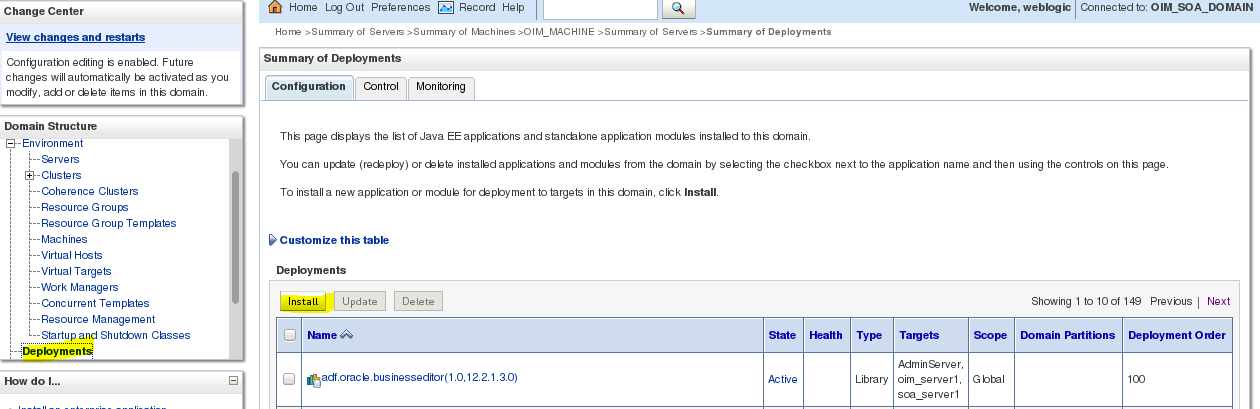
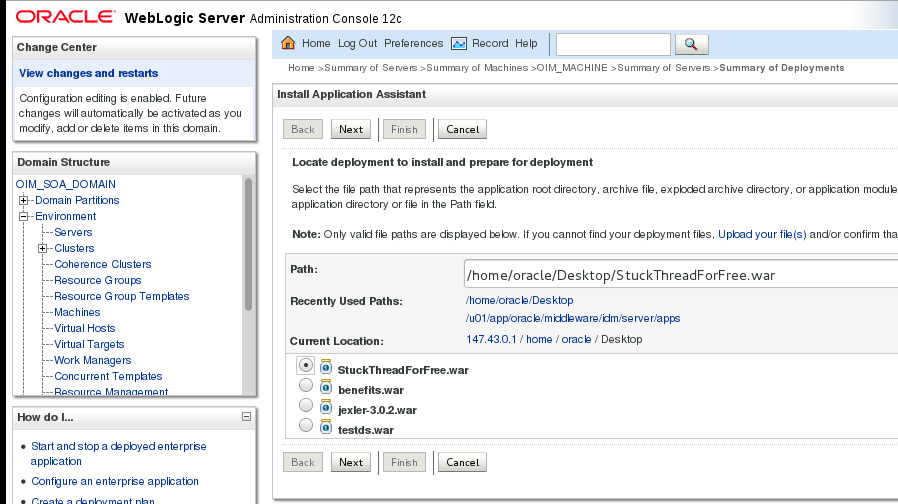
STUCK THREAD

