
Amazon Corretto

Corretto 8 User Guide



Amazon Corretto: Corretto 8 User Guide

Copyright © 2019 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

What Is Amazon Corretto 8?	1
Related Information	1
Contributing to the SDK	1
List of Patches for Amazon Corretto 8	2
Linux	5
Installing on Amazon Linux 2	5
Option 1: Use the yum Package Manager on Amazon Linux	5
Option 2: Download and Install RPMs Manually	5
Verify Your Installation	6
Uninstall Amazon Corretto 8	6
Installing on Debian-based and RPM-based Linux	6
Installing on Debian-based Linux	7
Installing on RPM-based Linux	8
Windows	9
Installing on Windows 7 or Later	9
Install Amazon Corretto 8	9
Uninstall Amazon Corretto 8	9
macOS	11
Installing on macOS 10.10 or later	11
Install Amazon Corretto 8	11
Uninstall Amazon Corretto 8	11
Docker	12
Getting Started	12
Using the official image for Amazon Corretto 8.	12
Build a Docker Image with Amazon Corretto 8	12
Create an Image	13
Downloads	14
Amazon Corretto 8	14
Signature Verification	15
Change Log	16
Corretto version 8.212.04.2 for Amazon Linux 2	16
Corretto version 8.212.04.2	16
April 2019 critical patch update: Corretto version 8.212.04.1	17
8.202.08.2: Amazon Corretto 8 RC.	18
8u202 PSU releases: Corretto version 8.202.08.1 for Amazon Linux 2.	18
8u202 PSU releases: Corretto version 8.202.08.1	19
New platform releases: Version 1.8.0_192-amazon-corretto-preview2-b12 and 1.8.0_192-amazon-corretto-preview2_1-b12	19
Bug fix releases: Version 1.8.0_192-amazon-corretto-preview2-b12	20
Initial Release: Version 1.8.0_192-amazon-corretto-preview-b12	21
Document History	26

What Is Amazon Corretto 8?

Amazon Corretto is a no-cost, multiplatform, production-ready distribution of the Open Java Development Kit (OpenJDK). Corretto comes with long-term support that includes performance enhancements and security fixes. Corretto is certified as compatible with the Java SE standard and is used internally at Amazon for many production services. With Corretto, you can develop and run Java applications on operating systems such as Amazon Linux 2, Windows, and macOS.

This guide includes a list of patches applied to the OpenJDK for this release of Amazon Corretto 8, and installation instructions for the platforms supported by this version.

Related Information

In addition to this guide, see the following resources for developers:

- [Amazon Corretto 8 Overview](#)
- GitHub:
 - [Documentation source](#)
 - [JDK Source](#)

Contributing to the SDK

Developers can contribute feedback in the following ways:

- Submit issues on GitHub:
 - [Submit documentation issues](#)
 - [Report a bug or request a feature](#)
- Submit pull requests in the documentation or JDK source GitHub repositories to contribute to the SDK development

List of Patches for Amazon Corretto 8

This section lists all the patches applied to OpenJDK for Amazon Corretto 8. We also provide links to the issues in the OpenJDK project.

[C8-1] Prevent premature `OutOfMemoryException` when G1 GC invocation is suspended by a long-running native call.

Programs that use the G1 GC could experience spurious out-of-memory (OOM) exceptions even when the Java heap was far from filled up. This happened when a spin loop that waited for long-running native calls gave up after only two rounds. This small patch makes this loop wait as long as it takes. Typically a few more rounds suffice. Worst case, a full GC would eventually occur (thanks to [JDK-8137099](#)) and also resolve the issue. The patch includes a unit test that provokes needing more than two rounds, and succeeds only if the patch is in place. See [JDK-8137099](#) for discussion.

[C8-2] Back port from OpenJDK 10, fixing [JDK-8177809](#): “`File.lastModified()` is losing milliseconds (always ends in 000)”.

The patch removes inconsistencies in how the last-modified timestamp of a file is reported. It standardizes the behavior across build platforms and Java methods so that the user receives second-level precision. See [JDK-8177809](#).

[C8-3] Back port from OpenJDK9, fixing [JDK-8150013](#), “ParNew: Prune `nmethods` scavengable list”.

This patch reduces pause latencies for the Parallel and the CMS garbage collector. GC “root scanning” speeds up by up to three orders of magnitude by reducing redundant code inspections.

[C8-4] Back port from OpenJDK 9, fixing [JDK-8047338](#): “`javac` is not correctly filtering non-members methods to obtain the function descriptor”.

This patch fixes a compiler bug that caused compile-time errors when a functional interface threw an exception that extend `Exception`.

[C8-5] Back port from OpenJDK 10, fixing [JDK-8144185](#): “`javac` produces incorrect `RuntimeInvisibleTypeAnnotations` length attribute”.

This problem made Findbugs, JaCoCo, and Checker Framework fail on some well-formed input programs.

[C8-6] Trigger string table cleanup in G1 based on string table growth.

