

Selective DNS Persistence on GTM



Colin Walker, 2008-24-06

The world of DNS is an interesting one to me. So much of what transpires on the internet these days relies on it, heavily, to connect from point A to B with ease, and yet it remains relatively unchanged, compared to the more complex and rapidly advancing application layer. Just about anything you do on the internet, from making an HTTP connection, receiving or sending mail, or talking to your friends over IM make use of DNS first to translate the human readable host name into a machine usable IP address. This is the same simple, straight-forward function that DNS has been serving for years.

As applications continually grow more complex and wide-spread, however, there is bound to be change even to foundational pieces of an environment such as DNS. In today's network it's not uncommon for a single application to span multiple data centers, possibly across multiple continents. Obviously the different systems in these widely separated networks won't have the same IP addresses, and as such DNS needs to be able to be intelligently switched between them, depending on any number of things from the location of the client making the request, to the health of the systems in a given DC.

That's been in place for a while, now. Global load balancing is what the GTM device at F5 is all about. We've been doing it for years, and there are plenty of bells and whistles that go along with that product. One such feature that's a little more recently available, is iRules. Admittedly it's a sub-set of the full iRules language available on the LTM, but it's still a very powerful tool when you're looking to do things that may be outside the box, such as, say, selective persistence at the DNS level.

What is DNS persistence? Well, it's very similar to normal stickiness or persistence that you'd deal with at an application level, so a given client always returns to the same server. The only real difference is that, since we're working at the DNS level, all we're doing is ensuring that a given client request gets sent to a specific IP address. What happens from there on out is the responsibility of the DC infrastructure and applications. By turning persistence on for some DNS requests we can ensure that the client making the request will continue returning to that same IP address for a configurable amount of time, theoretically without having to re-ask the nameservers for resolution.

I've run into a couple of situations where this kind of selective persistence could be extremely useful. Take, for instance, a deployment where there are many normal applications that we want people to operate normally with, requesting DNS based on a TTL and being served IP addresses via the GTM based on all the criteria we define in our Wide-IP ... normal behavior. Now add in a client application or two that needs to make an extremely long term connection to a back-end resource. Or even more importantly, an application that is going to make many requests over a long period of time, that you want to ensure is sent to the exact same node in the same DC, regardless of what other variables might change that would normally route that client to a different DC. All of a sudden selective DNS persistence sounds pretty useful, doesn't it?

Luckily this is quite straight-forward on the GTM with a simple iRule and a couple of Wide-IPs configured. As you can see in the below example, all you need to do is set out the cases in which you want to specifically point to the Wide-IP that has persistence enabled (or doesn't, depending on which is the more used resource) in an if statement, then use the cname command to ensure that the appropriate Wide-IP is used for the response. Take a look at the rule and two example Wide-IPs below:

```
rule "SelectivePersist-rule" {
  when DNS_REQUEST {
    if { [IP::addr [IP::client_addr]/16 equals "10.1.0.0" ] \
      or [IP::addr [IP::client_addr]/16 equals "10.2.0.0" ] } {
      cname "myService-noPersist.gtm.test.com"
    }
  }
}
```

```
}  
}  
  
wideip {  
    name          "myService.gtm.test.com"  
    persist       yes  
    persist_ttl   1800  
    pool_lbmode   rr  
    pool          "myPool"  
    rule          "SelectivePersist-rule"  
}  
  
wideip {  
    name          "myService-noPersist.gtm.test.com"  
    pool_lbmode   rr  
    pool          "myPool"  
}
```

Thanks on this example go to citizen_elah who submitted this example to the iRules codeshare. Check out the original example [here](#). Thanks a bunch elah, and keep up the good work!

Commands used: [cname](#) ; [IP::addr](#) ; [IP::client_addr](#) ; [DNS_REQUEST](#)

[Get the Flash Player](#) to see this player.
[20080624-SelectiveDNSPersistence.mp3](#)

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | [f5.com](#)

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com

©2018 F5 Networks, Inc. All rights reserved. F5, F5 Networks, and the F5 logo are trademarks of F5 Networks, Inc. in the U.S. and in certain other countries. Other F5 trademarks are identified at [f5.com](#). Any other products, services, or company names referenced herein may be trademarks of their respective owners with no endorsement or affiliation, express or implied, claimed by F5. CS04-00015 0113