being used. The PATH environment variable should be modified to include the bin directory of the Azul Zulu JDK installation as the first place where an executable called java is located.
2. The JAVA_HOME environment variable. Similar to PATH, this is also set differently depending on the operating system in use. JAVA_HOME indicates where the JDK is installed and should be set accordingly for the Azul Zulu JDK. Note that it is the installation directory so, unlike changing PATH, should not reference the bin sub-directory. JAVA_HOME is used by some applications (e.g. some application servers) but is not used universally by all applications.
3. For the Tomcat server, the default configuration file should be modified to reflect the location of the Azul Zulu JDK installation.



Phase 3: Testing - Functionally, there are no differences between the Oracle JDK and Azul Zulu OpenJDK (other than those already noted for desktop machines). This means that there will be no differences executing your Java application using either JDK.

However, it is recommended to run your standard tests for applications being used to ensure that none of the changes between JDK updates has affected application behavior. This is a good practice even for exact like for like version releases.

Back to Phase 1 Planning: Your approach to transition planning will depend on various factors:

- Centralized or decentralized approach to transition
- Availability (or not) of a Software Asset Management product (SAM)
 - If you have a SAM, producing a report of installed JDK's is relatively straightforward
 - If no SAM is in place, please see Appendix how to determine which JDK is installed on different operating systems. If you have a decentralized approach, individual owners may already have this information so this process may not be required.
- Are you requiring only certain versions of JDK be used and only the latest security releases?

Azul Platform Core provides genuine drop-in replacement builds for the Oracle JDK, providing considerable cost savings to users.

Transitioning: It's So Easy

Azul Platform Core provides JDK distributions that are a direct replacement for the Oracle JDK. Once Java versions are identified, transition simply requires the installation of the latest versions of the Azul Zulu JDK and minor changes to application configuration to reflect the new location of the JDK.

Appendix

How to determine what JDK version is deployed on various Operating Systems:

Ubuntu/Debian Linux

Assuming that the JDK has been installed using the package manager, installed versions can be listed using the command:

```
$ dpkg -l | egrep [jJ][rR][eE]\|[jJ][dD][kK]
```

This will produce output similar to this example from a machine running Ubuntu

```
ii default-jre-headless
2:1.11-68ubuntu1~18.04.1 amd64 Standard
Java or Java compatible Runtime
(headless)
ii jdk1.8 1.8.0202-1 amd64 Java
Platform Standard Edition Development Kit
ii openjdk-11-jre-headless:amd64 11.0.7
+10-2ubun-tu2~18.04 amd64 OpenJDK Java
runtime, using Hotspot JIT (headless)
ii zulu-8 8.48.0.51-1 amd64 Azul
Systems Zulu JDK 8.48.0.51 (8u262-b19)
```

This shows four Java runtimes are installed:

1. Default Ubuntu 18.04 Java Runtime Environment (JRE)
2. Oracle JDK 8 update 202 (Although Oracle is not stated explicitly, it is implied by the use of the Java(TM) trademark).
3. OpenJDK 11 update 7 from the Ubuntu Linux distribution
4. Azul Zulu OpenJDK 8 update 262.

An alternative approach is to search the whole filesystem, which will also find any JDKs that have been installed manually (such as by unpacking a zip file). Use this command to search for files named java that are executable.

```
# find / -perm u+x -type f -name java
```

NOTE: To enable a full scan of the file system, this should be run as the root user.

The paths listed will often indicate the exact version of the JDK. If, however, for example, the directory is `/opt/jdk8`, the exact version installed will need to be determined by running the java command:

```
$ /opt/jdk8/bin/java -version
```

This will produce output similar to this:

```
java version "1.8.0_202"

Java(TM) SE Runtime Environment (build 1.8.0_202-b08)
Java HotSpot(TM) 64-Bit Server VM (build 25.202-b08, mixed mode)
```

Again, this is the Oracle JDK 8 update 202.

