

# AWS EC2 M5 Instances Achieved 1.2x as Much Java Web Services Work as AWS EC2 M6g Instances



Java



**Handle 1.2x  
as much Java Web  
Services work with  
extra-large 64-vCPU  
M5 instances  
vs. M6g instances**



**Avoid the extra  
investment that can  
be necessary when  
changing from one CPU  
architecture to another**

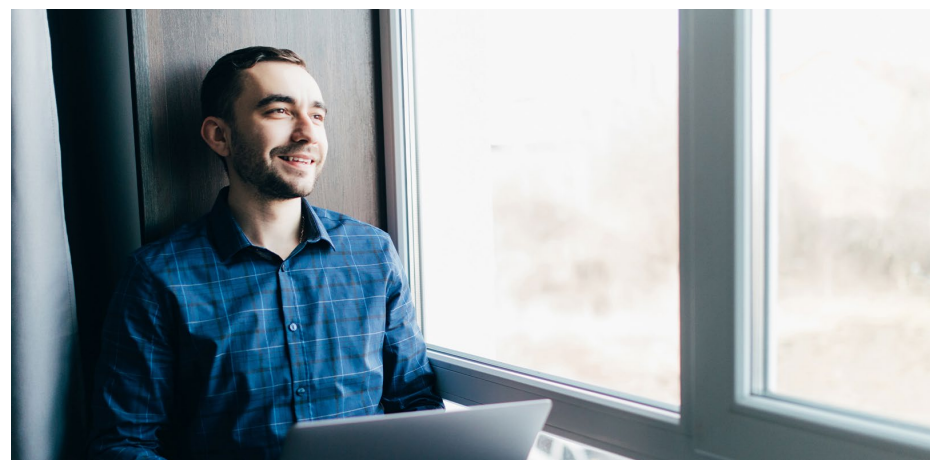
## Get Greater Performance by Selecting M5 Instances Featuring 2<sup>nd</sup> Gen Intel Xeon Scalable Processors

If your company runs Java web services workloads on AWS, you want to be sure to select the right instance type for the job. One consideration is performance, and another is continuity. By choosing AWS M5 instances enabled by 2<sup>nd</sup> Gen Intel Xeon Scalable processors rather than M6g instances powered by Amazon Graviton2 processors, you could reap advantages in both areas.

Testing with an Industry-standard Java server benchmark compared two extra-large AWS instances with 64 vCPUs: M5 instances enabled by 2<sup>nd</sup> Gen Intel Xeon Scalable processors and M6g instances powered by Amazon Graviton2 processors. The M5 instances did 1.2x the work of the M6g instances, which could translate to needing fewer instances to do a given amount of work.

Assuming you already run your Java web services workloads on VMs powered by 2<sup>nd</sup> Gen Intel Xeon Scalable processors, selecting AWS M5 instances can not only boost performance, but can provide a more seamless and predictable experience. Switching from the Intel architecture to the Graviton2 architecture could require you to re-optimize your workload, among other things.

For your Java web services needs, choose an M5 instance enabled by 2<sup>nd</sup> Gen Intel Xeon Scalable processors.



## Better Performance per Instance Can Reduce the Number of Instances You Need

If you've decided to host your Java web services workloads on AWS, you can get better performance by selecting M5 instances enabled by 2<sup>nd</sup> Gen Intel® Xeon® Scalable processors rather than M6g instances based on Amazon Graviton2 processors. As Figure 1 shows, an extra-large AWS 16-vCPU M5 VM instance enabled by Intel Xeon Scalable processors achieved 1.2x the Java work.

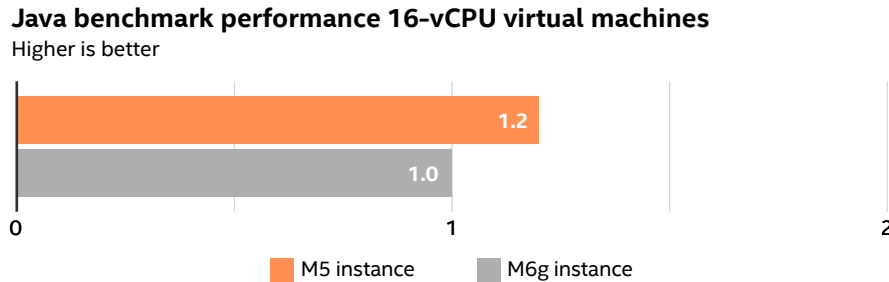


Figure 1. Relative test results comparing the Java web services performance of the extra-large (16-vCPU) M5 instance type to the extra-large M6g instance type.

## Sticking With a Known Processor Architecture Can Improve Continuity and Reduce Extra Work

If you've been running your Java web services workloads on Intel processors, you already have a set of baseline settings and best practices in place. You have likely optimized your workloads for that processor architecture. Changing to a new CPU architecture such as Amazon Graviton2 can require additional investments of time: your old assumptions and best practices might not apply. You'd have to perform testing, and might also have to tweak your workloads to get the best possible performance. By choosing M5 instances enabled by 2<sup>nd</sup> Gen Intel Xeon Scalable processors, you avoid all this extra work.

### Learn More

To begin running your Java services workloads on AWS M5 with 2<sup>nd</sup> Gen Intel Xeon Scalable processors, visit <http://intel.com/aws>.



Performance varies by use, configuration and other factors. Learn more at <https://intel.com/benchmarks>.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy. Your costs and results may vary.

Intel technologies may require enabled hardware, software or service activation.

© Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others

Printed in USA 0721/JO/PT/PDF US001

Please Recycle