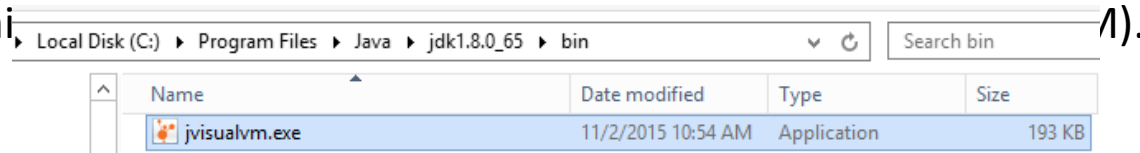


Requirements – summary

- [JVisualVM](#) – included with Java SDK

Java VisualVM is an intuitive graphical user interface that provides detailed information about Java .. applications ... whi

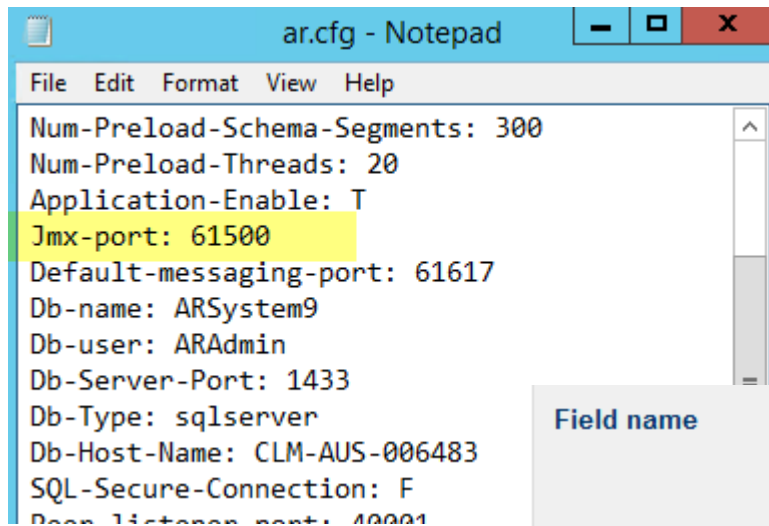


- AR Server – configured to allow monitoring connections

Some items are hardcoded in AR code

Requirements – AR configuration

- Configuration required (ar.cfg or Central Configuration)



```

ar.cfg - Notepad
File Edit Format View Help
Num-Preload-Schema-Segments: 300
Num-Preload-Threads: 20
Application-Enable: T
Jmx-port: 61500
Default-messaging-port: 61617
Db-name: ARSystem9
Db-user: ARAdmin
Db-Server-Port: 1433
Db-Type: sqlserver
Db-Host-Name: CLM-AUS-006483
SQL-Secure-Connection: F

```

Field name	AR System Configuration Generic UI form option	Description
JMX Port	Jmx-port	Defines the JMX port number that enables administrators to connect to JVM by using Java Messaging Extensions (JMX). The default port number is 61500.