This patch triggers “mixed” G1 collections needed to clean out the string table entries based on string table growth, not just Java heap use. The latter is an independent measurement and may trigger too rarely or even never, in some applications. Then the string table may grow without bounds, which is effectively a native memory leak. See [JDK-8213198](#).

[C8-7] Backport from OpenJDK 9, fixing [JDK-8149442](#): “`MonitorInUseLists` should be on by default, deflate idle monitors taking too long”.

This patch makes removing a performance bottleneck for highly thread-intensive applications the default setting. Enabling `MonitorInUseLists` allows more efficient deflation of only potentially in-use monitors, instead of the entire population of monitors.

[C8-8] Back port from OpenJDK 11, fixing [JDK-8198794](#): “Hotspot crash on Cassandra 3.11.1 startup with `libnuma 2.0.3`”.

This patch prevents Cassandra 3.11.1 from crashing at startup.

[C8-9] Back port from OpenJDK 11, fixing [JDK-8195115](#): “G1 Old Gen MemoryPool CollectionUsage.used values don’t reflect mixed GC results”.

Without this patch, it’s impossible to determine how full the heap is by means of JMX when using the G1 GC.

[C8-10] Speed up Class.getSimpleName() and Class.getCanonicalName().

Memorization greatly speeds up these functions. This patch includes correctness unit tests. See [JDK-8187123](#).

[C8-11] Back port of [JDK-8068736](#) from OpenJDK9, fixing “Avoid synchronization on Executable/Field.declaredAnnotations”.

Improves the performance of Executable/Field.declaredAnnotations() by result caching that avoids thread synchronization.

[C8-12] Back port from OpenJDK 9, fixing [JDK-8077605](#): “Initializing static fields causes unbounded recursion in javac”.**[C8-13] Fixed [JDK-8130493](#): “javac silently ignores malformed classes in the annotation processor”.**

javac silently swallowed malformed class files in an annotation processor and returned with exit code 0. With this patch, javac reports an error message and returns with a non-zero exit code.

[C8-14] Improved error message for the jmap tool.

Updated error messages to suggest additional approaches when the target process is unresponsive. See [JDK-8213443](#).

[C8-15] Fixed [JDK-8185005](#): “Improve performance of ThreadMXBean.getThreadInfo(long ids[], int maxDepth)”.

This patch improves the performance of a JVM-internal function that looks up a Java Thread instance from an OS thread ID. This benefits several ThreadMXBean calls such as getThreadInfo(), getThreadCpuTime(), and getThreadUserTime(). The relative performance improvement increases with the number of threads in the JVM, as linear search is replaced by a hash table lookup.

[C8-16] Back port from OpenJDK 12, fixing [JDK-8206075](#): “On x86, assert on unbound assembler Labels used as branch targets”.

Label class instances (used to define pseudo-assembly code) can be used incorrectly in both the C1 and Interpreter. The most common mistake for a label is being “branched to” but never defined as a location in code via bind(). An assert was added to catch these and thus triggered 106 jtreg/hotspot and 17 jtreg/jdk test failures. We then determined that the label backedge_counter_overflow was not bound when UseLoopCounter was True, but UseOnStackReplacement was False. This is now fixed and guarded by the above tests.

[C8-17] Improve portability of JVM source code when using gcc7.

This patch places up-to-date type declarations in all places where the gcc switch “-Wno-deprecated-declarations” would flag problems. It also enables the switch to catch future related issues. This makes the source code compile on all present Amazon Linux versions. This is a combination of much of [JDK-8152856](#), [JDK-8184309](#), [JDK-8185826](#), [JDK-8185900](#), [JDK-8187676](#), [JDK-8196909](#), [JDK-8196985](#), [JDK-8199685](#), [JDK-8200052](#), [JDK-8200110](#), [JDK-8209786](#), [JDK-8210836](#), [JDK-8211146](#), [JDK-8211370](#), [JDK-8211929](#), [JDK-8213414](#), and [JDK-8213575](#).

[C8-18] Back port from JDK 10, fixing [JDK-8195848](#): “JTREG test for StartManagementAgent fails”.

See <http://serviceability-dev.openjdk.java.narkive.com/cDFwZce9> for more details.

[C8-19] Re-enables a legacy/disabled cipher suite to pass two TCK tests that would otherwise fail.**Three backports from OpenJDK9 to support using preinstalled libraries.**

Backported items: [JDK-8043805](#) for libjpeg, [JDK-8035341](#) for libpng, and [JDK-8042159](#) for lcms2.

Integration of aarch64 support from IcedTea 3.8.

Updates to vendor-related metadata.

Identifies Amazon as the vendor of this OpenJDK distribution and adds hyperlinks for reporting issues.

Back port from OpenJDK 9, fixing [JDK-8048782](#): “OpenJDK: PiscesCache : xmax/ymax rounding up can cause RasterFormatException”.

