Part 6a: mod_proxy and Lenya

Table of contents

We had mentioned that we were going to do articles on usecases and doctypes, but as we were reconfiguring the newest version of Lenya (1.2.4 as of this writing) to be installed under the ROOT context of Tomcat, we were again interested in going back and fixing up our mod_proxy configurations so that the following would happen:

- You can access the authoring environment with http://lenya.client.com/
- Any login requests to and subsequent usages of the authoring environment are redirected to SSL for better security (no real certificate needed, since just using it on inside)
- Each publication is a directory under one virtualhost for http://www.client.com/, unless a new domain name or sub-domain name was needed, like http://publication.client.com/, where a separate virtualhost would be created pointing to that one publication
- A need for SSL pages on http://www.client.com/ (like form submissions)

It seemed like a lot to ask for, and we wasn't sure how to go about getting it done. Lucky for us (or so we thought), Lenya's documentation has a page on mod_proxy (http://wiki.apache.org/lenya/HowToModProxy) for a very similar configuration. As always, however, making sense of the documentation was harder than the actual configuration. Here's our attempt at explaining just how to get the above setup working.

1. Getting Lenya installed under the ROOT context

So, we were trying to make things a little cleaner for ourself and having Lenya be the only web application installed under Tomcat. To do this, you'll have to change a couple things in your local.build.properies file before building Lenya (see installing Lenya (installing_lenya.html) for more information):

1. Change the container from Jetty to Tomcat (new to 1.2.4) #web.app.server=Jettyweb.app.server=Tomcat
2. Change the tomcat.webapps.dir line to the following:tomcat.webapps.dir=${tomcat.home.dir}/webapps/ROOT
3. Change the tomcat.cache.dir line to the following:tomcat.cache.dir=${tomcat.home.dir}/work/Catalina/localhost

While the last line may seem strange, our thinking here was that since we are only using Tomcat for Lenya, if we ever reset Lenya, then we'll just clean out the whole work directory before starting Tomcat again.

Then install as always using ./build.sh. You'll have a shiny new installation under the ROOT context, where you can now access Lenya with http://lenya.client.com:8080/ (notice there's no 'lenya' at the end of the URL now).

2. A caveat on SSL

Yes, we know, there's always a caveat. Well, one thing the mod_proxy document doesn't mention is that you can't have two different domains or sub-domains with SSL ports opened on the same Apache instance and using the same IP address. For example, if we want to host http://lenya.client.com/ and http://www.client.com/ on the same Apache instance and they both have the same IP address (this is Name-based Virtual Hosting in Apache), you would think you could do this:

```
NameVirtualHost 192.168.1.100:80

<VirtualHost 192.168.1.100:80>
    ServerName lenya.client.com
    # rest of configuration goes here
</VirtualHost>

<VirtualHost 192.168.1.100:443>
    ServerName lenya.client.com
    # rest of configuration goes here
</VirtualHost>

<VirtualHost 192.168.1.100:80>
    ServerName www.client.com
    # rest of configuration goes here
</VirtualHost>

<VirtualHost 192.168.1.100:443>
```

```
    ServerName www.client.com
    # rest of configuration goes here
</VirtualHost>
```

But in fact, you can't. The explanation for it is [in Apache's documentation](http://httpd.apache.org/docs-2.0/ssl/ssl_faq.html#vhosts2) . The way around this would be to either get another IP address for www.client.com, or assign another port to www.client.com's SSL connection (instead of using the default 443). So, in this case, we are going to be using another IP address. To be perfectly honest, we are actually going to split the authoring and live servers on to two physical machines, but setting it up this way on one server using one Apache instance is good enough for demonstration purposes.

3. Authoring environment

So, for this first part, we want to be able to go to http://lenya.client.com/ and get the first page of Lenya, where we can choose the publication we need to edit. Here's what we have (it's nearly idential to the mod_proxy how-to document on Lenya's website):

```
NameVirtualHost 192.168.1.100:80

<VirtualHost 192.168.1.100:80>
    ServerName lenya.client.com
    ServerAlias lenya
    ProxyRequests Off
    RewriteEngine On
    RewriteLog logs/lenya.client.com.rewrite.log
    RewriteLogLevel 0
    RewriteRule ^/([^/\.]+)$ $1/ [R]
    RewriteRule ^/([^/\.]+)/$ http://lenya.client.com/$1/authoring/index.html [R,L]
    RewriteCond %{QUERY_STRING} lenya\.usecase=login(.*)
    RewriteRule ^/(.*) https://%{SERVER_NAME}/$1 [R,L]
    RewriteRule ^/(.*) http://lenya.client.com:8080/$1 [P,L]
    ProxyPassReverse / http://lenya.client.com:8080/
</VirtualHost>

<VirtualHost 192.168.1.100:443>
    ServerName lenya.client.com
    ServerAlias lenya
    ProxyRequests Off
    RewriteEngine On
    RewriteLog logs/ssl.lenya.client.com.rewrite.log
    RewriteLogLevel 0
    RewriteRule ^/([^/\.]+)$ $1/ [R]
    RewriteRule ^/([^/\.]+)/$ http://lenya.client.com/$1/authoring/index.html [R,L]
    RewriteRule ^/(.*) http://%{SERVER_NAME}:8080/$1 [P,L]
    ProxyPassReverse / http://lenya.client.com:8080/
</VirtualHost>
```

Let's step through this quickly. We setup our name-based virtual host for the IP address assigned to lenya.client.com:

```
NameVirtualHost 192.168.1.100:80
```

In the first virtual host (the non-SSL one on port 80), we give it a name of lenya.client.com, keep proxy requests off (so that it's only a reverse proxy, not a forwarding one), and then turn on the rewriting engine to enable to rewrite URLs. We also setup a log for the rewrites, but since the log level is 0, it won't actually log anything.

The first group of RewriteRules are for convenience's sake, really. They match everything after but up to the first forward slash that doesn't have a dot in it. In other words, it's trying to match a directory, like this:

```
http://lenya.client.com/default
```

If it matches, it resends it as http://lenya.client.com/default/authoring/index.html, meaning that it goes through the whole

rigamarole of matching again inside the VirtualHost. Well, whenever you access the authoring environment for the first time, Lenya checks to see if there's a session of you being logged in. Since there probably isn't, it forwards you an address where "lenya.usecase=login" something-or-another is appended to it. And that's where the second group of Rewrites comes in. See the RewriteCond? It checks to see if the query string of the URL has that pattern. If it does, it send it off to the SSL portion (see the https?). The rest of it gets sent back to the reverse proxy, where the content is grabbed from port 8080 where Tomcat is installed.

In the VirtualHost section for SSL on lenya.client.com, it's pretty much exactly the same. If you try to hit the SSL port with just a directory, like so:

```
https://lenya.client.com/default
```

It rewrites the URL as http://lenya.client.com/default/authoring/index.html. Here, again, it goes back to the first VirtualHost, and if you aren't logged in, it eventually takes you back to the SSL portion of the site to login. Otherwise, it matches everything and sends it through the reverse proxy on port 8080. So, once you login through SSL, you stay in SSL for all your editing.

4. Next part

This article turned out to be quite long, so the next time around, which should be very shortly, we will post the second half on configuring the live server mod_proxy config.