BIG-IP® Global Traffic Manager™: Load Balancing

Version 11.5

Table of Contents

Legal Notices...5

Acknowledgments..7

Chapter 1: Using Topology Load Balancing to Distribute DNS Requests to Specific Resources..11

How do I configure GTM to load balance DNS requests to specific resources?.............12

About Topology load balancing..12

Understanding topology records..12

Understanding how the BIG-IP system prioritizes topology records.....................13

Creating a topology record..15

Deleting a topology record..15

About Topology load balancing for a wide IP...16

Example configuration: Topology load balancing for a wide IP..............................16

Configuring a wide IP for Topology load balancing..18

About Topology load balancing for a pool...18

Example configuration: Topology load balancing for a pool..................................19

Configuring a pool for Topology load balancing...21

About Topology load balancing for both wide IPs and pools...................................21

About IP geolocation data...22

About topology records and IP geolocation data..22

Downloading and installing updates to the IP geolocation data............................22

Reloading default geolocation data using the Configuration utility........................23

Reloading default geolocation data using tmsh...23

Chapter 2: Preparing BIG-IP GTM for Static Persist Load Balancing for Transaction-Oriented Traffic..25

About preparing GTM to load balance transactions to a single transaction manager......26

About Static Persist load balancing...26

Legal Notices

Publication Date

This document was published on January 27, 2014.

Publication Number

MAN-0465-02

Copyright

F5 Networks, Inc. (F5) believes the information it furnishes to be accurate and reliable. However, F5 assumes no responsibility for the use of this information, nor any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent, copyright, or other intellectual property right of F5 except as specifically described by applicable user licenses. F5 reserves the right to change specifications at any time without notice.

Trademarks

Patents

This product may be protected by one or more patents indicated at:
http://www.f5.com/about/guidelines-policies/patents

Export Regulation Notice

This product may include cryptographic software. Under the Export Administration Act, the United States government may consider it a criminal offense to export this product from the United States.

RF Interference Warning

This is a Class A product. In a domestic environment this product may cause radio interference, in which case the user may be required to take adequate measures.

FCC Compliance

This equipment has been tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This unit generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.

Any modifications to this device, unless expressly approved by the manufacturer, can void the user's authority to operate this equipment under part 15 of the FCC rules.

Canadian Regulatory Compliance

This Class A digital apparatus complies with Canadian ICES-003.

Standards Compliance

This product conforms to the IEC, European Union, ANSI/UL and Canadian CSA standards applicable to Information Technology products at the time of manufacture.

Acknowledgments

This product includes software developed by Gabriel Forté.

This product includes software developed by Bill Paul.

This product includes software developed by Jonathan Stone.

This product includes software developed by Manuel Bouyer.

This product includes software developed by Paul Richards.

This product includes software developed by the NetBSD Foundation, Inc. and its contributors.

This product includes software developed by the Politecnico di Torino, and its contributors.

This product includes software developed by the Swedish Institute of Computer Science and its contributors.

This product includes software developed by the University of California, Berkeley and its contributors.

This product includes software developed by the Computer Systems Engineering Group at the Lawrence Berkeley Laboratory.

This product includes software developed by Christopher G. Demetriou for the NetBSD Project.

This product includes software developed by Adam Glass.

This product includes software developed by Christian E. Hopps.

This product includes software developed by Dean Huxley.

This product includes software developed by John Kohl.

This product includes software developed by Paul Kranenburg.

This product includes software developed by Terrence R. Lambert.

This product includes software developed by Philip A. Nelson.

This product includes software developed by Herb Peyerl.

This product includes software developed by Jochen Pohl for the NetBSD Project.

This product includes software developed by Chris Provenzano.

This product includes software developed by Theo de Raadt.

This product includes software developed by David Muir Sharnoff.

This product includes software developed by SigmaSoft, Th. Lockert.

This product includes software developed for the NetBSD Project by Jason R. Thorpe.

This product includes software developed by Jason R. Thorpe for And Communications, http://www.and.com.

This product includes software developed for the NetBSD Project by Frank Van der Linden.

This product includes software developed for the NetBSD Project by John M. Vinopal.

This product includes software developed by Christos Zoulas.

This product includes software developed by the University of Vermont and State Agricultural College and Garrett A. Wollman.

This product includes software developed by Balazs Scheidler (bazsi@balabit.hu), which is protected under the GNU Public License.

This product includes software developed by Niels Mueller (nisse@lysator.liu.se), which is protected under the GNU Public License.

In the following statement, *This software* refers to the Mitsumi CD-ROM driver: This software was developed by Holger Veit and Brian Moore for use with 386BSD and similar operating systems. *Similar operating systems* includes mainly non-profit oriented systems for research and education, including but not restricted to NetBSD, FreeBSD, Mach (by CMU).

This product includes software developed by the Apache Group for use in the Apache HTTP server project (http://www.apache.org/).

This product includes software licensed from Richard H. Porter under the GNU Library General Public License (© 1998, Red Hat Software), www.gnu.org/copyleft/lgpl.html.

This product includes the standard version of Perl software licensed under the Perl Artistic License (© 1997, 1998 Tom Christiansen and Nathan Torkington). All rights reserved. You may find the most current standard version of Perl at http://www.perl.com.

This product includes software developed by Jared Minch.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product contains software based on oprofile, which is protected under the GNU Public License.

This product includes RRDtool software developed by Tobi Oetiker (http://www.rrdtool.com/index.html) and licensed under the GNU General Public License.

