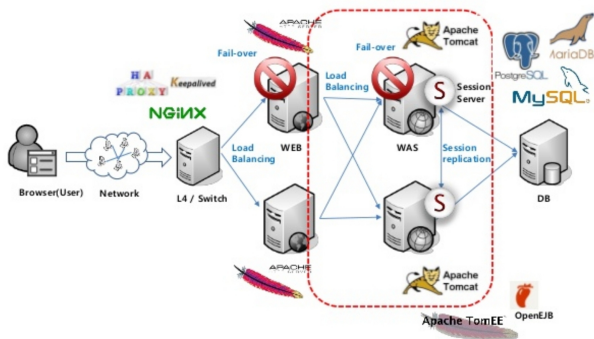
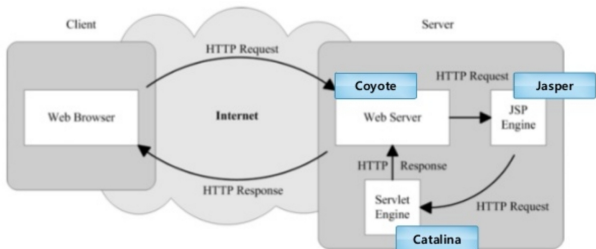


Application → Niveaux de Spec

ex: Servlet 3 } Regroupable
JSP 2.1 } en niveau JEE

Nécessaire: le Motive choisi
implémente au moins les specs
de l'appli



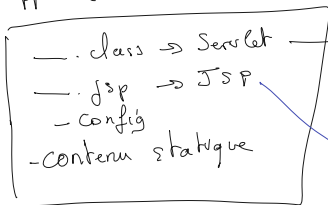


Coyote = connecteur http

Catalina = moteur de Servlet

Jasper = Compilateur JSP

Appli (.war)



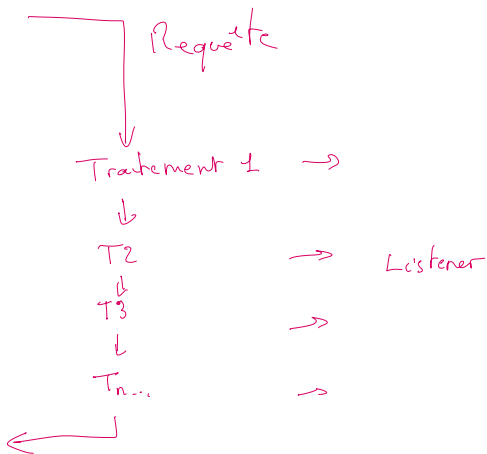
chargées,
exécutées par
Catalina

Compilées (Jasper)

JSP → source de
Servlet (.java)

Compilés en .class

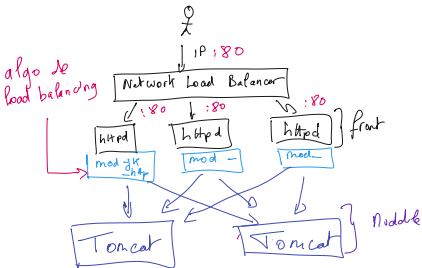
exécutés par

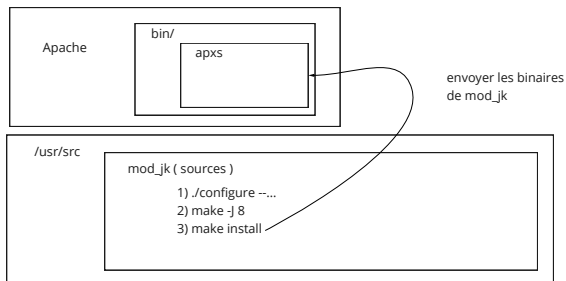


Procédure pour installer tomcat selon les bonnes pratiques et sous forme de Service:

Procédure pour installer tomcat sous forme de service avec gestionnaire disponible à distance:

[lien installation apache service gui distante](#)