The bug is related to `sun.java2d.pisces.PiscesCache` constructor that accepts '(int minx,int miny,int maxx,int maxy)' arguments: the internal 'bboxX1' and 'bboxY1' are set to values one greater than given maximum X and Y values.

Amazon Corretto 8 Guide for Linux

The topics in this section describe installation instructions for Amazon Corretto 8 on Linux platforms.

Topics

- [Amazon Corretto 8 Installation Instructions for Amazon Linux 2 \(p. 5\)](#)
- [Amazon Corretto 8 Installation Instructions for Debian-Based and RPM-Based Linux Distributions \(p. 6\)](#)

Amazon Corretto 8 Installation Instructions for Amazon Linux 2

This topic describes how to install and uninstall Amazon Corretto 8 on a host or container running the Amazon Linux 2 operating system.

Option 1: Use the yum Package Manager on Amazon Linux

1. Enable the yum repository in Amazon Linux 2.

Example

```
sudo amazon-linux-extras enable corretto8
```

2. You can install Amazon Corretto 8 as either the runtime environment (JRE) or the full development environment (JDK). The development environment includes the runtime environment.

Install Amazon Corretto 8 as JRE.

Example

```
sudo yum install java-1.8.0-amazon-corretto
```

Install Amazon Corretto 8 as JDK.

Example

```
sudo yum install java-1.8.0-amazon-corretto-devel
```

The installation location is `/usr/lib/jvm/java-1.8.0-amazon-corretto.<cpu_arch>`.

Option 2: Download and Install RPMs Manually

1. Download RPMs from the [Downloads \(p. 14\)](#) page for your CPU architecture. To install the JDK, you will need to download the RPMs for both the JDK and the JRE.
2. Install using `yum localinstall`.

Example

```
sudo yum localinstall java-1.8.0-amazon-corretto*.rpm
```

Verify Your Installation

In the terminal, run the following command to verify the installation.

Example

```
java -version
```

Expected output for 8u212:

```
openjdk version "1.8.0_212"  
OpenJDK Runtime Environment Corretto-8.212.04.2 (build 1.8.0_212-b04)  
OpenJDK 64-Bit Server VM Corretto-8.212.04.2 (build 25.212-b04, mixed mode)
```

If you see a version string that doesn't mention Corretto, run the following command to change the default java or javac providers.

Example

```
sudo alternatives --config java
```

If using the JDK you should also run:

```
sudo alternatives --config javac
```

Uninstall Amazon Corretto 8

You can uninstall Amazon Corretto 8 with the following commands.

Uninstall JRE:

Example

```
sudo yum remove java-1.8.0-amazon-corretto
```

Uninstall JDK:

Example

```
sudo yum remove java-1.8.0-amazon-corretto-devel
```

Amazon Corretto 8 Installation Instructions for Debian-Based and RPM-Based Linux Distributions

This topic describes how to install Amazon Corretto 8 on Debian-based and RPM-based Linux distributions.

If you need to install Amazon Corretto 8 on Amazon Linux 2, see [Installing on Amazon Linux 2 \(p. 5\)](#).

Install Amazon Corretto 8 on Debian-Based Linux

This section describes how to install and uninstall Amazon Corretto 8 on a host or container running a Debian-based operating system.

Download and Install the Debian Package Manually

1. Download the Linux x64 .deb file from the [Downloads \(p. 14\)](#) page. Before you install the JDK, install the `java-common` package.

Example

```
sudo apt-get update && sudo apt-get install java-common
```

2. Install the .deb file by using `sudo dpkg --install`.

Example

```
sudo dpkg --install java-1.8.0-amazon-corretto-jdk_8.212.04-2_amd64.deb
```

Verify Your Installation

In the terminal, run the following command to verify the installation.

Example

```
java -version
```

Expected output for 8u212:

```
openjdk version "1.8.0_212"  
OpenJDK Runtime Environment Corretto-8.212.04.2 (build 1.8.0_212-b04)  
OpenJDK 64-Bit Server VM Corretto-8.212.04.2 (build 25.212-b04, mixed mode)
```

If you see a version string that doesn't mention `Corretto`, run the following command to change the default `java` or `javac` providers.

Example

```
sudo update-alternatives --config java
```

If you're using the JDK, you should also run the following.

```
sudo update-alternatives --config javac
```

Uninstall Amazon Corretto 8

You can uninstall Amazon Corretto 8 by using the following command.

Uninstall JDK:

Example

```
sudo dpkg --remove java-1.8.0-amazon-corretto-jdk
```

Install Amazon Corretto 8 on RPM-Based Linux

Download and install RPM package manually

1. Download the Linux x64 .rpm file from the [Downloads \(p. 14\)](#) page.
2. Install the downloaded .rpm file using `yum localinstall`.

Example

```
sudo yum localinstall java-1.8.0-amazon-corretto-devel-1.8.0_212.b04-2.x86_64.rpm
```

Verify Your Installation

In the terminal, run the following command to verify the installation.