This product contains software licensed from Dr. Brian Gladman under the GNU General Public License (GPL).

This product includes software developed by the Apache Software Foundation (http://www.apache.org/).

This product includes Hypersonic SQL.

This product contains software developed by the Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, and others.

This product includes software developed by the Internet Software Consortium.

This product includes software developed by Nominum, Inc. (http://www.nominum.com).

This product contains software developed by Broadcom Corporation, which is protected under the GNU Public License.

This product contains software developed by MaxMind LLC, and is protected under the GNU Lesser General Public License, as published by the Free Software Foundation.

This product includes unbound software from NLnetLabs. Copyright ©2007. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of NLnetLabs nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE

LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes Intel QuickAssist kernel module, library, and headers software licensed under the GNU General Public License (GPL).

This product includes software licensed from Gerald Combs (gerald@wireshark.org) under the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or any later version. Copyright ©1998 Gerald Combs.

This product includes software developed by Thomas Williams and Colin Kelley. Copyright ©1986 - 1993, 1998, 2004, 2007

Permission to use, copy, and distribute this software and its documentation for any purpose with or without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Permission to modify the software is granted, but not the right to distribute the complete modified source code. Modifications are to be distributed as patches to the released version. Permission to distribute binaries produced by compiling modified sources is granted, provided you

1. distribute the corresponding source modifications from the released version in the form of a patch file along with the binaries,
2. add special version identification to distinguish your version in addition to the base release version number,
3. provide your name and address as the primary contact for the support of your modified version, and
4. retain our contact information in regard to use of the base software.

Permission to distribute the released version of the source code along with corresponding source modifications in the form of a patch file is granted with same provisions 2 through 4 for binary distributions. This software is provided "as is" without express or implied warranty to the extent permitted by applicable law.

This product contains software developed by Google, Inc. Copyright ©2011 Google, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This product includes software developed by Digital Envoy, Inc.

This product includes software developed by Jeremy Ashkenas and DocumentCloud, and distributed under the MIT license. Copyright © 2010-2013 Jeremy Ashkenas, DocumentCloud.

This product includes gson software, distributed under the Apache License version 2.0. Copyright © 2008-2011 Google Inc.

Chapter 1

Using Topology Load Balancing to Distribute DNS Requests to Specific Resources

- *How do I configure GTM to load balance DNS requests to specific resources?*
- *About Topology load balancing*
- *About Topology load balancing for a wide IP*
- *About Topology load balancing for a pool*
- *About Topology load balancing for both wide IPs and pools*
- *About IP geolocation data*

How do I configure GTM to load balance DNS requests to specific resources?

You can configure BIG-IP® Global Traffic Manager™ (GTM™) to load balance DNS requests to a resource based on the physical proximity of the resource to the client making the request. You can also configure GTM to deliver region-specific content, such as news and weather, to a client making a request from a specific location.

You can accomplish this by configuring GTM to perform Topology load balancing.

About Topology load balancing

Topology load balancing distributes DNS name resolution requests based on the proximity of the client to the data center housing the resource that responds to the request. When Topology load balancing is enabled, the BIG-IP® system uses topology records to make load balancing decisions.

Understanding topology records

A *topology record* is a set of characteristics that maps the origin of a DNS name resolution request to a destination. Each topology record contains the following elements:

- A *request source statement* that specifies the origin LDNS of a DNS request.
- A *destination statement* that specifies the pool or pool member to which the weight of the topology record will be assigned.
- A weight that the BIG-IP® system assigns to a pool or a pool member during the load balancing process.

Note: *In `tmsh`, the weight parameter is called score.*

Understanding user-defined regions

A *region* is a customized collection of topologies that defines a specific geographical location that has meaning for your network. For example, you can create two custom regions named Region_east and Region_west. Region_east includes the states on the east coast of the United States. Region_west includes the states on the west coast of the United States Then, you can use those custom regions as the Request Source or Destination of a topology record you create.

This table describes how the use of topology regions improves the load-balancing performance of the BIG-IP® system.

Faster load balancing configuration	Slower load balancing configuration
2 data centers	2 data centers
1000 pool members in each data center	1000 pool members in each data center
2 regions with 5000 CIDR entries each	
2 topology records:	10,000 topology records:
1 entry routes all requests from Region_east to data center1	5000 CIDR topology records route requests to data center1

Faster load balancing configuration	Slower load balancing configuration
1 entry routes all requests from Region_west to data center2	5000 CIDR topology records route requests to data center2

Creating a region for Topology load balancing

Create regions to customize the Topology load balancing capabilities of the BIG-IP® system. For example, you can create two regions to represent the data centers in your network: dc1_pools and dc2_pools. Alternatively, you can create a region to which you can add IP subnets as you expand your network. Then, when you create a topology record, you can use the custom regions as the Request Source or Destination of the record.

1. On the Main tab, click **DNS** > **GSLB** > **Topology** > **Regions**.
2. Click **Create**.
   The new record screen opens.
3. In the **Name** field, type a unique identifier for the region.
4. To add members to the region, do the following for each member you want to add to the region:
   a) From the **Member Type** list, select a type of identifier.
   b) Select an operator, either **is** or **is not**.
   c) From the **Continent** list, select the continent that contains the locations in the region you are creating.
   d) Click **Add**.
5. Click **Create**.

You can now create a topology record using the custom region you created.

