



**Azul Zulu builds of OpenJDK  
April 20, 2021 Update Release**

**Azul Zulu 16.30 (CA) for Arm 64-bit**

Document Version: 1.0

Updated: Apr 28, 2021

# Table of Contents

<b>What's New</b> .....	<b>2</b>
April 20, 2021 PSU Release .....	2
IANA time zone data version .....	2
New Features and Enhancements .....	2
Fixed Issues .....	3
JDK Common Vulnerabilities and Exposure (CVE) Fixes .....	3
Non-CVE Security Fixes .....	6
OpenJDK Bug Fixes .....	6
<b>About This Build</b> .....	<b>11</b>
Supported Platforms .....	11
Supported Functionality .....	11
HotSpot Compilers .....	11
<b>Getting Started with Azul Zulu</b> .....	<b>13</b>
<b>Legal Notice</b> .....	<b>14</b>

# Revision History

Revision	Date	Description
0.1	April 20, 2021	Initial version of the document.
1.0	April 20, 2021	Added information about CVEs fixed in this release.

# What's New

## April 20, 2021 PSU Release

This section describes new features and changes in behavior introduced in April 20, 2021 Azul Zulu PSU Update Release.

<b>Azul Zulu Version:</b>	16.30 (16.0.1+9)
<b>Release Date:</b>	April 20, 2021
<b>Based on Azul Zulu Version:</b>	16.28 (16+36)

A CPU (Critical Patch Update) release incorporates critical bug fixes and security vulnerability fixes. Azul Zulu CPU releases are based on the prior PSU release and are available commercially.

A PSU (Patch Set Updates) release is based on the current CPU release, i.e. it includes all bug fixes that have been fixed in the CPU release, and a number of [non-security bug fixes](#).

## IANA time zone data version

This release of Azul Zulu comes with IANA time zone data version 2021a. For more details, see <https://data.iana.org/time-zones/tzdb-2021a/NEWS>.

## New Features and Enhancements

### TLS 1.0 and 1.1 is turned off

TLS 1.0 and 1.1 is turned off in the PSU builds in this release.

## Fixed Issues

### JDK Common Vulnerabilities and Exposure (CVE) Fixes

This section summarizes Common Vulnerabilities and Exposure (CVE) fixes of the April 2021 OpenJDK release.

CVE #	Component	Protocol	Remote Exploit without Auth.	Base Score	Attack Vector	Attack Complexity	Privileges Required	User Interaction	Scope	Confidentiality	Integrity	Availability	Supported Zulu Versions Affected	Modules Changed to Address CVE	Notes
<a href="#">CVE-2021-2161</a>	Libraries	Multiple	Yes	5.9	Network	High	None	None	Unchanged	None	High	None	6, 7, 8, 11, 13, 15, 16	6, 7, 8: JDK 11, 13, 15, 16: java.base	Note 1
<a href="#">CVE-2021-2163</a>	Libraries	Multiple	Yes	5.3	Network	High	None	Required	Unchanged	None	High	None	6, 7, 8, 11, 13, 15, 16	6, 7, 8: JDK 11, 13, 15, 16: java.base	Note 2
CVE-2021-23841	Oracle GraalVM Enterprise Edition: Node (OpenSSL)	HTTPS	Yes	7.5	Network	Low	None	None	Unchanged	None	None	High	N/A	N/A	

CVE #	Component	Protocol	Remote Exploit without Auth.	Base Score	Attack Vector	Attack Complexity	Privileges Required	User Interaction	Scope	Confidentiality	Integrity	Availability	Supported Zulu Versions Affected	Modules Changed to Address CVE	Notes
CVE-2021-3450	Oracle GraalVM Enterprise Edition: Node (Node.js)	HTTPS	Yes	7.4	Network	High	None	None	Unchanged	High	High	None	N/A	N/A	

Metrics	Values
Attack Vector	Network (N), Adjacent (A), Local (L), and Physical (P)
Attack Complexity	Low (L) and High (H)
Privileges Required	None (N), Low (L), and High (H)
User Interaction	None (N) and Required (R)
Scope	Unchanged (U) and Changed (C)
Confidentiality Impact	High (H), Low (L), and None (N)
Integrity Impact	High (H), Low (L), and None (N)
Availability Impact	High (H), Low (L), and None (N)

Notes:

ID	Notes
1	This vulnerability applies to Java deployments that load and run untrusted code (e.g., code that comes from the internet) and rely on the Java sandbox for security. It can also be exploited by supplying untrusted data to APIs in the specified Component.
2	This vulnerability applies to Java deployments that load and run untrusted code (e.g., code that comes from the internet) and rely on the Java sandbox for security.

