BIG-IP LTM and Microsoft App-V

Ryan Korock, 2011-14-10

Over in the DevCentral Microsoft forums there has been quite a bit of noise around deploying BIG-IP LTM to provide load balancing for Microsoft's Application Virtualization (App-V). Attempting to live up to a commitment I made, here is some guidance on how to configure App-V load balancing using the BIG-IP.

Architecture:

There are several ways to architect the network for App-V, and this is one of the few instances in which F5 advocates the use of deploying with Direct Server Return (also known as DSR or nPath). In traditional load balancing implementations, both incoming client traffic and the return server traffic flow through the load balancer. With DSR, the incoming client traffic flows through the load balancer and to the application server, however the return traffic is routed around the load balancer and sent directly to the client. Since App-V relies on protocols that utilize a simple request in & large stream back model, this architecture has the benefit of eliminating the impact that the large amount of streaming traffic would have on your load balancer.

When implementing DSR, we're actually using asymmetric routing. Because of this, we need to make a few tweaks to the App-V Server & BIG-IP configurations. We need to let the BIG-IP know that it will only see one-half of the TCP connections, and need to configure the application servers to respond back to the client with the source IP address of the load balancers Virtual IP (VIP), instead of its own IP address. We can force the application servers to do this by binding the VIP address to their loopback adaptors. This step is very important, as without it, the connection will return to the client with destination/source IPs that don't match the original client connection. For a graphical explanation, take a look at the pic below.
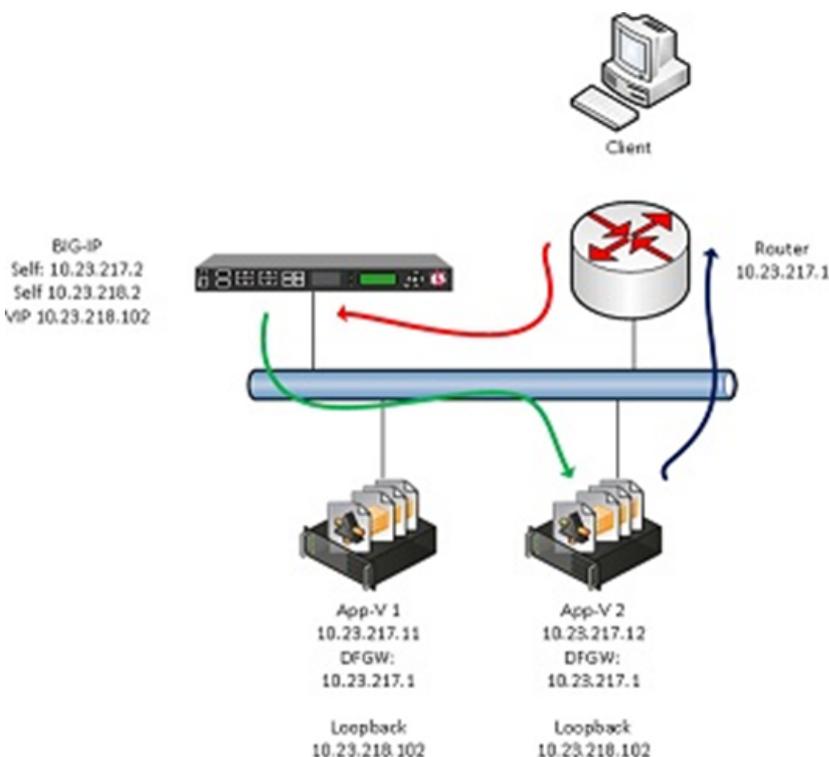
The Client generates the initial connection to the BIG-IP VIP (10.23.218.102)

The BIG-IP then selects an App-V server (10.23.217.12) and forwards the connection to the server.

The App-V server then sends the response to the Client directly through the default gateway router.

It's important to note that DSR is not required. It's just a recommended option. You can go with a traditional routed/snat'd configuration with both incoming and outgoing traffic flowing through the load balancer and this will actually work fine. It will actually reduce the complexity a bit (no loopback adaptors needed on the application servers); the only drawback being that you'll be sending a lot of traffic through the load balancer via the streaming.

