LoadBalanceWithoutStickyCookie

Load balancing with appservers who set a bad sticky cookie

Some application servers, or even dumb backends that only speak HTTP do not set a cookie that can be used to track user experience. Some times, though, these applications do require session persistence to the backend that created it. As an Apache HTTPD administrator, how can we give these developers what they want without requiring code changes to the backend?

Solution

You will need:
*mod_proxy
*mod_proxy_http
*mod_proxy_balancer
*mod_headers

And this configuration:

```
<Proxy balancer://Application_cluster>
   BalancerMember http://host1:80 route=member0
   BalancerMember http://host2:80 route=member1
   BalancerMember http://host3:80 route=member2
   ProxySet stickysession=Application_STICKY
</Proxy>
Header add Set-Cookie "Application_STICKY=sticky.%{BALANCER_WORKER_ROUTE}e;path=/;" env=BALANCER_ROUTE_CHANGED
```

Wait a minute.. what is going on here?

We are creating a load balancer using mod_proxy_balancer with the Proxy block. Inside that balancer we are creating three members and telling the balancer to look for a cookie called "Application_STICKY" as the cookie in question. This is pretty straight forward stuff.

The next line, however, is asking mod_headers to...
*Add a header named "Set-Cookie"
*The cookie is called "Application_STICKY" (remember, mod_proxy_balancer is looking for that name)
*The cookie value is "sticky." appended by the route of the member that served it
*The path of the cookie is set to "/", but can be more restrictive based on context-root
*And ONLY do this if mod_proxy_balancer has set the environment entry named "BALANCER_ROUTE_CHANGED"

This works perfectly because:
*On the first visit, mod_proxy_balancer will not detect the "Appliction_STICKY" cookie. This causes it to create the "BALANCER_ROUTE_CHANGED" environment entry
*Before responding to the user, mod_headers notices the "BALANCER_ROUTE_CHANGED" env entry and sets the cookie
*The user agent will then send the cookie back on the next request
*On the subsequent requests, mod_proxy_balancer identifies the cookie and route, and does not set the "BALANCER_ROUTE_CHANGED" cookie
*And since "BALANCER_ROUTE_CHANGED" is not set, mod_headers does not overwrite the cookie during that request cycle.