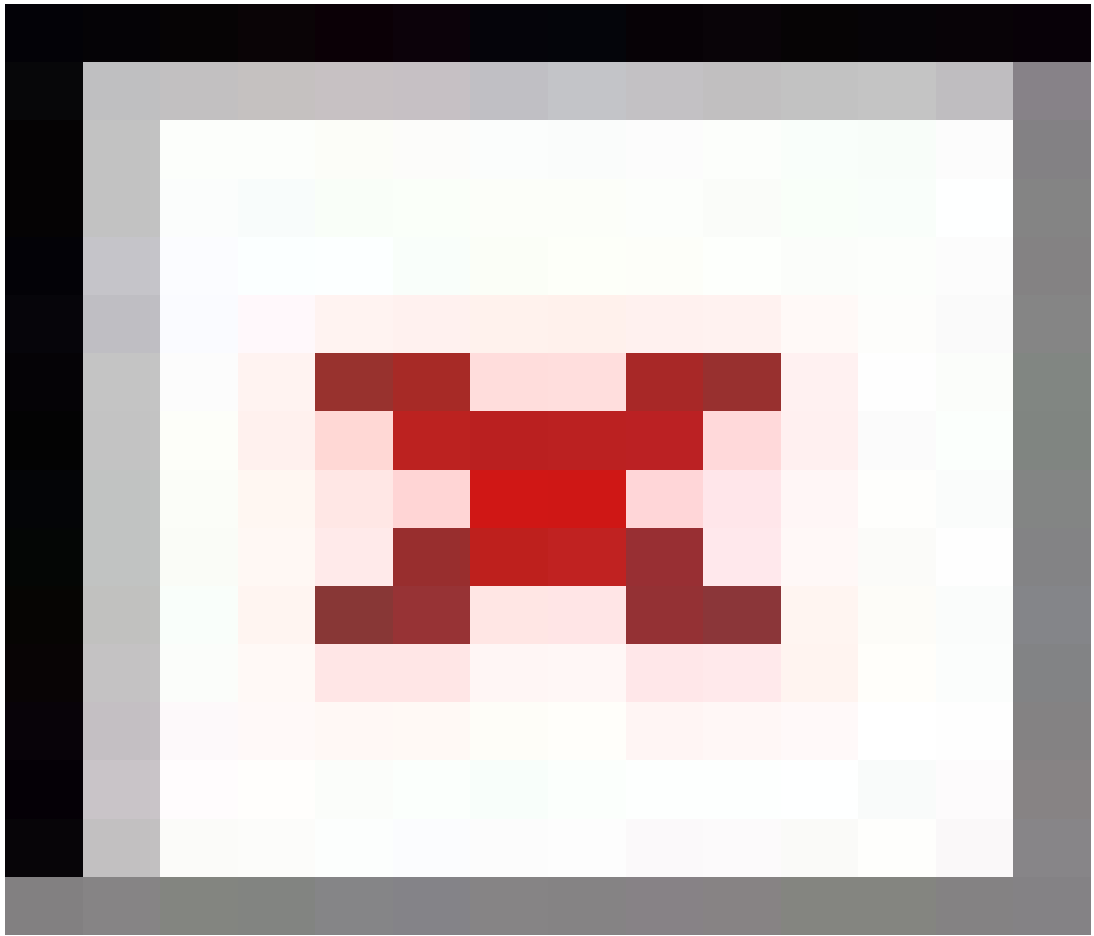




Portal > Knowledgebase > Databases > MSSQL > Basic setup of a JMeter tool for load test with MSSQL server

Basic setup of a JMeter tool for load test with MSSQL server

Shwetal Shah - 2017-09-08 - 0 Comments - in MSSQL



Release	Classification	Level	DB Platform	Category
3.2 and Later	How-to	TUNE SKILLED	MSSQL	Administration

Purpose:

This article will demonstrate steps in sequence which will be executed to configure JMeter load test on a MSSQL server for a Windows Domain Auth user.

Step-1.

Check whether java installed

a. `java -version`

b. java -XshowSettings:properties -version

If you do not have Java installed:

you need to download and install valid java JRE or JDK corresponding to your operating system. Java JRE or JDK can be download from [Java Download page](#) on Oracle website. After successful installation of JRE/JDK, open command prompt again and run the command `java -version` . It should now return java version number that you installed. If you still do not get the valid response after above steps, then you may look

into setting java environment variables under windows system properties.

Step-2.

Download and install JMeter. For this article we have used Apache-JMeter ver. 3.1.

To download JMeter simply go to official Apache JMeter website and download the zip or tgz package.

<https://archive.apache.org/dist/jmeter/binaries/>

After downloading the file, just unzip it anywhere in your system. In the unzipped folder there are all the

files which JMeter requires for its functioning.

Step-3.

Configure JMeter for better performance.

open jmeter.bat file in notepad and update following lines

```
// default configuration
```

```
set HEAP=-Xms512m -Xmx512m
```

```
set NEW=-XX:NewSize=128m -XX:MaxNewSize=128m
```

```
// new settings could be like this *depending* on your hardware and software specs
```

```
// note that, Max heap size should not exceed the 80% of total system memory
```

```
set HEAP=-Xms512m -Xmx2048m
```

set NEW=-XX:NewSize=128m -XX:MaxNewSize=1024m

Step-4.

Configure appropriate driver for MSSqlserver connectivity. There are two different types connectivities in SQL Server

SQL Server Authentication Mode

Windows Authentication Mode

For SQL Server Authentication mode we generally use SQLJDBC Driver. Download SQL JDBC Driver from

the following location [DownloadLink](#) . Click on Download Button and you will see the below pop up menu.