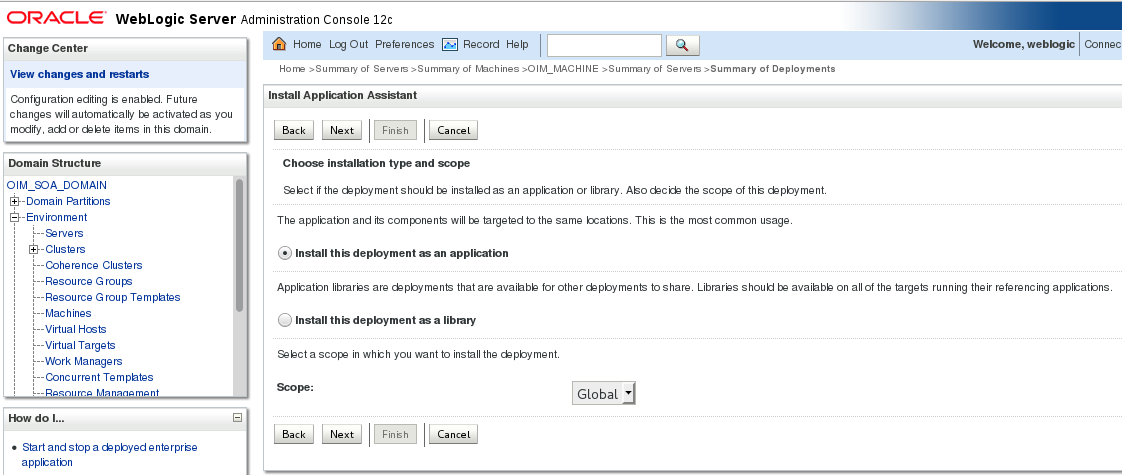
Deploy stuck thread app



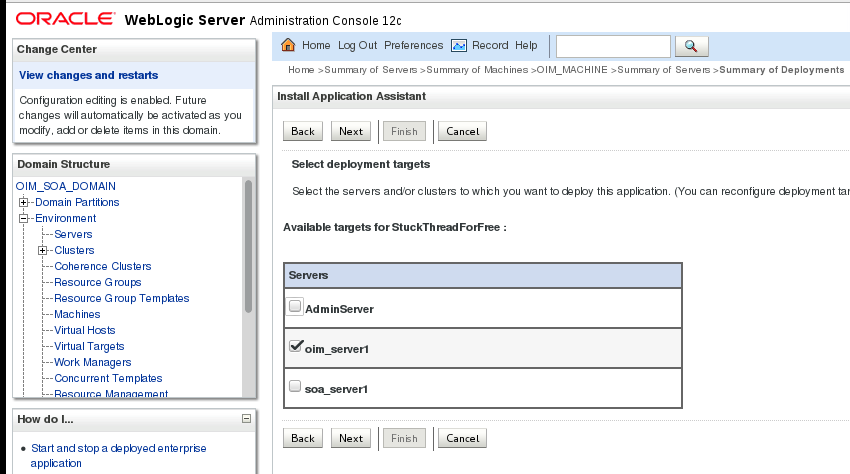
Install the stuck thread app



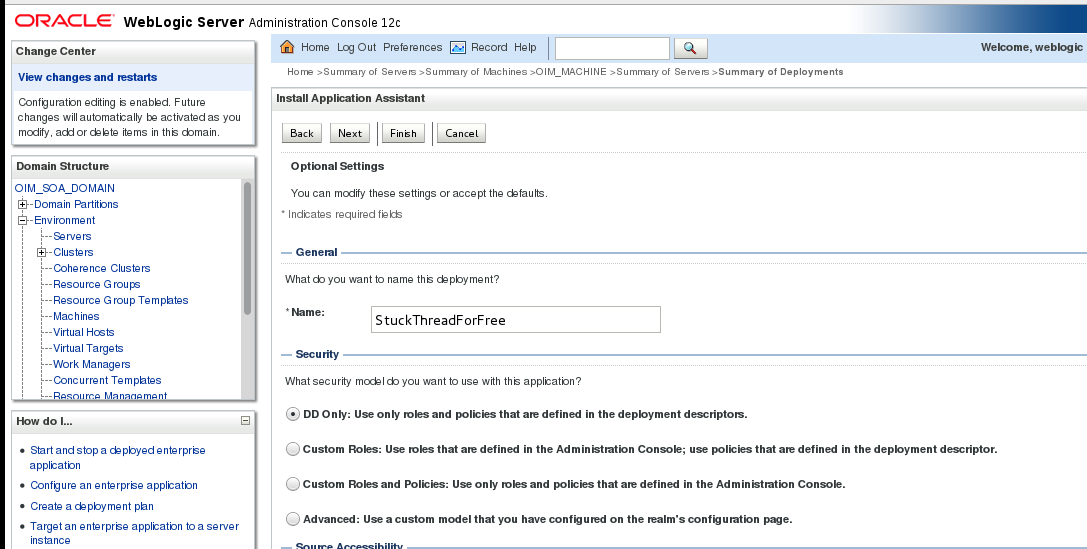
Install this deployment as an application



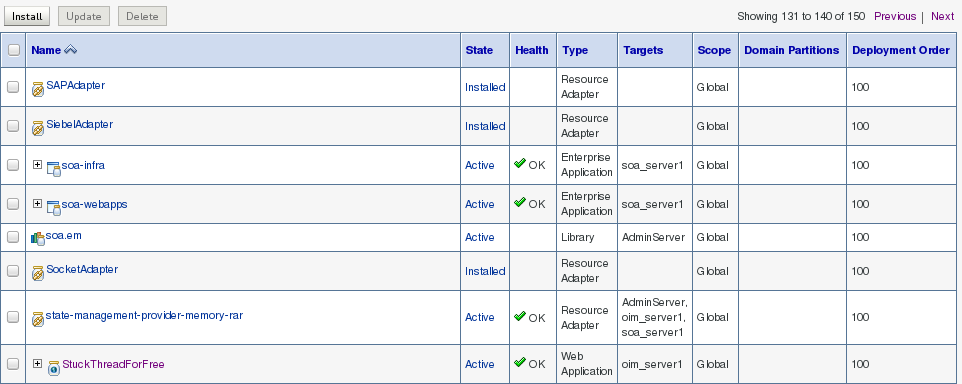
Choose the server



Deployment assistant code



Application is in active state



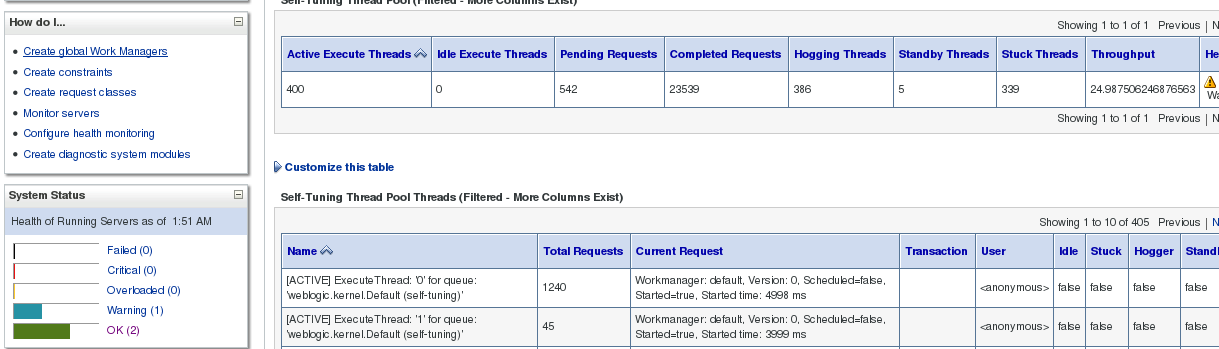
Launch the app



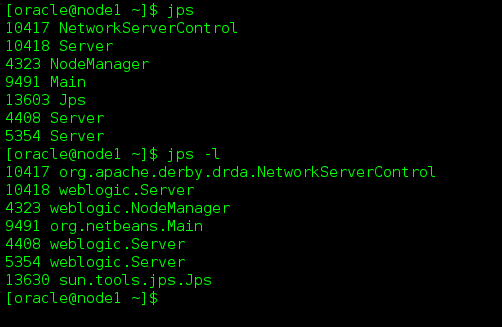
App runs



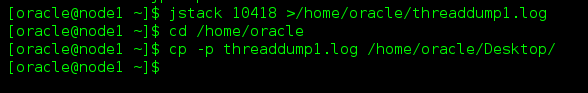
Now Server state changed into Warning state