Oracle/Red Hat Enterprise Linux/SUSE Linux Enterprise Server

Use the command (as root)

```
# rpm -qa --queryformat “%{NAME} %{VERSION} %{VENDOR}
\n” | egrep [jJ][rR][eE]\|[jJ][dD][kK]
```

This will produce output something like:

```
jdk1.8 1.8.0_202 Oracle Corporation
```

Alternatively, the same full file system search method can be used as described for Ubuntu/Debian.



Windows

From a command prompt run the command wmic, which starts an interactive shell. Type the command:

```
product get name
```

In this output, you will see something like this

```
Java 8 Update 202 (64-bit)
Java SE Development Kit 8 Update 202 (64-bit)
```

MacOS X

To list JDKs installed using the packaging system, use the command:

```
$ pkgutil --pkgs | egrep jre\jdk
```

This will produce output like this

```
com.oracle.jre
com.oracle.jdk-14.0.2
com.oracle.jdk8u202
```

The com.oracle.jre package does not provide a version number. This can be obtained using the command:

```
$ pkgutil -pkg-info com.oracle.jre
```

Which will produce output like this, showing it is Java SE 10.0.2

```
package-id: com.oracle.jre
version: 10.0.2.0.13
volume: /
location: Library/Internet Plug-Ins/
JavaAppletPlugin.plugin install-time: 1538383780
```

MacOS is a UNIX-based operating system, so the full file system search method described for Ubuntu/Debian Linux can also be used.

Solaris

To list JDKs installed using the packaging system, use these two commands:

```
$ pkg list | egrep [jJ][rR][eE]\|[jJ][dD][kK]
$ pkginfo | egrep [jJ][rR][eE]\|[jJ][dD][kK]
```

This will produce output like this: runtime/java/jre-8

1.8.0.181.12

and

```
system SUNwj8cfg   JDK 8.0 Host Config(1.8.0_202)
system SUNwj8dev   JDK 8.0 Dev. Tools (1.8.0_202)
system SUNwj8jmp   JDK 8.0 Man Pages: Japan
                  (1.8.0_202)
system SUNwj8man   JDK 8.0 Man Pages (1.8.0_202)
system SUNwj8rt    JDK 8.0 64-bit Runtime Env.
                  (1.8.0_202)
```



Solaris is a UNIX-based operating system, so the full `_le` system search method described for Ubuntu/Debian Linux can also be used.

Desktop Machine Transition

Prior to the release of JDK 11, Oracle JDK contained a number of commercial features that are not part of the OpenJDK source code. Two of these specifically apply to the deployment of applications on desktop systems - please note if these apply:

1. The Browser Plugin: This is used to enable Applets to be used through a web browser. There is no open-source alternative to this, and the Azul Zulu JDK does not include equivalent functionality. Most browser providers no longer support plugins, and Oracle ended support for the Browser Plugin in March 2019 (even for those with a commercial support contract). In these situations, the option is to continue using your existing JDK and accept the potential security risks of not being able to address known vulnerabilities.

2. Java Web Start: This deployment technology provides for applications to automatically update themselves when the user runs them. Although the Java Network Launch Protocol (JNLP), which is part of Web Start, has a JSR, no reference implementation was provided for this. Azul can provide builds of IcedTea-Web, which is an open-source alternative to Java Web Start. This is not a drop-in replacement, so some additional transition effort is needed - Azul has worked with other customers on this topic. IcedTea-Web transition is beyond the scope of this document. Download at <https://www.azul.com/products/components/icedtea-web/>.

Azul Platform Core is about Risk Mitigation.

Satisfy Corporate Compliance, Governance Requirements and SLAs.

- Guaranteed timely access to secured and updated binaries and out-of-band emergency patches - because hackers don't wait
- IP and licensing anti-contamination guarantees and indemnification - because we live in a litigious world
- Real Support 24x7x365 - because Java isn't perfect, and regressions cause outages and customer failures
- Whatever your Java versions: 17, 15, 13, 11, 8, - all Azul Zulu Builds of OpenJDK are ALWAYS up to date because engineering teams need flexibility on when to move to newer releases
- Linux (all flavors), Windows, macOS, Solaris platforms supported; 32-bit and 64-bit, AWS, Azure and GCP, because organizations don't want to be tied to a single O/S or Cloud Platform
- Azul Products are used by Barclays, ABN Amro, Federal Reserve, Netflix, Lombard Odier, National Australia Bank, SAS, and hundreds more organizations

