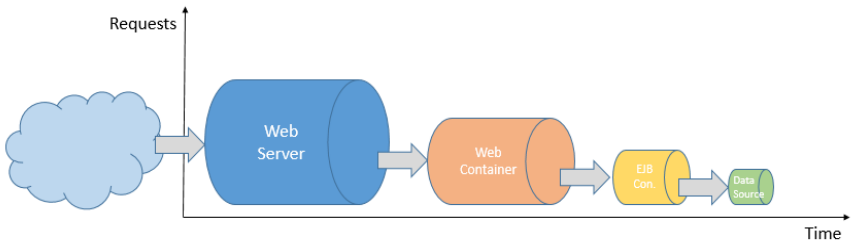
CHAPTER 15: PERFORMANCE TUNING

Theory

There are multiple areas in WebSphere Application Server to tune for better performance and there is no set of parameters that can fit into all systems. You need to consider your application, infrastructure and your workload to plan your environment. Furthermore, you need to test and analyze test results to find areas to tune.

The most common technique for performance tuning is considering queues. The idea is to keep requests as far as possible from the WebSphere to prevent overloading. To do that, it's better to keep the queues close the course of requests bigger and following queues relatively smaller.



It is not a good idea to keep queue sizes same, because we have always limited resources. Depending on your environment, you can consider having significantly bigger queues in your web server or in your data source pools. You need to analyze your architecture to understand the requirements of your environment, so that you can tune accordingly.

WebSphere Application Server uses connection pools for data source connections, because the initial connection to a database consumes higher resources. It is usually a better idea to set connection pool size lower than the "Max Connections" parameter of the "Web Container". A deadlock occurs when application requires multiple connections per thread and the connection pool size is not large enough. In order to avoid deadlock situations, it is better to develop the application to use one connection for each thread. If this is not possible, you need to set the data source connection pool size bigger than the number of threads of which already completed their first connection and waiting for the second. Furthermore, you have to always keep in mind the number of connections that the database is configured while changing the connection pool size.

There are also tweaks to tune EJB containers for better performance. Inactive pool cleanup interval is one of the options for tuning. This setting decides the frequency of cleaning of the EJB beans from the memory. You can set this interval to smaller

values if it takes longer time to create new EJB bean. EJB cache size is another parameter to improve EJB container performance.

IBM Tivoli Performance Viewer is very helpful tool to find out optimum values for performance tuning. With this tool:

- You can view PMI data of local and remote application servers
- You can get configuration advice for better performance
- You can log the performance data for later use such as comparison of the effects of the changed parameters.
- You can view and record server performance logs.

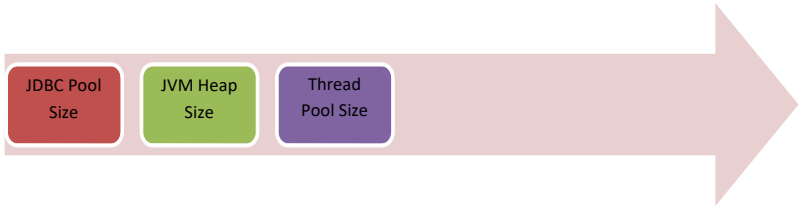
There are also other tools for analysis focused to Java such as IBM Pattern Modelling and Analysis Tool for Java Garbage Collector and IBM Monitoring and Diagnostic tools for Java.

AIM

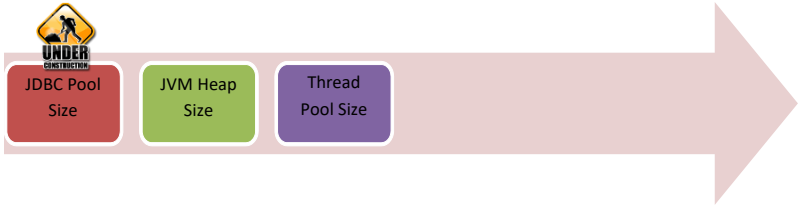
In this lab exercise, you will be familiar with the common performance tuning parameters. In order to achieve this goal, you need to dive into following tasks:

- Change JDBC connection pool size
- Change JVM heap size
- Change web container thread pool size.