## Non-CVE Security Fixes

OpenJDK Patch ID	Synopsis	CPU/PSU
<a href="#">JDK-8261183</a>	Follow on to Make lists of normal filenames	CPU
<a href="#">JDK-8259633</a>	compiler/graalunit/CoreTest.java fails with NPE after JDK-8244543	CPU
<a href="#">JDK-8259428</a>	AlgorithmId.getEncodedParams() should return copy	CPU
<a href="#">JDK-8258247</a>	Couple of issues in fix for JDK-8249906	CPU
<a href="#">JDK-8257001</a>	Improve HTTP client support	CPU
<a href="#">JDK-8253799</a>	Make lists of normal filenames	CPU
<a href="#">JDK-8244543</a>	Enhanced handling of abstract classes	CPU
<a href="#">JDK-8244473</a>	Contextualize registration for JNDI	CPU

## OpenJDK Bug Fixes

The following table describes the OpenJDK changes implemented in April 20, 2021 Zulu release.

OpenJDK Patch ID	Synopsis	CPU/PSU
<a href="#">JDK-8260356</a>	(tz) Upgrade time-zone data to tzdata2021a	CPU
<a href="#">JDK-8259048</a>	(tz) Upgrade time-zone data to tzdata2020f	CPU
<a href="#">JDK-8263968</a>	CDS: java/lang/ModuleLayer.EMPTY_LAYER should be singleton	PSU
<a href="#">JDK-8262018</a>	Wrong format in SAP copyright header of OsVersionTest	PSU
<a href="#">JDK-8261912</a>	Code IfNode::fold_compares_helper more defensively	PSU
<a href="#">JDK-8261860</a>	Crash caused by lambda proxy class loaded in Shutdown hook	PSU
<a href="#">JDK-8261779</a>	JCK test api/javax_crypto/EncryptedPrivateKeyInfo/Ctor4.html is failing with assertion error when assertions enabled	PSU
<a href="#">JDK-8261758</a>	[TESTBUG] gc/g1/TestGCLogMessages.java fails if ergonomics detect too small InitialHeapSize	PSU



OpenJDK Patch ID	Synopsis	CPU/PSU
<a href="#">JDK-8261753</a>	Test java/lang/System/OsVersionTest.java still failing on BigSur patch versions after JDK-8253702	PSU
<a href="#">JDK-8261752</a>	Multiple GC test are missing memory requirements	PSU
<a href="#">JDK-8261585</a>	Restore HandleArea used in Deoptimization::uncommon_trap	PSU
<a href="#">JDK-8261522</a>	[PPC64] AES intrinsics write beyond the destination array	PSU
<a href="#">JDK-8261483</a>	jdk/dynalink/TypeConverterFactoryMemoryLeakTest.java failed with "AssertionError: Should have GCd a method handle by now"	PSU
<a href="#">JDK-8261413</a>	Shenandoah: Disable class-unloading in I-U mode	PSU
<a href="#">JDK-8261397</a>	try catch Method failing to work when dividing an integer by 0	PSU
<a href="#">JDK-8261334</a>	NMT: tuning statistic shows incorrect hash distribution	PSU
<a href="#">JDK-8261310</a>	PPC64 Zero build fails with 'VMError::controlled_crash(int)::FunctionDescriptor functionDescriptor' has incomplete type and cannot be defined	PSU
<a href="#">JDK-8261261</a>	The version extra fields needs to be overridable in jib-profiles.js	PSU
<a href="#">JDK-8261251</a>	Shenandoah: Use object size for full GC humongous compaction	PSU
<a href="#">JDK-8261231</a>	Windows IME was disabled after DnD operation	PSU
<a href="#">JDK-8261209</a>	isStandalone property: remove dependency on pretty-print	PSU
<a href="#">JDK-8261170</a>	Upgrade to FreeType 2.10.4	PSU
<a href="#">JDK-8261022</a>	Fix incorrect result of Math.abs() with char type	PSU
<a href="#">JDK-8260934</a>	java/lang/StringBuilder/HugeCapacity.java fails without Compact Strings	PSU
<a href="#">JDK-8260933</a>	runtime/cds/serviceability/ReplaceCriticalClassesForSubgraphs.java fails without CompactStrings	PSU
<a href="#">JDK-8260864</a>	ProblemList two security/krb5 tests on Linux	PSU
<a href="#">JDK-8260592</a>	jpackage tests fail when Desktop is not supported	PSU
<a href="#">JDK-8260570</a>	Updating RE Configs for BUILD REQUEST 16.0.1+2	PSU