Compare Azul Platform Core to Oracle Java SE

Features	Azul Platform Core	Oracle Java SE
100% Open Source - no restrictions, no hassles	●	○
Production-quality support for Java 8, 11 and 17, older "LTS" versions Java 7 and 6, and newer versions Java 13 and 15	●	◐
Patent and Non-Contamination Indemnification	●	Not Applicable
Dedicated Global Java Support team, 24x7x365 live Support	●	●
Security-only Quarterly Updates	●	●
Commercial Support Production Lifecycle for Java 8 / Java 11	2030 / 2026	2030 / 2026
Engineering capacity to root-cause & fix bugs (independent of OpenJDK)	●	●
Proven to save millions in Java support costs	●	○

Azul Platform Core Security Key Benefits

Timely updates and security fix-dedicated releases

for Critical Vulnerability Exposures included with Azul Platform Prime: Only Azul and Oracle provide security-only "CPU" Releases.

Certified Compliance

Zulu builds are certified Java SE compliant and compatible, using 120,000+ Technology Compatibility Kit (TCK) tests, and tens of thousands of other Java quality tests developed by Azul over 20 years.

Dedicated to the Java Community for Security

Azul are active members of the OpenJDK Vulnerability Group. We were the first vendor to support TLS 1.3 standard for Java 8 and have back-ported several other key protocols.

What does Azul Platform Core Support include?

Azul Platform Core-supported OpenJDK binaries are fully supported by Azul, with contractually mandated SLAs. If you encounter a critical production issue with Azul Platform Prime, you will get the support you need, right away, for as long as it takes. This includes phone support. It also means that, if necessary, we will escalate an issue and work it 24x7 until resolution. By contrast, community builds of OpenJDK are not supported, and there is no time frame for security updates.



Azul Leaders with a passion for Java.

More performance. More value. More success.

Our leadership team has held key leadership positions in renowned companies across the world.



Scott Sellers
President, CEO & Co-Founder
Scott has more than 28 years of leadership experience, creating and growing technology companies.



Gil Tene
CTO & Co-Founder
Gill is an accredited technology specialist with over 40 technology patents and holds a coveted JavaOne “RockStar” status.

Together with the executive team they provide strategic, technical leadership and visionary direction.

“We have the people who carry the maintenance and development work, and employ over 100 full time engineers that work solely on JVMs and JDKs. We lead OpenJDK maintenance projects, and employ a multitude of active maintenance leads, reviewers, and contributors. We’ve been successful in supporting product Java deployments for well over a decade, and have the critical mass of capabilities, experience, expertise and resources.”

Gil Tene, CTO & Co-Founder, Azul

Java Champions. Leaders of the Java community.



@giltene



@GeertjanW

5

Gil Tene

Simon Ritter

Geertjan Wielenga

Gerrit Grunwald

Pratik Patel

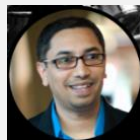
The Java Champion Program was founded in 2005 to recognize outstanding achievements and service of community members.



@speakjava



@hansolo_



@prpatel

Copyright© 2022 Azul Systems, Inc. 385 Moffett Park Drive Suite 115, Sunnyvale, CA 94089-1306. All rights reserved. Azul Systems, the Azul Systems logo, Zulu and Zing are registered trademarks, and ReadyNow! is a trademark of Azul Systems Inc. Java and OpenJDK are trademarks of Oracle Corporation and/or its affiliated companies in the United States and other countries. Monotype is a trademark of Monotype Imaging Inc. registered in the United States Patent and Trademark Office and may be registered in certain other jurisdictions. The Monotype logo is a trademark of Monotype Imaging Inc. and may be registered in certain jurisdictions. Other marks are the property of their respective owners and are used here only for identification purposes. Products and specifications discussed in this document may reflect future versions and are subject to change by Azul Systems without notice.

Contact Azul
385 Moffett Park Drive, Suite 115
Sunnyvale, CA 94089 USA
+1.650.230.6500
www.azul.com