Lab Exercise 15: PERFORMANCE TUNING

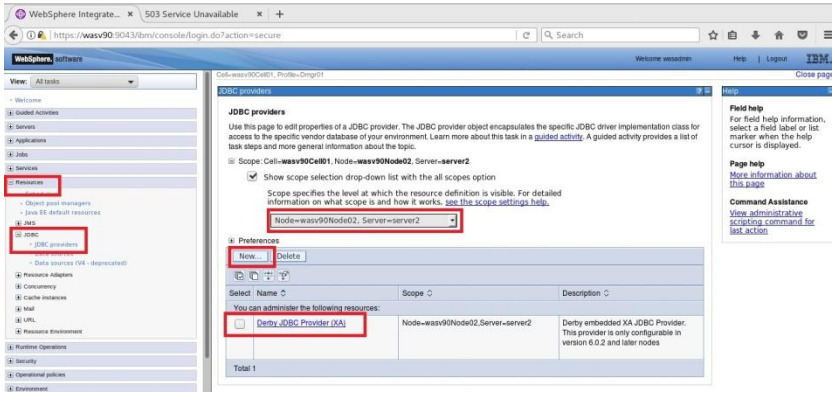


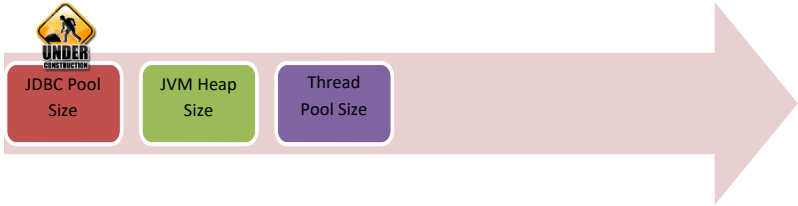
1. **Change JDBC connection pool size**
2. **Change JVM heap size**
3. **Change web container thread pool size**



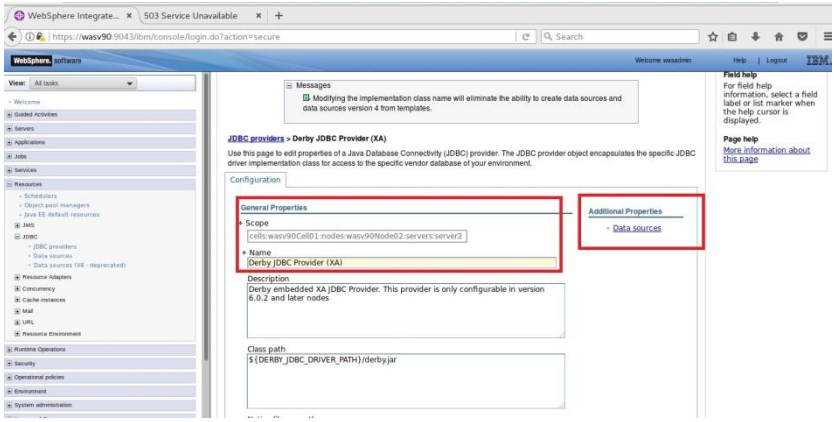
Task 1: Change JDBC connection pool size

Step 1: Navigate to “Resources>JDBC>JDBC providers” and click on the name of the JDBC provider you want to configure.