Understanding how the BIG-IP system prioritizes topology records

When Topology load balancing is configured, the order of the topology records is vital and affects how the BIG-IP® system scores the pools or pool members to which it load balances DNS name resolution requests. By default, the BIG-IP system prioritizes topology records using Longest Match sorting. As a result, topology records are automatically sorted based on a specific criteria each time the BIG-IP system configuration loads. Alternatively, you can disable Longest Match sorting and customize the order of the topology records in the list.

Understanding Longest Match topology record sorting

When Longest Match is enabled, the BIG-IP® system sorts the topology records by the LDNS request source statement, the destination statement, and the weight of the record.

The system first sorts the topology records by the type of LDNS request source statement using this order from highest to lowest:

1. IP subnet in CIDR format (the system places the most specific IP subnet at the top of the list; for example, 10.15.1.1/32, 10.15.1.0/24, 10.15.0.0/16, 10.0.0.0/8)
2. Region
3. ISP
4. State
5. Country
6. Continent

7. LDNS Request Source negation (record that excludes an LDNS)
8. Wildcard records (the system sorts records that include a wildcard to the bottom of the list, because these records are the least specific)

If the type of LDNS request source statement is the same in multiple topology records, the BIG-IP system then sorts these records by the type of destination statement using this order from highest to lowest:

1. IP subnet in CIDR format (the system places the most specific IP subnet at the top of the list; for example, 10.15.1.1/32, 10.15.1.0/24, 10.15.0.0/16, 10.0.0.0/8)
2. Data center
3. Pool
4. Region (customized collection of criteria)
5. ISP
6. State
7. Country
8. Continent
9. Destination negation (record that excludes a destination)
10. Wildcard records (the system sorts records that include a wildcard to the bottom of the list, because these records are the least specific)

If the type of LDNS request source statement is the same in multiple topology records and the type of destination statement is the same in those records, the system then uses the value of the weight from lowest to highest to sort the records.

The example shows a list of topology records sorted automatically using Longest Match. Note that the fourth and fifth records have the same LDNS subnet and the destinations are both of type State. Therefore, the weight determines the position in the list; thus, the record with the lower weight is first.

1. ldns: subnet 192.168.69.133/32 destination: subnet 10.15.1.1/32 weight: 500
2. ldns: subnet 192.168.69.133/32 destination: datacenter /Common/NorthAmerica weight: 400
3. ldns: subnet 192.168.69.0/24 destination: pool /Common/NorthAmerica weight 300
4. ldns: subnet 192.168.0.0/16 destination: state WA weight 100
5. ldns: subnet 192.168.0.0/16 destination: state NY weight 200

Customizing the sort order of topology records

Determine the order in which you want the topology records you create to be sorted.

Change the sort order of the topology records when you do not want the system to use the Longest Match sort order.

1. On the Main tab, click **DNS** > **GSLB** > **Topology** > **Records**.
2. Click the **Change Order** button.
3. Clear the **Longest Match** check box.
4. To change the order of the records in the Topology Record List, do the following:
   a) From the list, select a topology record.
   b) Click the **Up** or **Down** button to move the record to the preferred position in the list.
5. Click **Update**.

The BIG-IP system uses the customized Topology Record List for topology load balancing.

Important: *The BIG-IP system saves only one set of ordered topology records; if you re-enable Longest Match, your custom ordering will no longer be available.*

Configuring Longest Match

Ensure that topology records exist in the configuration.

Configure the BIG-IP® system to order the topology records using Longest Match.

1. On the Main tab, click **DNS** > **GSLB** > **Topology** > **Records**.
2. Click the **Change Order** button.
3. Select the **Longest Match** check box.
4. Click **Update**.

The BIG-IP system uses Longest Match sorting to order the topology records in a list.

Creating a topology record

Before you create topology records, it is essential that you understand how the system sorts the topology record list. Additionally, you must understand how the system uses the ordered list of records to assign scores to the pools or pool members, to which the BIG-IP® system load balances DNS requests.

Create topology records that instruct the BIG-IP system where to route DNS name resolution requests when Topology load balancing is enabled.

Tip: *The BIG-IP system is more efficient when using regions for Topology load balancing.*

1. On the Main tab, click **DNS** > **GSLB** > **Topology**.
2. Click **Create**.
   The new record screen opens.
3. To create an LDNS request source statement, use the **Request Source** settings:
   a) Select an origin type from the first list.
   b) Select an operator, either **is** or **is not**.
   c) Define the criteria for the request source statement based on the request source type you selected.
4. To create a destination (server object) statement, use the **Destination** settings:
   a) Select a destination type from the first list.
   b) Select an operator, either **is** or **is not**.
   c) Define the criteria for the destination statement based on the destination type you selected.
5. In the **Weight** field, specify the priority of this record.
6. Click **Create**.

Deleting a topology record

Delete existing topology records as your network changes. For example, when you add a new data center to your network, the topology records that the BIG-IP® system uses to distribute DNS name resolution requests can become obsolete, requiring deletion.

Note: You cannot modify topology records; you can delete records and create new ones that meet your needs.

