

Managing Apache vHosts configuration for reverse proxying to pollers

Pre-requisites - Understanding your environment

The Opmantek VM Appliance is CentOS 6 and ships with Apache v2.2

The latest version of Apache which ships with most recent OSs is Apache v2.4

There are differences in where Configuration files are stored between OSs. There is a difference in the syntax for configuration files between apache versions.

Note now what OS you are on and what Apache version you have.



Getting The Real Details

There are lots of useful things you can do with proxying and Apache. For example pushing URLs from HTTP to HTTPS, doing SSL offload, having different URLs for different customers, having the same server name for lots of your services hosted on different internal servers etc. To properly understand all the details refer to:

For Apache 2.4

http://httpd.apache.org/docs/current/howto/reverse_proxy.html

Apache 2.2 simply has this example [http://httpd.apache.org/docs/2.2/vhosts/examples.html#Using Virtual_host](http://httpd.apache.org/docs/2.2/vhosts/examples.html#Using_Virtual_host) and mod_proxy together



Best Practice Load Balancing

Of course your company may also have sophisticated proxying and **load balancing appliances** already in place, in which case **use those**.

Where are all my Apache configuration files?

<http://wiki.apache.org/httpd/DistrosDefaultLayout>

Centos

```
ServerRoot           :: /etc/httpd
Primary Config File  :: /etc/httpd/conf/httpd.conf
Other Config Files   :: /etc/httpd/conf.d
Module Locations     :: /usr/lib/httpd/modules
DocumentRoot        :: /var/www/html
ErrorLog             :: /var/log/httpd/error_log
AccessLog           :: /var/log/httpd/access_log
cgi-bin             :: /var/www/cgi-bin (empty and disabled by default)
binary              :: /usr/sbin/httpd
runtime directory    :: /etc/httpd/run
start/stop          :: /sbin/service httpd
{start|stop|restart|condrestart|reload|status|fullstatus|graceful|help|configtest}
```

Notes:

1. There is an extra config file in `/etc/sysconfig/httpd` which can be used to change to the worker mpm `/usr/sbin/httpd.worker`.
2. Extra config files named `*.conf` are loaded from `/etc/httpd/conf.d`. This directory is used by packages like `mod_python` for drop-in configs
3. If you're having issues with authorization and your permissions are correct, you might have problems with SELinux permissions. Take a look at [httpd_selinux\(8\)](#) and related documentation. Particularly `sealert(8)` can be used for analysis and suggested solutions.

Debian Debian, Ubuntu (Apache httpd 2.x):

```
ServerRoot          ::      /etc/apache2
DocumentRoot        ::      /var/www
Apache Config Files ::      /etc/apache2/apache2.conf
                   ::      /etc/apache2/ports.conf
Default VHost Config ::     /etc/apache2/sites-available/default, /etc/apache2/sites-enabled/000-default
Module Locations    ::      /etc/apache2/mods-available, /etc/apache2/mods-enabled
ErrorLog            ::      /var/log/apache2/error.log
AccessLog           ::      /var/log/apache2/access.log
cgi-bin             ::      /usr/lib/cgi-bin
binaries (apachectl) ::     /usr/sbin
start/stop          ::      /etc/init.d/apache2 (start|stop|restart|reload|force-reload|start-
htcacheclean|stop-htcacheclean)
```

Notes:

1. The Debian/Ubuntu layout is fully documented in `/usr/share/doc/apache2/README.Debian`
2. Debian/Ubuntu use symlinks to configure vhosts and load modules. Configuration files are created in their respective `sites-available` and `mods-available` directories. To activate vhosts and modules, symlinks are created in the respective `sites-enabled` and `mods-enabled` directories to the config files in either `sites-available` and `mods-available`. Debian provides scripts to handle this process called 'a2ensite' and 'a2enmod' which activates vhosts and modules.
3. The default vhost is defined in `/etc/apache2/sites-available/default`, and overrides the `DocumentRoot` set in the server context.

Proxy Configuration on Opmantek VM - CentOS6 - Apache 2.2

We need to add several Vhosts entries for the proxying. This means altering, adding deleting the configuration files (that we found above). We need to end up with vhost configuration entries for the localhost/Primary webservice and for the URLs proxied to the Pollers.

The shipped `04proxy.conf` file if you have one, should be removed and replaced with a new one as it will not work alongside new entries.

We already have several Location entries in operation for normal operation of OMK modules find these entries in the `04proxy.conf`

We should replace that configuration file with a new which specifies the URLs used by the Primary and by the pollers.

OMKproxy.conf example

```

## Centos /etc/httpd/conf.d/OMK-proxy.conf
## Debian /etc/apache2/sites-enabled/OMK-proxy.config

<IfModule mod_proxy.c>
ProxyRequests off
#LogLevel Debug

<IfModule mod_headers.c>
# if you are using the Opmantek applications behind an ssl-terminating apache vhost,
# then you should adjust the vhost configuration to add this header but with
# protocol "https".
# The Opmantek applications are location- and protocol-independent in almost all cases.
RequestHeader set X-Forwarded-Proto "http"
</IfModule>

### As this is an older version of of Apache we first need to enable the NameVirtualHost and associate it to IP
and port see https://httpd.apache.org/docs/2.2/vhosts/name-based.html
## In particular https://httpd.apache.org/docs/2.2/mod/core.html#namevirtualhost

NameVirtualHost *:80

## We can then specify the VirtualHosts and match to the required with ServerName {name} directive
<VirtualHost *:80>
## This should match the servername of the primary.
ServerName pimaryserver.example.com
  <Location "/omk">
    ProxyPass http://localhost:8042/omk retry=5
    ProxyPassReverse http://localhost:8042/omk
  </Location>
  <Location "/es">
    ProxyPass http://localhost:8042/es retry=5
    ProxyPassReverse http://localhost:8042/es
  </Location>
  <Location "/en">
    ProxyPass http://localhost:8042/en retry=5
    ProxyPassReverse http://localhost:8042/en
  </Location>
  <Location "/pt">
    ProxyPass http://localhost:8042/pt retry=5
    ProxyPassReverse http://localhost:8042/pt
  </Location>
</VirtualHost>

## This section is only used if you are using this server as a Primary server which is proxying to pollers.
<VirtualHost *:80>
  ## This server ServerName should match 'host' entry in /usr/local/nmis8/conf/Servers.nmis
  ServerName external-name-poller-1.example.com

  ProxyRequests off
  ## This URL should point to the internal name of the poller server so this Primary can create it's own
  connection to the poller
  ProxyPass / http://external-name-poller-1.example.com/ retry=5
  ProxyPassReverse / http://external-name-poller-1.example.com/

</VirtualHost>

## The second poller again these should match what is in Servers.nmis
<VirtualHost *:80>
  ServerName external-name-poller-2.example.com
  ProxyRequests off
  ProxyPass / http://external-name-poller-2.example.com/ retry=5
  ProxyPassReverse / http://external-name-poller-2.example.com/
</VirtualHost>
</IfModule>

```