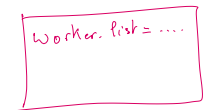


/opt/apache/conf

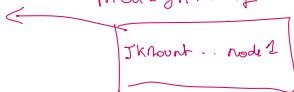
httpd.conf



workers.properties



mod-jk.conf



Télécharger mod_jk et décompresser les sources (/usr/src)
 dans native, lancer configure
 installer les prérequis (hint : yum grouplist)
 et installer le groupe outils de dev via yum groupinstall Development Tools
 ajouter les Tools apache : **yum install httpd-devel**
 configure avec **./configure --with-apxs=/usr/bin/apxs**

Activer le protocole AJP sur le Tomcat :
 avoir `<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />`
 dans le server.xml
 et `<Engine avec jvmRoute :`
`<Engine name="Catalina" defaultHost="localhost" jvmRoute="jvm1">`

Editer mod_jk.conf :

```
LoadModule jk_module "/etc/httpd/modules/mod_jk.so"
```

```
JkWorkersFile /etc/httpd/conf/workers.properties
# Where to put jk shared memory
JkShmFile /var/run/httpd/mod_jk.shm
# Where to put jk logs
JkLogFile /var/log/httpd/mod_jk.log
# Set the jk log level [debug/error/info]
JkLogLevel info
# Select the timestamp log format
JkLogStampFormat "[%a %b %d %H:%M:%S %Y]"
#JkRequestLogFormat "%w %V %T"
#JkEnvVar SSL_CLIENT_V_START worker1
```

Exemple de tuto :

http://www.diegoacuna.me/installing-mod_jk-on-apache-httpd-in-centos-6-x7-x/

Récapitulatif de httpd vers tomcat :

Ajouter / définir le virtualhost pour l'application dans /etc/httpd/conf.d/httpd.conf

ou **mieux : un autre fichier conf distinct et dans le même dossier pour ne pas manipuler le httpd.conf** et y mettre :

```
<VirtualHost *:80>
  ServerName app1.myhost.com
  ServerAdmin admin@myhost.com
  LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i\"" combined
  CustomLog /var/log/httpd/app1_access.log combined
  ErrorLog /var/log/httpd/app1_error.log
  <IfModule mod_jk.c>
    JkMount /* app1Worker
  </IfModule>
</VirtualHost>
```

mod_jk.conf fait référence à workers.properties

workers.properties détiens la liste des hosts

Modifier server.xml de tomcat pour spécifier la jvmRoute (balise en commentaires par défaut, le reste est ok)

Téléchargement du fichier archive de la configuration :

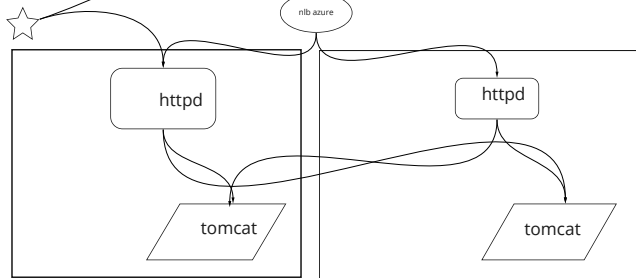
<https://1drv.ms/u/s!AvHBznCKgzcSnsAGYsW992RvJXFNDw?e=2MIMUA>

httpd.conf :
ajout de
Listen 81

mod_jk.conf :
vérifier LoadModule (le chemin)

app1.conf :
adapter ServerName à votre fqdn

server.xml
Il est conseillé, pour le sticky session, de spécifier le jvmRoute
unique par instance tomcat dans <Engine name="Catalina"



Lien vers le backup du fichier de conf :

<https://1drv.ms/u/s!AvHBznCKgzcSnsATOIUBUA66J4Rptg?e=F8dDAA>

/etc/httpd/conf/workers.properties :

```
workers.apache_log=/var/log/httpd  
worker.list=node1, node2, loadbalancer  
worker.stat1.type=status
```

```
worker.node1.type=ajp13  
worker.node1.host=tomcat018.francecentral.cloudapp.azure.com  
worker.node1.port=8009  
worker.node2.type=ajp13  
worker.node2.host=tomcat019.francecentral.cloudapp.azure.com  
worker.node2.port=8009
```

```
worker.loadbalancer.type=lb  
worker.loadbalancer.balanced_workers=node1, node2  
worker.loadbalancer.sticky_session=1
```