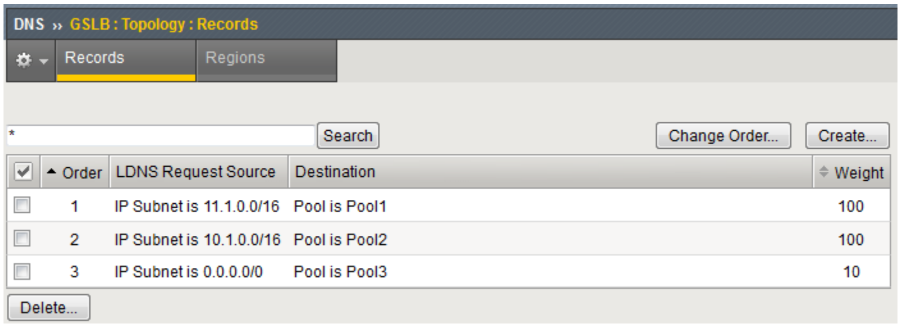
1. On the Main tab, click **DNS** > **GSLB** > **Topology**.
2. Select the topology record that you want to remove from the topology records list by selecting the corresponding **Select** check box.
3. Click **Delete**.
   A confirmation screen appears.
4. Click **Delete**.

About Topology load balancing for a wide IP

When you configure a wide IP for Topology load balancing, you can route DNS requests to the data center that is closest to the client making the request. With this configuration, the BIG-IP® system load balances DNS name resolution requests to the pools in a wide IP based on the topology records. (The members of each pool must be in the same data center.)

Example configuration: Topology load balancing for a wide IP

This example illustrates how DNS name resolution requests are load balanced when a wide IP is configured for Topology load balancing. An administrator configures the wide IP `www.siterequest.net` for Topology load balancing. The wide IP contains three pools: Pool1 and Pool3 are located in the North America data center; Pool2 is located in the South America data center. Next, the administrator creates topology records, as shown in this figure, and ensures that Longest Match is enabled on the BIG-IP® system.



☑	Order	LDNS Request Source	Destination	Weight
□	1	IP Subnet is 11.1.0.0/16	Pool is Pool1	100
□	2	IP Subnet is 10.1.0.0/16	Pool is Pool2	100
□	3	IP Subnet is 0.0.0.0/0	Pool is Pool3	10

Figure 1: Topology records for a wide IP configured for Topology load balancing

The first topology record directs all DNS name resolution requests from an LDNS in the IP subnet `11.1.0.0/16` to Pool1. The second topology record directs all DNS name resolution requests from an LDNS in the IP subnet `10.1.0.0/16` to Pool2. The third topology record is a wildcard. It directs DNS name resolution requests from an LDNS in any IP subnet to Pool3. However, it is important to note that the weight of the third topology record is lower than the weights of the other topology records.

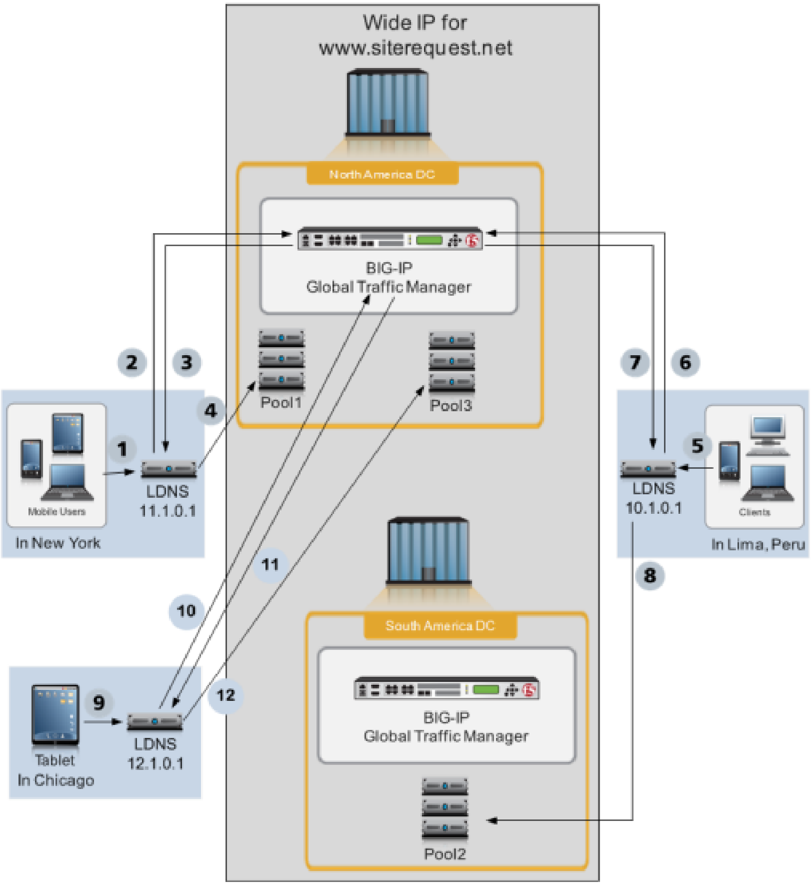


Figure 2: BIG-IP system load balancing DNS requests using a wide IP configured for Topology load balancing

