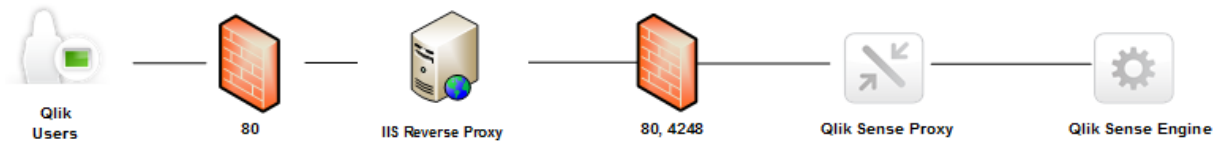# IIS as Reverse Proxy for Qlik Sense

Step by step guide to set up IIS to act as a Reverse Proxy for Qlik Sense.

The good news is that IIS supports all this using a combination of Application Request Routing (ARR) and URL Rewriting.

In most instances this is useful when a customer is adamant about only exposing port 80 or 443 to the external users (i.e. no 4248 or 4244), or when they want to use their corporate mandated web servers in their web tier, in which case we can just bundle the Qlik Sense proxy with the Engine.

So, how does one set this up?

Example setup

In my example I'm using the following machines:

sense-dev.sense.local: IIS Reverse Proxy

sense-pn.sense.local: Qlik Sense Proxy

sense-cn.sense.local: Client node for testing access

Step-by-step guide

First, I'll assume IIS is installed (I'm using IIS 8 on Windows Server 2012 R2 but 7.x should be fine as well).

1. Go here (http://blogs.technet.com/b/erezs_iis_blog/archive/2013/11/27/installing-arr-manually-without-webpi.aspx) and follow the steps for downloading and installing the:
    1. Web Farm Framework module
    2. External cache module
    3. URL Rewrite module
    4. ARR

    NOTE: I'm not linking to the downloads directly here as the above blog seems to

be doing a good job of keeping the download links current. If you have internet connection on your server it's easier to use the Microsoft Web Platform Installer and just download the required components. However, this is not always the case so the manual steps might come in handy.