Get the Process ID and take thread dump



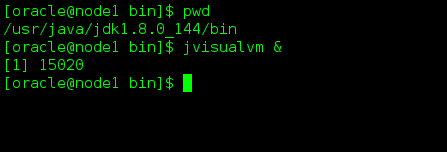
Take thread dump using Jstack



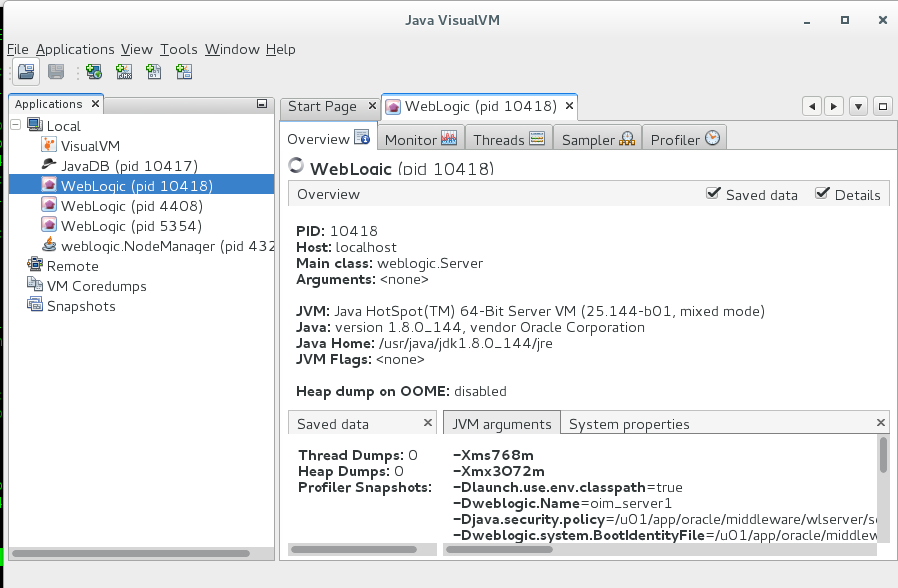
Take thread dump using Jcmd



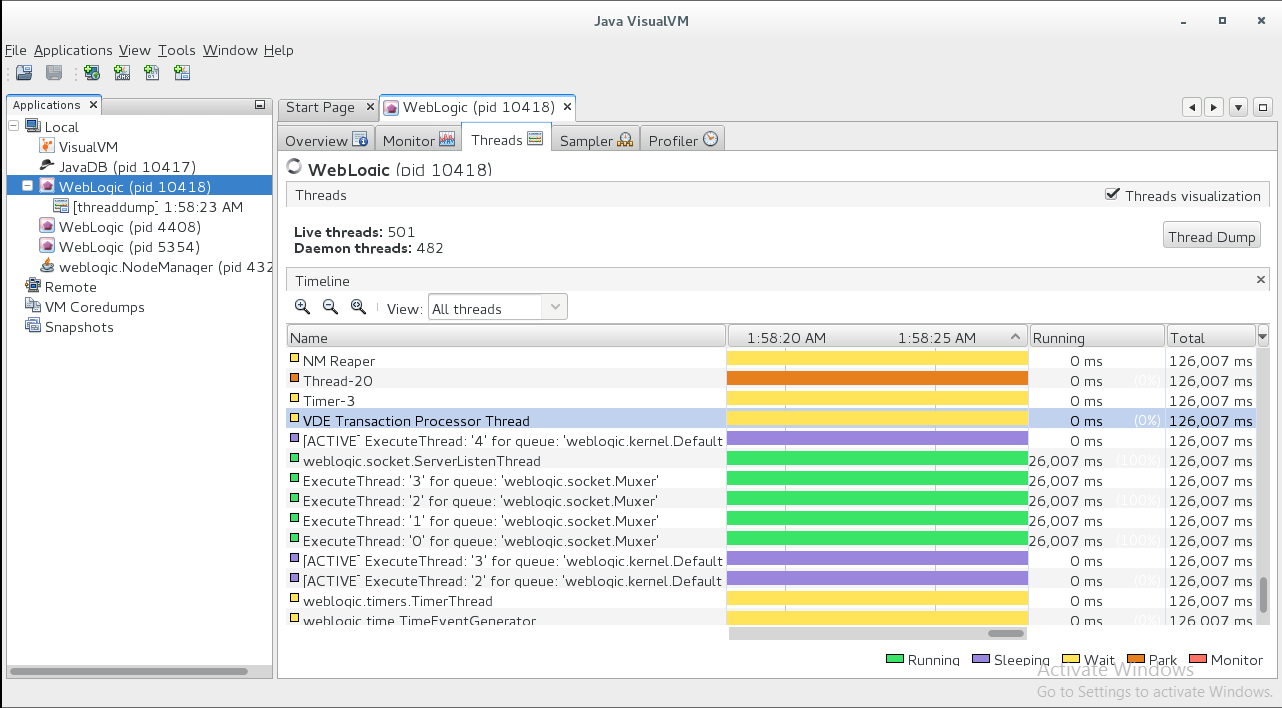
