<!--

- Resin 4.0 configuration file. Resin 4.0版本配置文件

-->

<!-- xml 根-->

<resin xmlns="http://caucho.com/ns/resin"

xmlns:resin="urn:java:com.caucho.resin">

<!-- property-based Resin configuration -->

<resin:properties path="${\_\_DIR\_\_}/resin.properties" optional="true"/>

<resin:if test="${properties\_import\_url}">

<resin:properties path="${properties\_import\_url}"

optional="true" recover="true"/>

</resin:if>

<!-- Logging configuration for the JDK logging API -->

<log-handler name="" level="all" path="stdout:"

timestamp="[%y-%m-%d %H:%M:%S.%s]"

format=" {${thread}} ${log.message}"/>

<!--

- Alternative pseudo-TTCC log format

-

- <log-handler name="" level="all" path="stdout:"

- timestamp="%y-%m-%d %H:%M:%S.%s"

- format=" [${thread}] ${log.level} ${log.shortName} - ${log.message}"/>

-->

<!--

- level='info' for production

- 'fine' or 'finer' for development and troubleshooting

-->

<logger name="" level="${log\_level?:'info'}"/>

<logger name="com.caucho.java" level="config"/>

<logger name="com.caucho.loader" level="config"/>

<!--

- Default configuration applied to all clusters, including

- HTTP, HTTPS, and /resin-admin configuration.

-->

<resin:import path="${\_\_DIR\_\_}/cluster-default.xml"/>

<!--

- health configuration

-->

<resin:import path="${\_\_DIR\_\_}/health.xml"/>

<!--

- Remote management requires at least one enabled admin user.

-->

<resin:AdminAuthenticator>

<user name="${admin\_user}" password="${admin\_password}"/>

<resin:import path="${\_\_DIR\_\_}/admin-users.xml" optional="true"/>

</resin:AdminAuthenticator>

<!--

- For clustered systems, create a password in as cluster\_system\_key

-->

<cluster-system-key>${cluster\_system\_key}</cluster-system-key>

<!--

- For production sites, change dependency-check-interval to something

- like 600s, so it only checks for updates every 10 minutes.

-->

<dependency-check-interval>${dependency\_check\_interval?:'2s'}</dependency-check-interval>

<!-- For resin.properties dynamic cluster joining -->

<home-cluster>${home\_cluster}</home-cluster>

<home-server>${home\_server}</home-server>

<elastic-server>${elastic\_server}</elastic-server>

<elastic-dns>${elastic\_dns}</elastic-dns>

<!--

- Configures the main application cluster. Load-balancing configurations

- will also have a web cluster.

-->

<cluster id="app">

<!-- define the servers in the cluster -->

<server-multi id-prefix="app-" address-list="${app\_servers}" port="6800"/>

<host-default>

<!-- creates the webapps directory for .war expansion -->

<web-app-deploy path="webapps"

expand-preserve-fileset="WEB-INF/work/\*\*"

multiversion-routing="${webapp\_multiversion\_routing}"

path-suffix="${elastic\_webapp?resin.id:''}"/>

</host-default>

<!-- auto virtual host deployment in hosts/foo.example.com/webapps -->

<host-deploy path="hosts">

<host-default>

<resin:import path="host.xml" optional="true"/>

</host-default>

</host-deploy>

<!-- the default host, matching any host name -->

<host id="" root-directory=".">

<!--

- webapps can be overridden/extended in the resin.xml

-->

<web-app id="/" root-directory="webapps/ROOT"/>

</host>

<resin:if test="${resin\_doc}">

<host id="${resin\_doc\_host}" root-directory="${resin\_doc\_host}">

<web-app id="/resin-doc" root-directory="${resin.root}/doc/resin-doc"/>

</host>

</resin:if>

</cluster>

<cluster id="web">

<!-- define the servers in the cluster -->

<server-multi id-prefix="web-" address-list="${web\_servers}" port="6810"/>

<host id="" root-directory="web">

<web-app id="">

<resin:LoadBalance regexp="" cluster="app"/>

</web-app>

</host>

</cluster>

<cluster id="memcached" xmlns:memcache="urn:java:com.caucho.memcached">

<!-- define the servers in the cluster -->

<server-multi id-prefix="memcached-" address-list="${memcached\_servers}" port="6820">

<!-- listen for the memcache protocol -->

<listen port="${memcached\_port?:11211}"

keepalive-timeout="600s" socket-timeout="600s">

<memcache:MemcachedProtocol/>

</listen>

</server-multi>

</cluster>

<cluster id="proxycache">

<!-- define the servers in the cluster -->

<server-multi id-prefix="proxycache-" address-list="${proxycache\_servers}" port="6830"/>

<host id="" root-directory="proxycache">

<web-app id="">

<resin:HttpProxy regexp=".\*">

<!-- backend HTTP servers to proxy to -->

<addresses>${backend\_servers}</addresses>

</resin:HttpProxy>

</web-app>

</host>

</cluster>

</resin>