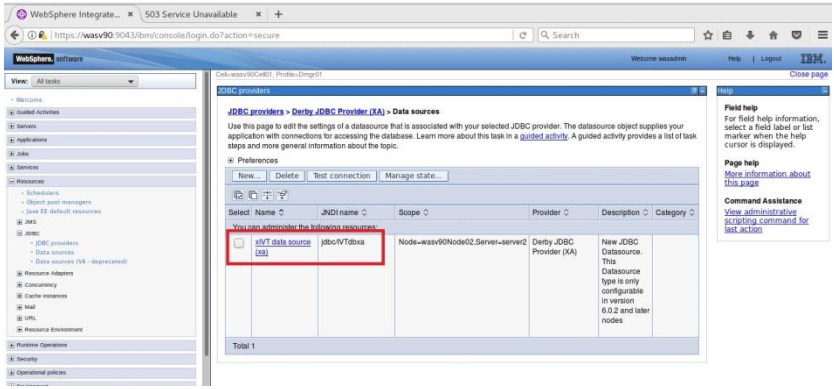


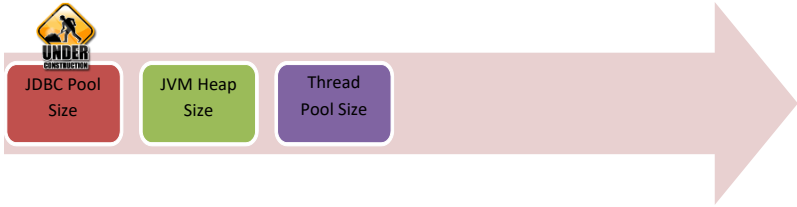


Step 2: Click on “Data sources” under “Additional Properties”.

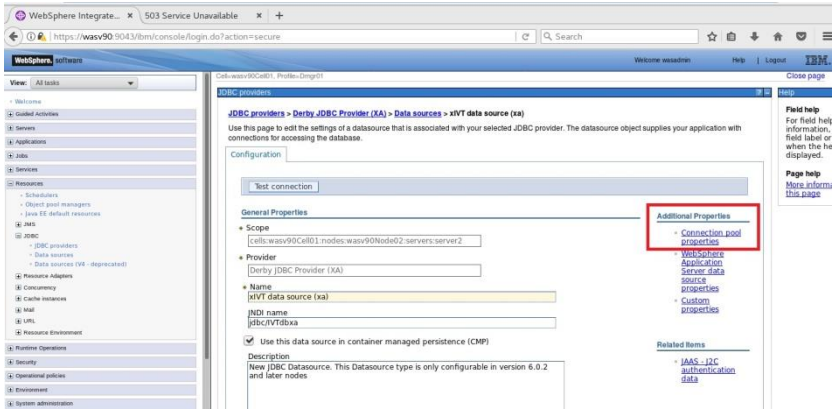


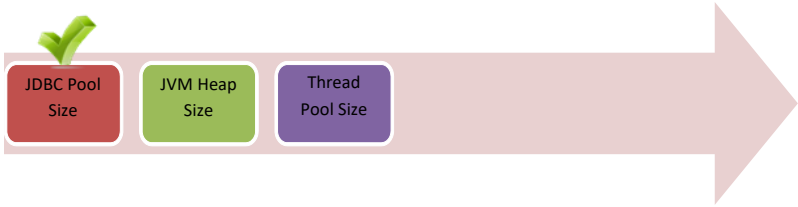
Step 3: Click on the data source name.



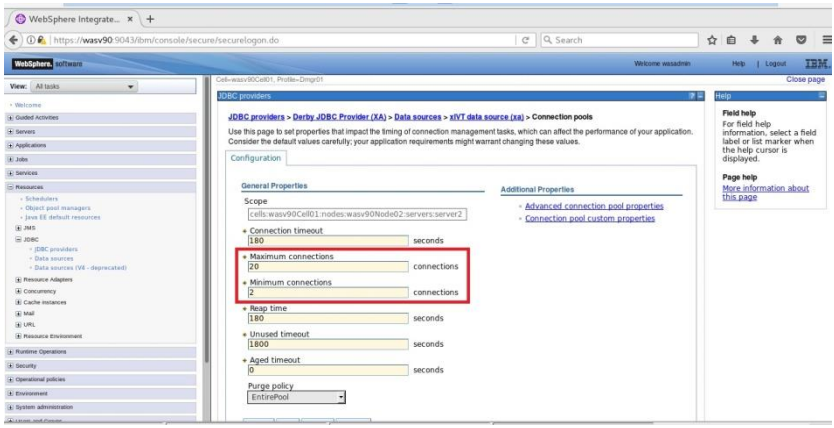


Step 4: Click on “Additional properties>Connection pool properties”.