Example

```
java -version
```

This is the expected output for 8u212:

```
openjdk version "1.8.0_212"  
OpenJDK Runtime Environment Corretto-8.212.04.2 (build 1.8.0_212-b04)  
OpenJDK 64-Bit Server VM Corretto-8.212.04.2 (build 25.212-b04, mixed mode)
```

If you see a version string that doesn't mention Corretto, run the following command to change the default `java` or `javac` providers.

Example

```
sudo alternatives --config java
```

If you're using the JDK, you should also run the following.

```
sudo alternatives --config javac
```

Uninstall Amazon Corretto 8

You can uninstall Amazon Corretto 8 by using the following

Uninstall JDK:

Example

```
sudo yum remove java-1.8.0-amazon-corretto-devel
```

Amazon Corretto 8 Guide on Windows

The topics in this section describe installation instructions for Amazon Corretto 8 on the Windows operating system. Windows builds are supported on Windows 7 and 10, and on Windows Server 2008, 2012, and 2016.

Topics

- [Amazon Corretto 8 Installation Instructions for Windows 7 or Later \(p. 9\)](#)

Amazon Corretto 8 Installation Instructions for Windows 7 or Later

This topic describes how to install and uninstall Amazon Corretto 8 on a host or container running the Windows 7 or later operating system.

Install Amazon Corretto 8

1. Download a Windows `.msi` file from the [Downloads \(p. 14\)](#) page.
2. Double-click the `.msi` file to start the installation wizard.
3. Follow the steps in the wizard.

You have the option of setting a custom installation path. By default, Amazon Corretto 8 is installed at `C:\Program Files\Amazon Corretto\`. If you set a custom path, make a note of it for the next step.

4. Once the install wizard is finished, set the `JAVA_HOME` and `PATH` environment variables.

Set `JAVA_HOME` to the installation location, noting that the directory contains the currently installed version. For example, if the default directory is used for 8u212, then set `JAVA_HOME` as `C:\Program Files\Amazon Corretto\jdk1.8.0_212`.

Add `%JAVA_HOME%\bin` to the current `PATH` variable.

5. Verify the installation by running `java -version` in a command prompt. You should see the following output.

Example

```
openjdk version "1.8.0_212"  
OpenJDK Runtime Environment Corretto-8.212.04.2 (build 1.8.0_212-b04)  
OpenJDK 64-Bit Server VM Corretto-8.212.04.2 (build 25.212-b04, mixed mode)
```

Uninstall Amazon Corretto 8

You can uninstall Amazon Corretto 8 by following the standard steps to uninstall an application from Windows.

1. Open **Programs and Features**.
2. Search for **Amazon Corretto 8** and then select it.
3. Choose **uninstall**.

Amazon Corretto 8 Guide for macOS

The topics in this section describe installation instructions for Amazon Corretto 8 on macOS.

Topics

- [Amazon Corretto 8 Installation Instructions for macOS 10.10 or later \(p. 11\)](#)

Amazon Corretto 8 Installation Instructions for macOS 10.10 or later

This topic describes how to install and uninstall Amazon Corretto 8 on a host running macOS version 10.10 or later. You must have administrator privileges to install and uninstall Amazon Corretto 8.

Install Amazon Corretto 8

1. Download the Mac .pkg file from the [Downloads \(p. 14\)](#) page.
2. Double click the downloaded file to start the installation wizard. Follow the steps in the wizard.
3. Once the wizard completes, Amazon Corretto 8 will be installed in `/Library/Java/JavaVirtualMachines/`.

You can run the following command in a terminal to get the complete installation path.

Example

```
/usr/libexec/java_home --verbose
```

4. Optionally, run the following commands in the terminal to set the `JAVA_HOME` variable.

Example

```
export JAVA_HOME=/Library/Java/JavaVirtualMachines/amazon-corretto-8.jdk/Contents/Home
```

Uninstall Amazon Corretto 8

You can uninstall Amazon Corretto 8 by running the following commands in a terminal.

Example

```
cd /Library/Java/JavaVirtualMachines/  
sudo rm -rf amazon-corretto-8.jdk
```

Amazon Corretto 8 Guide for Docker

This section describes simple use cases for using Amazon Corretto 8 in Docker containers.

Topics

- [Getting Started with Amazon Corretto 8 on Docker Images \(p. 12\)](#)

Getting Started with Amazon Corretto 8 on Docker Images

This topic describes how to build and launch a Docker image that uses Amazon Corretto 8. You must have the latest version of Docker installed.

Using the official image for Amazon Corretto 8.

Amazon Corretto 8 is available as an [official image on Docker Hub](#). The following example runs a container and displays Corretto's version.

Example

```
docker run amazoncorretto:8 java -version
```

Output:

Example

```
openjdk version "1.8.0_212"  
OpenJDK Runtime Environment Corretto-8.212.04.2 (build 1.8.0_212-b04)  
OpenJDK 64-Bit Server VM Corretto-8.212.04.2 (build 25.212-b04, mixed mode)
```

Build a Docker Image with Amazon Corretto 8

Run the following command to build an image that uses Amazon Corretto 8.