Requirements – AR configuration

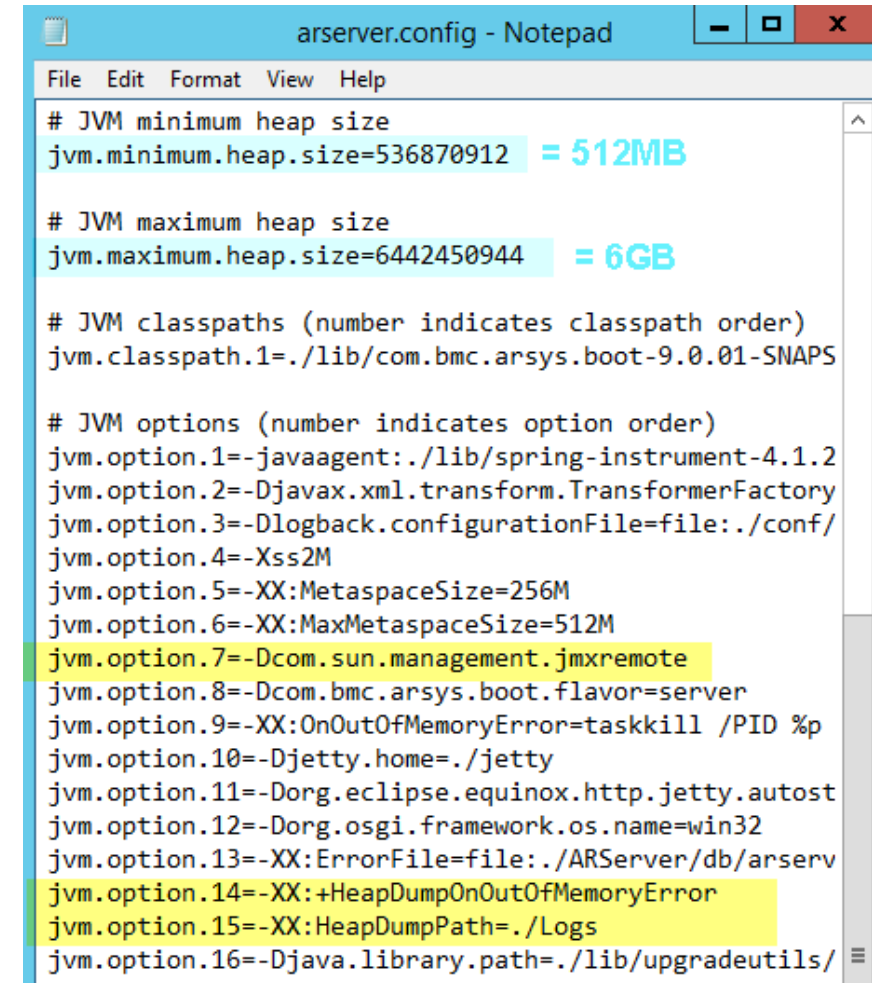
- Configuration required - arserver.config

7 – Allows for remote connection via “Java Management Extensions (JMX)”, like JVisualVM

14 and 15 - Tells the Java VM to create a heap dump of memory to the Path indicated, If Java exhausts its memory allocation

Note: minimum and maximum heap values are also set here

“take 512MB at start-up – allocate up to 6GB”



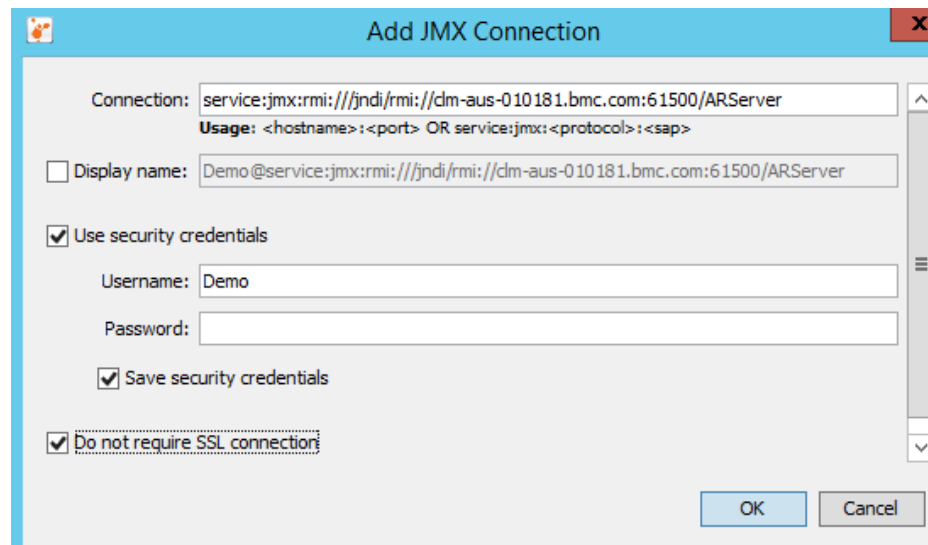
```
arserver.config - Notepad
File Edit Format View Help
# JVM minimum heap size
jvm.minimum.heap.size=536870912 = 512MB
# JVM maximum heap size
jvm.maximum.heap.size=6442450944 = 6GB
# JVM classpaths (number indicates classpath order)
jvm.classpath.1=./lib/com.bmc.arsys.boot-9.0.01-SNAPS
# JVM options (number indicates option order)
jvm.option.1=-javaagent:./lib/spring-instrument-4.1.2
jvm.option.2=-Djavax.xml.transform.TransformerFactory
jvm.option.3=-Dlogback.configurationFile=file:./conf/
jvm.option.4=-Xss2M
jvm.option.5=-XX:MetaspaceSize=256M
jvm.option.6=-XX:MaxMetaspaceSize=512M
jvm.option.7=-Dcom.sun.management.jmxremote
jvm.option.8=-Dcom.bmc.arsys.boot.flavor=server
jvm.option.9=-XX:OnOutOfMemoryError=taskkill /PID %p
jvm.option.10=-Djetty.home=./jetty
jvm.option.11=-Dorg.eclipse.equinox.http.jetty.autost
jvm.option.12=-Dorg.osgi.framework.os.name=win32
jvm.option.13=-XX:ErrorFile=file:./ARServer/db/arserv
jvm.option.14=-XX:+HeapDumpOnOutOfMemoryError
jvm.option.15=-XX:HeapDumpPath=./Logs
jvm.option.16=-Djava.library.path=./lib/upgradeutils/
```