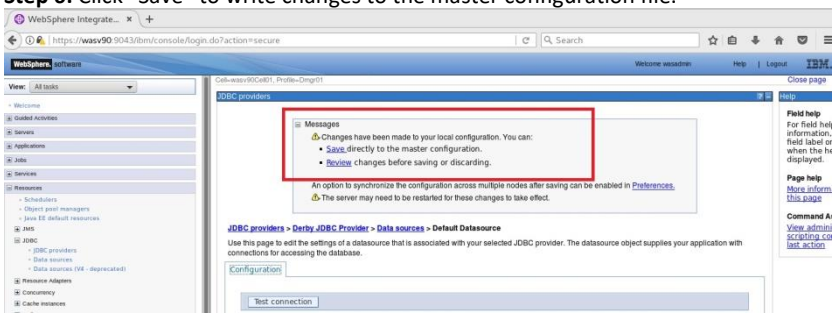




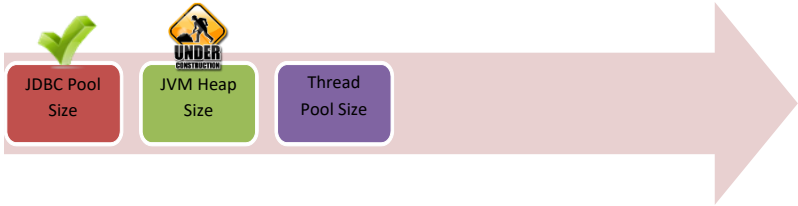
Step 5: Set the maximum and minimum number of connections, then click “OK”.



Step 6: Click “Save” to write changes to the master configuration file.

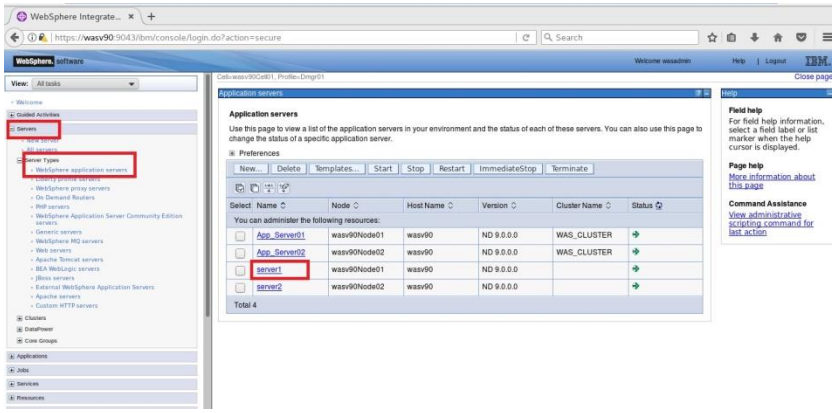


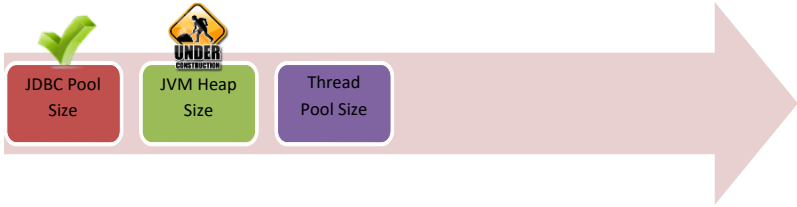
Task 1 is complete!