Example

```
docker build -t amazon-corretto-8 github.com/corretto/corretto-8-docker
```

After the command completes, you have an image called *amazon-corretto-8*.

To launch this image locally, run the following command.

Example

```
docker run -it amazon-corretto-8
```

You can also push this image to Amazon ECR. See the [Pushing an Image](#) topic in the *Amazon Elastic Container Registry User Guide* for details.

Create an Image

You can create a new Docker image using [Corretto's official Docker Hub image](#).

1. Create a Dockerfile with the following content.

Example

```
FROM amazoncorretto:8
RUN echo $' \
public class Hello { \
public static void main(String[] args) { \
System.out.println("Welcome to Amazon Corretto!"); \
} \
}' > Hello.java
RUN javac Hello.java
CMD ["java", "Hello"]
```

2. Build the new image.

Example

```
docker build -t hello-app .
```

3. Run the new image.

Example

```
docker run hello-app
```

You get the following output.

```
Welcome to Amazon Corretto!
```


Downloads for Amazon Corretto 8

This topic lists all the downloads available for Amazon Corretto 8. Use the link for the appropriate platform of your environment.

Amazon Corretto 8

Platform	Type	Download Link	Checksum (MD5)	Sig File
Linux x64 (p. 6)	JDK	java-1.8.0-amazon-corretto-jdk_8.212.04-2_amd64.deb	a04bc41d62ce8ed25bdb10d2a4fada88	
		java-1.8.0-amazon-corretto-devel-1.8.0_212.b04-2.x86_64.rpm	461739abc1fc08b89b5540d4fa05993b	
		amazon-corretto-8.212.04.2-linux-x64.tar.gz	782d5452cd7395340d910bd0f418a8	Download
Windows x64 (p. 9)	JDK	amazon-corretto-8.212.04.2-windows-x64.msi	e407008f9d0dba66727eebbd05c8f8c9	
		amazon-corretto-8.212.04.2-windows-x64-jdk.zip	b84eece357bbab8597ba3a415664fc3	Download
	JRE	amazon-corretto-8.212.04.2-windows-x64-jre.zip	deb7ec26424544ca079295ad1d31fe3d	Download
Windows x86 (p. 9)	JDK	amazon-corretto-8.212.04.2-windows-x86.msi	d815daace082388fd5d07579dde7039b	
		amazon-corretto-8.212.04.2-windows-x86-jdk.zip	0e69cdded96c99c65485ba7c5b569a4d6	Download
	JRE	amazon-corretto-8.212.04.2-windows-x86-jre.zip	e2e774344fc1ce0590450d6549a2bd08	Download
macOS x64 (p. 11)	JDK	amazon-corretto-8.212.04.2-macosx-x64.pkg	5df84d4c79503705da8d7f468e3b63f4	

Platform	Type	Download Link	Checksum (MD5)	Sig File
		amazon-corretto-8.212.04.2-macosx-x64.tar.gz	df2d5187c9cbfbbf56695883	56695883ae064c9d
Amazon Linux 2 x64 (p. 5)	JDK	java-1.8.0-amazon-corretto-devel-1.8.0_212.b04-2.amzn2.x86_64.rpm	74f156b10073bf19f754c3250294d0f2	
	JRE	java-1.8.0-amazon-corretto-1.8.0_212.b04-2.amzn2.x86_64.rpm	9dd471332016360a6ea8adccdba8bb44	
Amazon Linux 2 aarch64 (experimental) (p. 5)	JDK	java-1.8.0-amazon-corretto-devel-1.8.0_212.b04-2.amzn2.aarch64.rpm	240c93cc258d1b9c0f9091849d1336f9	
	JRE	java-1.8.0-amazon-corretto-1.8.0_212.b04-2.amzn2.aarch64.rpm	5b796b326737d87e749449addf941677	

Signature Verification

The public key to verify the SIGNATURE file can be downloaded from [here](#).

Change Log for Amazon Corretto 8

The following sections describe the changes for each release of Amazon Corretto 8.

Corretto version 8.212.04.2 for Amazon Linux 2

Release Date: May 02, 2019

The following platforms are updated in this release.

Target Platforms

- Amazon Linux 2, x64 and aarch64.

The following issues and enhancements are addressed in 8.212.04.2.

Issue Name	Platform	Description	Link
Update Corretto to 8.212.04.2.	All	Update Corretto 8 patch set to 8.212.04.2.	
Backport JDK-8048782: OpenJDK: PiscesCache : xmax/ymax rounding up can cause RasterFormatException	All	This patch fixes issue where sun.java2d.pisces.PiscesCache constructor that accepts min/max x and y arguments - the internal 'bboxX1' and 'bboxY1' are set to values one greater than given maximum X and Y values. This effectively causes an "off by 1" error.	corretto-8#94

Corretto version 8.212.04.2

Release Date: Apr 21, 2019

8.212.04.2 improves handling of TrueType fonts (JDK-8219066). The following platforms are updated:

Target Platforms

- RPM-based Linux using glibc 2.12 or later, x86_64
- Debian-based Linux using glibc 2.12 or later, x86_64
- Windows 7 or later, x86, x86_64
- macOS 10.10 and later, x86_64

April 2019 critical patch update: Corretto version 8.212.04.1

Release Date: Apr 16, 2019

The following platforms are updated in this release.