1. A client in New York makes a DNS request.
2. LDNS 11.1.0.1 queries the BIG-IP system in the North America data center.
3. The BIG-IP system directs the LDNS to Pool1. To determine this answer, for each pool, one at a time, the BIG-IP system iterates through the list of two topology records to find a match. Pool1 matches the first topology record in the list, because both the LDNS request source (11.1.0.1) and the Destination (Pool1) of the DNS request match the first topology record; therefore, the BIG-IP system assigns a score of 100 to Pool1. For Pool2, there is no matching topology record that contains both the LDNS request source (11.1.0.1) and the Destination (Pool2); therefore, the BIG-IP system assigns a score of zero to Pool2. Pool3, matches the third topology record in the list, because both the LDNS request source (11.1.0.1) and the Destination (Pool3) of the DNS request match the third topology record; therefore, the BIG-IP system assigns a score of 10 to Pool3. The BIG-IP system directs the LDNS to send the request to the pool with the highest score.
4. The LDNS sends the DNS request to Pool1 in the North America data center. How the system distributes the DNS requests to the members of Pool1 is not depicted in this illustration, but is based on the load balancing method configured for Pool1.
5. A client in Lima makes a DNS request.
6. LDNS 10.1.0.1 queries the BIG-IP system in the North America data center.
7. The BIG-IP system directs the LDNS to Pool2. To determine this answer, for each pool, one at a time, the BIG-IP system iterates through the list of two topology records to find a match. For Pool1, there is not a matching topology record that contains both the LDNS request source (10.1.0.1) and the Destination (Pool1); therefore, the BIG-IP system assigns a score of zero to Pool1. Pool2 matches the second topology record in the list, because both the LDNS request source (10.1.0.1) and the Destination (Pool2) of the DNS request match the second topology record; therefore, the BIG-IP system assigns a score of 100 to

Pool2. Pool3, matches the third topology record in the list, because both the LDNS request source (10.1.0.1) and the Destination (Pool3) of the DNS request match the third topology record; therefore, the BIG-IP system assigns a score of 10 to Pool3. The BIG-IP system directs the LDNS to send the request to the pool with the highest score.

8. The LDNS sends the DNS request to Pool2 in the South America data center. How the system distributes the DNS requests to the members of Pool2 is not shown in this illustration, but is based on the load balancing mode configured for Pool2.
9. A client in Chicago makes a DNS request.
10. LDNS 12.1.0.1 queries the BIG-IP system in the North America data center.
11. The BIG-IP system directs the LDNS to Pool3. To determine this answer, for each pool, one at a time, the BIG-IP system iterates through the list of two topology records to find a match. For Pool1, there is not a matching topology record that contains both the LDNS request source (12.1.0.1) and the Destination (Pool1); therefore, the BIG-IP system assigns a score of zero to Pool1. For Pool2, there is not a matching topology record that contains both the LDNS request source (12.1.0.1) and the Destination (Pool1); therefore, the BIG-IP system assigns a score of zero to Pool2. Pool3, matches the third topology record in the list, because both the LDNS request source (12.1.0.1) and the Destination (Pool3) of the DNS request match the third topology record; therefore, the BIG-IP system assigns a score of 10 to Pool3. The BIG-IP system directs the LDNS to send the request to the pool with the highest score.
12. The LDNS sends the DNS request to Pool3 in the North America data center. How the system distributes the DNS requests to the members of Pool3 is not depicted in this illustration, but is based on the load balancing mode configured for Pool3.

Configuring a wide IP for Topology load balancing

Before you configure a wide IP for Topology load balancing, ensure the following:

- At least two pools are associated with the wide IP that you are configuring for Topology load balancing.
- Topology records that define how you want the BIG-IP® system to load balance DNS name resolution requests are configured.

You can use Topology load balancing to distribute DNS name resolution requests among the pools in a wide IP based on the geographic location of both the client making the request and the pool that handles the response.

1. On the Main tab, click **DNS** > **GSLB** > **Wide IPs**.
   The Wide IP List screen opens.
2. Click the name of the wide IP you want to modify.
3. On the menu bar, click **Pools**.
4. From the **Load Balancing Method** list, select **Topology**.
5. Click **Update**.

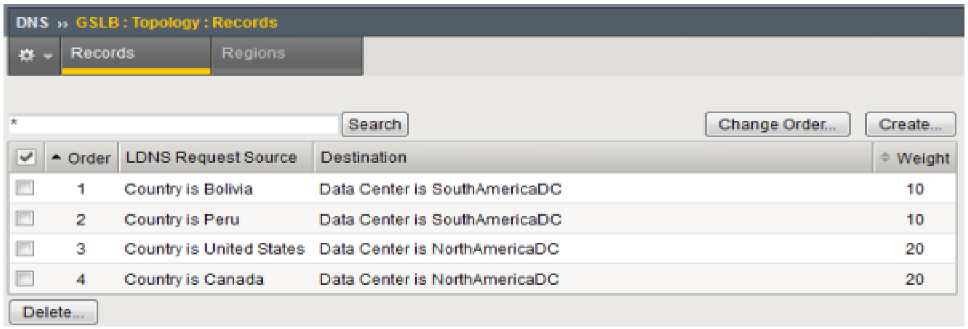
Repeat this process for each wide IP that you want to configure for Topology load balancing.

About Topology load balancing for a pool

When you configure a pool for Topology load balancing, you can route DNS requests to the data center that is closest to the client making the request. With this configuration, the BIG-IP® system load balances DNS name resolution requests to the members of the pool.

Example configuration: Topology load balancing for a pool

This example illustrates how DNS name resolution requests are load balanced when a pool is configured for Topology load balancing. An administrator configures pools in two different data centers: the North America data center (North America DC) and the South America data center (South America DC) for Topology load balancing. A server that contains the pool members 10.10.10.1 - 10.10.10.3 resides in the North America DC. The server that contains the pool members 11.10.10.1 - 11.10.10.3 resides in the South America DC. Next, the administrator creates topology records, as shown in the following figure, to load balance DNS requests to members of the pools, and ensures that Longest Match is enabled on the BIG-IP® system.