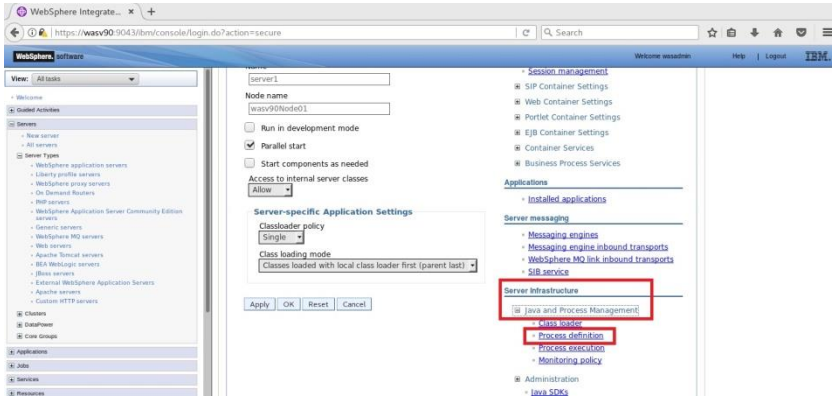
Task 2: Change JVM heap size

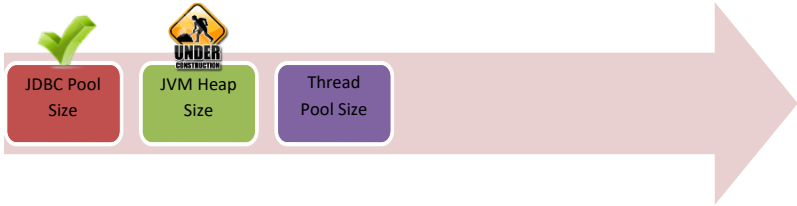
Step 1: Navigate to “Servers>Server Types>WebSphere application servers” and click on the name of the application server you want to configure.



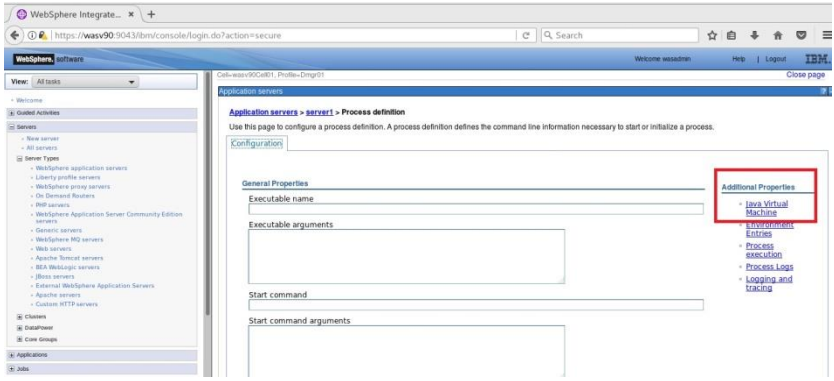


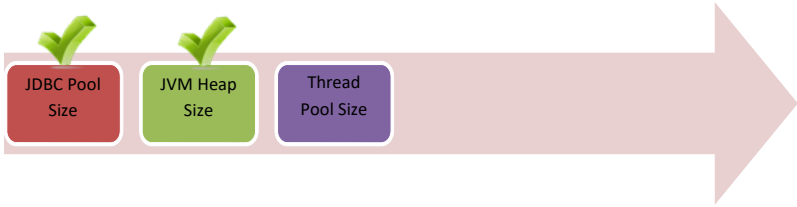
Step 2: Click on “Process definition” under “Java and Process Management”.



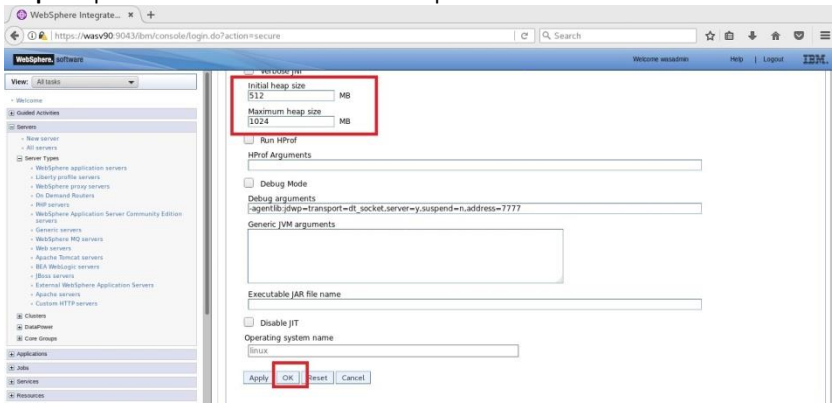


Step 3: Click “Java Virtual Machine” under “Additional Properties”.

