

Azul pitches Docker as alternative to virtualization for heavy-duty Java applications

Analyst: John Abbott

24 Sep, 2014

Responding to 'significant interest' from its enterprise customer base, the Java runtime company Azul Systems has released a certified version of its Zulu OpenJDK Java development kit on Docker. It's also preparing a release of its full-featured Zing runtime for high-end customers, due to reach general availability shortly. The Zulu release, which is freely distributable and commercially supported, works with Java SE versions 8, 7 and 6. The combination of Java and Docker is interesting because many customers with enterprise Java applications have resisted virtualizing their applications due to the high resource overhead and performance tax imposed by hypervisors. Azul reports seeing many real-world deployments among its customers, with some even ripping out their existing virtualization layer and replacing it with Docker.

The 451 Take

There is clearly a container bandwagon rolling, with Docker in the lead. While containerization is not new, both the increasing maturity of cloud within enterprises and the overall maturity of IaaS have helped project it to center stage. There is even a chance to apply the principles of cloud computing to containers and create a new offshoot of IaaS. For enterprise customers with heavy-duty Java applications wanting the flexibility of virtualization without the performance penalty and unpredictability, Docker could be something of a savior.

Context

The open source Docker containerization software, community and ecosystem have grown rapidly

from a few million downloads in June 2014 to more than 23 million today, according to the company. Docker and containerization are serving as de facto application deployment standards amid a lack of real standards for rapidly deploying and supporting applications and services on multiple infrastructures, including clouds. Docker the company has recently raised \$40m in venture funding (taking its total funding up to \$65.8m) as it attempts to leverage the opportunity. The Docker platform, based on mature Linux container technology, has moved beyond PaaS and is being used to build, ship and run distributed applications on multiple platforms, speeding up deployment, isolating problems and reducing implementation errors. More than 13,000 applications already support Docker.

Docker and Java

Part of the reason Docker is finding acceptance in the enterprise is that it's not seen as too much of a risk because it's based on mature Linux containerization technology and OS-level virtualization techniques. But the acceptance and widespread deployment of hypervisor-based virtualization, with many of the same concepts, has also helped pave its way. The real difference between Docker and traditional virtualization is that the Docker Engine container includes just the application and its dependencies – not an entire guest operating system as a virtual machine would. It runs as an isolated process in user space on top of the host operating system, thus enjoying 'native' performance, immediate startup and greater portability while retaining the isolation benefits of VMs.

Docker looks like just another operating system to OpenJDK, and the Java community has now fully tested and certified proven builds of OpenJDK on Docker for Java SE versions 8, 7 and 6. That means developers and DevOps engineers can be assured of compatibility for their deployments. Java's long-standing promise of 'write once, run anywhere' applications doesn't extend to the necessary multiple configurations that are required – with Docker, the promise becomes 'write once, configure once and deploy anywhere.' Rapid provisioning at the application level – improving productivity – is a key addition, and for cloud deployments, Docker can provide the basis for a common platform for on- and off-premises deployments with full portability between them. Currently, of course, most public clouds are hypervisor-based, but Docker support is on its way.

Products

Azul claims Zulu is the first official OpenJDK implementation built and supported by a commercial company to support Docker. Zulu is multi-platform with versions for Windows, Linux and Mac OS X. It has a 10-year lifecycle so that timing of upgrades can be controlled – an advantage when end of life for Oracle's Java 7 has already been set for April 2015. Azul also provides rapid 'out of release

cycle' access to bug fixes. There are no license or distribution restrictions, but enterprises that require support can get it by buying Zulu Enterprise. It's available immediately.

The high-end Zing runtime, Azul's flagship product supporting low-latency and garbage collection, needs a little more work to run, but general availability is expected shortly. It's here at the upper tier of the Java market that perhaps the greatest opportunity for Azul lies. Virtualization often can't be used for high-performance applications, particularly in the financial services sector where Zing is most popular.

Competition

There has been a spate of recent announcements regarding Docker support for developers. From the IaaS, cloud and service provider angle, Amazon Web Services has integrated Docker support in Elastic Beanstalk, CenturyLink Technology Services just announced a Docker service called Panamax, and startup Platform9 landed a relatively substantial round of funding to be a Docker service provider. Some low-cost, new-wave IaaS providers like DigitalOcean already have Docker as a portfolio item. The Kubernetes project is being hosted by Google with partners CoreOS, IBM, Mesosphere, Microsoft, Red Hat and SaltStack. VMware announced in August that it would partner with Docker, Google and Pivotal and contribute to Kubernetes. PaaS platforms now supporting Docker include Shippable, ActiveState's Stackato, Amazon Beanstalk, Google App Engine, Pivotal, Red Hat OpenShift and WaveMaker. Docker development and management tools include Cloudsoft with Clocker and New Relic with Centurion.

Other commercially supported releases of OpenJDK are available from Oracle, IBM and Red Hat. But the real competitive question here is how much impact Docker will have on VMware, Hyper-V, Xen and KVM. Of course, much of the value of the commercial virtualization products comes from the surrounding ecosystem and management tools – but those are likely to build up rapidly for Docker as well, as indicated by such open source Docker management projects as Decking, Fleet and Shipyard. Although most people would argue that Docker is not yet fully production-ready and is limited to the Linux OS and bare-metal servers, there is a school of thought developing that Docker could revitalize the use of lightweight container-based virtualization as an alternative to VMs. It's potentially a much more efficient mechanism for delivering native apps with equal or better isolation and resource control, at least on Linux-based operating systems.

SWOT Analysis

Strengths

Weaknesses

Docker provides many of the advantages of hypervisor-based virtualization but without the performance overhead, and with additional rapid provisioning at the application level.

Opportunities

The combination of Docker and Zing promises to tackle the scalability and performance-consistency concerns that have so far prevented many Java deployments on virtualized infrastructure.

Many would argue that Docker is not yet fully production-ready, and it is currently limited to the Linux OS and bare-metal servers.

Threats

Docker is the flavor of the moment – but its limitations need to be considered carefully before full-blown deployments. It may not always be a sensible replacement for hypervisor virtualization with its extensive high-level management stack already in place.

Reproduced by permission of The 451 Group; © 2014. This report was originally published within 451 Research's Market Insight Service. For additional information on 451 Research or to apply for trial access, go to: www.451research.com