Hosts Virtuales en Tomcat

Para configurar los Hosts Virtuales en Tomcat tendremos que editar el archivo server.xml que se encuentra en el directorio /var/lib/tomcat9/conf/.

Dentro de este .xml buscaremos la marca <Host> con su correspondiente marca de cierre </Host>, y copiaremos y/o editaremos el bloque entre ambas marcas, incluidas estas.

```
</Host>
   Guardar ≡ _ 🙃 😣
                          analyzes the HTTP headers included with the request, and passes them
                         on to the appropriate Host (virtual host)
127
128
139
131
132
133
135
136
137
138
139
140
141
142
144
145
150
151
152
153
154
157
158
159
161
162
163
164
165
166
167
168
                        Documentation at /docs/config/engine.html -->
              <!-- You should set jvmRoute to support load-balancing via AJP ie : <Engine name="Catalina" defaultHost="localhost" jvmRoute="jvm1">
              <Engine name="Catalina" defaultHost="localhost">
                  <Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster"/>
                 <!-- Use the LockOutRealm to prevent attempts to guess user passwords
    via a brute-force attack -->
<Realm className="org.apache.catalina.realm.LockOutRealm">
    <!-- This Realm uses the UserDatabase configured in the global JNDI
    resources under the key "UserDatabase". Any edits
    that are performed against this UserDatabase are immediately
    available for use by the Realm. -->
<Realm className="org.rpache.catalina.realm.UserDatabaseRealm"
    resourceAlmapse"/scriptabase"/>
    resourceAlmapse"/scriptabase"/>

                  "CENTRAL "UserDatabase"/>
                  Host name="localhost" appBase="webapps" unpackWARs="true" autoDeploy="true"
                      <!-- SingleSignOn valve, share authentication between web applications Documentation at: /docs/config/valve.html -->
                      <Valve className="org.apache.catalina.authenticator.SingleSignOn" />
                     <!-- Access log processes all example.
   Documentation at: /docs/config/valve.html
   Note: The pattern used is equivalent to using pattern="common"
<Valve className="org.apache.catalina.valves.AccessLogValve" director
    prefix="localhost access log" suffix=".txt"
    pattern="%h %l %u %t &quot;%r&quot; %s %b" />
                                                                                                                           LogValve" directory="logs
            /Engine>
171 </Server
                                                                                     XML ▼ Anchura del tabulador:8 ▼ Ln 135, Col 52 ▼ INS
```

De todo esto lo único que nos va a interesar son un par de atributos de <Host>:

- name: indica la URL de nuestro Host Virtual. Esta URL debe poder ser resuelta por nuestro interfaz de red, o bien mediante el servidor DNS o bien editando el archivo /etc/hosts/.
- appBase: es el directorio donde se alojarán los archivos de la aplicación a desplegar. Al contrario que cuando editamos Hosts Virtuales en Apache, en DocumentRoot, aquí establecemos una ruta relativa, que en realidad es un subdirectorio perteneciente al directorio /var/lib/tomcat9/ o bien también podemos escribir una ruta absoluta comenzando por /. Para desplegar aplicaciones en el dominio raíz, debemos crear dentro del appBase un directorio llamado ROOT y descomprimir/instalar allí el .war.

Creamos carpetas donde está webapps y le damos permisos: en /var/lib/tomcat9/ sudo mkdir nuevovh.es sudo chown tomcat:tomcat nuevovh.es sudo chmod 755 nuevovh.es

modifico el archivo de los hosts para añadir mi nuevo VH sudo gedit /etc/hosts 127.0.0.1 nuevovh.es

Reiniciamos tomcat sudo service tomcat9 restart

Hay que tener en cuenta que *Tomcat* usa por defecto el puerto 8080, por lo tanto si en name hemos establecido que la *URL* es midominio.es, en el navegador habrá que escribir http://midominio.es:8080/, añadiendo a esta dirección el subdirectorio de la aplicación dentro del subdirectorio establecido en appBase, a excepción de si hemos introducido la aplicación en el subdirectorio ROOT, quedando esa dirección sin ninguna añadidura.

Otra explicación

Alojamiento virtual

Vamos a configurar Tomcat para que sirva peticiones de varios dominios.

1. En primer paso añadimos las siguientes entradas en el fichero /etc/hosts

```
127.0.0.1 asir.com
```

127.0.0.1 daw.com

Recuerda que esto lo hacemos por simplicidad y no tener que configurar un servidor DNS !!!

- 2. Vamos a crear para cada dominio un directorio de despliegue.
 - Para asir.com crearemos /var/lib/tomcat9/webapps-asir
 - Para daw.com crearemos /var/lib/tomcat9/webapps-daw
- 3. Cambiamos el propietario de los directorios:
 - chown tomcat:tomcat webapps-asir
 - chown tomcat:tomcat webapps-daw
- 4. Cambiamos permisos por seguridad:
 - chmod 775 webapps-asir
 - chmod 775 webapps-daw
- 5. Editar el fichero /var/lib/tomcat9/conf/server.xml y añade un nuevo host para asir.com:

- 6. Reinicia Tomcat (sudo systemctl restart tomcat9)
- 7. Observa que se ha creado la ruta /var/lib/tomcat9/conf/Catalina/asir.com

- 8. <u>Copia</u> el fichero /var/lib/tomcat9/conf/Catalina/localhost/manager.xml a /var/lib/tomcat9/conf/Catalina/asir.com para habilitar en el Tomcat Web Manager el nuevo host virtual
- 9. Accede a http://asir.com:8080/manager y despliega la aplicacion *mi_aplicacion.war*
- 10. Ahora vamos a habilitar el host virtual daw.com siguiendo los mismos pasos que para asir.com. También tienes que desplegar la aplicación sample y habilitar el Tomcat Web Manager.
- 11. Ahora vamos a publicar una página estática en el "raíz" de daw.com. Para ello crea en webapps-daw un directorio ROOT y dentro una página index.html. Comprueba que funciona.