*

☑	Order	LDNS Request Source	Destination	Weight
□	1	Country is Bolivia	Data Center is SouthAmericaDC	10
□	2	Country is Peru	Data Center is SouthAmericaDC	10
□	3	Country is United States	Data Center is NorthAmericaDC	20
□	4	Country is Canada	Data Center is NorthAmericaDC	20

Figure 3: Topology records for a pool configured for Topology load balancing

The first topology record directs all DNS name resolution requests from an LDNS in Bolivia to the South America DC. The second topology record directs all DNS name resolution requests from an LDNS in Peru to the South America DC. The third topology record directs all DNS name resolution requests from an LDNS in the United States to the North America DC. The fourth topology record directs all DNS name resolution requests from an LDNS in Canada to the North America DC.

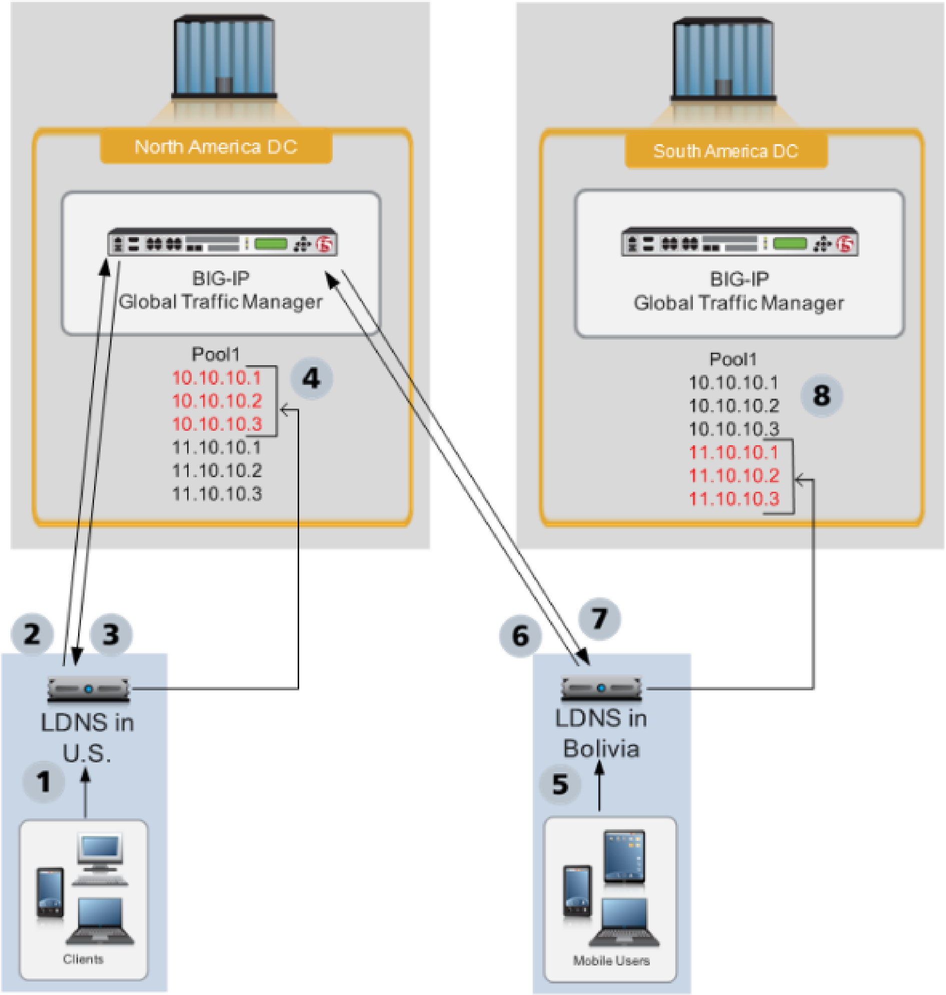


Figure 4: Pool configured for Topology load balancing

1. A client in the U.S. makes a DNS request.
2. An LDNS in the U.S. queries the BIG-IP system in the North America DC.
3. The BIG-IP system directs the LDNS to a member of Pool1 in the North America DC. To determine this answer, for each pool member, one at a time, the BIG-IP system iterates through the list of topology records to find a match. Pool members 10.10.10.1 - 10.10.10.3 each match the third topology record in the list, because both the LDNS request source (U.S.) and the Destination (North America DC) of the DNS request match the third topology record; therefore, the BIG-IP system assigns a score of 20 to each of those pool members. For each of the pool members 11.10.10.1 - 11.10.10.3, there is no matching topology record that contains both the LDNS request source (U.S.) and the Destination (South America DC); therefore, the BIG-IP system assigns a score of zero to each of those pool members. The BIG-IP system directs the LDNS to send the request to the pool member with the highest score.
4. The LDNS sends the DNS request to a pool member in the North America DC. Because all of the pool members in the North America DC have the same score, the system distributes the DNS requests to the pool members in a round robin fashion.
5. A client in Bolivia makes a DNS request.
6. An LDNS in Bolivia queries the BIG-IP system in the North America DC.
7. The BIG-IP system directs the LDNS to a pool member in the South America DC. To determine this answer, for each pool member, one at a time, the BIG-IP system iterates through the list of topology records to find a match. For each of the pool members 10.10.10.1 - 10.10.10.3 there is no matching

topology record that contains both the LDNS request source (Bolivia) and the Destination (North America DC); therefore, the BIG-IP system assigns a score of zero to each of those pool members. Pool members 11.10.10.1 - 11.10.10.3 each match the first topology record in the list, because both the LDNS request source (Bolivia) and the Destination (South America DC) of the DNS request match the first topology record; therefore, the BIG-IP system assigns a score of 10 to each of those pool members. The BIG-IP system directs the LDNS to send the request to the pool member with the highest score.