Target Platforms

- RPM-based Linux using glibc 2.12 or later, x86_64
- Debian-based Linux using glibc 2.12 or later, x86_64
- Windows 7 or later, x86, x86_64
- macOS 10.10 and later, x86_64

The following issues and enhancements are addressed in 8.212.04.1.

Issue Name	Platform	Description	Link
Update Corretto to 8.212.04.1.	All	Update Corretto 8 patch set to 8.212.04.1.	
Backport JDK-8048782: OpenJDK: PiscesCache : xmax/ymax rounding up can cause RasterFormatException	All	This patch fixes issue where sun.java2d.pisces.PiscesCache constructor that accepts min/max x and y arguments - the internal 'bboxX1' and 'bboxY1' are set to values one greater than given maximum X and Y values. This effectively causes an "off by 1" error.	corretto-8#94
Add jinfo file to Corretto Debian package.	Debian-based Linux	This patch fixes Corretto 8 does not provide a .jinfo file, which used by update-java-alternatives to switch all java related symlinks to another distribution.	corretto-8#63
Include /jre/lib/applet directory in rpm and deb packaging	RPM-based Linux , Debian-based Linux	/jre/lib/applet directory is missing in Corretto8 generic Linux deb and rpm, which makes it inconsistent with generic Linux tgz and other artifacts. This patch adds it back to deb and rpm.	

8.202.08.2: Amazon Corretto 8 RC.

Release Date: Jan 25, 2019

The following platforms are updated in this release.

Target Platforms

- RPM-based Linux using glibc 2.12 or later, x86_64
- Debian-based Linux using glibc 2.12 or later, x86_64
- Windows 7 or later, x86, x86_64
- macOS 10.10 and later, x86_64

The following issues and enhancements are addressed in 8.202.08.2.

Issue Name	Platform	Description	Link
Update java.vendor/ java.vm.vendor.	All	Vendor-related metadata has been updated to identify Amazon as the vendor of this OpenJDK distribution.	corretto-8#3
The Windows Installer should set file association for .jar files.	Windows	Windows users will now be able to run executable JARs using the file explorer.	corretto-8#43
Javapackager fails to load DLLs.	Windows	The JavaFX Packager on Windows has been fixed to allow bundling of MSVC DLLs.	corretto-8#47

8u202 PSU releases: Corretto version 8.202.08.1 for Amazon Linux 2.

Release Date: Jan 25, 2019

The following new platforms are supported.

New Platforms

- Experimental support for aarch64 on Amazon Linux 2.

The following platforms are updated in this release.

Target Platforms

- Amazon Linux 2

8u202 PSU releases: Corretto version 8.202.08.1

Release Date: Jan 23, 2019

The following platforms are updated in this release.

Target Platforms

- RPM-based Linux using glibc 2.12 or later, x86_64
- Debian-based Linux using glibc 2.12 or later, x86_64
- Windows 7 or later, x86, x86_64
- macOS 10.10 and later, x86_64

The following issues and enhancements are addressed in this release.

Issue Name	Platform	Description	Link
Remove DIZ files in Windows distribution.	Windows	Previous releases of Corretto on Windows contained debugging-related DIZ files. We received feedback that removing these files would benefit resource-constrained environments.	corretto-8#33
Improvements to JAVA_HOME-related variables on Windows.	Windows	Two fixes that will improve the ability for Windows applications to detect and use Amazon Corretto.	corretto-8#39 and corretto-8#40

New platform releases: Version 1.8.0_192-amazon-corretto-preview2-b12 and 1.8.0_192-amazon-corretto-preview2_1-b12

Release Date: Jan 16, 2019

The following new platforms are supported.

New Platforms

- RPM-based Linux using glibc 2.12 or later, x86_64
- Debian-based Linux using glibc 2.12 or later, x86_64

The following platforms are compatible with this release.

Target Platforms

- RPM-based Linux using glibc 2.12 or later, x86_64

- Debian-based Linux using glibc 2.12 or later, x86_64
- Windows 7 or later, x86, x86_64
- macOS 10.10 and later, x86_64

The following issues and enhancements are addressed in this release.