2. Ensure that you have added support for WebSockets when adding the IIS role to the server:
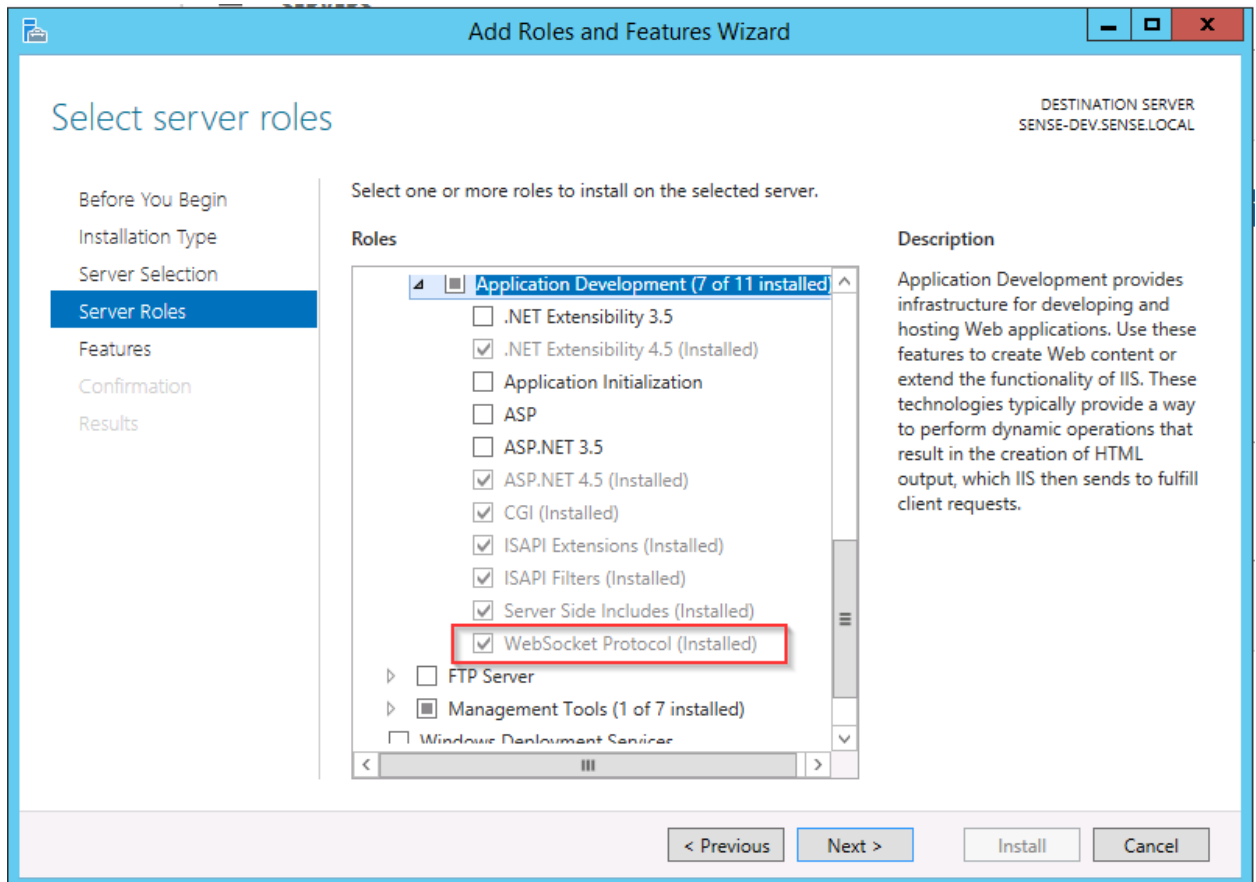
3. Go to your IIS Manager, select your Server and go to Application Request Routing Cache.

1. Set it up as a reverse proxy

Application Request Routing

Use this feature to configure proxy settings for Application Request Routing.

☑ Enable proxy

Proxy Setting

HTTP version:

| Pass through | ⌄ |

☑ Keep alive

Time-out (seconds):

| 120 |

☑ Reverse rewrite host in response headers

Custom Headers

Preserve client IP in the following header:

| X-Forwarded-For |

☑ Include TCP port from client IP

Forwarding proxy header value:

| |

4. Go back to your server view and double click URL Rewrite.
   1. Click Add Rule

2. Call the new rule **QlikReverseProxyAll** and add "(.*)" under Pattern:

3. Set "**http://sense-pn.sense.local/{R:0}**" (*NOTE: Use your own Qlik Sense Proxy URL)* under Rewrite URL:

NOTE: Make sure you uncheck the "Stop processing of subsequent rules" checkbox since we want to catch calls to the /form/ section for routing to the 4248/4244 port which we will set up in the next URL Rewrite rule below.

4. Apply the rule and create a new rule

5. Call the new rule **AuthForwarding** and add "form/" under Pattern:

6. Set "**http://sense-pn.sense.local:4248/{R:0}**" (*NOTE: Use your own Qlik Sense Proxy URL)* under Rewrite URL:

NOTE: This rule is for wrapping up the port 4248 behind port 80. You want the URL to point to your Qlik Sense Proxy URL and the port 4248 is the HTTP authentication port for Sense. If using HTTPS, use 4244.

7. If you are NOT using the Qlik Sense login page but are leveraging on Windows authentication, you also need to map the /windows_authentication/ path to port 4248.

8. Add a new rule with the Pattern: **/windows_authentication/** and Rewrite URL: **http://sense-pn.sense.local:4248/{R:0}**

5. Now, from my client node, I can access http://sense-dev.sense.local/hub and have the login rolled in under the standard HTTP port for the client: