Apache Httpd (Ver 2.4)

Background

Apache HTTPD Log Analysis App is an Http server for Windows and Unix machines that automatically Collect - Read - Parse - Analyzes - Reports all machine generated log data of the server and presents a comprehensive automatic predefined set of Reports, Dashboards and Gadgets. Once you Setup and configure the Apache HTTPD App, you will be redirected to the dashboards where yuo will have graphs about: errors occured, geographic data of users and requests, Browsers related analytics, Pages and hits analysis, resources and many statistics about your servers' performance. You later use XpoLog built in Analytics features to zero in on errors and take actions to improve your system's uptime. Apache HTTP server logs data can be viewed, filtered and searched via the main XpoLog console.

Steps

1. Add Log Data In XpoLog, When adding a log to XpoLog you can now select the Log Type (logtype) for Apache Httpd the are the following logtypes:
    a. *httpd*
    b. *w3c*
    c. *webserver*
        i. in addition select not only httpd but also the log type - *access* or *error*
2. Once all required information is set click next and edit the log pattern, this step is crucial to the accuracy and deployment of the Apache Httpd App. Use the following conversion table to build the XpoLog pattern out of the access log format.

Example

In the Apache Httpd configuration file, usually httpd.conf by default, located under the conf/ directory (Linux "/etc/httpd/conf") search for the LogFormat directive:

LogFormat "%h %l %u %t \"%r\" %>s %b" common

In XpoLog such pattern will be translated into:

{text:RemoteHost,ftype=remoteip} {text:logname,ftype=remotelog} {text:Remote User,ftype=remoteuser} [{date:Date,locale=en;,dd/MMM/yyyy:HH:mm:ss z}] "{choice:Method,ftype=reqmethod;,GET;POST;HEAD} {url:URL,paramsFtype=querystring;ftype=requrl;paramsName=Query;,} {string:reqprotocol,ftype=reqprotocol;,}" {number:ResponseStatus,ftype=respstatus} {number:Bytes Sent,ftype=bytesent}{eoe}

for more information see below:

Apache Https Access Log Format Conversion Table

logtypes should be set to: *httpd,w3c,webserver,access*

Format String	Description	XpoLog Pattern
%a	Remote IP-address	{geoip:Remote IP,ftype=remoteip}
%{c}a	Underlying peer IP address and port of the connection	{geoip:Remote IP,ftype=remoteip}
%A	Local IP-address	{geoip:LocalIP,ftype=localip}
%B	Size of response in bytes, excluding HTTP headers.	{number:Bytes Sent,ftype=bytesent}
%b	Size of response in bytes, excluding HTTP headers. In CLF format, *i.e.* a '-' rather than a 0 when no bytes are sent.	{number:Bytes Sent,ftype=bytesent}
%{Foobar}C	The contents of cookie Foobar in the request sent to the server. Only version 0 cookies are fully supported.	{string:Cookie_< FOOBAR >} Replace < FOOBAR > with cookie name

%D	The time taken to serve the request, in microseconds.	{number:ResponseTimeMicroSecs,ftype=processrequestmicrosecs}
%{FOOBAR}e	The contents of the environment variable FOOBAR	{string:EnvVariable_< FOOBAR >} Replace < FOOBAR > with variable name
%f	Filename	{text:FileName}
%h	Remote host	{text:Remotehost,ftype=remoteip}
%H	The request protocol	{text:RequestProtocol,ftype=reqprotocol}
%{Foobar}i	The contents of Foobar: header line(s) in the request sent to the server. Changes made by other modules (e.g. mod_headers) affect this. If you're interested in what the request header was prior to when most modules would have modified it, use mod_setenvif to copy the header into an internal environment variable and log that value with the %{VARNAME}e described above.	{text:<FOOBAR>} https://en.wikipedia.org/wiki/List_of_HTTP_header_fields and so on it goes for the different headers.
%{Referer}i	The referer which is associated with the request	{text:RefererQuery,ftype=refererquery;,}{regexp:Referer,ftype=referer;refName=RefererQuery,^([\w-]+://[^?]+
%{User-agent}i	The User Agent which is associated with the request	{text:User-agent,ftype=useragent}
%{X-Forwarded-For}i	Method for identifying the originating IP address of a client connecting to a web server through an HTTP proxy or load balancer.	{text:X-Forwarded-For,ftype=forwardforip}
%k	Number of keepalive requests handled on this connection. Interesting if KeepAlive is being used, so that, for example, a '1' means the first keepalive request after the initial one, '2' the second, etc...; otherwise this is always 0 (indicating the initial request). Available in versions 2.2.11 and later.	{number:KeepAlive}
%l	Remote logname (from identd, if supplied). This will return a dash unless mod_ident is present and IdentityCheck is set On.	{text:logname,ftype=remotelog}
%m	The request method	{choice:Method,ftype=reqmethod;,GET;POST;HEAD}
%{Foobar}n	The contents of note Foobar from another module.	{string:<FOOBAR>}
%{Foobar}o	The contents of Foobar: header line(s) in the reply.	{string:<FOOBAR>}
%p	The canonical port of the server serving the request	{number:ServerPort,ftype=serverport}

%{format}p	The canonical port of the server serving the request or the server's actual port or the client's actual port. Valid formats are canonical, local, or remote. %{canonical}p %{local}p %{remote}p	{number:ServerPort,ftype=serverport} {number:LocalServerPort,ftype=localserverport} {number:RemotePort,ftype=remoteport}
%P	The process ID of the child that serviced the request.	{text:ProcessID,ftype=processid}
%{format}P	The process ID or thread id of the child that serviced the request. Valid formats are pid, tid, and hextid. hextid requires APR 1.2.0 or higher.	{text:ProcessID,ftype=processid} Valid formats are pid, tid, and hextid.
%{pid}P		{text:ProcessID,ftype=processid}
%{tid}P		{text:ThreadID,ftype=threadid}
%{hextid}P		{text:HexThreadID,ftype=hexthreadid}
%r	First line of request	1. {choice:Method,ftype=reqmethod;,GET;POST} {url:URL,paramsFtype=querystring;ftype=requrl;paramsName=Query;,} {string:reqprotocol,ftype=reqpro
%R	The handler generating the response (if any).	{text:ResponseHandler}
%s	Status. For requests that got internally redirected, this is the status of the *original* request --- %>s for the last.	{number:ResponseStatus,ftype=respstatus} . For requests that got internally redirected, this is the status of the *original* request --- %>s for the last.
%t	Time the request was received (standard english format)	{date:Date,locale=en,dd/MMM/yyyy:HH:mm:ss z}

<table>
<tr>
<td>%{format}t</td>
<td>The time, in the form given by format, which should be in an extended strftime(3) form at (potentially localized). If the format starts with begin: (def ault)

the time is taken at the beginning of the request processing. If it starts with end : it is the time when the log entry gets written, close to the end of the request

processing. In addition to the formats supported by strfti me(3), the following format tokens are supported:

<table>
<tr><td>sec</td><td>number of seconds since the Epoch</td></tr>
<tr><td>msec</td><td>number of milliseconds since the Epoch</td></tr>
<tr><td>usec</td><td>number of microseconds since the Epoch</td></tr>
<tr><td>msec_frac</td><td>millisecond fraction</td></tr>
<tr><td>usec_frac</td><td>microsecond fraction</td></tr>
</table>
These tokens can not be combined with each other or s trftime(3) formatting in the same format string. You can use multiple %{format}t token s instead.

The extended strftime(3) t okens are available in 2.2.30 and later.</td>
<td>{date:Date,locale=en,dd/MMM/yyyy:HH:mm:ss z}

sec number of seconds since the Epoch
msec number of milliseconds since the Epoch
usec number of microseconds since the Epoch
msec_frac millisecond fraction
usec_frac microsecond fraction</td>
</tr>
<tr>
<td>%T</td>
<td>The time taken to serve the request, in seconds.</td>
<td>{number:ResponseTimeSecs,ftype=processrequestsecs}</td>
</tr>
<tr>
<td>%{UNIT}T</td>
<td>The time taken to serve the request, in a time unit given by UNIT. Valid units are ms for milliseconds, us for microseconds, and s for seconds. Using s gives the

same result as %T without any format; using us gives the same result as %D. Combining %T with a unit is available in 2.2.30 and later.</td>
<td>1. {number:ResponseTimeMilliSecs,ftype=processrequestmilli}
2. {number:ResponseTimeMicroSecs,ftype=processrequestmicrosecs}
3. {number:ResponseTimeSecs,ftype=processrequestsecs}</td>
</tr>
<tr>
<td>%u</td>
<td>Remote user (from auth; may be bogus if return status (%s) is 401)</td>
<td>{string:Remote User,ftype=remoteuser;,}

Remote user (from auth; may be bogus if return status (%s) is 401)</td>
</tr>
<tr>
<td>%U</td>
<td>The URL path requested, not including any query string.</td>
<td>{text:RequestURL,ftype=requrl}</td>
</tr>
<tr>
<td>%v</td>
<td>The canonical ServerName of the server serving the request.</td>
<td>{text:ServerName,ftype=servername}</td>
</tr>
</table>