Analyse using GUI



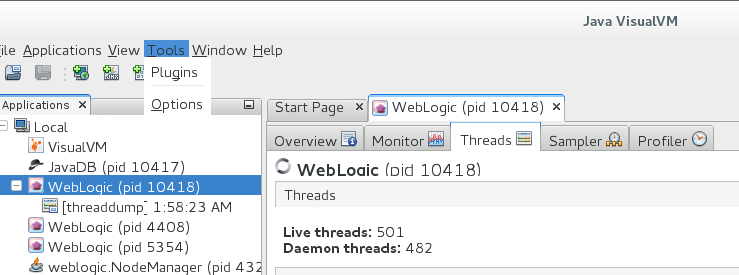
Java Visual Manager



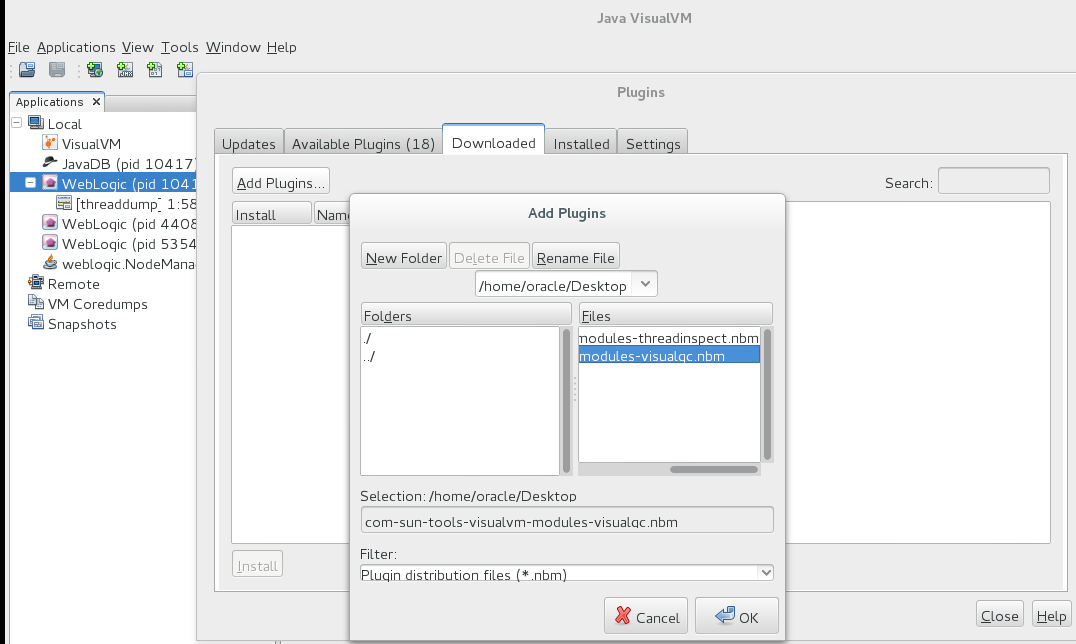
Monitor threads



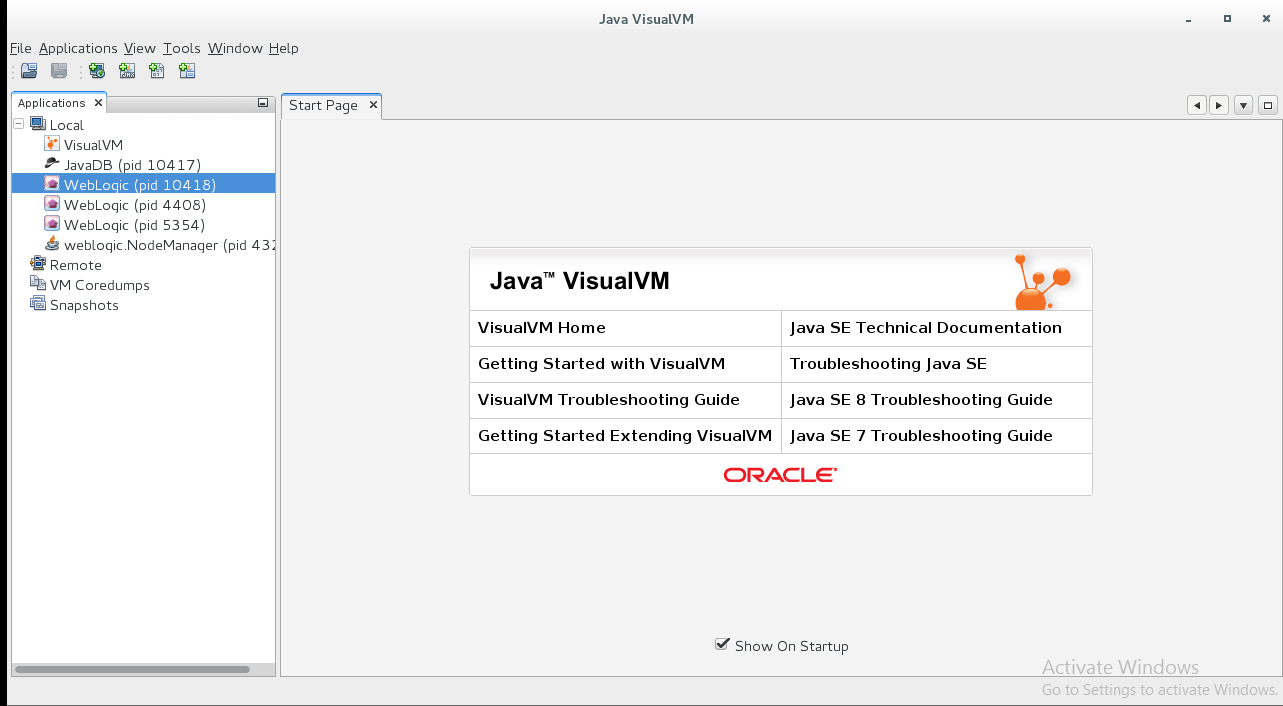
Install GC and thread inspector plugin and monitor