If you want to read up on Direct Server Return (nPath) on the BIG-IP, take a look here -> http://support.f5.com/kb/en-us/products/big-ip_ltm/manuals/product/ltm_implementation/sol_npath.html?sr=17063074

The next portion of this post will focus on the actual configuration. The following assumes a basic understanding of the BIG-IP, and that you have your App-V servers configured as a farm. These instructions cover the load balancing of the APP-V publishing and streaming RTSP(S) services.
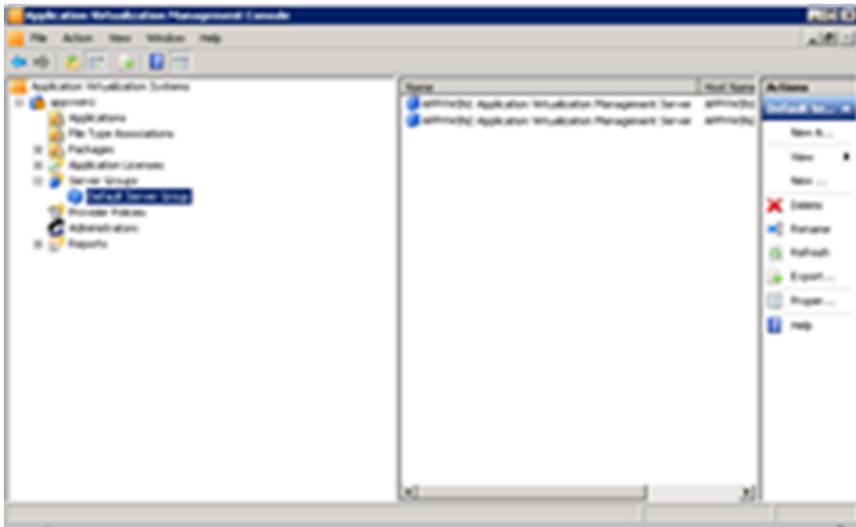
APP-V Configuration Notes

1. When you sequence the application, make sure that the hostname for the deployment package is set to the hostname of the load balancer VIP. In my case, I used appvserver.appv.f5demo.net
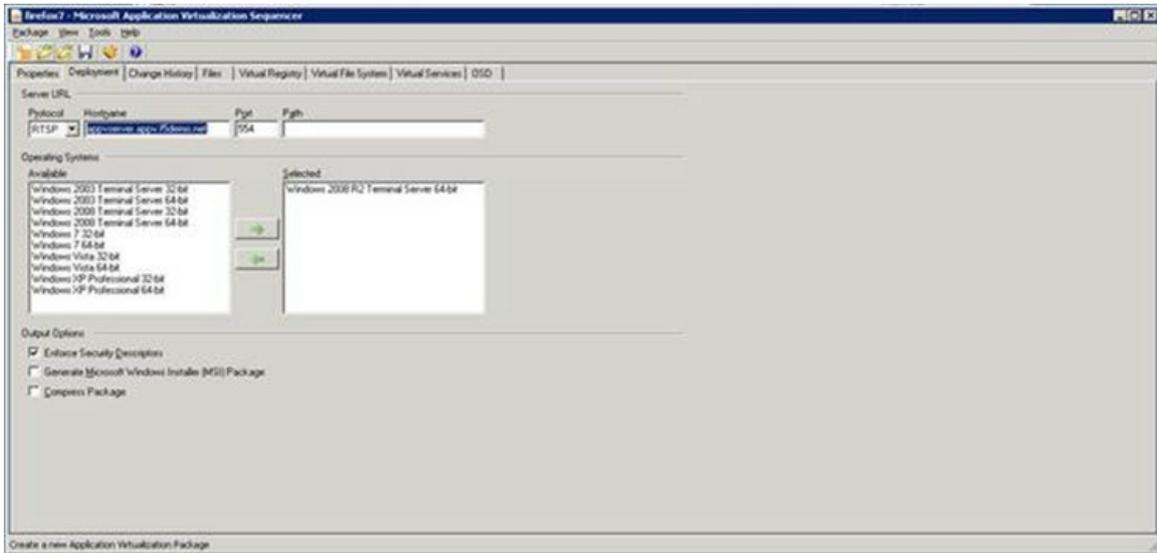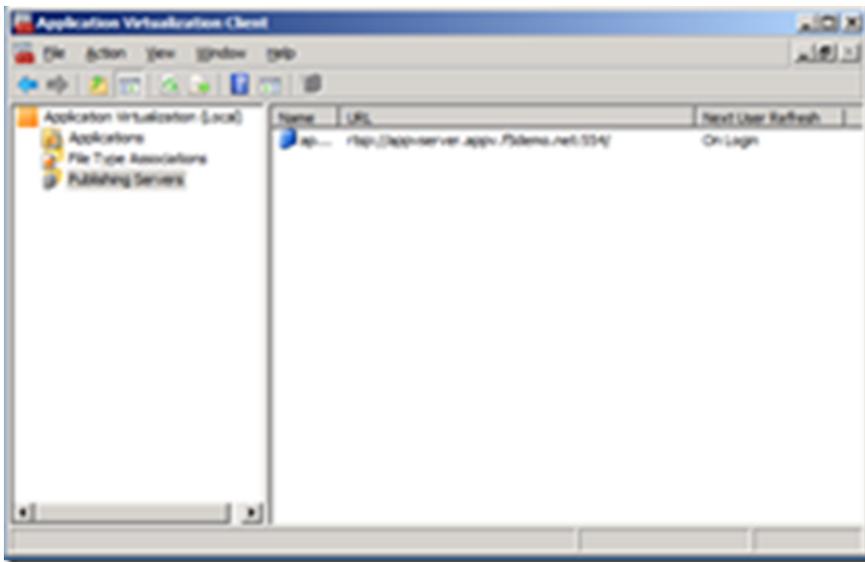