For accessing the database with windows authentication mode , using domain account JTDC driver will be used.

Download JTDC driver from [Here](#) . You will see the below image.



For example you will see -jtds-1.3.1-dist.zip. Extract into a folder and then place the "jtds-1.3.1.jar" file in

the lib folder of JMeter Installation.

This step has accomplished one of the pre-requisites for running a JMeter test with a Windows Domain

user.

Step-5.

Ensure JMeter can access sqljdbc_auth.dll.

The Microsoft Driver's integrated authentication option requires access to the sqljdbc_auth.dll file provided with the driver package. This dll comes in x86 and x64 flavors and you need to ensure your driver is accessing the correct version. Downlad this driver from

<https://docs.microsoft.com/en-us/sql/connect/jdbc/download-microsoft-jdbc-driver-for-sql-server>.

For our example we selected this file - "sqljdbc_4.1.8112.100_enu.tar.gz (x64)"

Extract this to a folder and copy sqljdbc_auth.dll to JMeter's /lib folder.

Other way is to point directly to the dll by adding a *java.library.path* startup flag in the jmeter.bat file.

Edit the jmeter.bat file located in JMeter's /bin folder by adding the following lines to create a new %LIBPATH% environment variable:

```
REM Setting the java.library.path to include the path to the MS  
REM integrated authentication dll. If you're using this approach  
REM be sure you specify the x86 or x64 folder as appropriate  
REM for your environment:
```

```
SET LIBPATH=%LIBPATH% -Djava.library.path="C:\JDBCers\  
Microsoft JDBC Driver 4.1 for SQL Server\sqljdbc_4.1\enu\  
auth\x64"
```

Now modify the %ARGS% environment variable by tacking the %LIBPATH% variable to the end of the string:

```
REM Collect the settings defined above  
SET ARGS=%DUMP% %HEAP% %NEW% %SURVIVOR% %TENURING%  
%PERM% %CLASS_UNLOAD% %DDRAW% %LIBPATH%
```

%JM_START% %JM_LAUNCH% %ARGS% %JVM_ARGS% -jar

"%JMETER_BIN%ApacheJMeter.jar" %JMETER_CMD_LINE_ARGS%

Step-6.

Now open JMeter in a GUI interface and begin with creating a test plan.

a. To open JMeter GUI, just navigate to apache-jmeter-2.13>bin directory and double-click

ApacheJMeter.jar or jmeter.bat file.

b. Now the New test plan window appears. Execute following steps to begin with.

First click on the Test plan and you see at the bottom on window a column to "Add a directory or jar to

classpath". Add the path for the

jtds driver jar file for this value as shown below:



Step-7.

Add "Thread Group" element under the Test plan by right clicking on test

plan->Add->Threads(Users)->Thread Group



Now you will see a window like below:



As you can see we have created a plan with name test-jdbc. When you further right-click this plan you

then create the JDBC-requests item.

Step-8.

Add "JDBC Request" element under thread group by right clicking on thread group->Add->Sampler->JDBC

Resuest(left click). Name it with relevant to your query eg. "Simulate-loop"

The images are just an example for demonstration. Give names as you prefer during creation.



b.

You will now get a screen as below where we input the value for Variable name bound to pool as "stest" to

identify the configuration to be used for differentiation.

Also that in image below you see the Query type which will be used for load generation. We used the

Select Statement as such type and embedded a few Queries as you see.



c. Adding a JDBC configuration element.

Add "JDBC Connection Configuration" element by right clicking on thread group->Add->Config Element->



You get a screen like below now:



In the above step you can see that we created a config element with name stest and filled in various

values as required. Most importantly the last part of this window contains

the values for Database url, Jdbc driver class, username and password. We filled in those values to have a

Domain user authentication on the SQL server via JMeter. In our example

we used:

Database url:

`jdbc:jtds:sqlserver://win2k8m1.mywoods.edu;DatabaseName=master;domain=mywoods.edu`

JDBC Driver class: net.sourceforge.jtds.jdbc.Driver

Step 9.

The results of Queries which you will load test will be gathered in a results pane. For thies we need to add

the Listeners. We will add a basic listener "View Results Tree".



Step-10.

Finally we now get ready to run this plan but it is good to save it first with some name. We saved as

myplan.jmx for this Demo. To run the plan now:

Now we save the plan with a good name in a jmx file for future use. To run this plan now:

a. Click "View results tree and get the Results window ready to view success/failures.

b. Click the Run on main menu --> Start button



As you can see above the test plan runs successfully for Queries selected. If there are any issues then the

results will appear in red.

On similar lines as above and with minor changes in the configuration steps you can run this test with the

SQL Auth user as well instead of the Windows Domain User which was demonstrated. It is worth

noting the most of the load runners need to be installed and configured on a windows station joined to the

Domain for running with a Windows Domain user. But this example demonstrates the configuration

which runs with windows domain auth user which from a Windows desktop which is not a Domain member.

You can also configure the ScaleArc ip/Vip to continue on these lines and configure NTLM/Kerberos load

tests via JMeter. Please check link below for such configurations.

If you are experiencing issues with ScaleArc or with any of its features, please contact

ScaleArc Support.

We are available 24x7 by phone at 855 800 7225 or +1 408 412 7315.

For general support inquiries, you can also e-mail us at support@scalearc.com.

Copyright ©2014
Cognate, Inc. All rights
reserved. Cognate
is a registered
service mark of
Cognate, Inc.
2000 Tanager
Drive Santa Clara, CA
95051
Email: sales@cognate.com

Permalink:
<https://support.scalearc.com/kb/articles/4403>