



Configuring a Denodo Cluster with BigIP F5

Revision 20200910

NOTE

This document is confidential and proprietary of **Denodo Technologies**. No part of this document may be reproduced in any form by any means without prior written authorization of **Denodo Technologies**.

Copyright © 2020
Denodo Technologies Proprietary and Confidential

CONTENTS

1 GOAL.....	3
2 CONTENT.....	4
2.1 HEALTH CHECK CONFIGURATION IN BIGIP F5.....	4
2.2 USING DYNAMIC RATIO AS A LOAD BALANCING STRATEGY IN BIGIP F56	
2.3 CHECK MEMORY HEAP SIZE USAGE WITH AN SNMP AGENT.....	6
2.4 CHECK CPU USAGE IN A WINDOWS NODE WITH AN SNMP AGENT.....	6
2.5 SENDING TCP KEEP-ALIVES TO CLIENT APPLICATIONS IN BIGIP F5 USING	
TCP PROFILES.....	7
3 REFERENCES.....	9

1 GOAL

This document explains how to configure a Denodo cluster in BigIP F5 in order to distribute the load between several nodes.

2 CONTENT

2.1 HEALTH CHECK CONFIGURATION IN BIGIP F5

In order to set up the health check in BigIP F5, we need to define a monitor in the configuration of the load balancer. The external monitors need to return anything to the standard output when the server is healthy. We will use the ping script of Denodo to build the custom health monitor.

The Denodo [ping](#) script is distributed with the Denodo Platform and it allows to check whether a Virtual DataPort server is running. It can be found not only in the \$DENODO_HOME/bin folder but also packed with other administration tools as being part of the *denodo-db-tools*.

The syntax for invoking this script requires to provide at least the Denodo VDP server that is going to be pinged:

```
ping [-t timeout] [ -q <query> -l <login> -p <password> ] [-v]
<host>[:<port>][/<database name>
```

Syntax of the ping script

To use Denodo's health check script in BigIP F5 we will need to create an "External Monitor" in the Monitors section of the Main tab of the configuration screen:

1. Unzip \$DENODO_HOME/tools/db/denodo-db-tools.zip in /usr/bin/monitors in the load balancer's file system. Permissions on the file *ping.sh* (located in the folder *denodo-db-tools/bin*) must be 700 or better; granting root rwx privileges to the file.
2. Create a new file containing the code below in /usr/bin/monitors in the load balancer's file system, then name it as "monitor_denodo" for instance. Permissions on the file must be 700 or better; granting root rwx privileges to the file

```
#!/bin/sh
# These arguments are supplied automatically for all external pingers:
# $1 = IP (nnn.nnn.nnn.nnn notation or hostname)
# $2 = port (decimal, host byte order)
# $3 and higher = additional arguments
#
node_ip=`echo $1 | sed 's/::ffff://'\`
pidfile="/var/run/pinger.$node_ip..$2.pid"
```

```

if [ -f $pidfile ]
then
    kill -9 `cat $pidfile` > /dev/null 2>&1
fi

cd /usr/bin/monitors/denodo-db-tools/bin
JAVA_HOME=/usr/java/openjdk
# JAVA_HOME=/usr/java/jre_version/ in case we have installed a new JVM

#echo "sh /usr/bin/monitors/denodo-db-tools/bin/ping.sh -t 2000 -v
$node_ip:9999 " >$pidfile
status=` /usr/bin/monitors/denodo-db-tools/bin/ping.sh -t 2000 -v
$node_ip:9999 2>/dev/null | grep OK | wc -l`

if [ "$status" -eq "1" ]
then
# Remove the pidfile before the script echoes anything to stdout and is
killed by bigd
    rm -f $pidfile
    echo "up"
fi

# Remove the pidfile before the script ends
rm -f $pidfile
    
```

Denodo External Monitor Script Template

3. Once the script is ready it can be imported by browsing to the following menu and selecting the monitor file:

System > File Management > External Monitor Program File List

2.1.1 Java compatibility

The Denodo Platform 6.0 uses Java 1.7 which matches the latest versions of BigIP F5. However, the newer versions of the Denodo Platform have updated the version of its embedded JVM, for instance, the Denodo 7.0 ping script and the libraries it uses are compiled with Java 1.8. Therefore, a BigIP F5 balancer using Java 1.7 will not be able to execute them.

Hence, for installations of Denodo Platform 7.0 or Denodo Platform 8.0, the previous template can be modified so the JAVA_HOME variable points to a Java 1.8 installation.

Tuning the script for Java 1.8

1. Install Java 8 in the load balancer machine. The following link contains the instructions to install Java 8 on a linux environment:
https://java.com/en/download/help/linux_x64rpm_install.xml#download
2. Then, update the external monitor file replacing the value for JAVA_HOME:
 JAVA_HOME=/usr/java/openjdk

with
 JAVA_HOME=/usr/java/jre1.8.0_xxx/ /* the new path */

2.2 USING DYNAMIC RATIO AS A LOAD BALANCING STRATEGY IN BIGIP F5

Dynamic Ratio load balancing is a special load balancing mode for which the BigIP system actively polls pool members and assigns a weight value to each individual member based on a set of default or user-defined threshold values. The weight assigned to an individual pool member determines how much traffic the pool member receives. The higher the value of the weight, the more traffic is directed to the pool member and the lower the value, the less traffic is directed to the pool member. The weights assigned to the pool members are dynamic and can change with each polling cycle that the **snmp_dca** or the **snmp_dca_base** monitors perform.