2. On the App-V servers themselves, make sure that the content path is also refers to the hostname of the VIP. In my case, I used \\appvserver.appv.f5demo.net\content\
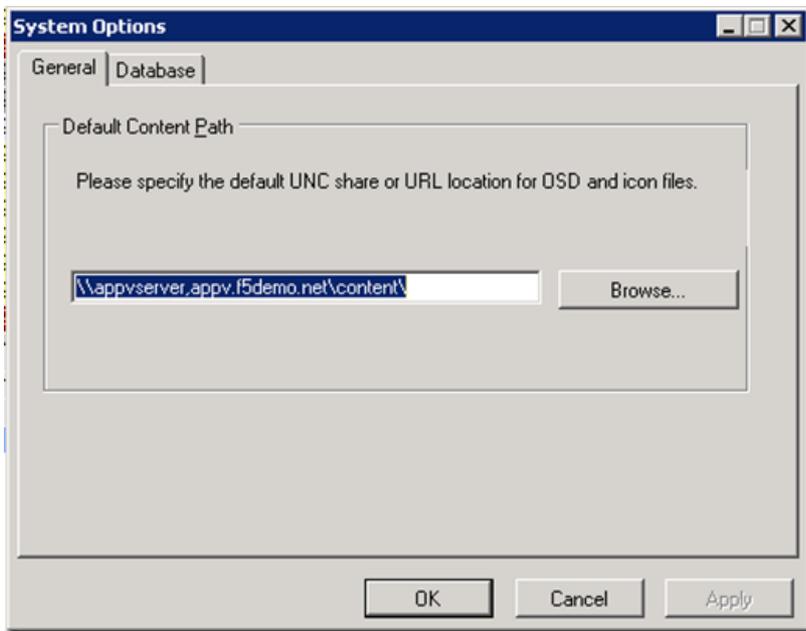
3. Ensure that the App-V Servers are added to the Server Group in the Management Console

4. On the App-V client, enter in the hostname of the VIP address as the Publishing Server. In this case, I use appvserver.appv.f5demo.net

App-V Server configuration for Direct Server Return

Now you will need to configure the App-V servers to send response traffic using a source IP that matches the VIP address. If you are using Windows Server 2008, you will need to follow the second step below.