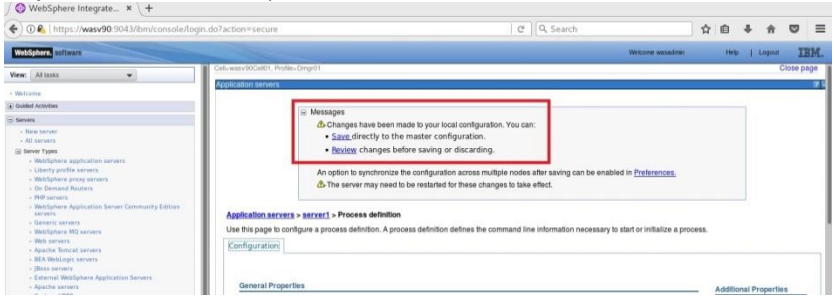




Step 4: Update the initial and maximum heap size and then click “OK”.



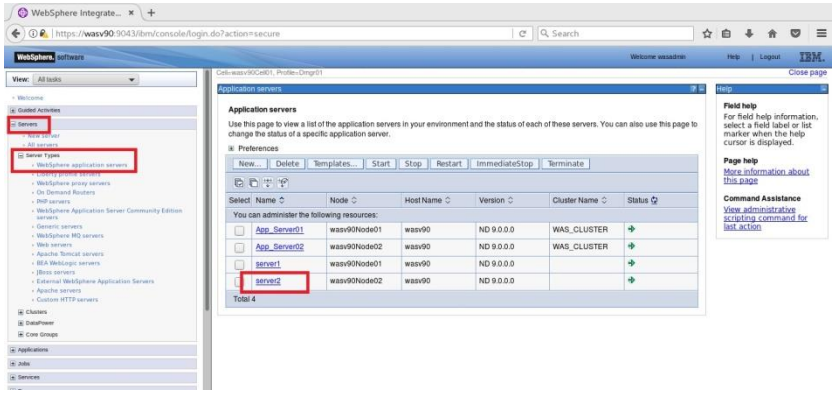
Step 5: Click “Save” to complete.



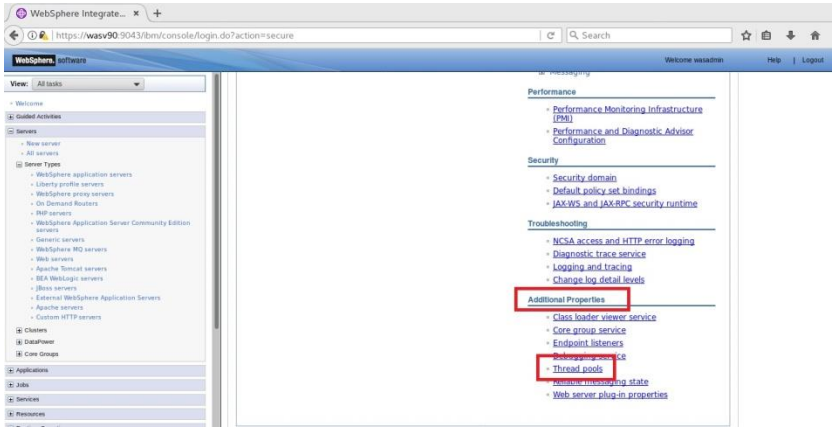
Task 2 is complete!

Task 3: Change web container thread pool size

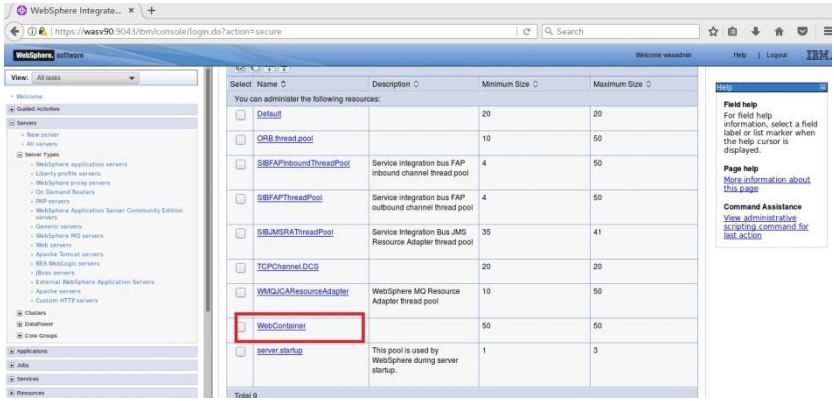
Step 1: Navigate to “Servers>Server Types>WebSphere application servers” and click on the name of the application server you want to configure.