OpenJDK Patch ID	Synopsis	CPU/PSU
<a href="#">JDK-8260380</a>	Upgrade to LittleCMS 2.12	PSU
<a href="#">JDK-8260349</a>	Cannot programmatically retrieve Metaspace max set via JAVA_TOOL_OPTIONS	PSU
<a href="#">JDK-8260338</a>	Some fields in HaltNode is not cloned	PSU
<a href="#">JDK-8260048</a>	Shenandoah: ShenandoahMarkingContext asserts are unnecessary	PSU
<a href="#">JDK-8260029</a>	aarch64: fix typo in verify_oop_array	PSU
<a href="#">JDK-8260010</a>	UTF8ZipCoder not thread-safe since JDK-8243469	PSU
<a href="#">JDK-8260009</a>	InstanceKlass::has_as_permitted_subclass() fails if subclass was redefined	PSU
<a href="#">JDK-8259954</a>	gc/shenandoah/mxbeans tests fail with -Xcomp	PSU
<a href="#">JDK-8259949</a>	x86 32-bit build fails when -fcf-protection is passed in the compiler flags	PSU
<a href="#">JDK-8259849</a>	Shenandoah: Rename store-val to IU-barrier	PSU
<a href="#">JDK-8259777</a>	Incorrect predication condition generated by ADLC	PSU
<a href="#">JDK-8259773</a>	Incorrect encoding of AVX-512 kmovq instruction	PSU
<a href="#">JDK-8259706</a>	C2 compilation fails with assert(vtable_index == Method::invalid_vtable_index) failed: correct sentinel value	PSU
<a href="#">JDK-8259679</a>	GitHub actions should use MSVC 14.28	PSU
<a href="#">JDK-8259628</a>	jdk/net/ExtendedSocketOption/AsynchronousSocketChannelNAPITest.java fails intermittently	PSU
<a href="#">JDK-8259620</a>	Bump release strings for JDK 16.0.1	PSU
<a href="#">JDK-8259619</a>	C1: 3-arg StubAssembler::call_RT stack-use condition is incorrect	PSU
<a href="#">JDK-8259580</a>	Shenandoah: uninitialized label in VerifyThreadGCState	PSU
<a href="#">JDK-8259576</a>	Misplaced curly brace in Matcher::find_shared_post_visit	PSU
<a href="#">JDK-8259451</a>	Zero: skip serviceability/sa tests, set vm.hasSA to false	PSU

OpenJDK Patch ID	Synopsis	CPU/PSU
<a href="#">JDK-8259446</a>	runtime/jni/checked/TestCheckedReleaseArrayElements.java fails with stderr not empty	PSU
<a href="#">JDK-8259392</a>	Zero error reporting is broken after JDK-8255711	PSU
<a href="#">JDK-8259339</a>	AllocateUninitializedArray C2 intrinsic fails with void.class input	PSU
<a href="#">JDK-8259231</a>	Epsilon: improve performance under contention during virtual space expansion	PSU
<a href="#">JDK-8259049</a>	Uninitialized variable after JDK-8257513	PSU
<a href="#">JDK-8258946</a>	Fix optimization-unstable code involving signed integer overflow	PSU
<a href="#">JDK-8258909</a>	update jdk16u jcheck conf	PSU
<a href="#">JDK-8258534</a>	Epsilon: clean up unused includes	PSU
<a href="#">JDK-8258490</a>	Shenandoah: Full GC does not need to remark threads and drain SATB buffers	PSU
<a href="#">JDK-8258471</a>	"search codecache" clhsdb command does not work	PSU
<a href="#">JDK-8258457</a>	testlibrary_tests/ctw/JarDirTest.java fails with InvalidPathException on windows	PSU
<a href="#">JDK-8258243</a>	C2: assert failed ("Bad derived pointer") with -XX:+VerifyRegisterAllocator	PSU
<a href="#">JDK-8258077</a>	Using -Xcheck:jni can lead to a double-free after JDK-8193234	PSU
<a href="#">JDK-8257746</a>	Regression introduced with JDK-8250984 - memory might be null in some machines	PSU
<a href="#">JDK-8257513</a>	C2: assertconstant_addr - _masm.code()&const()&start( == con.offset())	PSU
<a href="#">JDK-8256421</a>	Add 2 HARICA roots to cacerts truststore	PSU
<a href="#">JDK-8256215</a>	Shenandoah: re-organize saving/restoring machine state in assembler code	PSU
<a href="#">JDK-8253910</a>	UseCompressedClassPointers depends on UseCompressedOops in vmError.cpp	PSU