Actions – connecting to AR from JVisualVM

- run JVisualVM from a system other than the AR server in question (best practice)

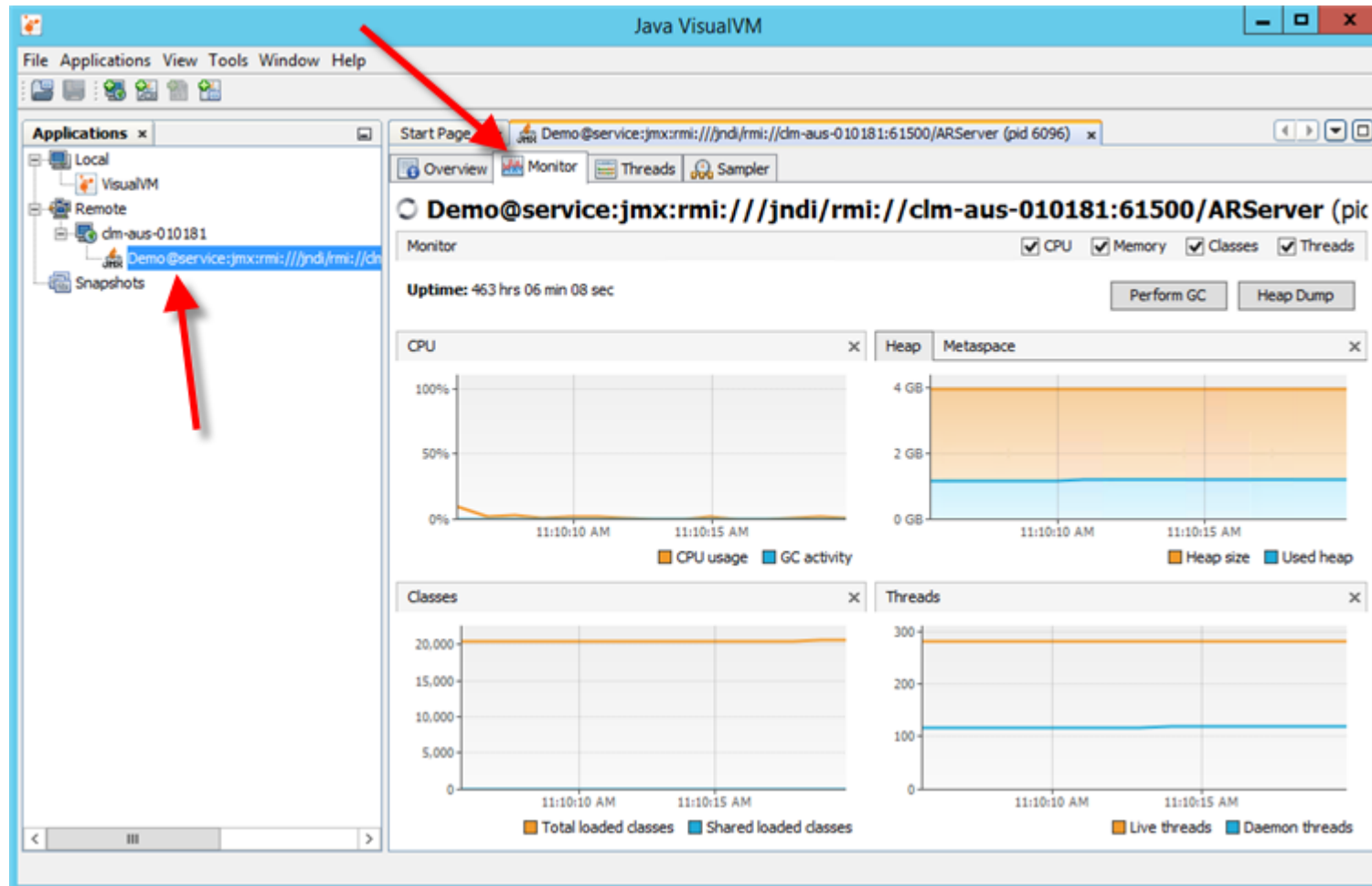
AR Server should have (example) 2+Gig of free space on C:\ for each heap dump generated

