

Leonardo Lima, Chief Architect and Founder at TechStone

3 Tips to Make it Easy to Migrate from Oracle

Over 15 years at V2COM, a leader in IoT solutions based in Sao Paulo, Brazil, I built and managed numerous applications using both server-side and embedded Java, was nominated for a Java Innovation Award, joined the Java Executive Committee, became a Java champion, and became CTO. Earlier this year I founded a boutique tech consultancy named TechStone that now has a thriving Java practice.

It was at V2COM that I became a happy customer and passionate supporter of Azul when we switched from Oracle Java SE to Azul Platform Core and greatly reduced our Java TCO. In my experience in migrating both embedded and server-side Java, I learned that it's a simple, straightforward process, especially if you follow these 3 tips:

1. Develop a baseline of expectations

Know how your application performs pre-switch, and compare after you switch so you can guarantee that you have not slowed it down. Key metrics are RAM, CPU usage, and processor speed. This information may not be known to your organization for any number of reasons, but you need it. Establish your baseline performance metrics, then test.

Oracle Java and Azul Zulu Builds of OpenJDK are based on the same open-source OpenJDK code, and have identical functionality. So, 99% of the time, it will work perfectly. The 1% of the time that it doesn't is usually about minor details, and that's why you have Azul Support. Case in point: we ran a proof of concept at V2COM and Azul's performance was slightly worse. I reported this to Azul Support, who helped us figure out what the issue was and issued a new version of its JVM which fixed the issue. Azul Support is always responsive, fast, helpful and effective.

2. Test the waters first with minimal impact, before you switch over completely

Azul's JDK is free to test, and it's easy. Most of the time you can have both Oracle and Azul JVMs installed at the same time so the risk truly is minimal. You can download the free version, install it side by side with Oracle's, and compare apples to apples. It's a simple matter to change the Java binary, or the Docker image if you're using Docker, that goes either to Oracle or to the Azul Zulu Build of OpenJDK, and then you're off. You can switch between them, run each one, measure, and compare.

3. Don't migrate and upgrade all at once. First, the vendor; then, the version

Changing JVM versions can sometimes have a bigger impact than you expect because of contingencies and complexities. You may be running many versions of Java, for example. So, it's better to make the change in stages. First, change from Oracle 8 to Azul 8, for example, rather than jumping to Azul 11. Then, when you are ready and it makes sense from a business standpoint, upgrade between the Java versions. First, switch vendors; then, switch versions.

When you switch to Azul, you get world-class Java support at substantial savings. And it's easy, fast and risk-free if you follow these three tips based on experience.

Contact Azul

385 Moffett Park Drive, Suite 115

Sunnyvale, CA 94089 USA

☎ +1.650.230.6500

www.azul.com