1. Bind the VIP IP address to the loopback adaptor of the App-V servers you are load balancing.

Hint: On Windows 2008 R2, you'll need to add a loopback adaptor. Use the device manager to 'add a legacy device'. Select loopback adaptor to install it, and then assign it the same IP that belongs to your BIG-IP VIP.
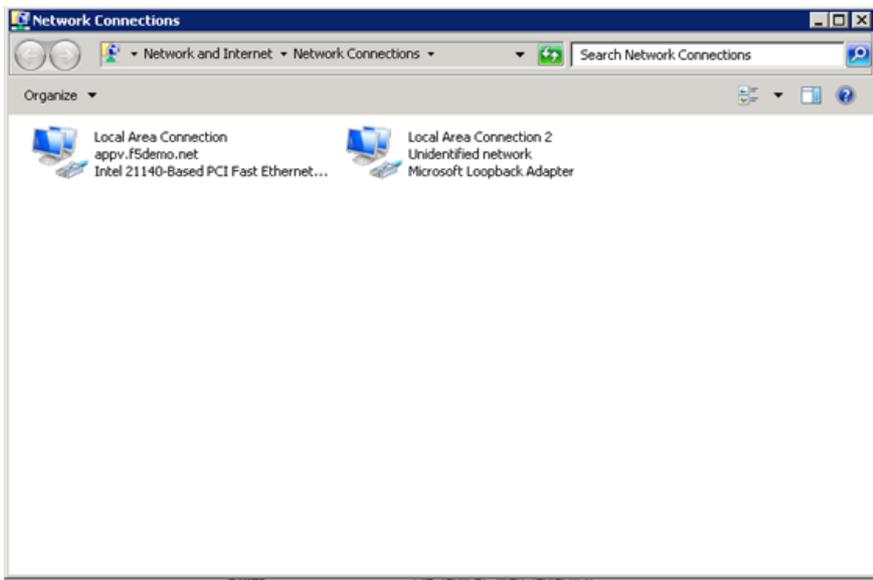
2. We are effectively asking the App-V server to accept (and send) traffic on an external interface for an IP address that is bound to the loopback adaptor. While this works natively on Windows 2003, Windows 2008 adopted a strong host model (RFC 1122) that prohibits one network interface accepting/sending traffic on behalf of another.

In order to get DSR to work with Windows 2008, you must re-enable the weak host model.Assuming your physical NIC is "Local Area Connection" and your loopback adaptor is "Local Area Connection 2", to enable weak host you must issue the following

netsh interface ipv4 set interface "Local Area Connection" weakhostreceive=enabled

netsh interface ipv4 set interface "Local Area Connection 2" weakhostreceive=enabled

netsh interface ipv4 set interface "Local Area Connection 2" weakhostsend=enabled

For more details on this, read RFC 1122, or this great article - http://technet.microsoft.com/en-us/magazine/2007.09.cableguy.aspx

Now onto the BIG-IP configuration….

BIG-IP Configuration

In our configuration, we will create 3 VIPs, all of which use the same IP address (just different ports). This is the IP that all App-V clients will send their publishing and streaming requests to. This is the IP address that appvserver.appv.f5demo.net resolved to in my example.The 3 VIP/Pools will use the following

Port 332 for RTSPS traffic (or Port 554 if you are using RTSP)

Port 445 for file transfers

Port 0 (all ports) for the RTP/RTCP traffic

A client will initiate the connection via RTSPS or RTSP, but will follow that up with file transfer and RTP/RTCP connections. We will want all of these follow on connections to be sent to the same server that the original connection was sent to. So we will use Source IP persistence for all 3 VIPs, and select 'persist across services' so that the persistence is kept across all 3 VIPs.

We will also create a custom fastl4 profile that will assist with the Direct Server Return. By enabling 'Loose Close' and upping the Idle Timeout, we cover for the fact BIG-IP only gets to see one side of the TCP connection.

Below are the actual steps needed to take to configure the BIG-IP.

Step 1. Pool Creation

In this step you will create 3 new pools for the 3 different types of traffic we will be load balancing. All 3 pools will have the same App-V server members; they'll just have different ports assigned to them.

Health Monitors Port 0 : Gateway ICMP

Port 445: TCP

Port 332 or 554: TCP

The graphic below is should match your config if you are using RTSP. If you are using RTSPS you will have a port 332 pool instead of port 554.