The configuration of the dynamic ratio strategy in BigIP F5 requires several steps:

1. Configure the SNMP agents in the JVM and operative system.
2. In the load balancer, create SNMP DCA BASE monitors.
3. Select "Dynamic Ratio (node)" as the balancing strategy.

In the following sections you can find the details for each one of these steps. More information on this balancing strategy can also be found here:

<http://support.f5.com/kb/en-us/solutions/public/9000/100/sol9125.html>

2.3 CHECK MEMORY HEAP SIZE USAGE WITH AN SNMP AGENT

Edit the management.properties file located in DENODO_HOME/jre/lib/management with the following values

- com.sun.management.snmp.port=161 (agent port)
- com.sun.management.snmp.interface=0.0.0.0 (allow petitions from any machine)
- com.sun.management.snmp.acl=false

To configure a SNMP monitor to check the heap size used by the JVM running in the node:

- In the navigation panel, click on 'Local Traffic' and select 'Monitors' → 'Create'.
- Select type 'SNMP DCA BASE'.
- Add following user variables to check used heap in remote JVM.
 - jvmHeapUsed = .1.3.6.1.4.1.42.2.145.3.163.1.1.2.11.0
 - jvmHeapUsed_threshold = 80000000
 - jvmHeapUsed_coefficient = 1

2.4 CHECK CPU USAGE IN A WINDOWS NODE WITH AN SNMP AGENT

First we have to install the SNMP agent on Windows. The following instructions work for Windows Server 2003, 2003 SP1, SP2 and 2003 R2. Other versions of Windows may require a different process:

1. Start → Control Panel → Programs and Features → Turn Windows features on or off.
2. Select 'Simple Network Management Protocol (SNMP)→ WMI SNMP Provider' and click 'OK'.
3. In the components wizard click 'Next' and later 'Finish'.

4. SNMP agent configuration:
 - a. The installation process creates the services 'SNMP Service' and 'SNMP Trap Service'. To modify the configuration of these services go to Start → Control Panel → Administrative Tools → 'Services'.
 - b. Right-click on one of the services and click 'Properties'.
 - c. In the security tab, add 'public' community with READ ONLY privileges and the hosts list from which SNMP packets will be accepted.
 - d. If the services are not started automatically, Right-click → 'Start'

To configure the SNMP monitor to check CPU usage in a windows server node with the SNMP agent provided by Microsoft:

1. Open the navigation panel → Click on 'Local Traffic' and select 'Monitors' → 'Create'.
2. Select type 'SNMP DCA BASE'.
3. Add following user variables to check cpu usage.
 - cpu = .1.3.6.1.2.1.25.3.3.1.2.2
 - cpu_threshold = 40
 - cpu_coefficient = 1

2.5 SENDING TCP KEEP-ALIVES TO CLIENT APPLICATIONS IN BIGIP F5 USING TCP PROFILES

When the BigIP system load-balances or forwards long-lived TCP connections, on environments with a lot of heavy processing and long-running queries, the long-lived TCP connections between client applications and the Denodo Platform Cluster need to be managed by the BigIP.

Some server applications send TCP Keep-Alive packets to maintain those long-lived connections with a client and, although this is not the approach used by the Denodo Servers, the BigIP system can be used as the server that sends the TCP Keep-Alive packets to the client applications and handle in this manner the long-lived TCP connections.

So, in order to prevent the clients from timing out or disconnecting because of a lack of response within a reasonable time frame, the BigIP systems will send a TCP Keep-Alive packet to the client playing the server role in this server-client TCP Keep-Alives communication. For that purpose F5 recommends configuring a custom client-side TCP profile as follows:

1. Log in to the Configuration utility.
2. Navigate to 'Local Traffic' → 'Profiles' → 'Protocol'.
3. Select 'TCP'.
4. Click 'Create'.
5. Type a name for the profile, for example: "Custom_Client_TCP".
6. Check the box next to 'Keep Alive Interval'.
7. Specify the amount of time, in seconds, that the BigIP system should send a TCP Keep-Alive packet to the client.
8. Select 'Finished'.
9. Apply the custom TCP profile to the virtual server that is handling the load balancing of the server application.

For further information regarding the implementation of TCP Keep-Alives for server-

client communication using TCP profiles see <https://support.f5.com/kb/en-us/solutions/public/8000/000/sol8049.html?sr=49036270>

3 REFERENCES

[LTM External Monitors: The Basics](#)

[F5 DevCentral Wiki: Advanced Design & Config - External Monitor](#)