- connect to the AR Server – over the JMX port, as an AR Administrator
`service:jmx:rmi:///jndi/rmi://<arserver-name>:61500/ARServer`



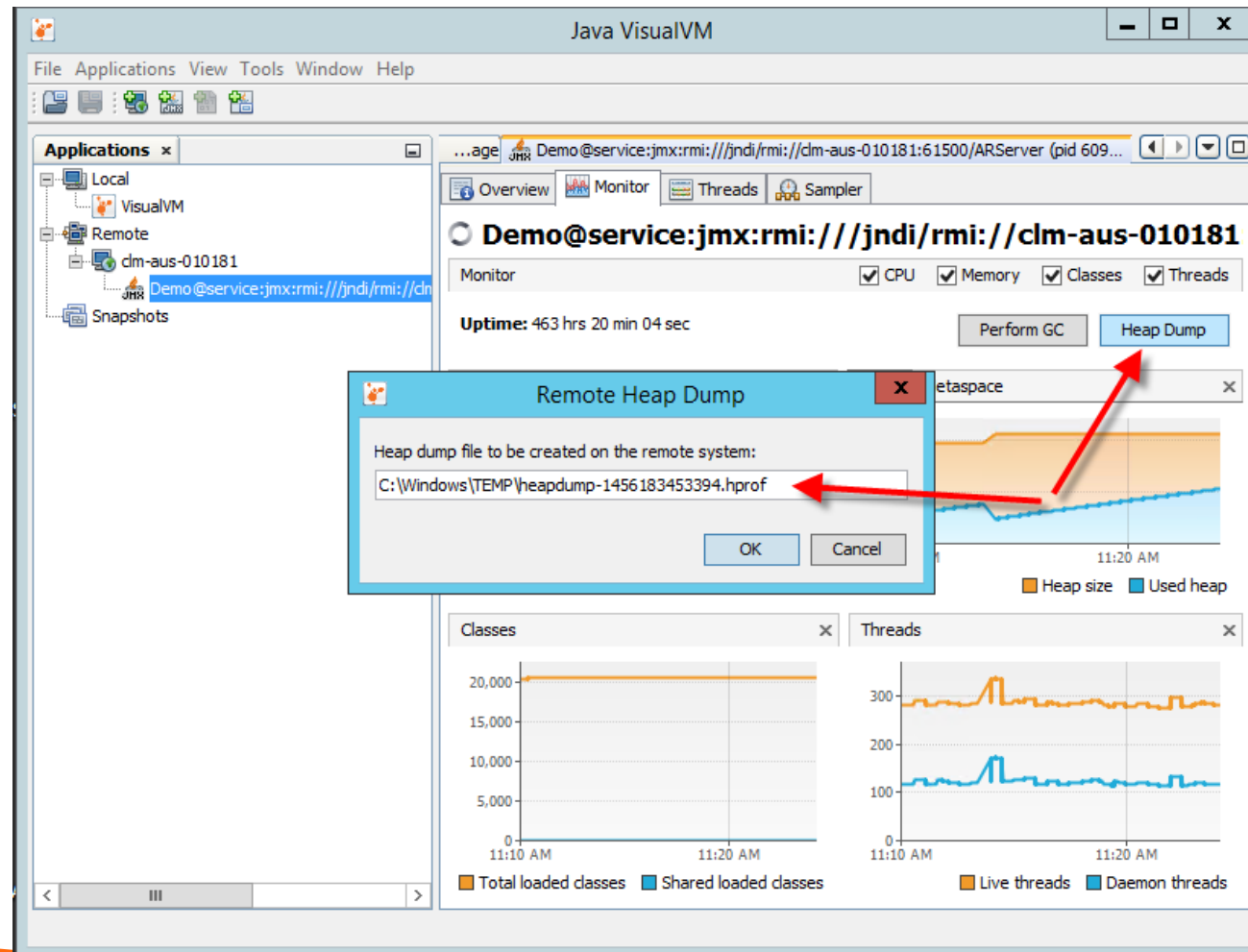
Actions – connecting to AR from JVisualVM

- double-click on the new connection, and select the Monitor tab:



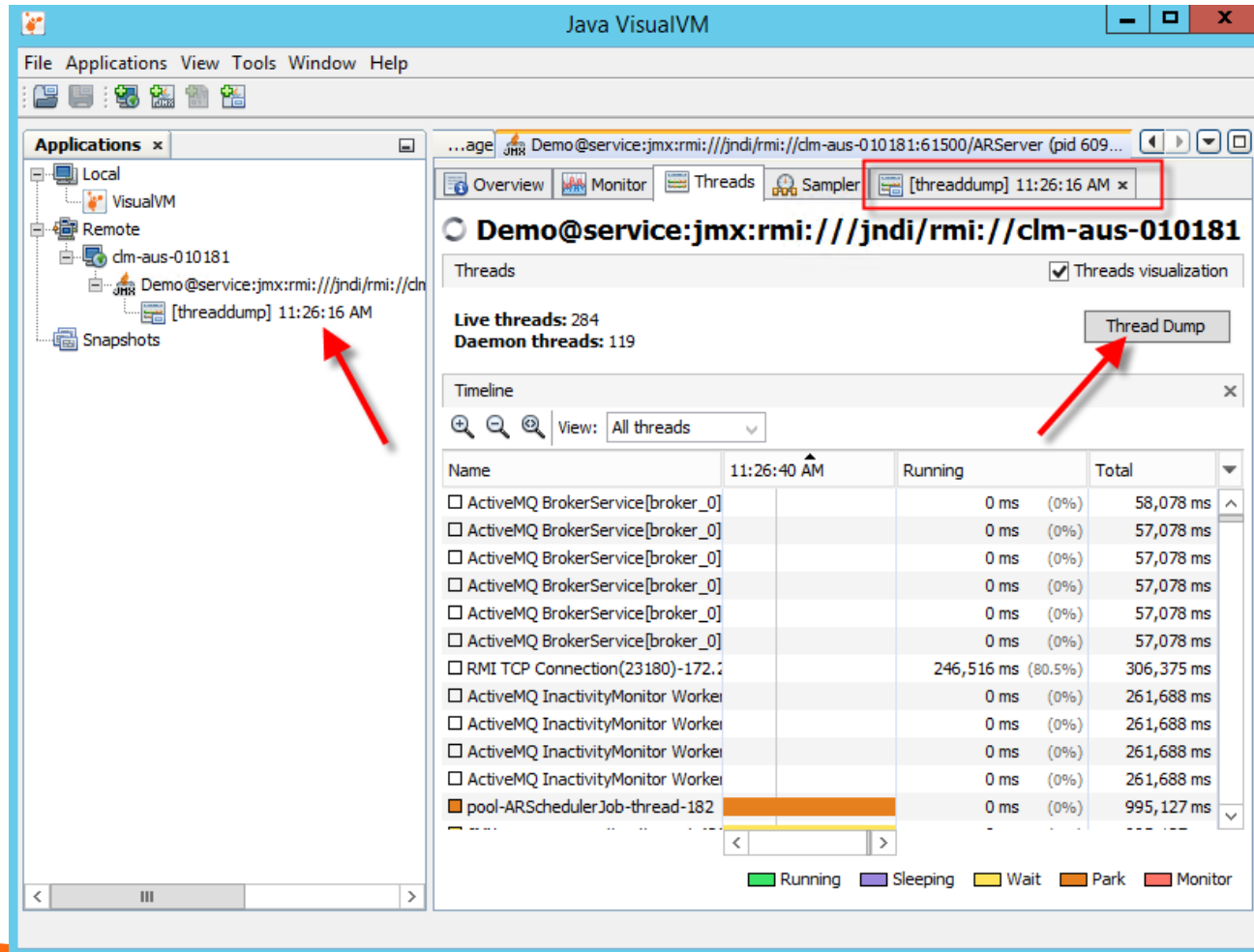
Actions – creating a Heap Dump file

- press the **Heap Dump** button – this action generates a large .hprof file on the AR Server



Actions – creating a Thread Dump file

- press the **Thread Dump** button – generates a dump of thread activity at that time:



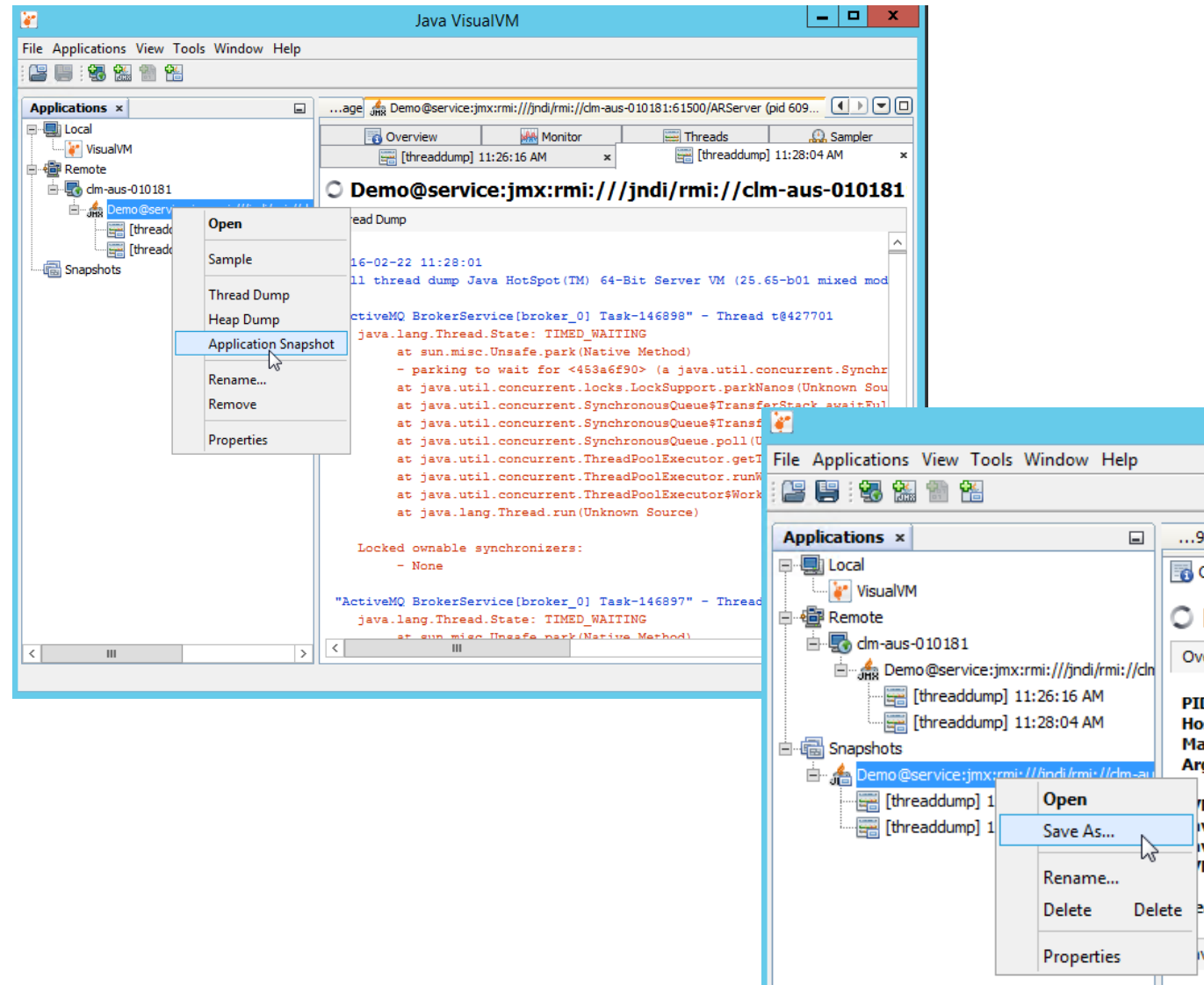
The screenshot shows the Java VisualVM interface. The 'Applications' tree on the left shows a remote application 'Demo@service:jmx:rmi:///jndi/rmi://clm-aus-010181' with a sub-entry '[threaddump] 11:26:16 AM' highlighted by a red arrow. The main window displays the 'Threads' panel for this application, showing 'Live threads: 284' and 'Daemon threads: 119'. A 'Thread Dump' button is visible in the top right of the 'Threads' panel, also highlighted by a red arrow. Below the 'Threads' panel is a 'Timeline' view showing a table of thread activity.

Name	11:26:40 AM	Running	Total
ActiveMQ BrokerService[broker_0]		0 ms (0%)	58,078 ms
ActiveMQ BrokerService[broker_0]		0 ms (0%)	57,078 ms
ActiveMQ BrokerService[broker_0]		0 ms (0%)	57,078 ms
ActiveMQ BrokerService[broker_0]		0 ms (0%)	57,078 ms
ActiveMQ BrokerService[broker_0]		0 ms (0%)	57,078 ms
ActiveMQ BrokerService[broker_0]		0 ms (0%)	57,078 ms
RMI TCP Connection(23180)-172.2		246,516 ms (80.5%)	306,375 ms
ActiveMQ InactivityMonitor Worker		0 ms (0%)	261,688 ms
ActiveMQ InactivityMonitor Worker		0 ms (0%)	261,688 ms
ActiveMQ InactivityMonitor Worker		0 ms (0%)	261,688 ms
ActiveMQ InactivityMonitor Worker		0 ms (0%)	261,688 ms
ActiveMQ InactivityMonitor Worker		0 ms (0%)	261,688 ms
pool-ARSchedulerJob-thread-182		0 ms (0%)	995,127 ms

Legend: Running (green), Sleeping (purple), Wait (yellow), Park (orange), Monitor (red)

Actions – creating an Application Snapshot file

- After several thread dumps, can also take an **Application Snapshot**, which saves the “point in time” environment details and thread dumps to a single file (which can also be shipped to Support)



NOTE: the hprofs are not included in this file