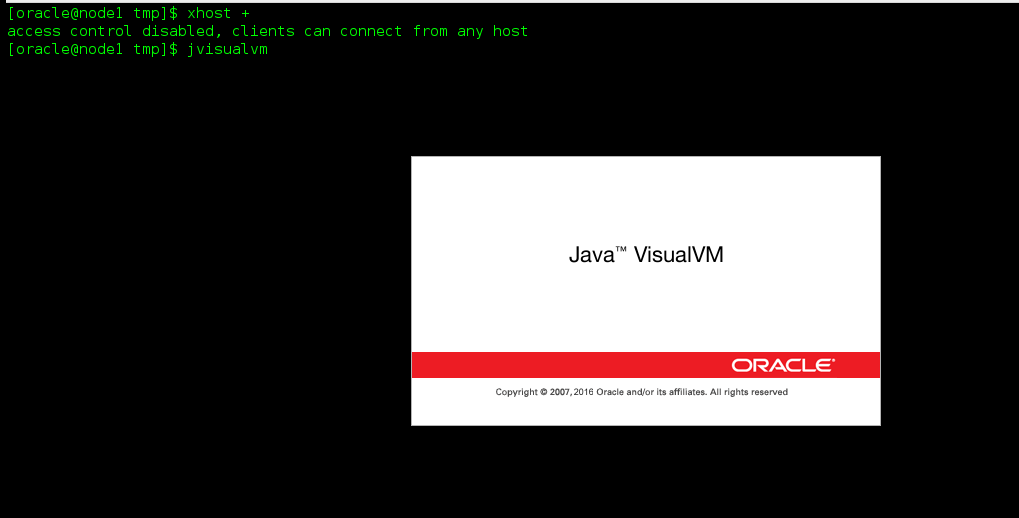
Install Visual GC



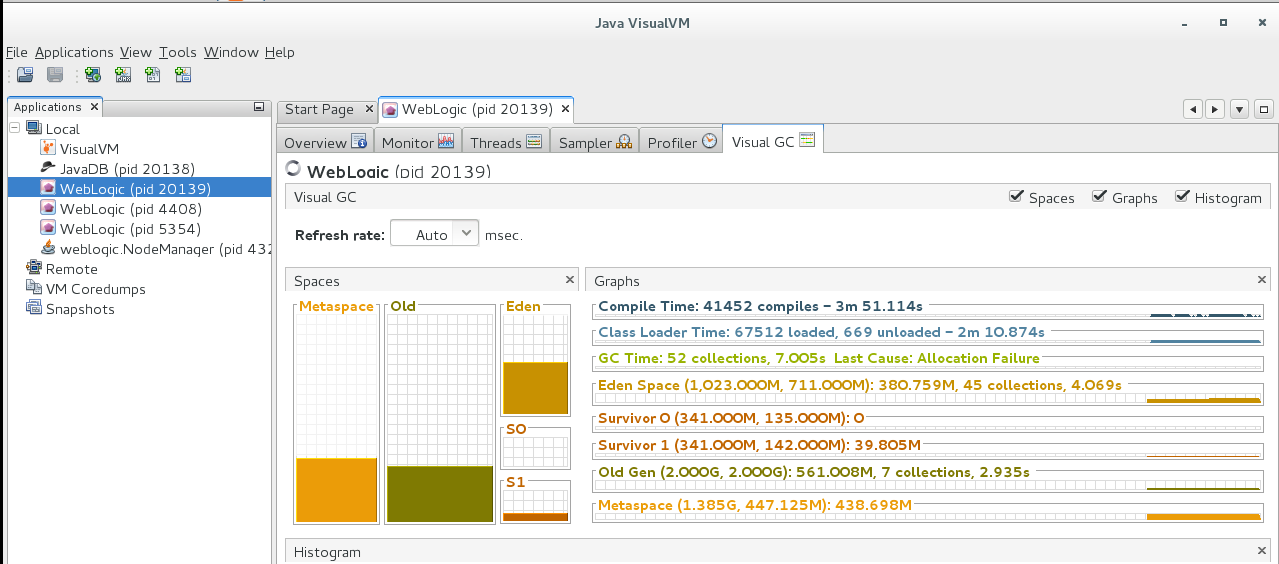
After installing Visual GC and thread inspector , reboot JvisualM