✓	Status	⬦ Pool/Member	Partition / Path
☐	🟢	appv_0_pool	Common
☐	🟢	-- 10.23.217.11:0	Common
☐	🟢	-- 10.23.217.12:0	Common
☐	🟢	appv_ds_pool	Common
☐	🟢	-- 10.23.217.11:445	Common
☐	🟢	-- 10.23.217.12:445	Common
☐	🟢	appv_rtsp_pool	Common
☐	🟢	-- 10.23.217.11:554	Common
☐	🟢	-- 10.23.217.12:554	Common

Step 2. L4 Profile Creation

For this implementation, you will want to create a custom "Fast L4" profile based upon the original fastL4 parent profile. Name this new profile "appvfastl4" and set the following custom settings

Idle timeout 1800 seconds

Loose Close Enabled

Step 3. Persistence Profile Creation

You will need to create a custom persistence profile based upon the original "source_addr" parent profile. Name this new profile "appv_source_pers" and set the following custom settings

Match Across Services Enabled

Timeout 1800 seconds

Step 4. VIP Creation

Now you will create the 3 VIPs to match up with your 3 Pools. These are standard VIPs with the following custom settings

The VIP type needs to be set to "Performance (Layer 4)"

The Protocol Profile needs to be set to "appvfastl4"

Address Translation option needs to be cleared (disabled)

Port Translation option needs to be cleared (disabled)

The Persistence Profile needs to be set to "appv_source_pers"

Port 0 VIP needs to point to your Port 0 Pool

Port 445 VIP needs to point to your Port 445 Pool

Port 332 VIP needs to point to your Port 332 Pool (if RTSPS used)

Port 554 VIP needs to point to your Port 554 Pool (if RTSP used)

		Status	▲ Name		≡ Application	≡ Destination	≡ Service Port	≡ Type	Resources
☐	○		appv_0_vs			10.23.218.102	0 (Any)	Performance (Layer 4)	Edit..
☐	○		appv_445_vs			10.23.218.102	445	Performance (Layer 4)	Edit..
☐	○		appv_554_vs			10.23.218.102	554	Performance (Layer 4)	Edit..

You should be good to go! Test with the client and give it a try!

Troubleshooting:

I plan on adding to this section as I find common setup issues and troubleshooting methods.

One of the common set issues regards the use of SNAT. If you are attempting to use Direct Server Return, SNAT will break DSR. Make sure that no VIP SNAT or Global SNATs are enabled.

One of the best troubleshooting tools is the BIG-IP Pool Statistics Page. If you look at graphic below, you'll notice a few things

Bits/Packets In counters are incrementing: That means traffic is being sent from the BIG-IP to the App-V servers. Good!

Bits/Packets Out counters are stuck at 0: That means that the BIG-IP is not seeing return traffic. This is also a good thing when using DSR!!

All traffic from our test client is being sent to the .11 node across all the pools. This means persistence is working. Good!!

Display Options

Statistics Type	Pools ▼
Data Format	Normalized ▼
Auto Refresh	Disabled ▼ Refresh

					Bits		Packets		Connections		
✓	Status	◆ Pool/Member	Partition / Path		In	Out	In	Out	Current	Maximum	Total
☐	○	appv_0_pool	Common		3.3M	0	9.8K	0	0	2	2
☐	○	-- 10.23.217.11:0	Common		3.3M	0	9.8K	0	0	2	2
☐	○	-- 10.23.217.12:0	Common		0	0	0	0	0	0	0
☐	○	appv_ds_pool	Common		63.3K	0	44	0	0	1	2
☐	○	-- 10.23.217.11:445	Common		63.3K	0	44	0	0	1	2
☐	○	-- 10.23.217.12:445	Common		0	0	0	0	0	0	0
☐	○	appv_rtsp_pool	Common		211.0K	0	69	0	0	4	9
☐	○	-- 10.23.217.11:554	Common		211.0K	0	69	0	0	4	9
☐	○	-- 10.23.217.12:554	Common		0	0	0	0	0	0	0

Future Blog Posts

Future Blog Posts

RTSP Application Layer Monitoring

RTCP/RTP Port Limiting

Non-DSR Implementations

Thanks for putting up with such a long post!! I hope this helps. Please, please send me any updates/omissions/criticism, etc. I will gladly update this post with corrections and additional information.

 r.korock@f5.com