Configuring RTMFP Unicast on EC2

Posted At : July 20, 2011 12:21 PM | Posted By : Stefan Richter
Related Categories: FMS, Windows

A few weeks ago I was testing some of the FMS4 features on a Windows EC2 instance. My goal was to use RTMFP (usually used for peer to peer communications in Flash) in a client-server mode, basically replacing RTMP in order to achieve lower latency.
In case you do not want to read the entire post below, here's were I went wrong:
1) I did not open port 1935 over UDP, only TCP. As it turned out, RTMFP does require port 1935 over UDP for the initial contact. 2) I didn't configure the HostPort directive in Adaptor.xml correctly. Instead of adding the public attribute I had added the IP just to the node valu.
Instead of

```
<HostPort public="50.19.224.164:19350-65535">:19350-65535</HostPort>
```

I had configured

```
<HostPort>50.19.224.164:19350-65535</HostPort>
```

After correcting that I was able to connect via RTMFP. This also works when the Windows firewall is turned on, all I configured there was to allow the FMS .exe files through.
So that's the solution - more detailed info follows below.

At first I could not get client-server RTMFP to work. Using the standard setup and install process, RTMP connections to my instance worked fine but RTMFP connections seemed to hang for a few minutes before eventually failing.
Initially I had tried to configure the elastic IP of my instance (hint: if you are new to EC2: the elastic IP is essentially the instance's static IP that is assigned at that time but which can be changed) using the hostport in fms.ini. Removing the IP from fms.ini's hostport directive re-enabled RTMP connections via the public elastic IP. RTMFP connections failed. The logs gave me these errors:

#Fields: date time x-pid x-status x-ctx x-comment
2011-06-10 09:07:41 3456 (i)2581173 FMS detected IPv6 protocol stack! -
2011-06-10 09:07:41 3456 (i)2581173 FMS config -
2011-06-10 09:07:41 3456 (i)2581173 FMS running in IPv4 protocol stack mode! -
2011-06-10 09:07:41 3456 (i)2581173 Host: ip-0A50DD0A IPv4: 10.80.221.10 -
2011-06-10 09:07:41 3456 (e)2631013 Failed to create listener for adaptor _defaultRoot_, IP 50.19.224.164, port 80: TCCommBridge::createListener 50.19.224.164:80/v4: bind failed!!!. -
2011-06-10 09:07:41 3456 (e)2631013 Failed to create listener for adaptor _defaultRoot_, IP 50.19.224.164, port 443: TCCommBridge::createListener 50.19.224.164:443/v4: bind failed!!!. -

2011-06-10 09:07:41 3456 (e)2631013 Failed to create listener for adaptor _defaultRoot_, IP 50.19.224.164, port 1935: TCCommBridge::createListener 50.19.224.164:1935/v4: bind failed!!!. -
2011-06-10 09:07:41 3456 (i)2631174 Listener started (_defaultRoot__edge1) : localhost:19350/v4 -
2011-06-10 09:07:42 3456 (e)2631114 Failed to start listeners for adaptor _defaultRoot__edge1. -
2011-06-10 09:07:42 3456 (e)2791225 Failed to start edge : _defaultRoot__edge1

Luckily Seth Hodgson from Adobe stepped in and provided tons of great info,. Here's what he told me.

In the Adaptor.xml config file for your FMS server, you'll find an section. Make sure that this is enabled, and then take a look at the following sub-element: /Adaptor/RTMFP/Core/HostPortList/HostPort

When you're running an RTMFP adaptor behind a NAT you need to specify the in-front-of-NAT IP address that your clients will be connecting to in the optional 'public' attribute for this element. Like so:

:19350-65535

Additional background: The port range here, by default 19350-65535, is defined because each FMSCore process that starts up (and the number of these depends on the number of app instances you have running and how they're configured to distribute across FMSCore processes) has its own dedicated RTMFP listener/adaptor. Assume you have 3 app instances and each is configured to run in its own process; then you'll have three FMSCore processes running and each will have its own RTMFP listener running. These listeners acquire the next available port from this range as they come online. So in this case, they should bind 19350, 19351 and 19352 (assuming those ports are available).

When a new client connects to the FMS, it initially is communicating with the FMSEdge process on UDP port 1935, and depending on the app instance the client is trying to connect to, it will be redirected (this is a low-level RTMFP capability, not a traditional RTMP redirect) to the port bound by the RTMFP listener for the FMSCore process hosting the target app instance. As part of this redirect, the RTMFP listener that the client is being redirected to is also notified that a client is being redirected to it, and it sends a packet out toward the client which should be sufficient to punch a hole in the firewall allowing the redirected clients traffic back in over the new port (say, 19351).

When you're NATed, if you don't specify the 'public' attribute above then the server has no way of knowing or discovering its in-front-of-NAT-address and the RTMFP redirect will point at the right port (e.g. 19351) but will use the behind-NAT address, which the client can't reach.

Also, while allowing UDP inbound on ports 19350-65535 (or whatever smaller range you define in your FMS configs) in your EC2 firewall rules doesn't hurt, I don't think you necessarily need to do this. You're probably OK with just allowing UDP in on 1935 and UDP out on any port (allowing traffic outbound on any port may be the default behavior for EC2, but I'm not personally familiar with that).

So there you go - does your head hurt yet? :-)