Issue Name	Platform	Description	Link
Support distribution via tar.gz/ZIP archives	Linux	To support other distribution systems (eg: Docker images, SDKMan), Corretto should also be offered in "plain" archives (tar.gz, ZIP).	corretto-8#2 and corretto-8#10
Debian Package distribution	Linux	Corretto should be offered in Debian package format in favor of customers using Debian-based Linux.	corretto-8#16
Support older versions of GLIBC	Linux	Current RPM for Amazon Linux 2 contains binaries that require GLIBC_2.26.	corretto-8#20
File javafx-src.zip missing: source code debugging for OpenJFX not enabled	Windows	The preview release is missing file javafx-src.zip with the compressed source code of OpenJFX.	corretto-8#19
Easy identification x86/x64 in Apps & Features of Windows	Windows	The Windows installer/uninstaller of Corretto should clearly display the architecture (x86/x64) information.	corretto-8#37
macOS Corretto installer without root access	macOS	Corretto Mac installer requires root access during the installation however some environments will require Corretto to be installed without necessarily having root access.	corretto-8#31

Bug fix releases: Version 1.8.0_192-amazon-corretto-preview2-b12

Release Date: Dec 17, 2018

The following platforms are compatible with this release.

Target Platforms

- Windows 7 or later, x86, x86_64
- macOS 10.10 and later, x86_64

The following are the bugs and enhancements addressed in this release.

Issue Name	Platform	Description	Link
libfreetype.dylib is incorrectly packaged	macOS	The libfontmanager in Corretto is linked to the libfreetype in X11 and breaks when X11 is not installed.	corretto-8#6
Eclipse and Eclipse-installer fail to run when using Amazon Corretto 8	macOS	The libjli.dylib under amazon-corretto-8.jdk/Contents/MacOS should be a symlink to ../Home/jre/lib/jli/libjli.dylib but was dereferenced. This causes the native JVM invoker in Eclipse failed to locate the JRE.	corretto-8#18
Enhance Installer to add standard registry keys on Windows for compat	Windows	Enhance the Windows installer to add registry keys for Corretto during the installation process.	corretto-8#14
Allow JAVA_HOME and PATH to be configured from installer	Windows	Enhance the Windows installer to add JAVA_HOME environment variable and also update the PATH environment variable with the Corretto installation location.	corretto-8#15
Support Windows 32bit binaries	Windows	Provide certified build for Windows 32bit OS.	corretto-8#22

Initial Release: Version 1.8.0_192-amazon-corretto-preview-b12

Release Date: Nov 14, 2018

The following platforms are compatible with this release.

Target Platforms

- Amazon Linux 2, x86_64
- Windows 7 or later, x86_64
- macOS 10.10 and later, x86_64

The following are the changes for this release.

Patch	Description	Release Date
[C8-1] Prevent premature OutOfMemoryException when G1 GC invocation is suspended by a long-running native call.	Programs that use the G1 GC could experience spurious out-of-memory (OOM) exceptions even when the Java heap was far from filling up. This happened when a spin loop that waited for long-running native calls gave up after only two rounds. This small patch makes this loop wait as long as it takes. Typically a few more rounds suffice. Worst case, a full GC would eventually occur (thanks to JDK-8137099) and also resolve the situation. The patch includes a unit test that provokes needing more than two rounds and succeeds only if the patch is in place. See JDK-8137099 for discussion.	2018-11-14
[C8-2] Back port from OpenJDK 10, fixing JDK-8177809: "File.lastModified() is losing milliseconds (always ends in 000)".	This patch removes inconsistencies in how the last-modified timestamp of a file is reported. It standardizes the behavior across build platforms and Java methods so that the user receives second-level precision.	2018-11-14
[C8-3] Back port from OpenJDK9, fixing JDK-8150013, "ParNew: Prune nmethods scavengable list".	This patch reduces pause latencies for the Parallel and the CMS garbage collector. GC "root scanning" speeds up by up to three orders of magnitude by reducing redundant code inspections.	2018-11-14
[C8-4] Back port from OpenJDK 9, fixing JDK-8047338: "javac is not correctly filtering non-members methods to obtain the function descriptor".	This patch fixes a compiler bug that caused compile-time errors when a functional interface threw an exception that extended Exception.	2018-11-14
[C8-5] Back port from OpenJDK 10, fixing JDK-8144185: "javac produces incorrect	This problem made Findbugs, JaCoCo, and Checker Framework	2018-11-14

