<?xml version="1.0" encoding="UTF-8"?>

<configuration>

<!-- See: https://logback.qos.ch/manual/configuration.html#LevelChangePropagator -->

<contextListener class="ch.qos.logback.classic.jul.LevelChangePropagator">

<resetJUL>true</resetJUL>

</contextListener>

<include resource="logging/logback-all-include.xml" />

<property name="ALERTABLE\_MESSAGE\_FORMAT" value="ALERTABLE: %msg"/>

<!-- The following appender is used to log all important logs to a single file called

"tweetypie-important.log". The filter "OnlyImportantLogsLoggingFilter" is used by this

appender to categorize logs into important or not. Anything from warnings, errors and

criticals are logged here. Logs that are tweet creation, deletions, undeletions or

updates are not logged to reduce noise from the logger.

NOTE: This appender (and all file based loggers) is not persistent through host restarts.

All logs are lost on machine restarts. -->

<appender name="IMPORTANT-FILE" class="ch.qos.logback.core.rolling.RollingFileAppender">

<file>tweetypie-important.log</file>

<append>true</append>

<rollingPolicy class="ch.qos.logback.core.rolling.FixedWindowRollingPolicy">

<fileNamePattern>tweetypie-important-%i.log</fileNamePattern>

<minIndex>1</minIndex>

<maxIndex>20</maxIndex>

</rollingPolicy>

<triggeringPolicy class="ch.qos.logback.core.rolling.SizeBasedTriggeringPolicy">

<maxFileSize>800MB</maxFileSize>

</triggeringPolicy>

<encoder>

<pattern>

%date [%thread] %-5level %logger{36} - %msg%n

</pattern>

</encoder>

<!-- Allow all INFO+ messages from tweetypie logs, ERROR+ messages from all others.

These filters do not change the log levels for the loggers, they only filter what is

allowed to be logged to the important file. -->

<filter class="com.twitter.tweetypie.util.logging.OnlyImportantLogsLoggingFilter" />

</appender>

<!-- The following appender is effectively a copy of the appender above in order to

persist logs through restarts, instead of writing the content to a file, this appender writes

it to our persistent storage "splunk". LOGLENS was the name of old library to write these logs

into a persistent storage and was later replaced by "splunk". So, any appender that mentions

loglens now is actually sending data to splunk. Due to limitations of logback, we cannot use

the same appender to write data at multiple places (file and splunk). So, we have 2 similar

appenders that write to 2 separate places. -->

<appender name="LOGLENS-BASE" class="com.twitter.loglens.logback.LoglensAppender">

<mdcAdditionalContext>true</mdcAdditionalContext>

<tag>${log.lens.tag}</tag>

<index>${log.lens.index}</index>

<category>loglens</category>

<encoder>

<pattern>%msg</pattern>

</encoder>

<!-- Allow all INFO+ messages from tweetypie logs, ERROR+ messages from all others.

These filters do not change the log levels for the loggers, they only filter what is

allowed to be logged to loglens. -->

<filter class="com.twitter.tweetypie.util.logging.OnlyImportantLogsLoggingFilter" />

</appender>

<!-- The important logs may sometimes get too noisy to find relevant exceptions and errors

occurring in tweetypie. So, this appender is used to only writes Errors and RuntimeExceptions

to a file called "alertable-exception.log". -->

<appender name="ALERTABLE-EXCEPTION-FILE" class="ch.qos.logback.core.rolling.RollingFileAppender">

<file>alertable-exception.log</file>

<append>true</append>

<rollingPolicy class="ch.qos.logback.core.rolling.FixedWindowRollingPolicy">

<fileNamePattern>alertable-exception-%i.log</fileNamePattern>

<minIndex>1</minIndex>

<maxIndex>17</maxIndex>

</rollingPolicy>

<triggeringPolicy class="ch.qos.logback.core.rolling.SizeBasedTriggeringPolicy">

<maxFileSize>100MB</maxFileSize>

</triggeringPolicy>

<encoder>

<pattern>

%date [%thread] %-5level %logger{36} - %msg%n

</pattern>

</encoder>

<!-- Only log alertable exceptions to this file -->

<filter class="com.twitter.tweetypie.util.logging.AlertableExceptionLoggingFilter" />

</appender>

<!-- This is the persistent version of the appender above for logging Errors and RuntimeExceptions

to splunk. To make it easier to search for these exceptions/errors in splunk we prefix the

exceptions with "ALERTABLE:" string. -->

<appender name="LOGLENS-ALERTABLE-BASE" class="com.twitter.loglens.logback.LoglensAppender">

<mdcAdditionalContext>true</mdcAdditionalContext>

<tag>${log.lens.tag}</tag>

<index>${log.lens.index}</index>

<category>loglens</category>

<encoder>

<pattern>${ALERTABLE\_MESSAGE\_FORMAT}</pattern>

</encoder>

<!-- Allow all Runtime Exceptions and Error messages from tweetypie logs.

These filters do not change the log levels for the loggers, they only filter what is

allowed to be logged to loglens. -->

<filter class="com.twitter.tweetypie.util.logging.AlertableExceptionLoggingFilter" />

</appender>

<!-- Each of the appenders above are called Asynchronously to make sure logging operations

are non-blocking. Also, one AsyncAppender should only wrap one appender, appending multiple

appenders under one AsyncAppender does not work. -->

<appender name="LOGLENS" class="ch.qos.logback.classic.AsyncAppender">

<appender-ref ref="LOGLENS-BASE" />

</appender>

<appender name="LOGLENS-ALERTABLE" class="ch.qos.logback.classic.AsyncAppender">

<appender-ref ref="LOGLENS-ALERTABLE-BASE" />

</appender>

<appender name="IMPORTANT" class="ch.qos.logback.classic.AsyncAppender">

<appender-ref ref="IMPORTANT-FILE" />

</appender>

<appender name="ALERTABLE-EXCEPTION" class="ch.qos.logback.classic.AsyncAppender">

<appender-ref ref="ALERTABLE-EXCEPTION-FILE" />

</appender>

<!-- Memcached and ObservableCache log a lot of messages at the WARN level. These messages

are mostly not useful, so lets filter them out. If you would like to see them,

you can change the log level on the admin page. -->

<logger name="com.twitter.tweetypie.cache" level="error"/>

<logger name="memcache" level="error"/>

<logger name="com.twitter.tweetypie.config.MemcacheExceptionLoggingFilter" level="debug" />

<!-- Prune the log of noisy DarkTrafficFilter exceptions. If you would like to see them,

you can change the log level on the admin page. -->

<logger name="DarkTrafficFilter" level="off"/>

<!-- At last, we combine all the active AsyncAppenders under a single Root node. Any appender

that is missing from the node below will not be called for logging. -->

<root level="INFO">

<appender-ref ref="ALL" />

<appender-ref ref="IMPORTANT" />

<appender-ref ref="LOGLENS" />

<appender-ref ref="LOGLENS-ALERTABLE" />

<appender-ref ref="ALERTABLE-EXCEPTION" />

</root>

</configuration>