

OutOfMemoryError - tier1app.com

NO.	OutOfMemoryError	Frequency	Cause	Solution	COMMENTS
1	Java heap space	5 Stars ★★★★★	<ol style="list-style-type: none"> Object could not be allocated in the Java heap Increase in Traffic volume Application is unintentionally holding references to objects which prevents the objects from being garbage collected Application makes excessive use of finalizers. Finalizer objects aren't GCed immediately. Finalizers are executed by a daemon thread that services the finalization queue. Sometimes finalizer thread cannot keep up, with the finalization queue. 	<ol style="list-style-type: none"> Increase Heap size '-Xmx'. Fix memory leak in the application 	GB -> G, g. MB -> M, m. KB -> K, k.
2	GC overhead limit exceeded	5 Stars ★★★★★	<ol style="list-style-type: none"> Java process is spending more than 98% of its time doing garbage collection and recovering less than 2% of the heap and has been doing so far the last 5 (compile time constant) consecutive garbage collections 	<ol style="list-style-type: none"> Increase heap size '-Xmx' GC Overhead limit exceeded can be turned off with '-XX-UseGCOverheadLimit' Fix the memory leak in the application 	
3	Requested array size exceeds VM limit	2 Stars ★★	<ol style="list-style-type: none"> Application attempted to allocate an array that is larger than the heap size 	<ol style="list-style-type: none"> Increase heap size '-Xmx' Fix bug in application. code attempting to create a huge array 	
4	Permgen space	3 stars ★★★	<ol style="list-style-type: none"> Permgen space contains: <ol style="list-style-type: none"> Names,Fields, methods of the classes Object arrays and type arrays associated with a class Just In Time compiler optimizations When this space runs out of space this error is thrown 	<ol style="list-style-type: none"> Increase Permgen size '-XX:MaxPermSize' Application redeployment without restarting can cause this issues. So restart JVM. 	
5	Metaspace	3 Stars ★★★	<ol style="list-style-type: none"> From Java 8 Permgen replaced by Metaspace. Class metadata is allocated in native memory (referred as metaspace). If metaspace is exhausted then this error is thrown 	<ol style="list-style-type: none"> If '-XX:MaxMetaSpaceSize', has been set on the command-line, increase its value. Remove '-XX:MaxMetsSpaceSize' Reducing the size of the Java heap will make more space available for MetaSpace. Allocate more memory to the server Could be bug in application. Fix it. 	
6	Unable to create new native thread	5 Stars ★★★★★	<ol style="list-style-type: none"> There isn't sufficient memory to create new threads. Threads are created in native memory. It indicates there isn't sufficient native memory space 	<ol style="list-style-type: none"> Allocate more memory to the machine Reduce Java Heap Space Fix thread leak in the application. Increase the limits at the OS level. ulimit -a max user processes (-u) 1800 Reduce thread stack size with -Xss parameter 	

NO.	OutOfMemoryError	Frequency	Cause	Solution	COMMENTS
7	Kill process or sacrifice child	1 Stars 	1. Kernel Job – Out of Memory Killer. Will kill processes under extremely low memory conditions	1. Migrate process to different machine. 2. Add more memory to machine	Unlike all other OOM errors, this is not triggered by JVM. But by OS.
8	reason stack_trace_with_native_ method	1 Stars 	1. Native method encountered allocation failure 2. a stack trace is printed in which the top frame is a native method	1. Use OS native utilities to diagnose	