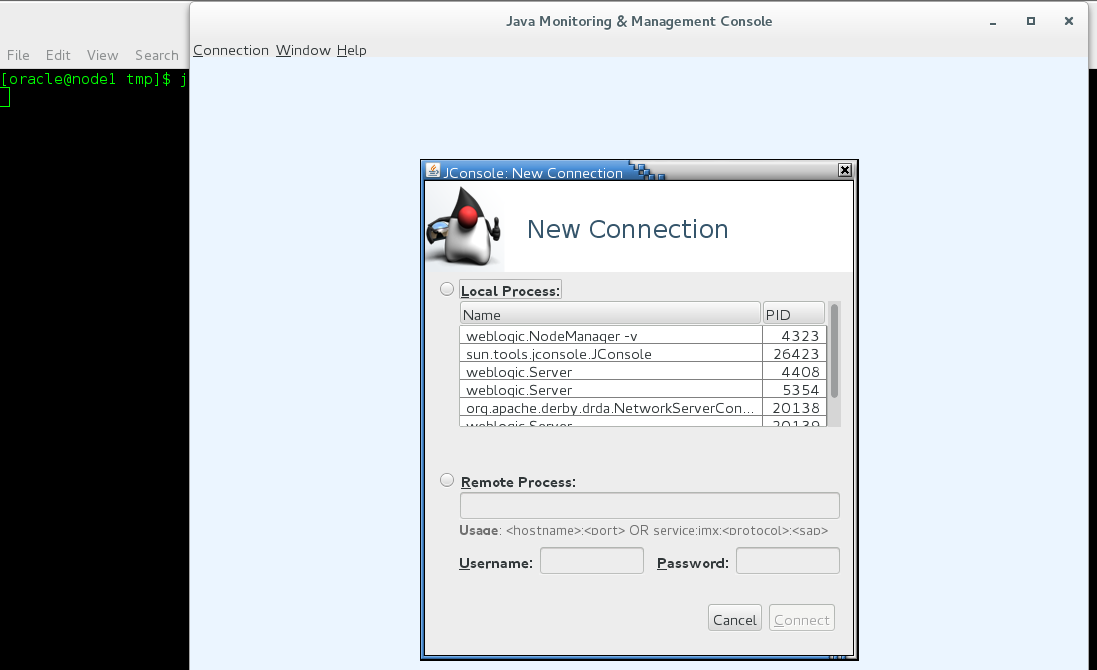
Launch JvisualVm

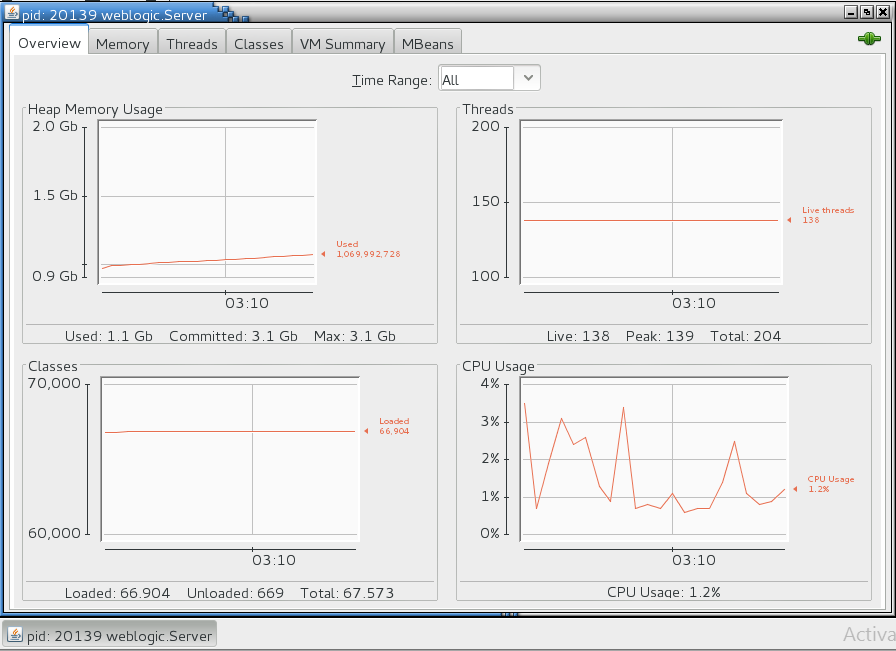


Visual GC



Launch Jconsole



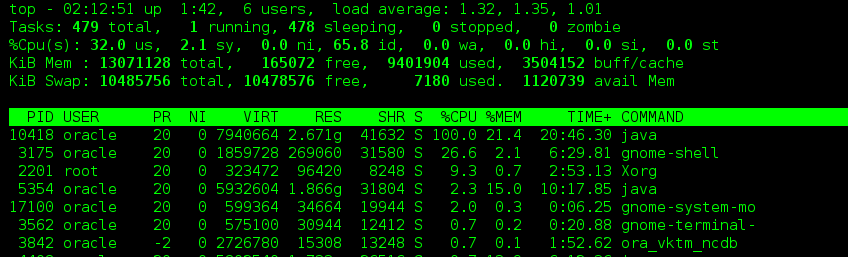


Launch JMC



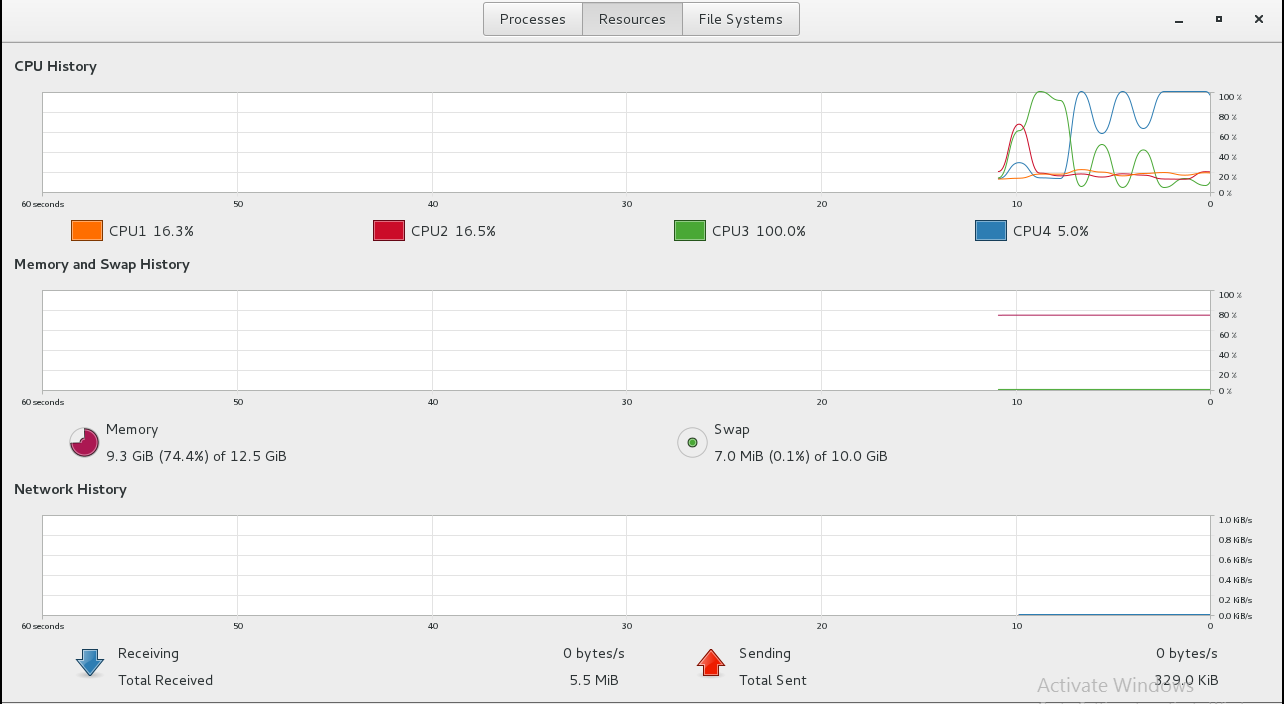


100% CPU used by Managed Server OIM\_server1 (10418)