%V	The server name according to the UseCanonicalName setting.	{text:ServerName,ftype=servername} The server name according to the UseCanonicalName setting.
%X	Connection status when response is completed: X = connection aborted before the response completed. + = connection may be kept alive after the response is sent. - = connection will be closed after the response is sent. (This directive was %c in late versions of Apache 1.3, but this conflicted with the historical ssl %{var}c syntax.)	{text:ConnectionStatus} Connection status when response is completed: X = connection aborted before the response completed. + = connection may be kept alive after the response is sent. - = connection will be closed after the response is sent. (This directive was %c in late versions of Apache 1.3, but this conflicted with the historical ssl %{var}c syntax
%I	Bytes received, including request and headers, cannot be zero. You need to enable mod_logio to use this.	{number:TotalBytesWHeadersReceived,ftype=reqbyteswheaders} (with headers)
%O	Bytes sent, including headers, cannot be zero. You need to enable mod_logio to use this.	{number:TotalBytesWHeadersSent,ftype=respbyteswheaders} (with headers – can help compute header size)
%{VARNAME}^ti	The contents of VARNAME: trailer line(s) in the request sent to the server.	{text:Req_<*VARNAME*>} The content of *VARNAME*: trailer line(s) in the request sent to the server.
%{VARNAME}^to	The contents of VARNAME: trailer line(s) in the response sent from the server.	{text:Resp_<*VARNAME*>} The contents of VARNAME: trailer line(s) in the response sent from the server.
%{FOOBAR}^ti	The contents of FOOBAR: trailer line(s) in the request sent to the server.	{text:Req_<FOOBAR>} The content of FOOBAR: trailer line(s) in the request sent to the server.
%{FOOBAR}^to	The contents of FOOBAR: trailer line(s) in the response sent from the server.	{text:Resp_<FOOBAR>} The contents of FOOBAR: trailer line(s) in the response sent from the server.

Error Log

In the Apache Httpd configuration file, usually httpd.conf by default, located under the conf/ directory (Linux "/etc/httpd/conf") search for the LogFormat directive:

```
ErrorLogFormat "[%{u}t] [%m:%l] [pid %P:tid %T] %F: %E: %M"
```

In XpoLog such pattern will be translated into:

[{date:Date,locale=en,EEE MMM dd HH:mm:ss.SSSSSS yyyy}] [{text:Module}:{priority:Level,ftype=status;,}] [pid {text:ProcessID,ftype=processid;,}:tid {text:ThreadId,ftype=threadid;,}]{text:ErrorCode,ftype=errorcode;,}:{block,start,emptiness=true} {text:SourceFileName}:{block,end,emptiness=true} {string:Message,ftype=Message;,}

for more information see below:

Apache Https Error Log Format Conversion Table

logtypes should be set to: httpd,w3c,webserver,*error*

Format String	Description	XpoLog Pattern
%a	Client IP address	{geoip:Remote IP,ftype=remoteip}

`%{c}a`	Underlying peer IP address and port of the connection (see the `mod_remoteip` module)	{geoip:Remote IP,ftype=remoteip}{block,start,emptiness=true}:{number:Port,ftype=remoteport}{block,end,emptiness=tru
`%A`	Local IP-address	{ip:LocalIP,ftype=localip}
`%{name}e`	Request environment variable *name*	{string:EnvVariable_name}
`%E`	APR/OS error status code and string	{number:Error Status Code,ftype=errcode}
`%F`	Source file name and line number of the log call	{text:FileName}
`%{Referer}i`	Referer of the call	{text:RefererQuery,ftype=refererquery}{regexp:Referer,ftype=referer;refName=RefererQuery,^([\\w-]+://[^?]+|/[^?]+)}
%{User-Agent}i	User-Agent of the call.	{text:User-agent,ftype=useragent}
`%k`	Number of keep-alive requests on this connection	{number:KeepAlive}
`%l`	The level of the message	{priority:Level,emerg;alert;crit;error;warn;notice;info;debug;trace1;trace2;trace3;trace4;trace5;trace6;trace7;trace8;ftype=
`%L`	Log ID of the request	{text:LogId,ftype=logid}
`%{c}L`	Log ID of the connection	{text:LogId,ftype=logid}
`%{C}L`	Log ID of the connection if used in connection scope, empty otherwise	{text:LogId,ftype=logid}
`%m`	Name of the module logging the message	{text:Module,ftype=module}
`%M`	The actual error message	{string:Message,ftype=Message}
`%{name}n`	Request note *name*	{Text:NOTE name}
`%P`	Process ID of current process	{text:ProcessID,ftype=processid}
`%T`	Thread ID of current thread	{number:ThreadID,ftype=thread}
`%{g}T`	System unique thread ID of current thread (the same ID as displayed by e.g. `top`; currently Linux only)	{number:System Thread ID,ftype=systemthread}
`%t`	Date	{date:Date,locale=en,EEE MMM dd HH:mm:ss yyyy}
`%{u}t`	The current time including micro-seconds	{date:Date,locale=en,EEE MMM dd HH:mm:ss.SSSSSS yyyy}
`%{cu}t`	The current time in compact ISO 8601 format, including micro-seconds	{date:Date,locale=en,EEE MMM dd HH:mm:ss.SSSSSS yyyy}
`%v`	The canonical `ServerName` of the current server.	{text:ServerName,ftype=servername}
`%V`	The server name of the server serving the request according to the `UseCanonicalName` setting.	{text:ServerName,ftype=servername} The server name according to the UseCanonicalName setting.