/etc/httpd/conf.d/app1.conf :

```
<VirtualHost *:81>  
    ServerName tomcat018.francecentral.cloudapp.azure.com  
    ServerAdmin admin@myhost.com  
    LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i\""  
    combined  
    CustomLog /var/log/httpd/app1_access.log combined  
    ErrorLog /var/log/httpd/app1_error.log  
    JkMount /* loadbalancer  
</VirtualHost>
```

Element Host dans `/opt/apache-tomcat/conf/server.xml` :

```
<Host name="localhost" appBase="webapps"
      unpackWARs="true" autoDeploy="false">

  <Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster"
    channelSendOptions="6">
    <Manager className="org.apache.catalina.ha.session.BackupManager"
      expireSessionsOnShutdown="false" notifyListenersOnReplication="true"
      mapSendOptions="6"/>
    <Channel className="org.apache.catalina.tribes.group.GroupChannel">
      <Membership
        className="org.apache.catalina.tribes.membership.McastService"
        address="228.0.0.4" port="54446" frequency="500" dropTime="3500"/>
      <Receiver
        className="org.apache.catalina.tribes.transport.nio.NioReceiver" address="auto"
        port="6000" selectorTimeout="100" maxThreads="6"/>
      <Sender
        className="org.apache.catalina.tribes.transport.ReplicationTransmitter">
        <Transport
          className="org.apache.catalina.tribes.transport.nio.PooledParallelSender"/>
        </Sender>
      </Channel>

    <Deployer className="org.apache.catalina.ha.deploy.FarmWarDeployer"
      tempDir="/opt/tomcat-temp/"
      deployDir="/opt/tomcat-deploy/"
      watchDir="/opt/tomcat-listen/"
      watchEnabled="false"/>
    <ClusterListener
      className="org.apache.catalina.ha.session.ClusterSessionListener"/>
    </Cluster>
  </Host>
```


<http://tomcat018.francecentral.cloudapp.azure.com/> : examples
<http://tc8.discimus.fr> : samples

Contenu du server.xml pour le multihost :

```
<Host name="tc8.discimus.fr" appBase="/install/samples"
unpackWARs="true" autoDeploy="true" >
    <Alias>tc9.discimus.fr</Alias>
    <Valve
className="org.apache.catalina.valves.AccessLogValve"
directory="logs"                prefix="tc8_access_log" suffix=".txt"
                                pattern="%h %l %u %t \"%r\" %s %b"
/>
    <Context path="" docBase="/install/samples"
reloadable="true" />
</Host>
<Host name="tomcat018.francecentral.cloudapp.azure.com"
appBase="/install/examples" unpackWARs="true"
autoDeploy="true" >
    <Valve
className="org.apache.catalina.valves.AccessLogValve"
directory="logs"
    prefix="examples_access_log" suffix=".txt"
    pattern="%h %l %u %t \"%r\" %s %b" />
    <Context path="" docBase="/install/examples"
reloadable="true" />
</Host>
</Engine>
</Service>
</Server>
```

tc1.discimus.fr ->
tomcat000.francecentral.cloudapp.azure.com
tc2 -> 002
tc3 -> 004
tc4 -> 006
tc5 -> 008
tc6 -> 010
tc7 -> 012
tc8 -> 018

Application de la sécurité Tomcat :

Chercher dans la documentation la page "Security manager"

Copiez le code Jsp qui plante la JVM (.exit())

Dans une application web accessible et éclatée, ajoutez un fichier jsp contenant ce code

Ex :

```
[root@tomcat018 samples]# cat test.jsp
```

```
<html>
```

```
<body>
```

```
ok
```

```
<% System.exit(1); %>
```

```
</body>
```

```
</html>
```

Affichez la page via un navigateur, le serveur doit s'arrêter.