8. The LDNS sends the DNS request to a pool member in the South America DC. Because all of the pool members in the South America DC have the same score, the system distributes the DNS requests to the pool members in a round robin fashion.

Configuring a pool for Topology load balancing

Before you configure a pool for Topology load balancing, ensure the following:

- The pool you are configuring for Topology load balancing contains at least two pool members.
- Topology records that define how you want the BIG-IP® system to load balance DNS name resolution requests are configured.

You can use Topology load balancing to distribute DNS name resolution requests among the members of a pool based on the geographic location of both the client making the request and the member of the pool that handles the response.

1. On the Main tab, click **DNS** > **GSLB** > **Pools**.
   The Pools list screen opens.
2. Click the name of the pool you want to modify.
3. On the menu bar, click **Members**.
4. In the Load Balancing Method area, from the **Preferred** list, select **Topology**.
5. In the Load Balancing Method area, from the **Alternate** list, select **Round Robin**.
6. In the Load Balancing Method area, from the **Fallback** list, select **None**.
7. Click **Update**.

Repeat this process for each pool that you want to configure for Topology load balancing.

About Topology load balancing for both wide IPs and pools

You can configure a wide IP for Topology load balancing. You can also configure each pool in the wide IP for Topology load balancing. When you configure both a wide IP and the pools in the wide IP for Topology load balancing, the BIG-IP® system uses topology records to load balance DNS name resolution requests first to a pool in the wide IP, and then, to a member of the pool.

Note:

When configuring both the wide IP and the pools in the wide IP for Topology load balancing, it is important to set the **Fallback** load balancing method for each pool to **None**. If you do not, GTM™ can send a DNS request to a pool in the wide IP even when no pool members are available. In this case, the load balancing algorithm for the pool would then fall back to BIND (static DNS). When you set the **Fallback** load balancing method for each pool to **None**, if no members of a pool are available, GTM sends the DNS request to another pool in the wide IP.

About IP geolocation data

The BIG-IP® system uses an IP geolocation database to determine the origin of DNS requests. The database included with the BIG-IP system provides geolocation data for IPv6 addresses at the continent and country levels. It also provides geolocation data for IPv4 addresses at the continent, country, state, ISP, and organization levels. The state-level data is worldwide, and thus includes designations in other countries that correspond to the U.S. state-level in the geolocation hierarchy, such as, provinces in Canada.

Note: *If you require geolocation data at the city-level, contact your F5® Networks sales representative to purchase additional database files.*

About topology records and IP geolocation data

The BIG-IP® system uses an IP geolocation database to determine the IP addresses that match the geographic names that you define in a topology record, such as continent and country.

Downloading and installing updates to the IP geolocation data

You can download a monthly update to the IP geolocation database from F5® Networks. The BIG-IP® system uses the IP geolocation database to determine the origin of DNS name resolution requests.

1. Log in to the F5 Networks customer web site at `http://downloads.f5.com`, and click **Find a Download**.
2. In the F5 Product Family column, find **BIG-IP**, and then in the Product Line column, click either **BIG-IP v11.x/Virtual Edition**.
3. Select a version from the list preceding the table.
4. In the Name column, click **GeolocationUpdates**.
5. Click **I Accept** to accept the license.
6. In the Filename column, click the name of the most recent compressed file that you want to download.
7. In the Ready to Download table, click the download method that you want to use.
8. In the dialog box, click **OK**.
9. Select the directory in which you want to save the compressed file, and then decompress the file to save the RPM files on the system.
10. To install and load one of the RPM files, run this command (the path and file name are case-sensitive):

    `geoip_update_data -f </path to RPM file and file name >`.

    The system installs and loads the specified database file.
11. Repeat step 10 for each of the RPM files that you saved to the system in step 9.

You can access the ISP and organization-level geolocation data for IPv4 addresses only using the iRules® `whereis` command.

Reloading default geolocation data using the Configuration utility

Before you reload the default geolocation data, delete the RPM files that are in the `/shared/GeoIP` directory.

To uninstall an update to the IP geolocation database, reload the default geolocation database files using the Configuration utility.

1. At the BASH prompt, run this command to query the RPM database and determine what geolocation data is installed:

   `rpm -qa --dbpath /shared/lib/rpm/`

   The system returns a list of RPMs, for example:

   ```
   geoip-data-ISP-1.0.0-20110203.61.0
   geoip-data-Region2-1.0.0-20110203.61.0
   geoip-data-Org-1.0.0-20110203.61.0
   ```

2. To uninstall the RPMs, run this command for each RPM in the list:

   `rpm -e --dbpath /shared/lib/rpm/ <name of file>`

   For example, to uninstall `geoip-data-ISP-1.0.0-20110203.61.0`, run this command: `rpm -e --dbpath /shared/lib/rpm/ geoip-data-ISP-1.0.0-20110203.61.0`

3. To remove the symlink in the `/shared/GeoIP` directory, run this command:

   `rm -f /shared/GeoIP/*`

4. Log on to the Configuration utility.
5. On the Main tab, click **System** > **Configuration**.
6. In the Geolocation area, click **Reload** in the **Operations** setting.
   The system reloads the default geolocation database files that are stored in `/usr/share/GeoIP`.