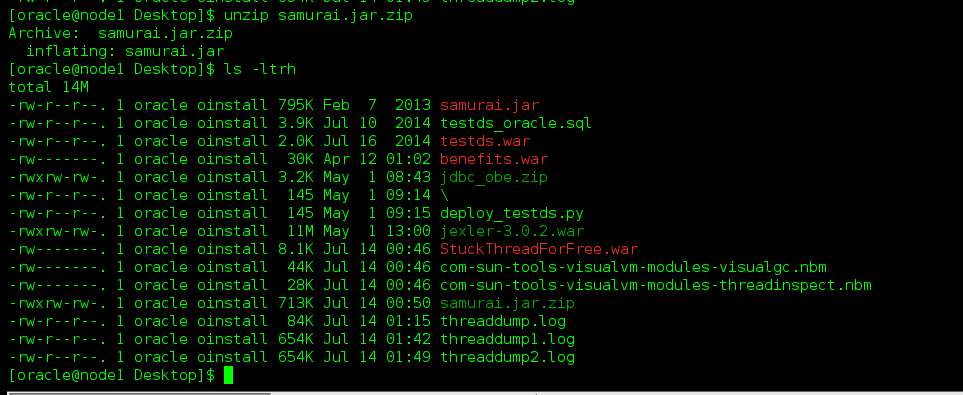


Monitor in GUI

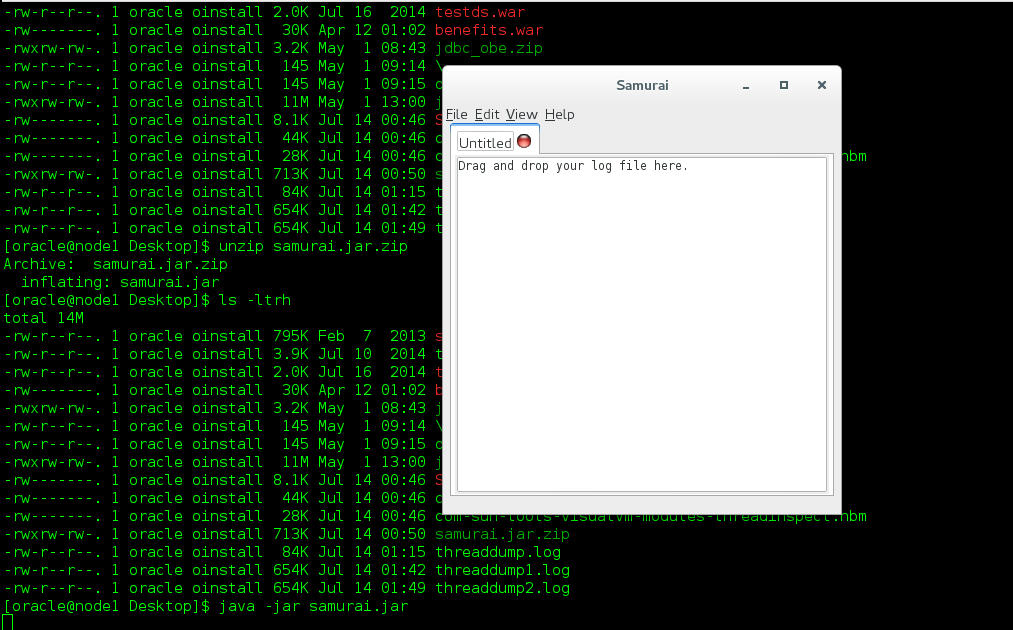




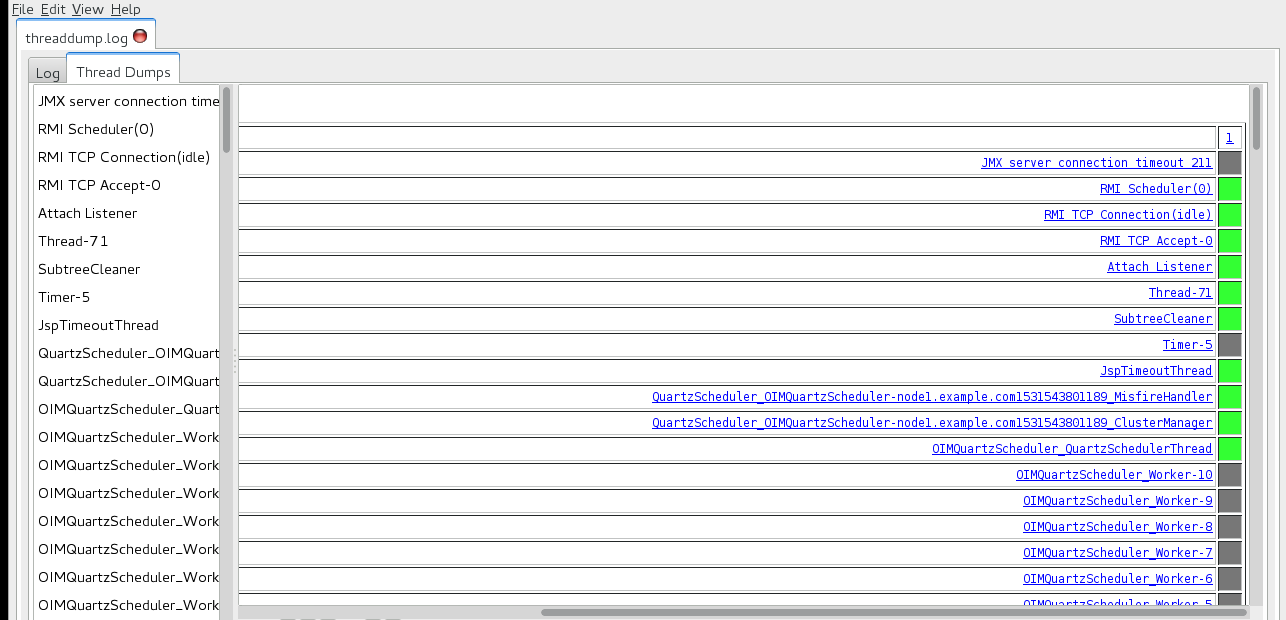
Analyze thread dump using Samurai



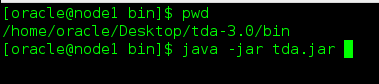
Java –jar samurai.jar