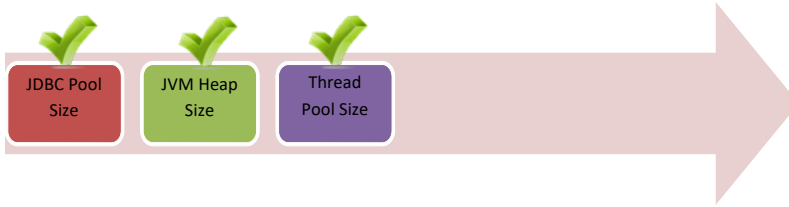


Step 2: Click on “Thread pools” under “Additional Properties”.

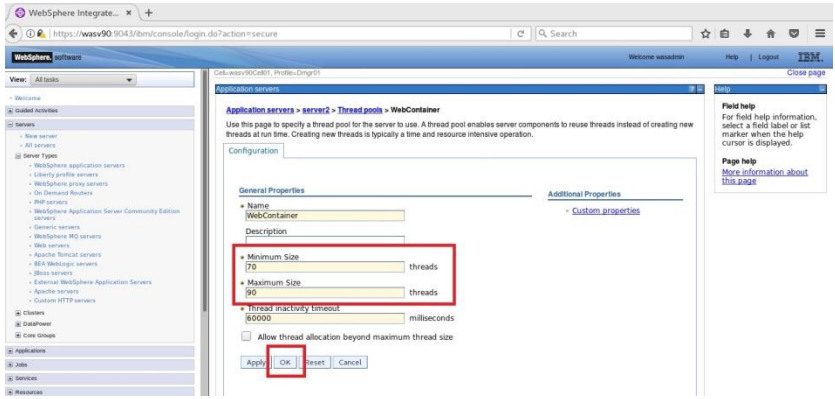


Step 3: Click on “WebContainer” to configure.

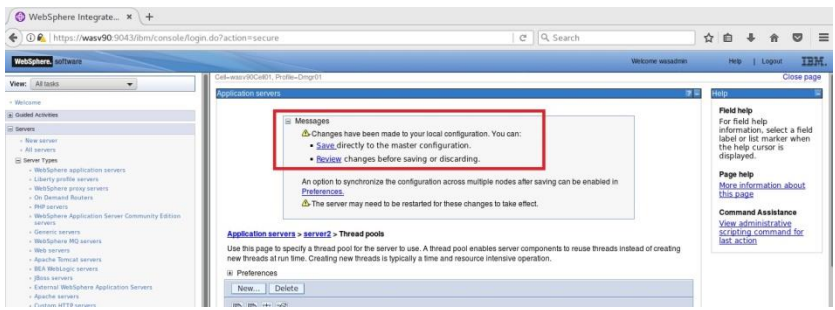




Step 4: Update the number of minimum and maximum threads and click “OK”.



Step 5: Click “Save” to write changes directly to the master configuration file.



Task 3 is complete!

SUMMARY

WebSphere Application Server provides multiple areas for tuning for better performance. Better tuning comes with better knowledge of your application, infrastructure and your workload. It is a continuous cycle of testing and analyzing the test results and tuning different parts of the application server environment. For that purpose, you can use the tools that come with WebSphere Application Server or other tools which are focused to different parts. Performance tuning is one of the most critical operations since we are always limited with resources.

REFERENCES

- http://www.ibmsoftwareservicesindia.com/basic-performance-tuning-parameters-for-websphere-application-server/
- http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.nd.doc/ae/welc_howdoi_tprf.html?lang=en

INDEX