Reloading default geolocation data using tmsh

To uninstall an update to the IP geolocation database, delete the RPM files, and then reload the default geolocation database files using `tmsh`.

1. At the BASH prompt, to query the RPM database and determine what geolocation data is installed, run this command:

   `rpm -qa --dbpath /shared/lib/rpm/`

   The system returns a list of RPMs, for example:

   ```
   geoip-data-ISP-1.0.0-20110203.61.0
   geoip-data-Region2-1.0.0-20110203.61.0
   geoip-data-Org-1.0.0-20110203.61.0
   ```

2. To uninstall the RPMs, for each RPM in the list, run this command:

   `rpm -e --dbpath /shared/lib/rpm/ <name of file>`

   For example, to uninstall `geoip-data-ISP-1.0.0-20110203.61.0`, run this command: `rpm -e --dbpath /shared/lib/rpm/ geoip-data-ISP-1.0.0-20110203.61.0`

3. To remove the symlink in the `/shared/GeoIP` directory, run this command:

   `rm -f /shared/GeoIP/*`

4. Log on to `tmsh`.
5. Run this command:

   ```
   load / sys geoip
   ```

   The system reloads the default geolocation database files that are stored in `/usr/share/GeoIP`.

Chapter 2

Preparing BIG-IP GTM for Static Persist Load Balancing for Transaction-Oriented Traffic

- *About preparing GTM to load balance transactions to a single transaction manager*

About preparing GTM to load balance transactions to a single transaction manager

You can configure BIG-IP® Global Traffic Manager™ (GTM™) to provide static persistence of local DNS servers to virtual servers. You can accomplish this by configuring GTM to perform Static Persist load balancing. With this configuration, GTM guarantees that certain transactions are routed through a single transaction manager (for example, Local Traffic Manager™ or other server array manager). This is beneficial for transaction-oriented traffic, such as e-commerce shopping carts, online trading, and online banking.

About Static Persist load balancing

The Static Persist load balancing method uses the persist mask with the source IP address of the LDNS in a deterministic algorithm to send requests to a specific pool member (virtual server). Using this method, BIG-IP® GTM™ sends DNS name resolution requests to the first available pool member based on a hash algorithm that determines the order of the pool members. This algorithm orders the pool members differently for each LDNS that is sending requests to BIG-IP GTM, taking into account the Classless Inter-Domain Routing (CIDR) of the LDNS. As BIG-IP GTM distributes requests across all pool members, requests from each LDNS (and thus, each client) are generally sent to the same pool member. When the selected pool member becomes unavailable, BIG-IP GTM sends requests to another pool member. When the original pool member becomes available again, BIG-IP GTM sends requests to that pool member.

Configuring CIDR options for Static Persist load balancing

The Static Persist load balancing mode uses the Classless Inter-Domain Routing (CIDR) system to identify which IP addresses require a persistent connection to a virtual server. You can configure the range of IP addresses by configuring the Static Persist CIDR setting for IPv4 and IPv6 addresses.

1. On the Main tab, click **DNS** > **Delivery** > **Settings** > **GSLB** > **Load Balancing**.
   The GSLB Load Balancing configuration screen opens.
2. For the **Static Persist CIDR (IPv4)** setting, type the number of bits that determine the IP address range for IPv4 addresses.

   Note: *The default value is **32**.*

3. For the **Static Persist CIDR (IPv6)** setting, type the number of bits that determine the IP address range for IPv6 addresses.

   Note: *The default value is **128**.*

4. Click **Update**.

You can now configure a pool for Static Persist load balancing.

Index

C

CIDR
- configuring for Static Persist load balancing *26*

D

database
- installing geolocation *22*

G

geolocation data
- about *22*
- and topology records *22*
- downloading and installing updates *22*
- reloading default data *23*
- reloading default data using tmsh *23*

I

IP geolocation data
- and topology records *22*
- downloading and installing updates *22*

L

load balancing
- about Topology *12*
- and sort order of topology records *13*

load balancing methods
- Static Persist *26*

load balancing modes
- about Static Persistence *26*
- about Topology *12*

Longest Match sorting
- and topology records *13*
- disabling *14*
- enabling *15*

P

pools
- and example Topology load balancing configuration *19*
- and Topology load balancing *21*
- and Topology load balancing mode *18*

R

regions
- and Topology load balancing *12*
- creating for Topology load balancing *13*

S

Static Persist load balancing
- and configuring CIDR options *26*

Static Persist load balancing method
- about *26*

Static Persist load balancing mode *26*

T

tmsh, and reloading default geolocation data *23*

Topology load balancing
- about *12*
- and an example for pools *19*
- and an example for wide IPs *16*
- and Longest Match sorting of records *13*
- and pools *21*
- and regions *12*
- and wide IPs *16*, *18*

Topology load balancingand creating regions *13*

Topology load balancing mode *12*

topology records
- about *12*
- and IP geolocation data *22*
- and Longest Match sorting *13*
- changing sort order *14*
- configuring Longest Match sort order *15*
- creating *15*
- deleting *15*

U

user-defined regions
- about *12*
- creating for Topology load balancing *13*

W

wide IPs
- and example Topology load balancing configuration *16*
- and Topology load balancing *16*, *21*