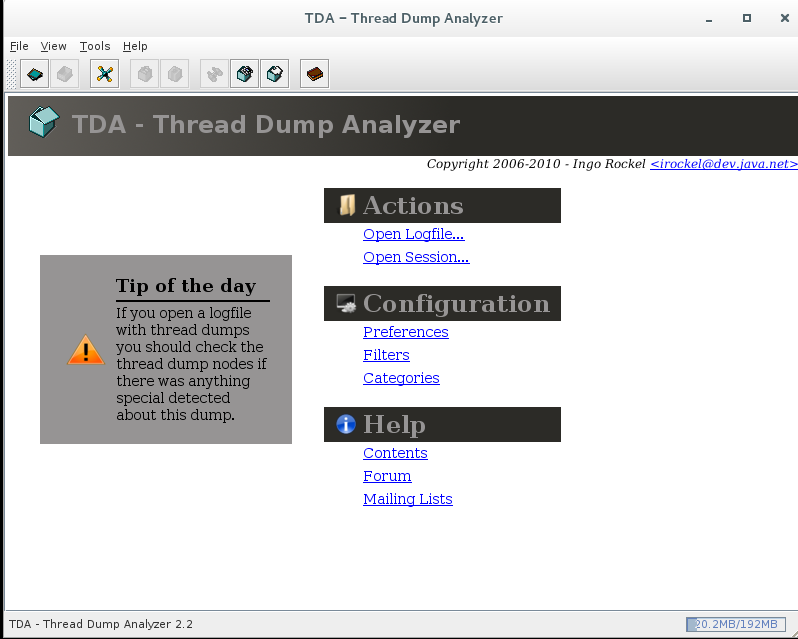


Drap and drop the file

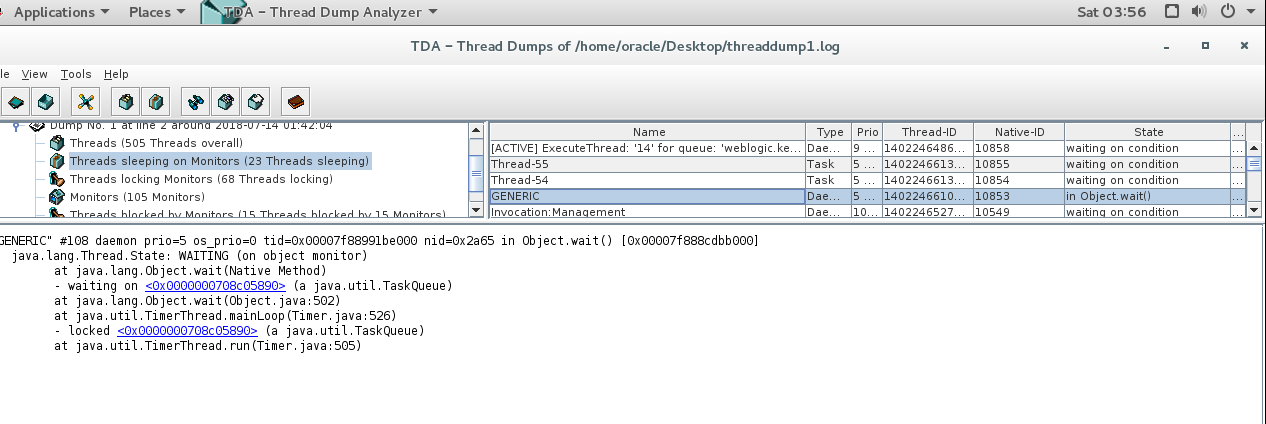


Analyze using thread dump analyzer





Thread dump analyzer



<http://allthingsmdw.blogspot.com/2012/02/analyzing-thread-dumps-in-middleware_9856.html>