Relancez le serveur avec le gestionnaire de sécurité

ex :

```
./catalina.sh run -security
```

Refaites le test

Vous devez avoir une erreur 500

Notez l'exception, et la permission associée

Ajoutez la permission dans la section grant { généraliste de Catalina.policy

ex :

```
grant {
```

```
// Required for JNDI lookup of named JDBC DataSource's and
```

```
// javamail named MimePart DataSource used to send mail
```

```
permission java.util.PropertyPermission "java.home", "read";
```

```
permission java.util.PropertyPermission "java.naming.*", "read";
```

```
permission java.util.PropertyPermission "javax.sql.*", "read";
```

```
permission java.lang.RuntimePermission "exitVM.1";
```

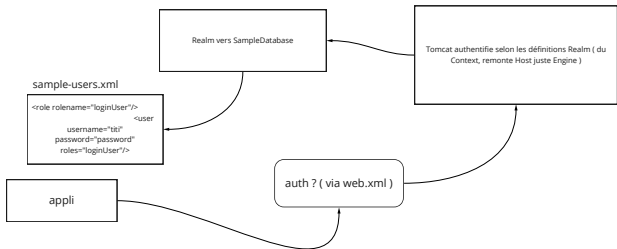
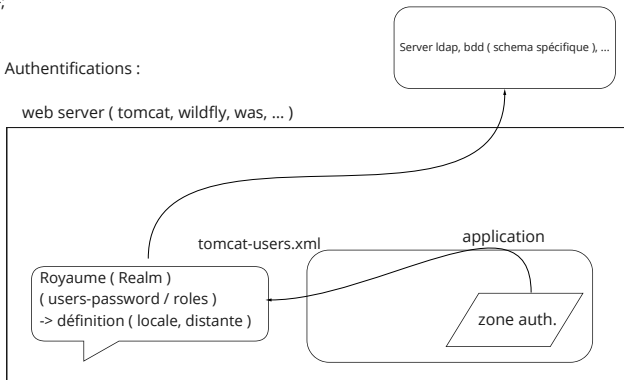
Refaites le test.

Vous avez appliqué la sécurité, mais autorisé l'appel (dangereux) à cette méthode système.

Appliquer de la sécurité pour une application en particulier :
ex de format à ajouter :

```
grant codeBase "file:${catalina.home}/webapps/app1/-" {  
  permission java.lang.RuntimePermission "getenv.APP1_HOME", "";  
  permission java.lang.RuntimePermission "otherpackage.classes.*";  
  permission java.security.AllPermission;  
};
```

Authentifications :



```

<!-- FORM-BASED LOGIN AUTHENTICATION EXAMPLE -->
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/javaee" version="2.5"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
  http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">

  <display-name>hello1_formauth</display-name>
  <servlet>
    <display-name>index</display-name>
    <servlet-name>index</servlet-name>
    <jsp-file>/index.jsp</jsp-file>
  </servlet>
  <security-constraint>
    <display-name>SecurityConstraint</display-name>
    <web-resource-collection>
      <web-resource-name>WRCollection</web-resource-name>
      <url-pattern>/*</url-pattern>
    </web-resource-collection>
    <auth-constraint>
      <role-name>loginUser</role-name>
    </auth-constraint>
    <user-data-constraint>
      <transport-guarantee>NONE</transport-guarantee>
    </user-data-constraint>
  </security-constraint>
  <login-config>
    <auth-method>FORM</auth-method>
    <form-login-config>
      <form-login-page>/logon.jsp</form-login-page>
      <form-error-page>/logonError.jsp</form-error-page>
    </form-login-config>
  </login-config>
  <security-role>
    <role-name>loginUser</role-name>
  </security-role>
</web-app>

```

logon.jsp :

```
<html>
<head>
  <title>Login Page</title>
</head>

<h2>Hello, please log in:</h2>
<br><br>
<form action="/j_security_check" method=post>
  <p><strong>Please Enter Your User Name: </strong>
  <input type="text" name="/j_username" size="25">
  <p><p><strong>Please Enter Your Password: </strong>
  <input type="password" size="15" name="/j_password">
  <p><p>
  <input type="submit" value="Submit">
  <input type="reset" value="Reset">
</form>
</html>
```