Patch	Description	Release Date
RuntimeInvisibleTypeAnnotations length attribute".	fail on some well-formed input programs.	
[C8-6] Trigger string table cleanup in G1 based on string table growth.	This patch triggers "mixed" G1 collections needed to clean out the string table entries based on string table growth, not just Java heap usage. The latter is an independent measurement and can trigger too rarely or even never, in some applications. Then the string table can grow without bounds, which is effectively a native memory leak. See JDK-8213198.	2018-11-14
[C8-7] Backport from OpenJDK 9, fixing JDK-8149442: "MonitorInUseLists should be on by default, deflate idle monitors taking too long".	This patch makes removing a performance bottleneck for highly thread-intensive applications the default setting. Enabling MonitorInUseLists allows more efficient deflation of only potentially in-use monitors, instead of the entire population of monitors.	2018-11-14
[C8-8] Back port from OpenJDK 11, fixing JDK-8198794: "Hotspot crash on Cassandra 3.11.1 startup with libnuma 2.0.3".	This patch prevents Cassandra 3.11.1 from crashing at startup.	2018-11-14
[C8-9] Back port from OpenJDK 11, fixing JDK-8195115: "G1 Old Gen MemoryPool CollectionUsage.used values don't reflect mixed GC results".	Without this patch, it's impossible to determine how full the heap is by means of JMX when using the G1 GC.	2018-11-14
[C8-10] Speed up Class.getSimpleName() and Class.getCanonicalName().	Memorization greatly speeds up these functions. This patch includes correctness unit tests. See JDK-8187123.	2018-11-14
[C8-11] Back port of JDK-8068736 from OpenJDK9, fixing "Avoid synchronization on Executable/Field.declaredAnnotations".	Improves the performance of Executable/Field.declaredAnnotations() by result caching that avoids thread synchronization.	2018-11-14
[C8-12] Back port from OpenJDK 9, fixing JDK-8077605: "Initializing static fields causes unbounded recursion in javac".	N/A	2018-11-14

Patch	Description	Release Date
[C8-13] Fixed JDK-8130493: "javac silently ignores malformed classes in the annotation processor".	javac silently swallowed malformed class files in an annotation processor and returned with exit code 0. With this patch, javac reports an error message and returns with a non-zero exit code.	2018-11-14
[C8-14] Improved error message for the jmap tool.	Suggests additional approaches when the target process is unresponsive. See JDK-8213443.	2018-11-14
[C8-15] Fixed JDK-8185005: "Improve performance of ThreadMXBean.getThreadInfo(long ids[], int maxDepth)".	This patch improves the performance of a JVM-internal function that looks up a Java Thread instance from an OS thread ID. This benefits several ThreadMXBean calls such as getThreadInfo(), getThreadCpuTime(), and getThreadUserTime(). The relative performance improvement increases with the number of threads in the JVM, as linear search is replaced by a hash table lookup.	2018-11-14
[C8-16] Back port from OpenJDK 12, fixing JDK-8206075: "On x86, assert on unbound assembler Labels used as branch targets".	Label class instances (used to define pseudo-assembly code) can be used incorrectly in both the C1 and Interpreter. The most common mistake for a label is being "branched to" but never defined as a location in code via bind(). An assert was added to catch these and thus triggered 106 jtreg/hotspot and 17 jtreg/jdk test failures. We then determined that the label backedge_counter_overflow was not bound when UseLoopCounter was True but UseOnStackReplacement was False. This is now fixed and guarded by the above tests.	2018-11-14

Patch	Description	Release Date
[C8-17] Improve portability of JVM source code when using gcc7.	This patch places up-to-date type declarations in all places where the gcc switch “-Wno-deprecated-declarations” would flag problems. It also enables the switch to catch future related issues. This makes the source code compile on all present Amazon Linux versions. This is a combination of much of JDK-8152856, JDK-8184309, JDK-8185826, JDK-8185900, JDK-8187676, JDK-8196909, JDK-8196985, JDK-8199685, JDK-8200052, JDK-8200110, JDK-8209786, JDK-8210836, JDK-8211146, JDK-8211370, JDK-8211929, JDK-8213414, and JDK-8213575.	2018-11-14
[C8-18] Back port from JDK 10, fixing JDK-8195848: “JTREG test for StartManagementAgent fails”.	See http://serviceability-dev.openjdk.java.narkive.com/cDFwZce9 for more details.	2018-11-14
[C8-19] Re-enables a legacy/disabled cipher suite to pass two TCK tests that would otherwise fail.	N/A	2018-11-14

Document History for User Guide

The following table describes the documentation for this release of Amazon Corretto 8.

update-history-change	update-history-description	update-history-date
Corretto version 8.212.04.2. (p. 26)	8.212.04.2 released for Amazon Linux 2.	May 2, 2019
Corretto version 8.212.04.2. (p. 26)	Improves handling of TrueType fonts (JDK-8219066).	April 21, 2019
Quarterly Corretto update 8.212.04.1. (p. 26)	April 2019 update of security fixes for Corretto 8.	April 16, 2019
Amazon Corretto 8 is now in GA. (p. 26)	No changes have been made from the RC.	January 31, 2019
8u202: Add links for 8.202.08.2 RC and Amazon Linux 2 8.202.08.1. (p. 26)	Updates artifacts and instructions to point to current RC.	January 25, 2019
8u202 PSU releases: Corretto version 8.202.08.1 (p. 26)	Updates Amazon Corretto 8 to 8u202.	January 23, 2019
New Platform Release (1.8.0_192) (p. 26)	Bug fix release of Amazon Corretto 8 Developer Preview.	January 14, 2019
Bug fix Release (1.8.0_192) (p. 26)	Bug fix release of Amazon Corretto 8 Developer Preview.	December 17, 2018
Initial Release (1.8.0_192) (p. 26)	Initial release of Amazon Corretto 8 Developer Preview.	November 14, 2018