OpenJDK Patch ID	Synopsis	CPU/PSU
<a href="#">JDK-8253702</a>	BigSur version number reported as 10.16, should be 11.nn	PSU
<a href="#">JDK-8253409</a>	Double-rounding possibility in float fma	PSU
<a href="#">JDK-8253368</a>	TLS connection always receives close_notify exception	PSU
<a href="#">JDK-8252971</a>	WindowsFileAttributes does not know about Unix domain sockets	PSU
<a href="#">JDK-8252883</a>	AccessDeniedException caused by delayed file deletion on Windows	PSU
<a href="#">JDK-8251944</a>	Add Shenandoah test config to compiler/gcbarriers/UnsafeIntrinsicsTest.java	PSU
<a href="#">JDK-8249867</a>	XML declaration is not followed by a newline	PSU
<a href="#">JDK-8241372</a>	Several test failures due to javax.net.ssl.SSLException: Connection reset	PSU
<a href="#">JDK-8226810</a>	Failed to launch JVM because of NullPointerException occurred on System.props	PSU
<a href="#">JDK-8198540</a>	Dynalink leaks memory when generating type converters	PSU
<a href="#">JDK-7146776</a>	Deadlock between URLStreamHandler.getHostAddress and file.Handler.openConnection	PSU

# About This Build

Azul Zulu for Arm 64-bit is a binary build of OpenJDK that Azul builds for the platforms based on the Arm 64-bit architecture. Azul Zulu binary builds are distributed as bundles. A bundle is a package that includes specific components of the binary build (e.g. headless JRE, Compact Profiles, specific CPU types, etc.). This section details the target platforms and the bundles included with this Azul Zulu build.

Azul Zulu 16.30 for Arm 64-bit provides the following bundles:

- Java Runtime Environment:

```
zulu16.30.15-ca-jre16.0.1-linux_aarch64.tar.gz
```

- Java Development Kit:

```
zulu16.30.15-ca-jdk16.0.1-linux_aarch64.tar.gz
```

## Supported Platforms

Azul Zulu 16.30 for Arm 64-bit is built for the platforms that meet the following requirements:

- Linux-based operating system with a kernel version of 3.10.x and higher.
- Arm v8 CPU with 64-bit support.
- Linux Arm 64-bit EABI.

## Supported Functionality

### HotSpot Compilers

In addition to the optimized template interpreter, Azul Zulu includes the following HotSpot just-in-time (JIT) compiler(s):

- Client Compiler (C1)
- Server Compiler (C2)

Use the following command-line options to change compilation behavior:

- `-Xint` – Runs the application in interpreted-only mode.
- `-Xcomp` – Enforces compilation of methods on first invocation.
- `-Xbatch` – Disables background compilation so that compilation of all methods proceeds as a foreground task until completed.
- `-XX:[+/-]TieredCompilation` – Enables or disables the tiered compilation (enabled by default). When the tiered compilation is disabled, only the server compiler is used.
- `-XX:TieredStopAtLevel=X` – Limits the compilation level (0 - interpreted, 1 - only the client compiler is used, 4 - full tiered compilation up to C2).

For more information on how to fine-tune compilation behavior, refer to the extended list of [Advanced JIT Compiler Options](#).

# Getting Started with Azul Zulu

To start using Azul Zulu, follow the steps given below.

1. Extract the installation archive to a dedicated directory. The name of the installation archive depends on the type of bundle:

- JDK bundle:

```
zulu16.30.15-ca-jdk16.0.1-linux_aarch64.tar.gz
```

- JRE bundle:

```
zulu16.30.15-ca-jre16.0.1-linux_aarch64.tar.gz
```

You can extract the archive by running the following command in the terminal:

```
$ tar -xzf zulu16.30.15-ca-jdk16.0.1-linux_aarch64.tar.gz
```

The command will create a new directory named after the archive but without the extension (`.tar.gz`). This directory contains all the files of your Azul Zulu bundle.

We will refer to this directory as `<ZULU_HOME>`.

2. Verify the Java version of your Azul Zulu installation.

Run `<ZULU_HOME>/bin/java -version` command and verify that the output is similar to the example below:

```
$ <ZULU_HOME>/bin/java -version
openjdk version "16.0.1" 2021-04-20
OpenJDK Runtime Environment Zulu16.30+15-CA (build 16.0.1+9)
OpenJDK 64-Bit Server VM Zulu16.30+15-CA (build 16.0.1+9, mixed
mode)
```

# Legal Notice

© 2005–2021, Azul Systems, Incorporated, 385 Moffett Park Drive, Suite 115, Sunnyvale, CA 94089. All rights reserved.

Products and specifications discussed in this document may reflect future versions and are subject to change without notice. Azul Systems assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this guide.

No part of this document may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Azul Systems. Please note that the content in this document is protected under copyright law even if it is not distributed with software that includes an end user license agreement.

Azul Systems, Azul Zulu, and the Azul logo are trademarks or registered trademarks of Azul Systems, Inc. Linux is a registered trademark of Linus Torvalds. Java is a registered trademark of Oracle Corporation. Microsoft and Windows are registered trademarks of Microsoft Corporation. Other marks are the property of their respective owners and are used here only for identification purposes.